

Incremental Acquisition of Conceptual Indices for Multimedia Design Documentation

Catherine Baudin* Barney Pell**

Artificial Intelligence Research Branch
NASA Ames Research Center
Mail Stop 269-2
Moffett Field, CA 94035
{baudin,pell}@ptolemy.arc.nasa.gov

Smadar Kedar

Institute for the Learning
Sciences
Northwestern University
Evanston, IL 60201
kedar@ils.nwu.edu

Abstract

Information retrieval systems based on conceptual indexing can access the underlying meaning of text, graphics or videotaped documents. Since conceptual indices represent the semantics of a piece of information, it is difficult to extract them automatically from a document, and it is tedious to build them manually. We present a method to acquire and refine conceptual indices in the context of Dedal, a system that facilitates the indexing and retrieval of text, graphics and videotaped design documents in the mechanical engineering domain. Our approach is to use an underlying model of the domain covered by the documents to constrain the user's queries. This facilitates question-based acquisition of conceptual indices: converting user queries into indices which accurately model the content of the documents, and can be reused. We demonstrate the relevance and coverage of the acquired indices through experimentation.

1. Motivation

Information retrieval systems based on conceptual indexing can access the underlying meaning of text, graphics or videotaped documents. Conceptual indices focus on the important concepts of a domain (the semantics) rather than on the multiple ways these concepts are represented in a document (the syntax). This facilitates information retrieval because: (1) the number of concepts in a document is smaller than the number of their possible syntactic representations, thus facilitating vocabulary selection when formulating queries to a system, and (2) since conceptual indices represent the *content* of a piece of information they can be used by a reasoning component to facilitate the match between a query and the information in the documents [Baudin et al. 93b][Tong et al. 89]. The following example, extracted from a technical design report, illustrates the difference between conceptual and syntactic indexing.

"The inner hub holds the steel friction disks and causes them to rotate when the road input is transmitted through the connecting link to the rotating inner shaft...

This paragraph can be indexed by words from the text such as *inner hub, friction disks, inner shaft, connecting links*. However, the content of this text refers to concepts like the *function of the inner-hub*, or the *relation between the road input and the way the device works*. Accessing these concepts enables an information retrieval system to accurately answer questions about the function of each part of the device, their operation and the way they interact. Conceptual indexing combined with knowledge of the relations among the objects in a domain can be used by a reasoning component to draw *inferences* about how to locate a piece of information. In this example, the content of the above paragraph can be summarized by one concept: "operation of disk stack" to convey the fact that it describes how the disk stack device works. A reasoning component can then infer that the paragraph might describe the function of each part of the disk stack and the way they interact. In this case, the component *hub* is a subpart of the *disk stack* mechanism and its function is referenced in the paragraph (the purpose of the inner hub is to hold the rotating friction disks).

Since conceptual indices represent the underlying meaning of a piece of information, the language used to build these indices is usually different from the language in the documents. This abstraction level mismatch between the indexing language and the language used to convey the information makes it difficult to automatically extract conceptual indices directly from multimedia documents. On the other hand, the creation of conceptual indices by human indexers is a labor intensive task that is difficult to perform exhaustively. This is particularly true for a large corpus of documentation where concepts are closely interrelated, as is the case for technical documents that describe the operation, diagnosis or design of complex artifacts.

2. Approach

Our approach is to use a *conceptual query language* plus feedback from the user on the relevance of the documents retrieved in response to a query, to incrementally acquire new conceptual indices for that document. The user formulates a query to the system. If no document description exactly matches the query, the system approximates the retrieval and prompts the user for

(*) RECOM Technologies.

(**) RIACS Research Institute for Automated Computer Science.

feedback on the relevance of the references retrieved. If a reference is confirmed, the query is turned into a new index. This extends *relevance feedback* techniques [Salton et al. 90] to the acquisition of *conceptual indices*.

This approach uses a *question-based indexing* paradigm [Osgood et al. 91][Schank 91][Mabogunje 90] where the query language and the indexing language have the same structure and use the same vocabulary. The assumption is that the questions asked by users indicate the objects and relationships that are relevant to describe the content of the documents at a conceptual level appropriate for a class of users. However, in order to use the queries to acquire new indices the following conditions must be met by the query language:

1. **Reusability:** The query language must be general enough to create indices that will match a class of queries.
2. **Relevance:** The query language must be able to describe the information that the user is interested in. Articulating queries to acquire information in order to achieve a goal is in general a difficult task [Croft et al. 90][Graesser et al. 85]. In our approach, the query formulation is *constrained* by a model of the domain covered by the documents and a model of the type of information of interest to designers.

In the next section we provide background on Dedal, a system that acquires conceptual indices to facilitate the reuse of multimedia design documents in the mechanical engineering domain. Section 4 describes the index acquisition and refinement process. In section 5 we discuss two experiments where conceptual indices were created by Dedal while mechanical engineers used the system to access information about a shock absorber design. Section 6 presents a discussion and a set of areas for future work.

3. Indexing and Retrieval in Dedal

An engineering team can generate a large amount of information in different forms. For instance, a designer's notebook is a good place to look for alternatives considered and the analysis performed to evaluate them. Design meeting minutes are likely to refer to the main issues, decisions and rationale that led to a particular solution. Technical reports provide details on selected solutions while tests performed on a prototype might be best documented by a videotape showing its operation and the test setting. Canned-text design information or videotapes of meetings are easy to capture but difficult to retrieve by systems that have no representation of the information *content*.

We developed Dedal, an information retrieval system that uses conceptual indexing to represent the content and the form of multimedia text, graphics and videotaped design information. in mechanical engineering design. It is an interface to records such as meeting summaries, pages of a designer's notebook, technical reports, CAD drawings and videotaped conversations between designers.

This type of knowledge-based retrieval requires: (1) a conceptual language to describe and query documents, (2) domain knowledge about the indexing concepts and their

relationships and (3) heuristic retrieval strategies for matching a query with an index.

3.1 Conceptual Indexing Language

Based on studies of the information seeking behavior of designers conducted at Stanford's Center for Design Research and NASA Ames, we identified a language to describe and query design information [Baya et al. 92]. This language combines concepts from a model of the artifact being designed with a task vocabulary representing the classes of design topics usually covered by design documents. For instance, "function," "operation," or "alternative" are topics of the task vocabulary.

A conceptual index can be seen as a structured entity made of two parts: the *body* of the index which represents the content of a piece of information and the *reference* part that point to a region in a document. In Dedal the body of an index has the following form: <topic **T** subject **S** level of detail **L** medium **M**> where S is a list of subjects from a domain model and T, L and M are member of the task vocabulary. The reference part of an index contains a pointer to the *record* and *segment* corresponding to the starting location of the information in a document (e.g. document name and page number or video counter). A segment of information is described by several conceptual indices, each of which partially describing its content.

For instance: "The inner hub holds the steel friction disks and causes them to rotate" is part of a paragraph in page 20 of the record: report-333. It can be described by two indexing patterns:

```
<topic function subject inner-hub level-of-detail
configuration medium text segment-size:
paragraph in-record report-333 segment 20>.
```

```
<topic relation subject inner-hub and steel-friction-
disks level-of-detail configuration medium text
segment-size: paragraph in-record report-333
segment 20>
```

The queries have the same structure as the body of an index and use the same vocabulary. A question such as: "How does the inner hub interact with the friction disks?" can be formulated in Dedal's language as:

```
<get-information-about topic relation regarding subject
inner-hub and steel-friction-disks with preferred
medium equation>.
```

3.2 The Domain Model

The domain model defines and organizes the indexing terms and their relations. This organization is used both in query formulation, to help users select terms which are understood by the system, and in retrieval, to the system find terms related to those in a given query.

In the mechanical engineering design domain, the model includes a representation of the artifact structure, some aspects of its function, the main decision points and alternatives considered. It also includes concepts that are part of the problem but external to the device representation. The main relations in the model are *isa*,

part-of, *attribute-of*, and *depends-on* (see Figure 1). For example, in Figure 1, *metal-disk* is part of the *disk-stack*, and the value of the *resistive force of the disk-stack* depends on the value of *torque of the rotary damper*.

While elaborate domain models increase the power of the retrieval by enabling sophisticated retrieval strategies (see Section 3.3), it should be noted we can start an indexing effort with a simple structural model (involving only *part-of* relations), and extend the relations over time as the base of indices grows.

3.3 Retrieval

The retrieval module takes a query from the user as input, matches the question to the set of conceptual indices and returns an ordered list of references related to the query. The retrieval proceeds in two steps: (1) exact match: find the indices that exactly match the query and return the associated list of references. If the exact match fails: (2) approximate match: activate the *proximity retrieval heuristics*.

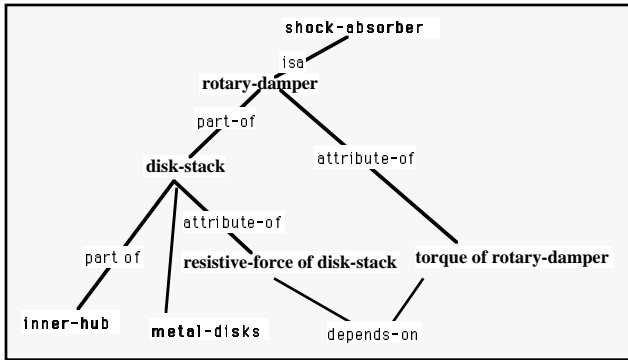


Figure 1: Objects and relations in the domain model

The analysis of different types of design records such as engineering notebooks and structured progress reports led to the identification of a set of heuristics to help match a user's query with the set of indexing patterns describing the documents. These heuristics are activated when no index exactly matches a query or when the user is not satisfied with the references retrieved. The goal is to direct the user to regions in the documentation that contain related information, on the assumption that the required information will be found near these regions (on the same page, or in the same subsection depending on the type of record). Dedal currently uses twenty proximity retrieval heuristics to find related answers to a query. For instance:

equation-to-schemata: In an engineering notebook, an equation describing a mechanism will usually be found next to a drawing representing this mechanism.

Some heuristics reason from the relations between the concepts in the task vocabulary:

performance-to-analysis: Information about the *performance* of a particular assembly and the *analysis* of this assembly are likely to be located in nearby regions of the documentation.

Other rules exploit the hierarchical relations in the domain model:

operation-to-function: In a structured document such as a progress report, the *function* of a component X in a mechanical assembly Y might be found near where the *operation* of assembly Y is described.

The heuristics use knowledge of the relations in the domain model to compensate for missing indexing patterns. For example, an information segment can be summarized by an indexing pattern using terms attached to a high level of a subject hierarchy. This enables the system to infer the likely existence of more detailed indexing patterns within the same segment. For instance, in our example, instead of the two indexing patterns pointing to paragraphs describing the interactions between the parts of the disk-stack mechanism, the segment of information could have been "summarized" by one indexing pattern:

<topic **operation** subject **disk-stack** level-of-detail **configuration** medium **text** segment-size: **page** in-record **report-333** segment **20**>.

to state that page 20 in report-333 describes how the disk-stack mechanism works. From this, the system can infer (using the operation-to-function heuristic above in conjunction with the model portion shown in Figure 1) that this information segment may include information about the function of the subparts and the way they interact.

While these heuristics have been shown to significantly increase the recall of the system when compared with the system without the heuristics and with a base-line boolean retrieval system [Baudin et al. 93b], they are not always able to make correct predictions and sometimes retrieve irrelevant references. For instance, there is no information in segment 20 about metal-disks, although metal-disks are also part of the disk-stack. This motivates the need for the system to refine its indexing knowledge through user feedback.

4. Incremental Acquisition of Indexing Knowledge

The system starts with a core of conceptual indices entered manually and extends this core of indices using the queries of the end-user and feedback about the relevance of the documents retrieved. This section describes this incremental acquisition process.

Dedal acquires new indexing knowledge in three phases: (1) an index creation phase, (2) an index monitoring phase, and (3) an index refinement phase. Figure 2 illustrates with an example how Dedal acquires a new index. The index creation phase goes through the following steps:

1. Query formulation: The user's intended question is "what is the function of the hub?". In the query formulation phase the user must express the question in the conceptual language of Dedal. This is partly done by navigating the domain model (the *part-of* hierarchy or the *class-subclass*

hierarchy for instance) to select appropriate subjects for the query. In this case the user selects the subject *inner-hub* from the domain model and the topic *function* from the task vocabulary, the corresponding query in Dedal is: < topic: function of subject: inner-hub> (In the following paragraphs we will use a shortened syntax for queries where the words topic and subjects are omitted and where domain concepts are indicated in bold).

2. Query-Index mapping: Dedal tries to find an index that exactly matches the query. In this case, it does not find an exact match and applies a proximity heuristic to guess where the required information may be located. The heuristic states that any information describing how a mechanism works might also describe the function of its parts. In this case, given that *inner-hub* is a subpart of the *disk-stack* mechanism, Dedal matches the query "function of **inner-hub**" with two indices I1 and I2 pointing to two information regions describing the "operation of **disk-stack**".

3. Relevance Feedback: The user looks at the two references retrieved, finds that the reference pointed to by the index I2 (page 12 in the record report-333) describes the function of the inner hub while the document associated with index I1 does not. The user rates the reference I1 as irrelevant and I2 as relevant.

4. Index Acquisition: The query: "function of inner-hub" is more specific (see section 3) than the index "operation of disk-stack". In this case Dedal creates a new index I3. The system now knows that page 12 of report-333 explicitly describes the function of the inner-hub.

Each time a reference is retrieved by the approximate match and is relevant, Dedal attaches the reference of the selected index to the query, turning the query into a new

index (as shown in step 4 in figure 2). In addition, the procedure records the type of inference that relates each subject of the new question to the subject of the matching index [Baudin et al, 1993a].

4.2 Index Monitoring

Two factors may impact the ability of an acquired index to accurately describe the associated information:

(1) *incompleteness of the domain model*: If the model is missing the particular subject the user is interested in and the user selects a related subject, the approximate match might still retrieve a relevant document. In this case the user query does not exactly describe the information required by the user and the resulting index will be inaccurate.

(2) *Noise in the relevance feedback*. In particular, when a query involves several subjects from the model, the user might feel satisfied with a document that refers to a *subset* of these subjects. For instance, if the query is of the form "relation between **outer-cage, solenoid and lever**" the user might feel satisfied with a reference which only describes the relation between outer-cage and solenoid, the third argument: lever will then incorrectly describes the content of the referenced document.

The index monitoring phase keeps track of the performance of newly acquired indices. Each time a query Q matches an acquired index I, the following procedure is activated: if the corresponding reference is relevant, the success count of the index is incremented. If the reference retrieved is irrelevant, the failure count of the index is incremented.

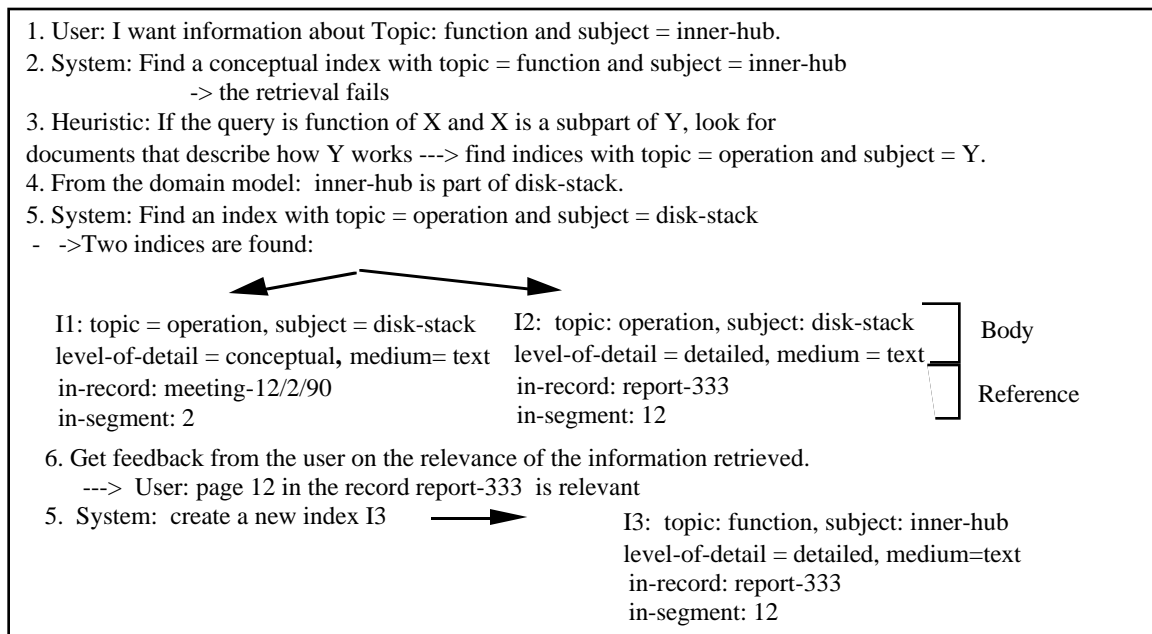


Figure 2: Acquiring a new index

4.3 Index Refinement

The purpose of this phase is to enable the system to compensate for errors in the newly acquired indexing knowledge. Our goal here is to prune the irrelevant references caused by noisy data, that is errors caused by ambiguity in the relevance feedback provided by the user and incompleteness of the model.

In order to refine the indexing knowledge in DEDAL we gather examples of relevant and irrelevant query/index pairs generated during user interaction. Figure 3 shows a positive example of query/index pair (a query/index match that retrieved a relevant reference). It is represented as a set of features. For instance in this example the subject of the query is inner-hub, the topic of the query is function, the subject of the matching index is disk-stack, and the name of the heuristic used is function-to-operation.

```
example E-122
type: positive
heuristic-name: operation-to-function
query-topic: function
index-topic: operation
query-subjects: inner-hub
index-subjects: disk-stack
medium-query: any
medium-index: text
level-of-detail-query: any
level-of-detail-index: configuration
segment-size: page
in-record: report-333
in-segment: 12
```

Figure 3: A query/index pair.

We use an off-the-shelf ID3 induction algorithm [Quinlan 86] on the examples of successful and failed retrievals to generate a decision tree. When a query/retrieval example is selected by the existing set of heuristics, this example is filtered by the decision tree in an attempt to detect irrelevant references and assign them low priority. The following rules correspond to two branches of the decision tree generated during an experiment where a mechanical engineer interacted with DEDAL to retrieve information about the "variable damper" design (see section 5).

```
If subject-query = suspension-system and
record= spring-drd-damper-89
segment = 15 then negative
```

```
If subject-query = arm
name-of-heuristic = function-to-operation then negative
```

The branch of the decision tree represented by the first rule above states that the reference segment 15 in document spring-drd-damper-89 is not relevant to any questions about *suspension-system*. The second rule above states that the heuristic: function-to-operation leads to irrelevant retrieval when the question is about *arm*. While the first rule suggest that segment 15 is incorrectly indexed, this second rule is actually pointing a flaw in the *retrieval heuristics*

themselves. This refinement step is described in detail in [Baudin et al. 94b].

5. Results

In this section we evaluate the effectiveness of Dedal's index acquisition. Index acquisition is considered effective if the system improves its retrieval *performance* following its interaction with users, and if this improvement follows from increased knowledge of the *content* of the document set. We first report on an experiment on the impact of the acquired indexing knowledge on retrieval performance. We then discuss the content of the knowledge acquired.

Impact of acquired indexing knowledge on retrieval performance.

We selected a set of examples generated by two mechanical engineers using DEDAL to retrieve answers about the "rotary-friction-damper" design, an innovative electromechanical shock absorber designed for Ford Motor Corporation [Baudin et al. 92]. The users queried the system while solving a redesign problem involving the modification of the shock absorber design. Each of the references retrieved was rated by the users as relevant or irrelevant.

We extracted 300 examples of relevant and irrelevant heuristic retrievals generated from the interaction with one user (user1) as our training set, and another set of 81 examples generated by another user (user2) as our test set. We measured the impact of the decision tree in terms of its ability to filter the negative examples in the test set while preserving the positive examples. The total number of examples generated by user2 is 81, with 54 negative examples and 27 positive examples. The cases that are undecided are counted as positive. The decision tree generated for this experiment would have reduced by 44% the number of irrelevant references retrieved by the heuristics, while assigning low priority to only 7% of the relevant references presented to user2. This represents a marked increase in precision, with only a minor decrease in recall.

Analysis of Content of Acquired Indexing Knowledge.

To understand the actual content of the acquired knowledge, we considered separately the content of the generated indices and the content of the decision trees generated during the index refinement phase.

To assess the content of the acquired indices, we presented the 71 indices created during interaction with user1 along with the associated information regions to a designer familiar with the shock absorber documentation. The designer rated each of these indices as relevant or irrelevant depending on his appreciation of the ability of the index to describe (part of) the associated information. The designer rated 86% of the acquired indices as relevant.

It should be noted that the designer was evaluating the relevance of the indices independently of the context in which they were generated. He was different from the users who conducted the experiment, he rated the indices

independently of any problem solving task, and he had no access to the English version of the questions that motivated the queries.

To assess the content of the decision trees, we presented the trees generated during this experiment to an expert familiar with the domain. As discussed in section 4, these trees pointed out flaws in the acquired indices as well as flaws in the retrieval heuristics themselves. The detection of these flaws and presentation in an intelligible fashion (decision trees) could enable the professional indexer to provide additional refinements to the indexing knowledge.

6 Discussion and Future Work

The ultimate goal of our work is to produce a system that has a large base of knowledge about the semantic content of a corpus of multimedia documents. The totally manual approach is both labor-intensive and difficult to perform exhaustively. The totally automated approach, which requires a program to extract document content directly from raw information and without apriori knowledge of the domain or the document contents, is difficult to realize with current technology. Our approach falls midway between these two extremes. The system starts with a core of manually entered indexing knowledge representing document content at a coarse level. It then incrementally extends this core through experience by turning user queries into indices that partially describe the information at a conceptual level appropriate for a given class of users in a given domain.

The power of our query-based index acquisition approach derives from: (1) constraining the query language: this requires studying the information needs of users in a given domain to identify generic types of questions of interest to this class of users, and (2) using a model of the domain to relate queries with more general or related conceptual indices.

Completeness of the Query Language: While a constrained query language ensures that the queries reflect the content of the required information, this raises issues about how complete the domain model must be before this type of index acquisition can be effectively used. In the index acquisition approach described in this paper, we made the assumption that there are enough concepts in the domain model to enable a user to formulate a query. At times this might prove to be a simplified assumption. A missing domain subject forces the user to fall back on a related subject and is a source of inaccuracy in the use of queries for indexing purposes. One way of alleviating this problem is to allow the end-user to define new indexing terms in the domain model when he cannot find a suitable concept to formulate a query. However, the possibility of defining new terms in the domain model during the query formulation phase raises the question of whether an end-user is qualified to update models or whether this task should be reserved for the professional indexer or the knowledge-engineer. We are investigating scenarios in which the system supports and derives feedback from *both*

professional *indexers* and *end-users* at different points in the system's growth [Baudin et al, 1994b]. Another solution under investigation is to integrate conceptual retrieval with syntactic retrieval methods capable of extracting features of the documents in response to queries. This would enable the user to refer to concepts which have not yet been defined in the model, and would provide an opportunity for the system to add such concepts to the model.

Domain Model: While much of the power of this approach derives from the use of a domain model, a model must be defined for each new design project. This model-building effort is a significant investment when compared to knowledge-free information retrieval approaches. However, there are three advantages of domains that relate to engineered artifacts. First, the scope of the domain model is usually well defined. For instance, in the engineering design domain a large part of a technical documentation can be indexed using terms of a structural model (*part-of* hierarchy of components) of the designed artifact. Second, parts of such models often exist for other purposes, such as simulation or CAD models. Third, the development of a model may have useful benefits outside of the retrieval task. For example, one useful by-product of this model-construction effort is that the domain model becomes a design glossary whose terms are related by different types of relations. It is interesting to note that this type of "super glossary" is actually useful to the members of a design project as it explicitly defines what is meant by the vocabulary used by each member of the team.

Acknowledgments

Thanks to Vinod Baya, Ade Mabogunje and Jody Gevins Underwood who participated in the development and evaluation of DEDAL. Thanks to Larry Leifer and to the other members of the GCDK group for their feedback and support on this project, to Wray Buntine and members of NASA Ames. Thanks to Michel Baudin for his help on earlier drafts.

References

- Baudin, C., Gevins, J., Baya, V., Mabogunje, A. "Dedal: Using Domain Concepts to Index Engineering Design Information", Proceedings of the Meeting of the Cognitive Science Society, Bloomington, Indiana. 1992
- Baudin, C., Kedar, S., Gevins, J., Baya, V., Question-Based Acquisition of Conceptual Indices for Multimedia Design Documentation. Proceedings of AAAI93 conference, Washington, D.C., 1993a.
- Baudin, C., Gevins, J., Baya, V., "Using Device Models to Facilitate the Retrieval of Multimedia Design Information", in proceedings of IJCAI 93 Chambéry, 1993b.
- Baudin, C., Kedar, S., Pell, B. "Increasing Levels of Assistance in Refinement of Knowledge-Based Retrieval Systems" in the Knowledge Acquisition Journal, Vol 6, 1994a.

Baudin, C., Pell, B., Kedar, S. "Using Induction to Refine Information Retrieval Strategies" in Proceedings of the AAAI94 conference. Seattle 1994b.

Baya, V, Gevins, J, Baudin, C, Mabogunje, A, Leifer, L., Toye, G., "An Experimental Study of Design Information Reuse", in proceedings of the 4th International Conference on Design Theory and Methodology, 1992.

Croft, W.B, Das, R., "Experiments with Query Acquisition and Use in Document Retrieval Systems". in Proceedings of SIGIR 1990.

Graesser, A.; Black, J., The Psychology of Questions. Lawrence Erlbaum associates 1985..

Mabogunje, A. "A conceptual framework for the development of a question based design methodology", Center for Design Research Technical Report (19900209), February 1990.

Mauldin, M. L., "Retrieval Performance in Ferret, A Conceptual Information Retrieval System". in Proceedings of SIGIR 1990.

Osgood, R., Bareiss, R. "Question-based indexing", Technical report 1991, The Institute for the Learning Sciences, Northwestern University 1991.

Quinlan, J.R., Induction of decision trees. Machine Learning 1(1):81-106 1986.

Salton, G., Buckley, C., Improving Retrieval Performance by Relevance Feedback. J. of ASIS. 41:288-297. 1990.

Schank, R., Ferguson, W., Birnbaum, L., Barger, J., Greising, M., "ASK TOM: An Experimental Interface for Video Case Libraries" ILS technical report, March 1991.

Tong, M. R., Appelbaum, A., and Askman V. "A Knowledge Representation for Conceptual Information Retrieval", International Journal of Intelligent Systems. vol. 4, 259-283, 1989.