Search and Retrieval in Object-Oriented Information Systems

Erich J. Neuhold, Karl Aberer,
Wolfgang Klas, Adelheit Stein, Ulrich Thiel

March 1993

Abstract

In this paper, we outline the features of a knowledge-based retrieval interface MERIT (Multimedia Extensions of Retrieval Interaction Tools), and VODAK, an object-oriented DBMS suitable for multimedia data. MERIT offers access to a subset of the CORDIS databases with data about European research programs, projects, and consortia in the field of information technology. The system features graphical presentations of retrieval results, employs interactive maps for geographical data, and provides scanned-in documents, e.g. photos of contact persons. The database management system functionality of VODAK is extended such that it treats multimedia data, like text, audio, or video, as an integral part of the databases. Furthermore VODAK is an extendible database management system which allows to integrate information from heterogeneous and distributed resources.

Published in:
Proceedings of the GI-EMISA Workshop: "Objektorientierte Methoden für Informationssysteme, Klagenfurt, 7.-9. Juni 1993". Berlin: Springer, 1993.

1 Introduction

Retrieving data that satisfies a complex information need is often an exploratory and incremental process. In addition, the handling of multimedia information places an additional cognitive workload on the human user. A guiding principle of system design is that the system should adapt to the needs of the human user, and not vice versa [cf. Norman & Draper 1986]. This means that the user with his strong and even more with his weak points has to be the central parameter in the interface design process. An ideal system would be one which presents the user a familiar environment, which minimizes the learning of the formal details of the system in order to work effectively with it. Therefore, the design of the information system must respond to the need for user assistance by incorporating components which address the following problems users of information systems have to face:

Finding a useful retrieval strategy: Information-seeking processes are not determined by an exactly predefined task structure, but by individual strategies and tactics of users to which the system should adapt. Taking this assumption into account, we use “case-based” dialogue plans to guide the user through the retrieval process.

Flexible dialogue control: The user must have the opportunity to alter the dialogue steps proposed by the system and by this way control the dialogue flexibly. The system should provide the user appropriate means for withdrawing dialogue steps, going back in the dialogue history, changing the retrieval strategy, etc. In addition, subdialogues have to be permitted for requesting more context information on the current state. This may result in very complex dialogue structures which should be made transparent to the user in order to avoid disorientation.

Clarification of information needs: The user should be able to access a database without knowing the particular terminology or conventions used during the compilation of the database. The system should support the user in viewing the concepts under consideration from different perspectives during the query formulation phases and the data inspection phases of the dialogue.

Visualization of information spaces: Human perceptual processes operate in a highly parallel fashion. The presentation of information in forms which are to be serially consumed by the user (such as lengthy texts) should be combined with visualizations that provide survey information at a glance (such as graphics, pictures).

Relevance assessment of retrieved items: In complex information systems, which are used for document retrieval or knowledge base access, the problem of relevance assessment arises. The system should employ interactive presentation forms which exhibit the cues that are required for this purpose.

In this paper, we outline a system architecture which is intended to meet these requirements, and show how an object-oriented DBMS, which is capable to cope with multimedia data, can support this approach. Throughout the paper, we will illustrate our ideas with examples from recent prototypes that have been developed at GMD-IPSI. In particular, we will refer to the features of

MERIT (**M**ultimedia **E**xtensions of **R**etrieval **I**nteraction **T**ools), a knowledge-based retrieval interface [cf. Stein, Thiel & Tißen 1992], and **VODAK,** an object-oriented DBMS suitable for multimedia data. MERIT offers access to a subset of the CORDIS databases with data about European research programs, projects, and consortia in the field of information technology[1]. The system features graphical presentations of retrieval results, employs interactive maps for geographical data, and provides scanned-in documents, e.g. photos of contact persons. The database management system functionality of VODAK is extended such that it treats multimedia data, like text, audio, or video, as an integral part of the databases. Furthermore VODAK is an extendible database management system which allows to integrate information from heterogeneous and distributed resources.

The paper is organized as follows: We start with an overview of a user-centered system architecture which is intended to tackle the problems mentioned above. Next, an outline of the object-oriented DBMS VODAK is given. In the subsequent sections, we discuss some aspects of the user-centered approach in detail, starting, just as in a MERIT dialogue session, with the user's determination of the information seeking strategy that is to be supported by the system. We then outline the system's components, which provide strategical and terminological assistance to the user. Next, some interactive presentation forms for retrieved data are discussed with respect to their effects on the user's relevance assessment. The paper concludes with short accounts of related and future work.

2 A System Architecture Supporting Search and Retrieval in Multimedia Databases

In order to achieve context sensitive and flexible dialogues the *Graphical Interface* (see figure 1) is augmented by dialogue and presentation components. The *Dialogue Manager* is able to support a global dialogue strategy as well as to maintain the conversational flow of information. The *Presentation Manager* generates the presentation of information and information structures to the user in a form which can most effectively be processed by the user's cognitive and perceptual capabilities. The *Database Interface* translates the user's requests into a formal query to the underlying database.

The *Dialogue Manager* operates on an internal representation of the dialogue. Thus, it is possible to handle the direct manipulative inputs of the user in a context-sensitive way. Retrieving information from an information system can be seen as an iterative process. A complex information need can be formulated in multiple ways, e.g. with a few, but complex queries in contrast to a step by step query formulation.

1. MERIT runs on SUN color workstations (SPARC stations) and is written in CommonLISP. As its graphical interface, MERIT uses HyperNeWS, a NeWS-based graphics environment developed at the Turing Institute in Glasgow. Its dynamic and static knowledge bases are based on CLOS, the object-oriented extension of CommonLISP, and on CRL, a part of Knowledge Craft. It accesses a subset of the CORDIS databases which are offered online by ECHO, the official host organization of the Commission of the European Community (CEC).

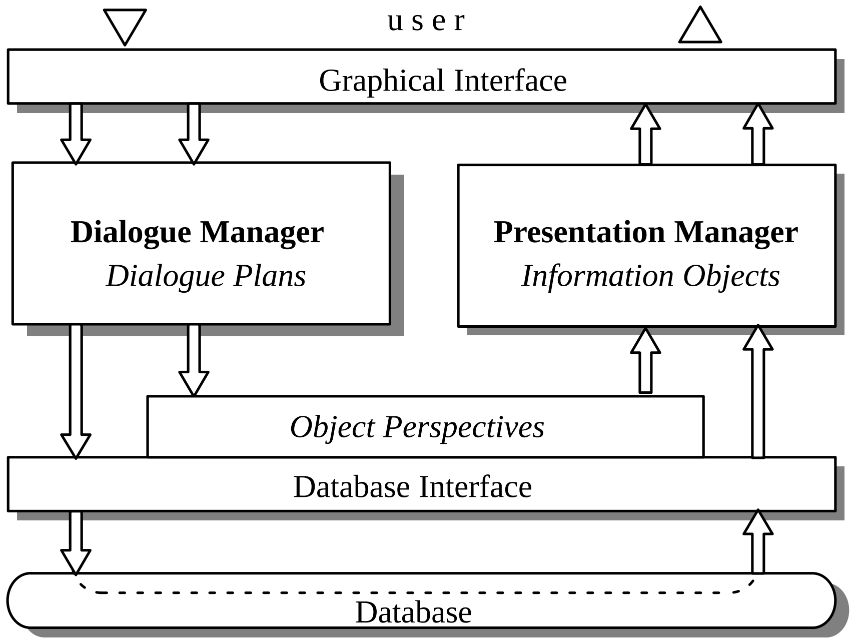

Figure 1: User-centered architecture

The user's actions (e.g. mouse clicks, menu selections) will not only send messages to the graphical surface objects which then execute methods, but will effect transformations of the internal representation of the dialogue. This representation is to be maintained by the dialogue manager employing a conversational model [cf. Sitter & Stein 1992, Stein & Thiel 1993] of the interaction to keep track of the ongoing dialogue. Interpreting the interaction in terms of a conversational model allows the system to realize its reactions as instances of generic dialogue acts like Inform, Offer, Request. In general, a dialogue act performed by the system will result in the visualization of new graphical dialogue objects on the screen.

A case-based approach to user guidance [cf. Tißen 1991, Stein, Thiel & Tißen 1992, Belkin et al. 1993] allows the system to adopt a dialogue strategy which is adequate to achieve the user's dialogue goals. The strategy is determined by a sequence of steps in a dialogue plan. Steps are represented as frames which contain slots for preconditions, actions and side-effects (like plan operators in traditional approaches to plan representation, extended by input-/output descriptions and perspectives). A perspective consists of the main concept, relevant query and presentation attributes, and navigation paths to related concepts.

The main function of the *Presentation Manager* is the automatic generation of system reactions using generic presentation forms. This component provides the display of retrieved results obtained from the database. While these data form the propositional content of the system's contributions, the Presentation Manager also selects the graphical means to express the purpose (illocution) of a dialogue act, e.g. to inform, explain, comment. For the generation of dialogue contributions that do not involve data from the database, e.g. explanations, help, context information, the dialogue manager uses its dialogue knowledge bases.

The *Database Interface* translates the propositional content of the user's input into a query formulated in the query language of the underlying database. Stating restrictions in a query form sheet allows the user to specify the set of items he is interested in. In order to assist users in formulating restrictions for textual information, we also provide a facility (*Knowledge Explorer*) for semi-automatic term extension that is based on a "fuzzy association network" [cf. Kracker 1992].

The database itself can be realized in VODAK [cf. Duchêne, Kaul & Turau 1988, Klas 1990, Turau & Rakow 1993], a behavioral object-oriented database management system. The schema of the database is expressed in VML (**V**ODAK **M**odeling **L**anguage) [cf. Klas et al. 1992] and provides the conceptual model for the information system. Since VML supports metaclasses as schema objects, i.e. classes and metaclasses are first class objects themselves, one can refer to classes appearing in the schema as to any other object in the database. This is the way the VODAK data dictionary is realized. The metaclass concept is also the basis for other important mechanisms. First, it allows to introduce arbitrary semantic relationships, like specialization, generalization or part-of, between classes and their corresponding objects. Second, it makes VODAK an extendible database management system as it allows to integrate schemas and databases of other database management systems. This can, for example, be used to build an object-oriented schema of the CORDIS databases.

The *query language* of VODAK, VQL [cf. Fischer & Aberer 1993], allows for a SQL-like set-oriented access to the database. In a VQL-statement arbitrary method-calls may appear and so arbitrary search strategies may be used in this way, as they may be necessary, for example, for multimedia datatypes. VODAK provides modelling primitives for representing noncontinuous multimedia data, e.g. text or graphics, as well as continuous multimedia data, e.g. audio and video. The goal is to support multimedia data in a way, that it is, like the conventional alphanumeric data, handled as an integral part of the database management system.

3 Searching Complex Objects: A Sample Domain and its Object-Oriented Representation

A typical example of mixing structured and unstructured data is the information offered by CORDIS (**C**ommunity **R**esearch and **D**evelopment **I**nformation **S**ervice). CORDIS provides information about the **E**uropean **C**ommunity (EC) **R**esearch and **T**echnological **D**evelopment (RTD) programs and related matters for organizations and individuals.[2] These data can be accessed online via a conventional full-text retrieval interface of the ECHO host.

However, in this domain the users typically do not search for documents, but for relevant *information objects*, e.g. research programs or projects which fulfill certain conditions, together with factual data, e.g. addresses, numerical data, e.g. project duration, and textual passages, e.g.

3. The corresponding databases are provided as CORDIS databases, e.g. *RTD-Programs, RTD-Projects, RTD-Acronyms, RTD-Comdocuments, RTD-Publications, RTD-Results, RTD-Partners* and so on.

a project's objectives. Given the functionality of a full-text retrieval system, the users must switch between the databases, and combine data sets retrieved from different isolated databases in order to obtain the complete information. From the users' perspective, however, it seems to be highly desirable to be able to formulate a complex query covering all relevant aspects of the information objects of interest, no matter in which particular RTD-databases the data is stored. Of course, this query formulation has to be assisted, e.g. via a form-based interface. Another prerequisite to a system capable to process such queries directly is a conceptual data model which defines the object classes of the domain of interest and their relationships.

Figure 2 provides a graphical overview of the conceptual model which serves as database schema and can be realized using VML. The conceptual model can be verbalized as follows: The EC research activities are grouped by **program**s (such as IMPACT, ESPRIT 2, etc.); each program contains a group of **project**s as *members*. Different EC **commission service**s are *responsible for* different programs. Programs and projects have their contractors, *i.e. primary contractors and member contractors*, called **organization**s *located in* some **countrie**s and some **citie**s. There are some **person**s in the role of *contact* person of program, project, or commission service. The **publication**s, including **report**s, **article**s, and **conference paper**s, are *issued by* organizations, programs and projects, with some person(s) as ***author(s)***. The programs, projects and publications can be classified to some ***subjects***.

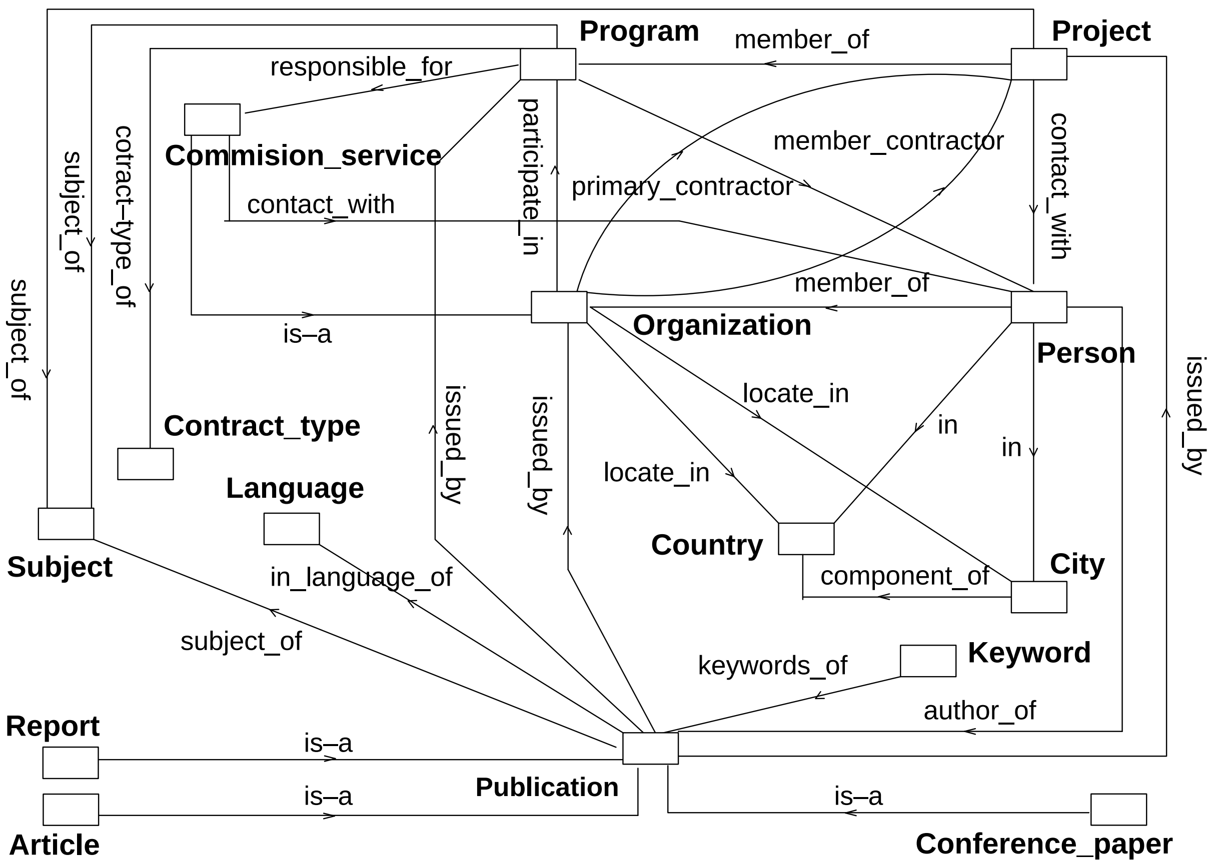

Figure 2: Conceptual Model

4 Retrieval Strategies and Flexible Dialogue Control

Information-seeking processes are characterized by the fact that information needs are not predefined distinctively, and that there is no objective underlying task-structure which strictly regulates the order of navigational steps. The information seeker may change his intentions or the thematic focus during the interaction. An intelligent system should be capable to cope with such topic shifts and changing user strategies. At the same time it is important to facilitate the user's construction of cognitive coherence by adequate guidance. Thus, we propose to establish a dialogue control which can adapt to some extent the system's behavior to changing dialogue goals and information needs of the user. Initially the user has only vague ideas about his information need [cf. Belkin & Marchetti 1990] and how to explore the information space. For guiding the user effectively, the system relies on **dialogue plans**, which are dynamically adapted to the ongoing dialogue, and it captures the focus of the dialogue by means of **object-oriented perspectives**.

At any time the user must have the opportunity to control the dialogue flexibly in order to alter the steps of the current dialogue plan. He can do this by directly manipulating graphical objects, e.g. special dialogue-icons [for details cf. Stein, Thiel & Tißen 1992]. However, most of these graphical actions are interpreted as "communicative acts" that express dialogue goals [cf. e.g. Maybury 1990, Stein & Thiel 1993]. The formal (illocutionary) model underlying the exchange of dialogue acts in MERIT has thoroughly been discussed elsewhere [cf. Sitter & Stein 1992].

Dialogue Plans: In order to achieve a flexible and adaptive management of dialogue strategies we employ a ***Ca****se-based* ***Di****alogue Manager,* CADI [cf. Tißen 1991]. CADI can be seen as a special adaptation of a case-based reasoner for guiding information-seeking dialogues. The basic idea of a case-based reasoner is to use the experience from old solutions for the generation of dialogues plans. Kolodner and Simpson describe the role of experience in problem solving: "*Individual experiences act as exemplars upon which to base later decisions. Analogies to previous cases guide and focus later decision making.*" [1986, p. 99 f.] In MERIT, previous cases are used to *guide* the user through the retrieval process and to *focus* the dialogue and the presentations considering his goals. In our application, the notion of 'cases' refers to protocols of prior executions of dialogue plans, which are represented by frame-like data structures. Dialogue plans structure an information-seeking dialogue according to principles of topical coherence. They model the thematic progression and serve as a means for reducing the relevant information space.

In accordance with this approach we offer the user a selection of basic dialogue plans at the beginning of the retrieval session that can get modified during the dialogue. The library of dialogue plans can be seen as a set of prototypical cases stored in the past. In the MERIT domain of research projects, programs, and publications there are for example cases like: "Projects about tutoring systems in the current RACE program", "Overview about ESPRIT projects concerning intelligent interfaces", "Looking for university partners with project experience in text-generation". Whenever the user wants the system to memorize the current dialogue path with all its modifications, he can use the CADI system to compose and store a new plan and hereby augment the library of dialogue plans.

Object Perspectives: By dialogue goals we do not refer to any real-world goals of the information seeker, but rather to the hyperthema of the dialogue or the expected scope of information, e.g. the amount of data or the granularity of information. Nevertheless, real-world goals or interests influence the thematic scope to a large extent. Therefore we use the concept of *"object perspectives"* [cf. Tißen 1991]. An object perspective can be described as a set of assumptions and expectations associated with a specific domain object which is in the focus of the dialogue (e.g. projects), and a closely related problem-solving task (e.g. looking for project partners).

From the user's point of view a perspective describes the relevant information space in a given dialogue situation. This includes the possibilities to navigate to specific information objects within this space. The system's definition of a perspective marks the relevant information space of the database – including relations of the central information object to other objects and their attributes. It also defines a set of related perspectives. Moving to one of these perspectives maintains the local topical coherence of the dialogue. Thus, our notion of perspectives combines *semantic aspects*, the 'view' onto the represented domain-objects, with temporal or *topical aspects* (the currently selected perspective determines possible navigation paths and subsequent topics).

A hierarchy of perspectives together with a set of functions enables the system to adapt a query or presentation step directly to a new situation. For instance, the perspective concerning project information, focusing on project partners, is a sub-perspective of organizational aspects of project information. In this perspective only a small subset of database attributes is relevant to the query. If the dialogue is in a presentation state, switching from the "organizational perspective" to the more restricted "organizational-partner perspective", only the generation of a new presentation of the data, which have already been retrieved from the database, is required. This is a very simple example of a modification of a dialogue plan influencing only the amount of presented information without side effects. However, most modifications in a dialogue cause side effects to be handled by the system. For example, it may be necessary to send a new internal query to the database, or to modify the proposed subsequent dialogue steps automatically (insert, delete, or replace steps which propagate changed parameters).

The *knowledge base for perspectives* is closely related to the domain knowledge base. This implies that the perspectives have to be adapted when the conceptual domain knowledge changes. The dynamic adaptation of dialogue plans during a dialogue session is initiated by the user and controlled by modification rules. These rules use the knowledge base of object perspectives. For each executed dialogue step the dialogue manager generates a state in the *dialogue history*, representing the step with all dynamic modifications, together with the input given by the user. This explicit representation is the basis for generating a new case to increase the library of good sample cases.

Due to the close relation of object perspectives to the domain knowledge base, which is a database whose schema is described in the VODAK modeling language VML, they are preferably expressed in VML. We give the sketch of the structural part of an implementation of perspectives in VML. Each perspective is an instance of the class Perspective.

```
CLASS Perspective
  INSTTYPE
    PROPERTIES
    main-concept: Class
    zoom-in:  {Perspective};
    zoom-out: {Perspective};
              // these two properties establish the hierarchy
    shift-to: {Perspective};
              // related perspectives
    query-attributes: {Property};
    presentation-attributes: {Property};
  END
END
```

In this class definition we make use of several typical features of VML. We have defined the structural part (properties) of the type definition for the instances of the class *Perspective*. This is signalled by the keyword INSTTYPE. The properties refer to instances or sets of instances of other classes. Such direct references are possible since each instance has a unique object identifier. The class definition of Perspective makes use of system-defined classes like Class and Property, which contain information about all classes and properties of a given schema. Note that each property "knows" to which class it belongs. The classes *Class* and *Property* are system-defined as they are already needed by the VODAK data dictionary.

It is clear that, in order to avoid a break in the system architecture and to avoid the introduction of complicated interfaces the other cognitive concepts, like dialogue plans, are preferably modeled within VML either. In this paper, however, we do not go into the details of this.

5 Clarification of Information Needs

5.1 Support of Query Formulation

Object perspectives [cf. e.g. Tou et al. 1982, McCoy 1986] describe subsets of the database relevant in a specific context, for the query component as well as for the presentation component. A hierarchical representation of perspectives together with a set of functions enables the system to adapt a query or presentation step directly to a new situation. For instance, the perspective concerning project information, focusing on project partners, is a sub-perspective of organizational aspects of project information. In this perspective only a small subset of database attributes are relevant to the formulation of a query.

After the case selection, the user is presented a query form [cf. e.g. McAlpine & Ingwersen 1989] listing attributes of the selected perspective (see figure 3). The user reduces the amount of relevant instances of the current concept class. This is achieved by stating *attribute restrictions*. Each line of the query form sheet represents a restriction consisting of a comparison operator selected from a menu that provides an attribute-specific choice, and a constant to which attribute values of instances in the knowledge base are to be compared. If all restrictions are fulfilled for a given instance, then this instance will belong to the set of responses.

Figure 3 shows the query form for the thematic perspective of projects, i.e. it does not include organizational attributes like project partners, contact persons, location, dates, financial data, etc. By the filled-in query all projects will be retrieved which contain the terms *"author"* or *"document"* or *"text"* in their general project information and at the same time the terms *"natural language"* or *"multi-media"* or *"hypermedia"* in their project objectives.

Figure 3: A sample query form

5.2 Support of Free Text Search

Query forms may contain multiple free text fields, for example a description of projects. The user may require that a text field of a retrieved instance must or must not contain a given search term.

If search terms are given in different lines of a multiple line field they are implicitly connected by the Boolean *OR*.

Originally, the user wants to do a *content*-based search, but instead he has to deal with *terms* which just are the surface representation of the content. In order to improve the recall of his query the user in general has to add several synonyms or similar concepts to each search concept. Instead of leaving the task of finding these additional search concepts to the user, we employ a module called *Knowledge Explorer* (*KX*) which has conceptual knowledge of the database domain and suggests such supplementary concepts.

This conceptual knowledge is stored as a fuzzy association network [cf. Kracker 1992]. The meaning of a concept is solely determined by its relationships to other concepts. There are four types of such relationships: A *positive association* connects concepts which are semantically similar or often used in the same context, a *negative association* is used to express some kind of opposition, a *generalization* links one concept to another which is more general in a semantic or partitive sense, and the *specialization* is the inverse of the generalization relationship (cf. figure 4).

		association		categorization	
		P	N	G	S
association	P	T_3	T_3	T_2	T_2
	N	T_3	—	T_2	T_2
categorization	G	T_2	T_2	T_3	T_1
	S	T_2	T_2	T_1	T_3

P: positive association
N: negative association
G: generalization
S: specialization

$T_1(a, b) = \max(0, a + b - 1)$
$T_2(a, b) = ab$
$T_3(a, b) = \min(a, b)$.

Figure 4: T-norm defining a calculus of similarity

Each pair in a relationship has a value out of [0, 1] assigned which is interpreted as the *strength* of the relationship. A positive association with a strength close to 1, for example, identifies a relation between two very similar concepts. We assume the positive association, the generalization and the specialization to be transitive relationships.

We can use the inherent transitivity of these relationships to compute relationships between two concepts even if they are not *directly* connected. There are many (in fact, an unlimited number of) T-norm functions $[0, 1] \times [0, 1] \rightarrow [0, 1]$ suitable for combining transitive relationships. They differ in the strength of the resulting relationship. Opposed to [Bezdek, Gautam & Hunag 1986], where a single T-norm function is used for all types of relationships, we employ three T-norm functions (see figure 4).[3]

The type of a derived relationship must also be specified. Again, this type is a priori determined depending on the types of the combined relationships. An extensive discussion of the inference process can be found in [Kracker 1992]. On demand the system computes all relationships that are stronger than a heuristically determined cutoff value for a given initial concept . This method is sufficient for our application described above and is fast enought for interactive use.

To give an example of how the user might get support in the domain *multimedia/ hypermedia* we again refer to figure 4. To increase the recall of that query, the user has to include additional terms in the free text field *project objectives* which he can disjunctively combine with the already specified search terms *author, document,* and *text*. Upon request, the system suggests a ranked list of semantically similar terms in a pull-down menu. Each item the user selects will be automatically inserted into the query form, e.g. *media* or *hypertext*.

6 Query Processing in VODAK

Multimedia data needs search strategies which are very different from those on conventional data. As an example we have considered free text search. Extendibility of the database primitives in VODAK allows to integrate the corresponding algorithms and data structures easily. The query language must then allow to use the new primitives without restrictions, which is obtained by allowing arbitrary method calls in query statements.

In the following we sketch how VODAK can support an extendible search strategy in the case of free text search. Assume a class is given which contains as a property a piece of text. We model the text not directly as String in the class Info but instead refer to an instance of the class Info_Text_Class which represents the text.

```
CLASS Info
  INSTTYPE
    PROPERTIES
    info_text: Info_Text_Class;
    .... (other properties)
END
```

As we want to make available a certain search algorithm on text, the class Info_Text_Class itself is an instance of a metaclass which provides these corresponding methods in its so called instance-instance-type (INSTINSTTYPE) via inheritance. In other words any instance of a class with metaclass Text_Metaclass will inherit the interface defined in the instance-instance-type of the metaclass. Note that the instances of the metaclass are classes themselves.

3. It is easy to show that $T1(a, b) \leq T2(a, b) \leq T3(a, b)$ holds. T1 is the most conservative function returning the smallest values. T3, on the contrary, makes the most optimistic conclusions and infers the strongest relationships. T2 assumes the arguments to be independent and treats them like independent probabilities. Depending on the types of the relationships to be combined the most suitable T-norm can be used.

```
CLASS Info_Text_Class METACLASS Text_Metaclass END

CLASS Text_Metaclass METACLASS Metaclass
  INSTINSTTYPE
    PROPERTIES
      text: BYTESTRING;
    METHODS
      search(s: STRING);
         // searches in text for the string s
END
```

The property text now contains the actual information. The BYTESTRING datatype is one of the new primitive datatypes provided by VODAK in order to support multimedia datatypes. Other primitive datatypes will be AUDIO and VIDEO. The method search can now be sent to any instance of a class whose metaclass is Text_Metaclass, e.g. to any instance of Info_Text_Class. We also could provide through the metaclass Text_Metaclass persistent indexing structures which can be used to speed up search.

In a VQL query we can use the method search, e.g. as in the following query:[4]

ACCESS i FROM i IN Info WHERE i.info_text–>search(”string”).

This query returns the set of instances of Info which contain in the text represented by *info_text* the string “string”.

Another feature of VQL that is important to support the interface techniques introduced earlier is the possibility of arbitrary nesting of queries and reuse of query results. Thus an iterative refinement as it can appear in the dialogue is possible.

A central part of query processing is concerned with query optimization. As we are facing in multimedia database management systems the necessity to support also search algorithms and access structures which can not be foreseen at the time the query optimizer is designed it must be extendible in the same way as the rest of the system is. This is a current focus of research in the development of the VQL query optimizer.

7 Presentation of Retrieval Results in MERIT

The mixture of textual, factual, and organizational data which is retrieved from the database as a response to a query has to be presented adequately. Our approach to the presentation of retrieval results is motivated by van Rijsbergen’s logical interpretation of relevance [van Rijsbergen 1989]. Given a query Q and a set of information items D we have to determine the probability P of the fact that D entails Q: P(D ->Q). A high probability P(D ->Q) indicates that D is relevant with respect to Q.

4. We use the keyword ACCESS since we allow arbitrary method calls in the query. Therefore a distinction between SELECT (no updates) and UPDATE is no longer meaningful [cf. Fischer 1992].

The purpose of presenting information objects as a response to a query in a retrieval dialogue is twofold: First, to inform the user about the retrieved data, second, to convince him that the system's reaction is relevant to the query, and, as a consequence, to the user's information need. As the data are organized as a complex structure additional information must be conveyed. As an example, the fact that a given part of the visualization pertains to a certain concept or search term in the query will be very useful for the user in her/his interpretation of the query result. In order to support the user's inferencing on presented data we employ cognitively motivated *presentation forms*.

For a discussion of the cognitive background of the visualization design we refer to [Kerner & Thiel 1991], in the following, however, we will concentrate on relevance assessment with regard to information presentation as a follow-up of querying a database.

7.1 Providing Survey Information

The *survey presentation form* reveals the relationships between the retrieved data items. In figure 5 the acronyms of retrieved projects are shown together with the subject index codes. These data items have been retrieved from the database in accordance to the query (cf. figure 3), and,

Figure 5: Survey presentation form

subsequently, transformed into an object-oriented representation. This allows the user to change the contents of the visualized presentation form. He may switch to a similar graph (or a table) depicting the same project acronyms, but instead of the index terms the prime contractor companies are shown. Note that this leads to a different perspective which overlays the one originally suggested by the chosen plan.

7.2 Presentation of Detailed Information and Additional Visual Information

If further attributes or details have to be presented, the survey presentation form fails to meet this information need. The presentation form shown in figure 6 responds to the detail-oriented information needs. In our example, the available attributes of projects like acronym, title, program, general information, objectives, prime contractor, contact person, etc. are on display. The current perspective determines which attributes are currently presented to the user. One project and its attached information items are visualized on the screen, other projects belonging to the same retrieval result are available on demand. The user selects the other projects by pressing the *next* or *previous* button.

Figure 6: Maps and photo integrated into the detail presentation form

In MERIT a subset of the project data was augmented by additional scanned-in or programmed images encoded in EPS (Encapsulated PostScript). Also audio annotations can be added in order to provide further *context information*. We will illustrate this issue giving two examples: photos and maps. Contextual information supports the associations of the user. In our example, the user can click on the contact-person button in order to get additional information about the person, i.e. a photo (cf. figure 6). When the user is interested in the location of a project partner he may click on a button in order to get the geographical information. In our example he gets first a map of Europe with the country (Germany) highlighted. Then he may click on the highlighted area to get a more detailed map of Germany displaying the city of the prime contractor (Darmstadt).

7.3 Realization of the Visualization

The presentation of multimedia data stored in VODAK cannot be considered separately from the database management system. In contrast to conventional alphanumeric data, for these data special mechanisms are provided by the database management system which allow to deal with the huge amounts of data which have to be represented. Especially for continuous data these mechanisms are crucial, as only they allow to bring the right piece of data, e.g. a video frame, at the right time to the right device, e.g. a window. Also the user interaction, e.g. mouse clicks, have to be part of this process in order to support browsing and control of the presentation by the user. These issues are currently investigated in the AMOS project [for more details see Rakow et al. 1993, Aberer & Klas 1992, Klas 1992].

8 Related Work

The approach presented in this paper is interdisciplinary in nature, since it involves not only different branches of computer science, but also builds upon work in information science, psychology and linguistics.

In particular, the design of MERIT addressed problems of cooperative graphical or multimodal interfaces [cf. Hayes 1987, Neal & Shapiro 1989; Cohen et al. 1989; Allgayer et al. 1989; Arens & Hovy 1990; Moore & Swartout 1990; Feiner & McKeown 1990; Bandyopadhyay 1990; Stock 1991]. Another body of related work aims at the plan-based or task-driven automatic generation of graphics [e.g. Wahlster et al. 1991; Oei et al. 1992] and the case-based composition of video presentations [MacNeil 1991]. However, these approaches are not interactive in the sense that the end user may influence the generation process, while in MERIT the sequence of visualizations of retrieved data items results from the cooperation between user and system.

The user of MERIT engages in a *visual dialogue*, in which the interaction, although realized by graphical means, complies with conversational patterns of exchange [cf. Stein & Thiel 1992]. Similar approaches to flexible graphical interaction based on the *conversational metaphor* [cf. Hutchins 1989, Reichman 1986, 1989, Thiel 1990] treat user inputs such as mouse clicks, menu selections, etc. not as invocations of methods that can be executed without regarding the dialogue context, but as dialogue acts expressing a discourse goal of the user.

The object-oriented database management system VODAK will serve as a platform for the MERIT Interface system. Other object-oriented database systems supporting multimedia data like ORION [cf. Woelk, Kim & Luther 1986] as well as experiences made with hypermedia systems like *NoteCards* [cf. Halasz, Moran & Trigg 1987, Halasz 1988], *Neptune* [cf. Delisle, N. & Schwartz 1986], *Intermedia* [cf. Garret, Smith & Meyrowitz 1986, Meyrowitz 1986], and *KMS* [cf. Akscyn, McCracken & Yoder 1988] may also influence the overall design of multimedia information systems with respect to their functionality. An object-oriented database system with a visual query interface is O_2 [Bancilhon, Cluet & Delobel 1989] but which provides only a presentation component, and OQL [Lam, Chen, Ty, Qiu, & Su 1990] is an interesting approach to a graphical interface for an object-oriented query language.

9 Conclusions

MERIT supports the user by offering a selection of sample retrieval strategies (*cases*) which are modifiable to meet situative requirements. Based on a selected *case* the system adopts an appropriate *perspective* while accessing the database. Thus, irrelevant attributes of the objects of interest can be neglected. The perspective enables the system to offer situation-dependent query forms. The process of selecting search terms is assisted by a semantic component proposing additional search terms that are derived from an initially given one by associative reasoning. MERIT generates graphical presentations of retrieval results which support – in accordance to the current case – a survey or detail-oriented reception of the retrieved data, thus providing visual cues for the user's relevance assessment. Additionally, the system employs interactive maps to display geographical data, and can provide scanned-in documents.

An extendible database system like VODAK is the ideal underlying database system. Due to its specific object-oriented features it allows to integrate a support for multimedia data, to integrate heterogeneous resources, and its query language is extendible for multimedia search strategies. Care has to be taken to provide a seamless interface and to exploit fully the support for presentation of multimedia data as provided by the database management system.

References

Aberer, K. & Klas, W.: *The Impact of Multimedia Data on Database Management Systems.* Technical Report TR-92-065, International Computer Science Institute (ICSI), Berkeley, CA, 1992.

Allgayer, J., Harbusch, K., Kobsa, A., Reddig, C., Reithinger, N. & Schmauks, D.: XTRA: A Natural Language Access System to Expert Systems. *Int. Journal of Man-Machine Studies, Vol. 31(2),* 1989, pp. 161-195.

Arens, Y. & Hovy, E.H.: How to Describe What? Towards a Theory of Modality Utilization. In: *CogSci '90, Proc. of the 12th Annual Conference of the Cognitive Science Society in Cambridge, MA.* Hillsdale, NJ: Erlbaum, 1990, pp. 487-94.

Akscyn, R., McCracken, D.L. & Yoder, E.: A Distributed Hypertext for Sharing Knowledge in Organizations. *Communications of the ACM, Vol. 31(7),* July 1988, pp. 820-835.

Bancilhon,F., Cluet S. & Delobel C.: A Query Language for the O_2 Object-Oriented Database System. In: *Proc. of the DBPL, Salishan Lodge, Oregon* 1989.

Bezdek, J.C., Gautam, B. & Hunag, Li-Ya: Transitive Closures of Fuzzy Thesauri for Information-Retrieval Systems. *Int. Journal of Man-Machine Studies, Vol. 25,* 1986, pp. 343-356.

Belkin, N., Cool, C., Stein, A. & Thiel, U.: Scripts for Information Seeking Dialogues. Paper presented at: *AAAI Spring Symposium on Case-Based Reasoning and Information Retrieval, Stanford University, CA, March 23-25, 1993.*

Belkin, N.J. & Marchetti, P.G.: Determining the Functionality and Features of an Intelligent Interface to an Information Retrieval System. In: Vidick, J.-L. (ed.): *SIGIR '90, Proc. of the 13th Int. Conference on Research and Development in Information Retrieval*, Brussels, Belgium, 1990, pp. 151-178.

Bandyopadhyay, S.: *Towards an Understanding of Coherence in Multi-Modal Discourse*, Technical Memo TM-90-01, Deutsches Forschungsinstitut für Künstliche Intelligenz, Saarbrücken, 1990.

Cohen, P. R., Dalrymple, M., Moran, D., Pereira, F., Sullivan, J., Gargan Jr., R., Schlossberg, J. & Tyler, S.: Synergistic Use of Direct Manipulation and Natural Language. In: Bice, K. & Lewis, C. (eds.): *Proc. of CHI '89, Austin, TX.* New York: ACM, 1989, pp. 227-233.

Duchêne, H., Kaul, M. & Turau, V.: VODAK Kernel Data Model. In: Dittrich, K.R. (ed.): *Advances in Object-Oriented Database Systems, Proc. of the 2nd Int. Workshop on Object-Oriented Database Systems, Bad Münster am Stein-Ebernburg, FRG.* Lecture Notes in Computer Science, No. 334. Berlin: Springer, 1988.

Delisle, N. & Schwartz, M.: Neptune: A Hypertext System for CAD Applications. In: *Proc. of ACM SIGMOD '86, Washington D.C., May 28-30.* New York: ACM, 1986, pp. 132-142.

Feiner, S.K. & McKeown, K.R.: Coordinating Text and Graphics in Explanation Generation. In: *AAAI '90, Proc. of the 8th National Conference on Artificial Intelligence, Vol. I.* Menlo Park et al.: AAAI Press / The MIT Press, 1990, pp. 442-449.

Fischer, G.: Updates in Object-Oriented Database Systems Caused by Method Calls in Queries. In: *Proc. of the 3rd ERCIM Database Research Group Workshop on Updates and Constraints Handling in Advanced Database Systems, Pisa, Italy, September 28-30, 1992.*

Fischer, G. & Aberer, K.: VODAK Query Language VQL 1.0. *Internal Report GMD-IPSI, Feb. 1993.*

Garret, L.N., Smith, K.E. & Meyrowitz, N.: Intermedia: Issues, Strategies, and Tactics in the Design of a Hypermedia Document System. In: *CSCW '86, Proc. of the Conference on Computer-Supported Cooperative Work, Austin, Texas, Dec 3-5,* 1986, pp. 163-174.

Halasz, F.G., Moran, T.P & Trigg, R.H.: NoteCards in a Nutshell. In: *CHI + GI '87, Proc. of the 1987 ACM Conference of Human Factors in Computer Systems, Toronto, Ontario, Apr 5-9.* 1987, pp. 45-52.

Halasz, F.G., Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems, *Communications of the ACM, Vol. 31(7),* July 1988.

Hayes, P.J.: Steps towards Integrating Natural Language and Graphical Interaction for Knowledge-Based Systems. In: du Boulay, B., Hogg, D., Steels, L. (eds.): *Proc. of the ECAI '86: Advances in Artificial Intelligence – II.* Amsterdam : North-Holland, 1987, pp. 543-552.

Hutchins, E.: Metaphors for Interface Design. In: Taylor, M.M., et al. (eds.): *The Structure of Multimodal Dialogue*. Amsterdam: North-Holland, 1989, pp. 11-28.

Kerner, A. & Thiel, U.: Graphical Support for Users' Inferences within Retrieval Dialogues. In: *Proc. of the IEEE Workshop on Visual Languages, Kobe, Japan.* Washington: IEEE Computer Society Press, 1991, pp. 211-216.

Klas, W.: *A Metaclass System for Open Object-Oriented Data Models.* PhD Thesis, Technical University of Vienna, Austria, January 1990.

Klas, W.: Tailoring an Object-Oriented Database System to Integrate External Multimedia Devices. In: *Proc. of the 1992 Workshop on Heterogeneous Databases & Semantic Interoperability, Boulder, CO, Feb. 10-12, 1992.*

Klas W. et al.: *VML – The VODAK Model Language Version 2.2,* Technical Report, GMD-IPSI, August 1992.

Kolodner, J.L. & Simpson R.L.: Problem Solving and Dynamic Memory. In: Kolodner, J.L. & Riesbeck, C.K. (eds.): *Experience, Memory, and Reasoning.* Hillsdale, NJ, 1986, pp. 99-114.

Kracker, M.: A Fuzzy Concept Network Model and Its Applications. In: *Proc. of the FUZZ-IEEE ’92, San Diego, CA, March 1992,* pp. 760-768.

Lam, H., Chen, H.M., Ty, F.S., Qiu, J. & Su, S.Y.W.: A Graphical Interface for an Object-Oriented Query Language. In: *Proc. of the 14th. Int. Computer Software & Application Conf., Chicago, Ill., Oct. 31 – Nov. 2, 1990,* pp. 231-237.

MacNeil, R.: Generating Multimedia Presentations Automatically using TYRO, the Constraint, Case-Based Designer’s Apprentice. In: *Proc. of the IEEE Workshop on Visual Languages, Kobe, Japan.* Washington: IEEE Computer Society Press, 1991, pp. 74-79.

Maybury, M.T.: Planning Multimedia Explanations Using Communicative Acts. In: *AAAI ’91, Proc. of the 9th National Conference on Artificial Intelligence, Anaheim, CA, July 1991*. Menlo Park: AAAI Press/ The MIT Press, 1991, pp. 61-66.

McAlpine, G. & Ingwersen, P.: Integrated Information Retrieval in a Knowledge Worker Support System. In: Belkin, N.J. & van Rijsbergen, C.J. (eds.): *SIGIR ’89, Proc. of the 12th Int. Conference on Research and Development in Information Retrieval.* Cambridge, MA, 1989, pp. 48-57.

McCoy, K.F.: The ROMPER System: Responding to Object-Related Misconceptions Using Perspective. In: *Proc. of the 24th. Annual Meeting of the Association for Computational Linguistics,* New York, 1986.

Meyrowitz, N: Intermedia: The Architecture and Construction of an Object-oriented Hypermedia System and Applications Framework. In: *OOPSLA ’86, Proc. of the Conference on Object-oriented Programming Systems, Languages, and Applications, Portland, Oregon, Sept. 29 – Oct 2.* ACM SIGPLAN No. 21, 11. 1986.

Moore, J.D. & Swartout, J.R.: Pointing: A Way Toward Explanation Dialogue. In: *AAAI ’90, Proc. of the 8th National Conference on Artificial Intelligence, Vol. I.* Menlo Park, CA: AAAI Press / The MIT Press, 1990, pp. 457-464.

Neal, J.G. & Shapiro S.C.: Intelligent Multi-Media Interface Technology. In: Sullivan, J.W. & Tyler, S.W. (eds.): *Proc. of the Workshop on Architectures for Intelligent Interfaces: Elements and Prototypes.* ACM/ Addison-Wesley, 1989, pp. 69-91.

Norman, D.A. & Draper, S.W. (eds.): *User Centered System Design. New Perspectives on Human-Computer Interaction.* Hillsdale, NJ: Erlbaum, 1986.

Oei, S., Smit, R., Schreinemakers, J., Marinos, L. & Sirks, J.: The Presentation Manager, A Method for Task-Driven Concept Presentation. In: Neumann, B. (ed): *Proc. of the ECAI ’92,* Chichester et al.: John Wiley & Sons, pp. 774-775.

Rakow, T.C., Löhr, M., Moser, F. & Neuhold, E.J.: Einsatz von objektorientierten Datenbanksystemen für Multimedia-Anwendungen. To be published in: *it + ti: Special Issue on “Multimedia”*. Oldenbourg, 1993 (forthcoming).

Reichman, R.: Communication Paradigms for a Window System. In: Norman, D.A. & Draper, S.W. (eds.): *User Centered System Design: New Perspectives on Human-Computer Interaction.* Hillsdale, NJ: Erlbaum, 1986, pp. 285-313.

Reichman, R.: Integrated Interfaces Based on a Theory of Context and Goal Tracking. In: Taylor, M.M., et al. (eds.): *The Structure of Multimodal Dialogue*. Amsterdam: North-Holland, 1989, pp. 209-228.

van Rijsbergen, C.J.: Towards an Information Logic. In: N.J. Belkin & C.J. van Rijsbergen (eds)*: SIGIR '89, Proc. of the 12th Int. Conference on Research and Development in Information Retrieval*, Cambridge, MA, 1989, pp. 77-86.

Sitter, S. & Stein, A.: Modeling the Illocutionary Aspects of Information-Seeking Dialogues. *Information Processing & Management, Vol. 28(2),* 1992, pp. 165-180.

Stein, A. & Thiel, U.: A Conversational Model of Multimodal Interaction in Information Systems. In: *AAAI '93, Proc. of the 11th National Conference on Artificial Intelligence, Washington DC, USA, July 11-16, 1993*. Menlo Park: AAAI Press/ The MIT Press, 1993 (in press). Extended version also available in: *Arbeitspapiere der GMD No 741,* Sankt Augustin: GMD, March 1993.

Stein, A., Thiel, U. & Tißen, A.: Knowledge-Based Control of Visual Dialogues in Information Systems. In: *AVI '92, Proc. of the 1st Int. Workshop on Advanced Visual Interfaces, Rome, Italy, May 27-29.* Singapore: World Scientific-Press, 1992, pp. 138-155. Also available as: *Arbeitspapiere der GMD, No 662,* Sankt Augustin: GMD, July 1992.

Stock, O.: Natural Language and the Exploration of an Information Space: The ALFresco Interactive System. In: *Proc. of the IJCAI '91.* Sydney, 1991, pp. 972-978.

Thiel, U.: *Conversational Graphical Interaction with Information Systems. An Approach Based on Speechact Theory* (PhD-Dissertation, in German). Konstanz, Germany, Universität Konstanz, 1990.

Tißen, A.: A Case-Based Architecture for a Dialogue Manager for Information-Seeking Processes. In: Bookstein, A., et al. (eds.): *SIGIR '91, Proc. of the 14th Int. Conference on Research and Development in Information Retrieval*, Chicago, IL, 1991, pp. 152-161.

Tou, F., Williams, M., Fikes, R., Henderson, D.A. & Malone, T.: RABBIT: An Intelligent Database Assistant. In: *AAAI '82, Proc. of the National Conference on Artificial Intelligence.*, 1982, pp. 314-318.

Turau, V. & Rakow, T.C.: A Schema Partition for Multimedia Database Management Systems. In: *Arbeitspapiere der GMD, No. 729.* Sankt Augustin: GMD, 1993.

Wahlster, W., André, E., Graf, W. & Rist, T.: Designing Illustrated Texts: How Language Production is Influenced by the Graphics Generation. In: *Proc. of the 5th Conference of the European Chapter of the Association for Computational Linguistics*, 1991.

Woelk, Darrel, Kim, W. & Luther, W.: An Object-Oriented Approach to Multimedia Databases; *ACM SIGMOD Record 1986*, pp. 311-325, ACM, 1986.