Nicholas J. Belkin, Colleen Cool,
Adelheit Stein & Ulrich Thiel

Cases, Scripts, and Information-Seeking Strategies: On the Design of Interactive Information Retrieval Systems

November 1994

Prof. Nicholas J. Belkin
Colleen Cool
School of Communication, Information and Library Studies
Rutgers University
4 Huntington Street
New Brunswick, NJ 08903
U.S.A.
e-mail: nick@belkin.rutgers.edu, cool@zodiac.rutgers.edu

Dr. Adelheit Stein,
Dr. Ulrich Thiel
GMD-IPSI
German National Research Center for Computer Science –
Integrated Publication and Information Systems Institute
Dolivostr. 15
D-64293 Darmstadt
Germany
e-mail: stein@darmstadt.gmd.de, thiel@darmstadt.gmd.de

Abstract

The support of effective interaction of the user with the other components of the system is a central problem for information retrieval. In this paper, we present a theory of such interactions taking place within a space of information-seeking strategies, and discuss how such a concept can be used to design for effective interaction. In particular, we propose a model of information retrieval system design based on the ideas of: a multi-dimensional space of information-seeking strategies; dialogue structures for information seeking; cases of specific information-seeking dialogues; and, scripts as distinguished prototypical cases. We demonstrate the use of this model by discussing in some detail the MERIT system, a prototype information retrieval system which incorporates these design principles.

To appear in:
Expert Systems with Applications, Vol 9, 1995

Also available as:
Arbeitspapiere der GMD No. 875, Sankt Augustin: GMD, Nov. 1994
and
SCILS Research Report No. 94-44, School of Communication, Information and Library Studies, Rutgers University, New Brunswick NJ, 1994

1 Introduction: Information Retrieval as Interaction

A central problem for information retrieval (IR) is designing for effective interaction. IR has traditionally been concerned primarily with the processes of *representation* of texts and queries, and of *comparison* of these representations. The former process has typically been exemplified by indexing or classification, the latter by any of a variety of retrieval techniques. Although these traditional issues are clearly still important, it is becoming increasingly evident that IR is an inherently interactive process, from a variety of points of view. This means, in particular, that supporting, and taking advantage of the interaction of the user with the other components of the IR system is crucial for effective IR system design (see, for example, Belkin, Marchetti & Cool, 1993). Although there is a substantial tradition of interest in this issue in IR (early concern with user interface design, e.g., Walker 1971, the long-standing concern with, and research program in relevance feedback, cf. Salton & Buckley 1990), only fairly recently has it been suggested that interaction is a central, and organizing process in IR (Oddy 1977, Belkin 1982, Belkin & Vickery 1985, Croft & Thompson 1987, Bates 1989, Thiel 1990, Ingwersen 1992 are all examples of this point of view).

Some common features of this viewpoint are that: asking people to specify their information ‘needs’ is in general unrealistic; people’s conceptions of their information problems change through their interactions with the IR system; there are different kinds of information problems, for which different kinds of interactions might be appropriate; the rest of the system can better represent the person’s information problem through interaction; and, entire information-seeking episodes can consist of a variety of types of interaction.

Explicitly, the interactive approach to IR has led to a focus on the user-oriented activities of query formulation and reformulation, and inspection and judgement of retrieved items, as processes of central interest to IR. We agree with this view. As such processes are inherently interactive, we therefore understand IR itself *as* interaction, and, in particular, as human-computer interaction. This notion of IR as interaction has two aspects which can be cast into the following questions:

- Given a certain situation in the dialogue session, what are the relevant documents in the database?
- How can the interaction be structured in a way that supports the user in inspection, judgement, and query formulation?

A major problem with the interactive approach to IR is being able to gain an understanding of the nature of the interaction itself, and then to move from such understanding to the specification of a system design and structure which supports and enhances it. This is the major theme of this paper. We take two, related approaches to resolving this issue. The first is to consider IR as interactive information seeking; we deal with this issue in some detail in the section 1.1. The second is to use concepts of case-based reasoning (CBR) (cf. Riesbeck & Schank 1989) as means to implement interactive IR system design.

The case-based reasoning approach has some obvious intuitive relevance to IR system design. We can consider the record of a person’s interaction in an IR system as a *case* of problem solving activity, and we can understand that a specific pattern of information-seeking behavior might be usefully engaged in by a person at different times, or by different people. Thus, such cases could be reused in new circumstances. This promises to be useful to us in designing interactive IR systems, since it suggests ways to understand and use structures of interaction. Unfortunately, there are significant problems in using CBR approaches to system design. One of the most important of these is knowing when a given stored case is relevant to a current situation. This problem can take several forms; most commonly, it is perceived as a problem of indexing the cases, and retrieving the relevant cases at appropriate times, for use in supporting the current situation.

In this paper we propose that the concept of a space of information-seeking strategies, combined with a view of IR interaction as dialogue, can lead to possible solutions to both the problem of understanding interaction, and the problem of representing, organizing and retrieving cases of information interaction.

1.1 Information-Seeking Strategies

Elsewhere (Belkin, Marchetti & Cool 1993), we have proposed a model of information-seeking behavior based upon the concept of a multidimensional space of information-seeking strategies (ISSs). According to this model, the variety of behaviors people engage in while searching for information in some knowledge resource can be viewed as ISSs, and these ISSs can be construed as interactions between the user and the other components of the IR system. These strategies arise from characteristics of the person's *problematic situation* (Wersig 1979); in particular, the user's state of knowledge and information-seeking goals. For instance, a person may or may not be able to specify a desired item, or may or may not know how to use an information resource effectively. A person may begin an information interaction with only a vague understanding of his or her information problem or need. Furthermore, a person's knowledge and goals may change over the course of a single information-seeking episode. The specific values of the ISS taken at any one time are contingent upon these factors, related to the larger context of a person's information problem.

We suggest that any single information-seeking interaction is a complex activity, which can be characterized according to its values on a relatively small set of factors, or dimensions. In our preliminary attempts to develop this model, we proposed the following four dimensions of ISS, based upon our own observations and on the empirical findings of others, including Belkin, et al. (1990), Ellis (1989) and Hancock-Beaulieu (1990):

Method of Interaction
(scanning ——————————— searching)

Goal of Interaction
(learning ——————————— selecting)

Mode of Retrieval
(recognition ——————————— specification)

Resource Considered
(information ——————————— meta-information)

The first of these, method of interaction, can be understood in terms of the classic distinction between *searching* for a known item and looking around, or *scanning*, for something interesting among a collection of items. The goal of the interaction may be *learning* about some aspect of an item or resource, or *selecting* useful items for retrieval. Furthermore, looking for identified items can be characterized as retrieval by *specification*, while identifying relevant items through stimulated association can be characterized as retrieval by *recognition*. And interaction with *information* items themselves can be contrasted with interaction with *meta-information* resources that describe the structure and contents of information objects.

We suggest that any single ISS can be described according to its location along these four dimensions. In the simplest case, we can consider each of these dimensions as orthogonal, with dichotomous values, resulting in sixteen distinct ISSs. Table 1 illustrates all of the possible ISSs that can be derived from our binary, four-dimensional model. At this point, it is unclear whether the dimensions suggested in our model can be treated as continuous rather than dichotomous variables, although it seems likely. If so, we have a multidimensional space of ISSs, in which certain regions might exemplify standard interaction strategies.

Table 1: Information-Seeking Strategies (cf. Belkin, Marchetti & Cool 1993, p. 326)

ISS	Method		Goal		Mode		Resource	
	Scan	Search	Learn	Select	Recognize	Specify	Information	Meta-information
1	x		x		x		x	
2	x		x		x			x
3	x		x			x	x	
4	x		x			x		x
5	x			x	x		x	
6	x			x	x			x
7	x			x		x	x	
8	x			x		x		x
9		x	x		x		x	
10		x	x		x			x
11		x	x			x	x	
12		x	x			x		x
13		x		x	x		x	
14		x		x	x			x
15		x		x		x	x	
16		x		x		x		x

According to our matrix, *ISS2* represents a situation in which a person needs to learn about characteristics of the knowledge resource before the information search can begin. This can also be understood as the ISS associated with an unformulated and unspecified information problem.

A person in this situation might begin an information-seeking episode by looking through a meta-information resource such as a classification scheme or thesaurus in order to learn about the organization of the knowledge resource.

In contrast, *ISS15* is a prototypical example of a well understood information problem, in which the goal of the interaction is not to learn about the system, but to select items which can be specified by the user. After learning about the information system by consulting the classification scheme, the person represented above in *ISS2* might choose some descriptors from that scheme as the specification of a topic of information items to be searched for. Similarly, *ISS1* represents a person who scans the current periodical shelves in order to learn what journals exist on a particular topic of interest. *ISS6* corresponds to a person scanning through a table of contents of a journal, with the goal of selecting articles relevant to a particular topic.

We do not wish to make the claim that the dimensions suggested in our model are exhaustive, but, on the basis of our examples, they appear to be necessary if not sufficient, and represent at least a useful starting point for characterizing ISSs. The matrix can be used to illustrate several less usual ISSs. For example, while scanning is typically associated with retrieval by recognition, and searching is typically associated with retrieval by specification, the ISSs identified in our matrix demonstrate that these are not inevitable associations. Scanning occurs in conjunction with retrieval by specification in *ISS4*. A person may know precisely what she is looking for, but not where it is located. Searching occurs in conjunction with retrieval by recognition in the case of knowing what one wants to retrieve, but not being able to specify it (*ISS13*). The inability to specify identifying characteristics of the item means that retrieval by recognition is the only mode available for this user.

According to our conceptualization, information-seeking behavior is characterized by movement from one strategy to another within the course of a single information-seeking episode, as a person's problematic situation changes (cf. Bates 1989). For example, a person may not be able to specify the title or author of a book she is looking for, but may remember its approximate shelf location. In order to recognize the item, this person might go to this location and scan the shelves (*ISS5*). While at the shelves, she may find some other book or item that seems relevant. This might prompt her to search for other similar items, but first she needs to know how to find these items. In order to learn how other items have been characterized in the information system, she may turn to a meta-information resource, such as a catalogue, to identify the subject terms under which the item has been indexed (*ISS12*). Having finished this step, this user would be able to continue the information-seeking episode by specifying these terms in a search for other relevant items (*ISS15*).

Having ISSs described by, and located in, the kind of space we suggest gives us a means to describe movement from ISS to ISS, as well as to describe the individual ISSs, and potentially the means to understand such movement well enough to devise methods for supporting it in a principled fashion. From this point of view, we can consider ISSs as types of user interactions within the IR system, rather than as queries or demands put to that system.

1.2 Dialogue Structures for Information Seeking

A formal model of information-seeking dialogues (the "Conversational Roles Model" (COR), cf. Sitter & Stein 1992, Stein & Thiel 1993) gives us the possibility of describing the interaction at the discourse act level. Basically, this model defines the types of dialogue acts (conversational acts) available and possible (local) patterns of exchange between the two participants, the information seeker and the information provider. For instance, an information offer or request for information can be accepted or rejected by the addressee, which would then require different possible responses; or the decision of how to proceed is postponed by inserting a meta-sequence or a clarification dialogue, where the conditions for this decision are negotiated and context information is exchanged.

The COR model was influenced by the "Conversation for Action" model proposed by Winograd and Flores (1986, p. 64 ff) where discourses are interpreted as "negotiations". We extended their model for the situation of information-seeking dialogues, applying some concepts of Systemic Linguistic approaches to discourse modeling (cf. Fawcett et al. 1988). Also, the framework of Rhetorical Structure Theory (RST) (cf. Mann & Thompson 1987) was used to enrich the COR model by specifying a set of possible semantic and pragmatic relations/ links which describe *how* discourse units are connected in a coherent way (see, for example, Stein & Maier 1993, 1995).

Figure 1 represents a recursive transition network as the basic schema of a dialogue: Circles and squares represent the *states* on the top-level of the dialogue (circles are within a dialogue, squares indicate terminal states). Arrows represent transitions between two states, i.e., the *dialogue contributions* or *moves*. A and B are the participants, A referring to the information seeker, B to the information provider. The order of the parameters indicates the speaker–hearer roles, the first parameter indicates the speaker, the second the addressee. A traversal of the basic 'dialogue' network from state <1> over any other state back to state <1> is called a *dialogue cycle*.

The presentation in figure 1 displays a variety of underlying categorizations of the dialogue contributions by graphical means (for example, orientation of arrows and placement of circles and squares). The main ideas can briefly be described as follows: The bold arrows between the states <1> and <5> represent two 'idealized' straightforward courses of a dialogue: A initiates the dialogue by a request for information, B promises to look it up (possibly skips the promise) and presents the information, A is satisfied and finishes the dialogue. Or, B initiates the dialogue with an offer to provide some information (anticipating an information need of A), A accepts the offer (or part of it), B provides the information, A finishes the dialogue. These two dialogue courses follow the role expectations set up by the two initiative dialogue acts. The dialogue cycles which end with a traversal from <4> to <1> ('continue') have a special status, because an information was given, but A is not yet contented and expresses the wish to continue the interaction.

However, such simple courses of actions are very rare in more problematic information-seeking situations. Participants often depart from that straight course which, besides, is not perceived by them as an unexpected deviation. Information-seeking dialogues are highly structured and usually contain a lot of corrections, retractions, and embedded clarifying sequences. In our work

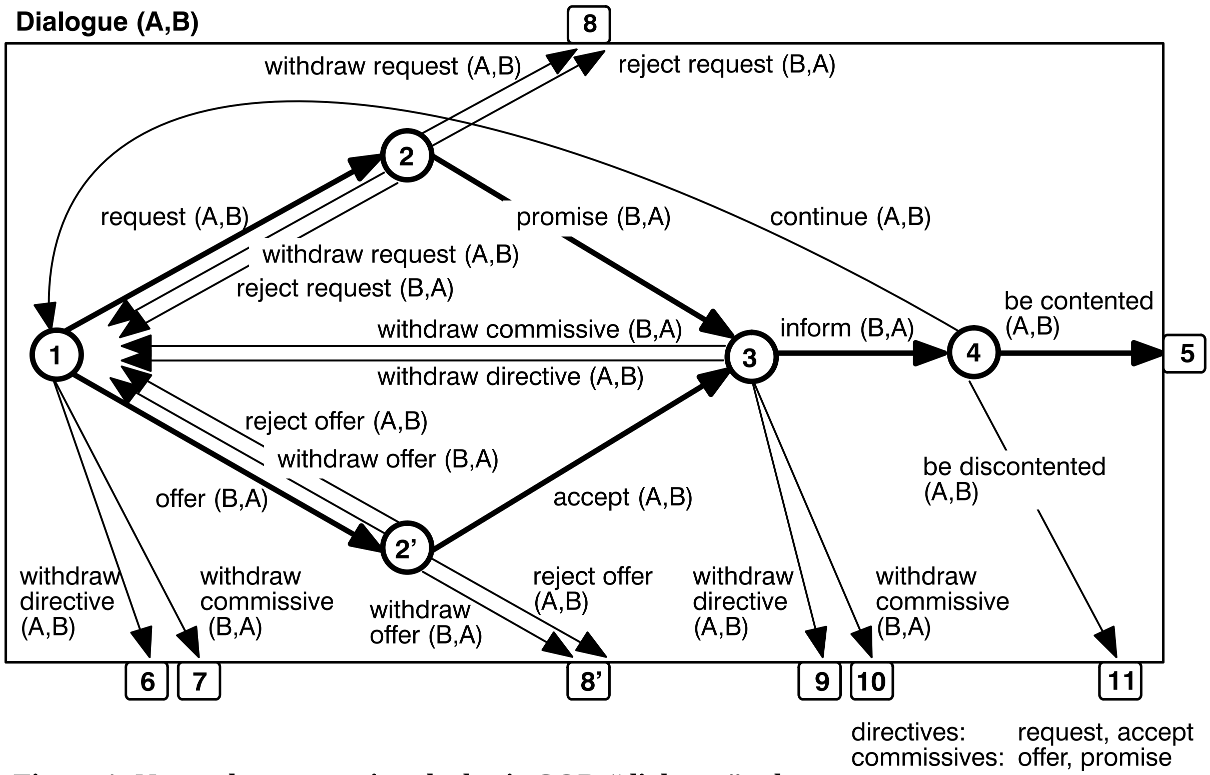

Figure 1: Network representing the basic COR "dialogue" schema

we attempt to overcome reductionist conceptions of dialogues as iterations of question – answer pairs which seem to build the basis for most of the classical interfaces to information systems.

According to the COR model the participants have basically two options of departing from the straight course:

- In each dialogue state they can reject or withdraw a given commitment finishing the whole dialogue (states <6> to <11>) or turning back to state <1>. This accounts for *iteration*.
- The transitions in figure 1 represent 'moves' that may be composed of several elements. A move either contains an 'atomic' dialogue act as the only element, or it (recursively) contains additional moves and subdialogues. Consider the following example: After an atomic request (A: "Search for xyz!") A adds some context information (A: "I'd like to get an overview, first."); or, instead, B asks for such additional information (B: "Would you like to see an overview or detailed information?") entering a sub-dialogue for clarification. In analogy to terms adopted from RST, the atomic act (the request) is interpreted as the "*nucleus*" which cannot be skipped, whereas the context information added and the sub-dialogue are optional "*satellites*". Thus, the transitions in figure 1 are themselves transition networks which may contain sub-dialogues of the basic schema type. This accounts for *recursion*. The COR model defines two types of transition networks for moves, which are not displayed here (for a more detailed discussion of the COR model and its integration with RST concepts see Stein & Maier 1995, and Fischer, Maier & Stein 1994).

Under the assumption that the two participants act cooperatively, COR describes their exchanges in terms of interpersonal behavior which follows conversational conventions and rules. Conversational phenomena, such as turn taking, and interactional functions, such as role taking and role expectations, are important to this model. However, not only the intentional (illocutionary) aspects are to be considered, but also the communicative effects of dialogue contributions (cf. Reichman 1985) and and their functional role during the dialogue development. Both, the roles of the participants as well as the role of their contributions in the thematic context, are to be taken into account. The COR model describes the formal *local* dialogue structures and abstracts away from the task/ domain and the strategic level, but it provides at the same time the general framework for further extensions on the topical and strategic level (cf. Stein & Thiel 1993).

Since the topical level governs the selection of the contents communicated in the dialogue acts, it plays an essential role in dialogue planning. If we want the system to engage in a meaningful cooperative interaction with the user, we have to address this question by supplying a prescriptive addition to the – so far descriptive – dialogue model. We perceive actual dialogues as instantiations of more abstract entities, called “scripts”, each of which represents a class of concrete dialogues. These are not arbitrary representatives of a class of interactions. Instead, they possess a ‘prototype’ property (in the sense of Rosch at al. 1976).

2 The Concept of Script-Based User Guidance

As well as characterizing ISSs according to the dimensions of the ISS space (effectively, locating them in that space), we also suggest that we can associate a prototypical, or ‘optimal’ interaction pattern with each region of the space, or each ISS. Such an interaction pattern could be construed as a general “script” (cf. Schank & Abelson 1977) or plan for a dialogue between the user and the rest of the system, when that region is relevant to the user. These scripts would characterize the most usual, or most effective, or in some sense standard means by which the user and the rest of the system interact, in order to accomplish the ISS associated with that region of the ISS space. Such scripts, based, for instance on, and abstracted from observations of people as they engage in information seeking, could be used as a means for structured human-computer interaction aimed at achieving the goal of that particular ISS. Of course, there could be more than one pattern associated with any one region, although the reverse is rather less likely. In section 2.1 we give examples of such prototypical interactions, limiting them, for simplicity reasons, to one script per region.

2.1 Scripts as Prototypical Interactions

In section 1.1 we have seen examples of different kinds of ISSs, each associated with a general region of our example ISS space. These examples were static, in the sense that they described situations, rather than activities. However, when the search for information is performed by means of an IR system, the strategy has to be performed by a sequence of interactions with the system. In the following, we give *example scripts* for human-computer interaction for some of these ISSs, in order to demonstrate the nature of such interaction patterns. These scripts are construed as patterns of moves in a two-party interaction; that is, a conversation. As such, these interactions are proper dialogues, and can be formally characterized as such by the COR model (see section 1.2). Scripts are idealized discourses which are easy to understand, because of a certain well-formedness. The real dialogues which are instances of these entities will be more complex (with cycles, sub-dialogues, branches, etc.). Note that although a script is a dialogue, it is abstract in the following sense: It reflects idealized defaults concerning the dialogue partners and the task. In a realistic dialogue which evolves in a non-idealized situation these defaults may be replaced, e.g., for a specific query there may be too much or no relevant data. The structure of the dialogue will then be changed accordingly.

As we intend to exploit these interaction patterns for controlling a human-machine dialogue, we have to be careful about their complexity. Obviously, an arbitrary dialogue is not a good model for the intended guidance, since it may consist of different kinds of moves. Some will be ‘necessary’ for the accomplishment of an ISS’s goal, while others will reflect particularities of the situation, e.g., the user’s knowledge of the system, or the availability of specific items. Thus, it seems reasonable to identify for each ISS a set of moves that are needed to achieve the desired results in the given setting. Then, each ISS can be assigned a ‘hierarchy of goals’ or a goal ‘tree’ which organizes the set of necessary moves. For reasons of simplicity, we do not further differentiate between a goal and the corresponding move in this section. However, as a discourse goal can be achieved in different ways, e.g., by a single (atomic) utterance or a sub-dialogue (see section 1.2), we will take this into account in the next section. We start elaborating the idea of a prototypical interaction by first discussing examples on a somewhat abstract descriptive level. Later on, we will introduce more concrete examples on the utterance level.

Consider *ISS5* which is appropriate for situations in which the user wants to find some item that is known about (or known to exist), but which cannot be specified by searchable characteristics. However, the general ‘location’ of the item may be known (by location we mean some conceptual region, such as red items, or items on the third floor, or like some other items). Thus, the overall goal of this ISS is to *select information items* by *recognizing* them while *scanning* the data available at the location (cf. table 1). A first step to accomplish this goal consists of accessing the location. After a decision to use this specific strategy has been met, an “ideal” interaction might commence with the system asking the user to *identify the location* of the item. The user is prompted to respond by suggesting the area that is known to be potentially relevant, which she does. The system then asks the user to *indicate aspects* of the item that would be useful for recognizing the right one (title, author, frontispiece, etc.), giving the user a choice of the features available to the system. The user *specifies* those that would be useful, and the system displays the items in the specified location, focusing on the characteristics which the user has indicated as helpful for recognition. The user *scans* through these items and, in the simplest case, *recognizes* the desired item.

The ISS realized by our first script example can be characterized by the following goal tree:

```
select_info (identify_location,
             specify_aspects,
             verify_offered_location (scan_items,
                                      recognize_desired_items))
```

Note that – in this example – the extracted moves only reflect the user's behavior. Due to her well-defined intention and a high degree of apriori knowledge, the user is able to *control the dialogue* efficiently in this situation. Therefore, in this goal hierarchy the system's responses could be neglected based on the assumption that in a prototypical or idealized interaction the system should detect and adopt the user's dialogue goals – or, at least support their accomplishment (cf. Kobsa 1985, Thiel 1990).

In a script-based dialogue, the system can anticipate the next move of the user, and offer alternative ways to perform it. This is controlled by the ISS's goal hierarchy: When user and system execute the goal hierarchy associated with the current ISS, one of the dialogue partners has to take the initiative whenever she enters a sub-tree in the goal structure. In our next example, we illustrate how this pattern yields a mixed-initiative interaction, because in this case the user lacks not only a specific information detail, but has only vague ideas about the relevant items.

Consider the situation in which a person would like to find items like a specified known item, but does not know how to characterize the desired items appropriately. The relevant strategy (*ISS12*) is then in the region of searching, learning, specifying, meta-resource. After the user has chosen the appropriate interaction script, the system begins by asking the user to identify or *specify the known item*, which the user does. The system then *asks* the user what *characteristics* of this item she would like to learn about. The user chooses one or more such characteristics. The system finds the specified item, and *displays* the requested characteristic associated with that item (e.g., the terms used to index it). The user can then choose from the display as a basis for a subsequent search, or, ask the system to display things related to one or more characteristics selected from the display (e.g., browse the thesaural structure for a given term).

The corresponding goal tree might be construed as follows:

```
learn_about_characteristics (specify_known_item,
                             define_characteristics (offer_choice,
                                                     select_characteristics),
                             verify_offered_characteristics (display_item,
                                                             assess_relevance),
                             switch_to_other_script)
```

Note that this example also shows that it is, of course, an oversimplification to expect that a single script will suffice to model the interaction needed in a realistic situation. For this reason, we use the meta-dialogic goal `<switch_to_other_script>` to indicate a shift to another script. We will see that meta-dialogic topics are essential in realistic scripts, when we analyze the more detailed examples.

As a last example on the abstract level, the script for *ISS15* responds to a quite different situation: one in which the user knows how to describe desired information items according to searchable characteristics, and desires to have such items retrieved. In this case, having entered the appropriate script, the system asks the user to *specify the value(s) of the desired characteristics*, from some general *choice of characteristics*. The user *chooses* one or more and specifies them, the system retrieves and displays items 'matching' the specification, and the user chooses those which are appropriate.

For this straightforward interaction the following goal hierarchy applies:

```
retrieve_specified_items (specify_characteristic (offer_choice,
                                                  select_and_specify),
                          recognize_desired_items)
```

From these few examples, we can see that the pattern of interaction between system and user differs greatly between the different regions of the ISS space. These differences arise in terms of what each partner asks of the other, and what each offers the other, and in the general structure of the interaction. Notice that the script in each case is tailored to taking advantage of what the user knows, in order to help the user to accomplish the desired goal.

Despite these differences, the scripts regarded so far have a specific feature in common: They directly implement the execution of an ISS's goal hierarchy in terms of a cooperative dialogue. Whenever a sub-goal is encountered, one of the partners takes the initiative by requesting or offering information (either in the data or in the meta-domain). In COR, this situation is reflected by dialogue state <1>, the starting point for a new dialogue cycle. Hence, we are able to assign each node in the goal tree to a cycle in COR. This implies two interesting consequences:

- First, a script can be 'constructed' from the goal tree of a strategy by deriving from the COR model the minimum amount of interaction required to achieve each of the goals.
- On the other hand, this sheds some light on the macrostructure of the dialogue since now we have 'links' between the COR cycles.

In the following, we will discuss the dialogue structure of scripts in a more detailed way. Any ISS interaction begins with a standard introductory section, in which the system informs the user of what classes of interactions it can support, the user chooses one, and the system informs the user of how that interaction will proceed. The various interaction choices (say, sixteen for the different ISSs in our simple model) will be described for the users according to the values of the dimensions, and especially according to the goals and knowledge that they assume of the user. The user having chosen, and learned about one such interaction support pattern, the specific interaction proper begins.

We start with a fragment of a script which is not associated with a specific ISS, but allows determination of an appropriate one (note that this sequence can be iterated several times, as necessary):

Standard Introductory Sequence for all Scripts

```
1  sys   Here's what we can do (offers choice).
2  user  Let's do this (chooses one).
3  sys   OK, here's how we'll do it
         (presents plan and means for accomplishing script). --> 4 or 1 or 5
4  user  a. OK.  --> 5
         b. No, I don't like this.  --> 1
```

On the formal level these steps are seen as a well-formed sequence of distinct discourse acts (A: offer –> B: accept –> A: inform –> B: continue) which build a complete dialogue cycle according to the COR model (traversal of the network back to state <1>). In this cycle the plan or script for the subsequent dialogue is negotiated. Thus, the focus is mainly on the dialogue planning itself (on a meta-level) rather than on the specification of information items of interest. This sequence may get extended or structured without changing this overall function, for example, when the user revises her decision and makes another choice, or when she initiates a sub-dialogue in order to get more information on the system's offer or the suggested plan.

The choices offered in this sequence correspond to ISSs associated with the relevant region of the space. Our first example continues in the region of the ISS space defined by interaction via *searching*, with the goal of *selecting*, by *specification*, in an *information* resource (ISS15). A prototypical interaction in this space would be based on the following schema:

Example Script 1 (ISS15)

```
 5  user  Find me something that corresponds to this
         (specification of kinds of items to be retrieved).
 6  sys   a. Here's something that you'll like
             (presents the one item found, in detail).
          b. Here are some things that you'll like
             (presents the set of items found, in overview).
          c. I can't find anything like what you asked for. --> 5 or 7c
 7  user  a. 1. I like this.--> 8
             2. I don't like this. --> 7c
    user  b. 1. Let's continue. --> 7c
             2. Let's quit. --> 10
             3. Let's look at this one (chooses one from list).
    sys      4. Here (presents item in specific detail).
    user     5. I like this. --> 8
             6. I don't like this. --> 7b.7
    sys      7. How about one of these (shows list)?
                --> 7b.2 or 7b.3, or, if nothing left in list --> 7c or 10)
    sys   c. Here are some ways that we might be able to find something you'd like
             (presents set of suggestions for continuing search, modifying
             specification, or changing script).
    user     1. Let's try this one (chooses). --> 5 or new relevant script
             2. Let's quit. --> 10
 8  sys   Shall we save this and continue? --> 9
 9  user  a. Yes. --> 7b.7 or 7c
          b. No, let's continue. --> 7b.7 or 7c
          c. No, let's just quit. --> 10 or 12
          d. No, let's just save this and quit. --> 10 or 12
10  sys   What would you like to do with what's been found
          (presents list of choices of action)?
          [conditional on something having been found]
11  user  This (chooses something).
12  sys   Goodbye.
```

This interaction sequence allows the user to specify a criterion according to which information items will be retrieved by the system, for judgement by the user as to whether or not they are relevant. It also allows the user to modify the initial specification, according to a set of tactics relevant to that particular situation. The set of tactics could be within the given script, or they could be an invocation of another script, such as that one for browsing through a meta-information structure for learning by recognition (e.g., display of a thesaurus).

According to the COR model (cf. figure 1) this sequence can be interpreted as follows:

- The steps 5-7a build a dialogue cycle (A: request –> B: inform –> A: continue (evaluation); *or:* A: request –> B: promise (skipped/ implicit) –> B: withdraw (6c)). Here, the main goal is the specification of items to be retrieved from the database.

- The 7b.1-4 steps are interpreted as a sub-dialogue (in 6b) initiated by a user's request for more detailed information on one of the retrieved items. 7b.5 is a system's offer which also initiates a sub-dialogue with a similar function, i.e., to elaborate on information already given in overview.

- With 7c the system turns again to the top-level of the dialogue offering a set of tactics for continuing the search or for proceeding with a new dialogue plan. The sequences 8-9 and 10-11 can also be seen as negotiations on a meta-level where the system requests advice on how to proceed. However, they are related to specific other steps, e.g., the user's relevance judgements or decision to quit the dialogue.

Our second example script is that of an ISS in the region defined by interaction via *scanning*, with the goal of *learning* by *recognition*, in a *meta-information* resource (*ISS2*). This dialogue begins with the initial four steps common to all information-seeking episodes, with the first step enumerating the possibilities in this region (e.g., learning about searching vocabulary; learning about relations among institutions), and the third specifying how any one task might be accomplished (e.g., scanning in a thesaural display). Our example continues after the user has chosen a task and method, as follows:

Example Script 2 (ISS2)

```
5  sys   a. You can look at the overall structure of the [name of meta-information
            resource]. --> 6a
         b. I can suggest a possible starting point from which to view the
            structure. --> 6b
        c. You can specify a starting point from which to view the structure.--> 6c
6  user  a. OK.  --> 7a
         b. Let's start from this one
            (selects a starting point from the display).--> 7a
         c. Here's a starting point. --> 7a, 7b
7  sys   a. Here's the part of the structure that you wanted to see. --> 8
         b. I couldn't find what you wanted in the structure.
            Here are some things that you could do. --> 5
8  user  a. Show me the structure around this item (selects from display). --> 7a
         b. I like this (selects from  display). --> 9
         c. Let's quit. --> 11
         d. Let's stop this and do something else. --> 1
```

```
 9  sys   Shall we save this and continue or shall we use this term for a query to
          the database?
10  user  a. Save and continue.  --> 8a
          b. Don't save, just continue. --> 8a
          c. Save, but quit. --> 11
          d. Don't save, and quit. --> 11
          e. Save, but let's do something else. --> 1
          f. Don't save, and let's do something else. --> 1
          g. Start a query with this term. --> script 15 or other relevant script
11  sys   Goodbye.
```

This interaction sequence allows the user to learn about some aspect of the data, and data structures within the system, for instance, about search term vocabulary. The description of this sequence is somewhat abbreviated, in comparison to the previous one, as we have not included some possible courses of action which might be relevant in case of a failure. Notice that in this sequence, there is no idea of doing a specified search, but rather the user browses from one term to another, saving those that she likes, until enough has been learned, for whatever purpose brought the user to this ISS.

Of course, our examples are simplified in that they are successful at each point in which there is a possibility of non-success, each ISS script is independent and self-contained, and the user's problem, goal and knowledge are constant. These are clearly unrealistic simplifications. Relaxing these simplifying assumptions leads us to the concept of a single information-seeking episode consisting of a sequence of several such scripts.

2.2 Information-Seeking Episodes

A realistic concept of human information-seeking behavior would respond to the observation that people, in such activities, routinely change from one ISS to another. This can be due to their goals, problems, and knowledge changing, through the course of the interaction, or because their overall information-seeking plan for the particular episode required a sequence of ISSs to achieve the overall goal which led them to the activity. From the point of view of our model, this can be seen as the combination of scripts into patterns of information-seeking episode interactions. Such combinations might be pre-planned or pre-stored in a system, but they are much more likely to arise in response to the specifics of any particular episode. Below, we offer an example of an interaction which combines the two scripts described above, and then discuss how it might be possible to predict and support potential moves from one place in the space of ISSs to another.

Consider the case of a person whose goals and knowledge correspond to those associated with ISS5. This person would like to look at a book that she had seen last week, but cannot remember its author or title. She does remember that it was on the bottom shelf of some set of stacks on the third floor of the library. Being in a progressive institution, the library catalog allows her to engage in an ISS5 script with the specified location as region to be scanned. While looking over these items, the user sees an interesting item that she didn't know about before. This phenomenon is usually called serendipity, and is something that every information system tries to support.

This item looks so interesting, that the user would like to see more like it. Unfortunately, the other items nearby, that she is scanning, do not appear to be very similar to it, at least in ways that are relevant to this user. Now the user finds herself in the position of wanting to learn how to describe this item in a way that will help her to find others like it. This is the situation that ISS12 responds to. So the user selects the item of interest, and informs the system that she would like to engage in a new ISS. This might be accomplished, for instance, by selecting the appropriate ISS script from a menu of alternative scripts that the system keeps displayed. Once engaged in the ISS12 script, the user asks to see the subject headings associated with this book, and the headings related to them. She finds one of the terms in the display to be relevant to her interests, and so wishes to see what other documents indexed by this term look like. This brings her to ISS15, which supports her in searching to select items in an information resource according to some specified characteristic(s). In this case, ISS15 might be invoked by the user selecting the appropriate term from the display, and informing the system that this should be used as a search key, again from the menu of ISS scripts. The system, having been informed of the user's desires, performs the search as specified, and displays the other items indexed by the chosen term. Clearly, such movement from ISS to ISS could continue indefinitely, but we'll bring it to a halt here.

Predicting, or constructing such sequences is clearly a problem for the design of a system based on these principles. One way to address this problem is to make each script in the system always available for selection by the user, at each interaction move. A slight variant of this approach has been followed in the interface design suggested by Belkin, Marchetti and Cool (1993). But with many available scripts, and with the many ways in which they could be implemented highly dependent upon the context of the interaction, this might become rather cumbersome.

An alternative approach is to base the script combinations on likely branching points, such as points in the interaction where the system might not be able to display the desired response because of problems with the input, or no match between what the user is able to specify, and what is in the database. A typical example of this situation is when a user specifies a search term which is not used in the database. Such failure points can be anticipated by detailed analysis of the moves in each script, and potential remedial strategies (e.g., movement to other scripts) can be enumerated. In the example given, for instance, the user might be prompted to begin again with a different search characteristic, or it might be suggested that she consults a thesaurus in order to find different terms. In our experimental implementation of script-based user guidance using the MERIT environment (cf. section 3.1) such hints are coupled with the system's report on success or failure of the current activities (cf. figure 4).

This leaves us still with the problems of supporting movement from ISS to ISS based upon the user's overall goal and plan in entering the system, or upon unanticipated changes in user's problem, goal and knowledge which arise during the course of the interaction. Some of these can again be anticipated, based upon analyses of what people are likely to do when presented with particular kinds of information, and what problems they are likely to face in attempting to achieve particular classes of goals. Regard a leaf in a goal tree of an ISS as representing a basic goal, e.g., `<enter_appropriate_terms>`. If this goal cannot be accomplished by the user in an atomic action – entering a sequence of terms – she can 'expand the tree' by entering another strategy which allows her to learn about the domain, e.g., by exploring a thesaurus. In our model this modification of the user's intentional structure is represented by an operation on the current goal tree: The new strategy's goal tree replaces the leaf node. This approach requires the explicit representation of the goal hierarchies associated with the scripts available in the system.[1] In our enhanced system design currently under development (see section 2.3), we intend to go beyond predefined scripts, therefore the dialogue manager will be able – in a critical situation – to propose not only standard problem solutions, but also to suggest scripts derived from the user's previous sessions. We elaborate this idea in the following section.

2.3 On Deriving Scripts: A CBR Approach

We have suggested that the concept of a space of ISSs can lead to the specification of a relatively small number of prototypical interaction sequences, or scripts, which can be used to guide effective user interaction in IR systems. A general characterization of information-seeking goals, and a related cognitive task analysis is one way to address the problem (cf. Belkin, Marchetti & Cool 1993). Another is through empirical observation of instances of interaction patterns, or storing of such patterns for use in guiding similar new interactions. This, of course, suggests the use of case-based reasoning techniques for helping to structure and organize interaction sequences.

The problem can be addressed in the case-based approach by collecting numbers of cases, analyzing them to determine to which region of the ISS space their different parts are relevant, and characterizing each such sequence as a series of dialogue moves. With sufficient data, or as data are gathered, general patterns for the different regions can be induced. That is, some empirical observation in typical information-seeking situations is required in order to begin the original system design. The concept of ISSs provides a means for structuring such observation, and for analysis of the data which would lead to standard interaction sequences, or scripts. And once the system is used on a regular basis, collecting and maintaining the cases of interaction could lead to learning about new interaction sequences, both within a single region of the ISS space, and among them.

In order to derive scripts from a collection of cases, the following prerequisites are required: First, each episode has to be analyzed and represented according to the COR model. Further, an analysis of the rhetorical structure based on the Rhetorical Structure Theory (RST, cf. Mann & Thompson 1987) is needed to identify the nuclei (cf. section 1.2), i.e., the essential parts of components of a dialogue move. An abstraction process might then yield the goal structure of the given sequence of moves, and finally, an ISS might be associated according to the goal.

The dialogue analysis aiming at the identification of candidates for scripts is a long-term goal, requiring interdisciplinary efforts in the field of discourse analysis, information science, knowledge representation, and CBR. Steps towards this goal involving a systematic extension of RST

---

1. This approach to dialogue planning is quite similar to the operator-based approach to text planning, cf., for example, Moore & Swartout 1990, Moore & Paris 1993.

to cope with the phenomena occurring in dialogical texts are reported in Sitter & Maier 1992, Fischer, Maier & Stein 1994, and Stein & Maier 1995. Since a detailed account of the linguistic results is beyond the scope of this paper, we refer to the related publications, and return to our main topic, that is, the application of cases, or scripts, in interactive information retrieval.

3 The Use of Scripts and Cases in Interactive Information Retrieval

The notion of scripts or schemas has been employed in several top-down approaches to discourse planning. However, it was also used in *bottom-up* case-based planning of multimedia presentations (cf. MacNeil 1991). In the domain of information-seeking dialogues, we have a certain variety of possible ways to realize a given strategy with the functions/ options of a specific system. These "ways" may be obtained from dialogue sessions, and may therefore be regarded as 'cases' in the CBR sense. Since each of the cases can be related to (at least) one of the strategies, the set of cases in the library is partitioned. Now, we are in a position to ask: can we distinguish one of the cases related to the same strategy as a prototype? Such a prototype – a *script* – may be used as a starting point in the planning process, since it is reasonable to start with a prototypical case, when we want to realize a certain strategy. As the specific problem may require modifications of the solution proposed in the first place, we have to look for another member of the class of cases to solve the problem. However, this may not suffice, since from the user's behavior we might conclude that she is actually preferring some other strategy. Then we have to switch to some other class of cases.

In our experimental work, we have already started to adapt the ideas of CBR to the requirements of a user-guidance component developed as part of the prototype multimedia information system MERIT (for details, cf. Tissen 1991, 1993a,b, Stein, Thiel & Tissen 1992). Section 3.1 describes the general structure of MERIT, and how the concepts of dialogues, scripts and cases are integrated in that structure. Section 3.2 gives an extended example of interaction in the MERIT system, in which two different scripts are used in a single information-seeking episode, or case.

3.1 An Overview of the MERIT System

The idea of regarding the information retrieval interaction as a 'conversation' between the user and the system was the basis for the design of the MERIT system. Although the interaction is mainly graphical, i.e., based on direct manipulations of screen objects, it is nevertheless structured according to conversational patterns (cf. Reichman 1989 for a similar approach). The COR model provides the general scheme for interpreting menu selections, etc. as dialogue acts of the user, and, it allows identifying appropriate continuations of an ongoing dialogue. While this aims at a coherent dialogue guidance on the tactical level, the overall structure of the dialogue is determined by a dialogue manager (CADI, cf. Tissen 1991), which applies a 'case-based' approach to user guidance. In order to "... solve new problems by adapting solutions that were used to solve old problems." (Riesbeck & Schank 1989, p. 25) the user is offered a selection of cases that were stored in the past representing successful retrieval sessions. The proper cases, which are stored in the case library, are somewhat simpler (avoiding unnecessary loops, etc.) but since they are derived from real dialogue sessions, they may be not so ideal or elegant as a script. Remember that they reflect the search behavior of a certain person who may not have found the theoretically best solution, but a working or successful one. Of course, such a case has to be adapted to the user's current information need, e.g., by modifying the sample query, or, the proposed presentation form of the retrieval result. The MERIT system interfaces to a large (relational) database containing information about European research projects, funding programs, and project publications.

At the beginning of a dialogue session, the user can either choose among an offered preselection of basic cases, or she may use the CADI system to retrieve a case in the case library (cf. Tissen 1991). Then she usually gets a query form representing the relevant sub-set of attributes which are specific to the current case and dialogue state. After this query step (user's request), the retrieved data are shown in one or more subsequent steps at various levels of detail (the system's inform/presentation steps). The number of presentation steps and selection of the generic presentation form is pre-defined by the current case, but the user also has the opportunity to change the presentation form or the level of detail in each step, thereby altering the current case to some extent. Several additional functions for modifying cases, perspectives, etc. are provided by the CADI component (cf. Tissen 1993a,b).

After the case selection, the user is presented a query form listing attributes of the selected perspective. Each line of the query form sheet represents an *attribute 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. The process of finding query concepts is crucial for the quality of the retrieval result. In MERIT, the user is supported in this task by a module called *Knowledge Explorer (KX)*, which incorporates conceptual knowledge of the domain and suggests concepts which may be used to expand the query in a meaningful way (cf. Thiel, Kracker & Stein 1993). This conceptual knowledge is represented as a fuzzy association network (cf. Kracker 1992). featuring four types of such relationships: *positive* and *negative association* between concepts, *generalization*, and *specialization* relationships can be represented. Each pair in a relationship is assigned a value out of [0, 1], 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.

Stating restrictions in a query form sheet allows accessing the database without noticing its internal relational structure. The presentation component of MERIT employs a set of generic forms (graphs, tables, lists etc.) which allow for an adequate visualization (cf. Kerner & Thiel 1991). Whenever the user wants the system to memorize the current dialogue path with all its modifications, she can use the CADI system to compose and store a new plan and thereby augment the library of dialogue cases. As one can generate a very large number of possibly useful dialogue cases this way, a classification of dialogue types considering information-seeking strategies is desirable to enable the user to find an appropriate strategy.

The functionality of the MERIT system and the use of cases to guide the interaction can be illustrated by the first steps of a sample session. At the beginning the system offers a list of tasks (*step 1 of the standard introductory sequence* as described in section 2.1):

Here's what we can do now:
1. Find information about European research programs
2. Find a good theme for a project
3. Find good partners for a proposal in a program
4. Write and submit a proposal
5. Find projects in a specific field
6. Find publications of specific projects
7. ...

The user chooses 5 “Find projects in a specific field” (*step 2*); the system informs her (showing a text) that in this situation the only starting point is to specify the field or topic (*step 3*), and the user says “OK”. Then the system displays the relevant query form (see figure 4, next section) that contains slots such as project title, project objectives and topic description, but no additional slots such as contact person or duration of projects, project partners or their location, etc. The user inserts ”information processing” in the topic description field and presses “search & show results” (*step 5 of example script 1 for ISS15* as described in section 2.1).

As defined by the case and underlying script, the retrieved results (28 projects) are first displayed in overview (*step 6b*), that is, in the form of a table (see figure 2). Projects and their funding programs are represented by acronyms to give a first orientation. In this situation the user has some “standard” options to proceed: She may click on “change content” or “change presentation form” and select from the automatically generated menu-entries an alternative “view” of the retrieved items; she may go back one step (arrow at the left hand side) to the previous state in order to reformulate her query; she has the options of “changing the script” or to insert another type of query (e.g., “query on programs or persons”). These are either global or local (situation-dependent) options, which correspond to transitions in the COR network.

In our example the user does not choose/select a project to look at details, but decides to go on, and clicks on the “continue” arrow at the right hand side (*step 7b.1*). As the current script foresees a branching point in this situation, some of the available options are explicitly recommended to the user (the dialogue box “MERIT offers” pops up, *step 7c*). Choosing one of the three options may invoke similar or the same functions as clicking on other interface objects (e.g., change presentation form to select a presentation form that shows more details). This redundancy was considered important to support various interaction styles, that is, the user may choose between a guided interaction style and an unguided/ less guided one. On the other hand, the invoked functions often differ from the functions of the other standard buttons and icons. An explicit offer of recommended options gives a better opportunity for clarifying situation-dependent goals or intentions. For example, after clicking “Look at one item in detail” the user has the chance to define the degree of detail she is interested in, or, she may enter a subdialogue (e.g., to get information about the next step). In our case the user chooses “Modify query or pose new query” (*step 7c.1*), and the dialogue continues.

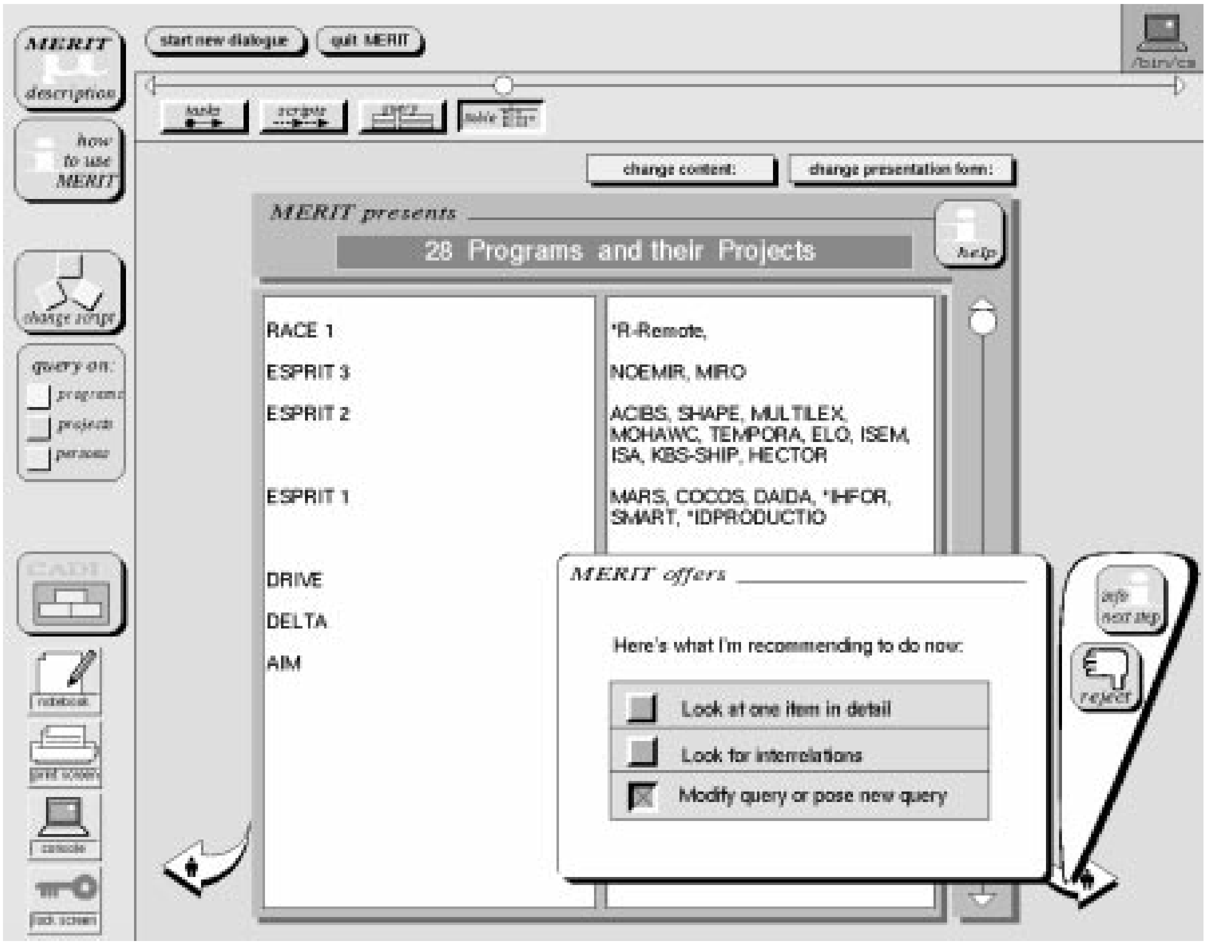

Figure 2: Example screen of MERIT (retrieved items in overview and a system's offer)

As we can see from this example, MERIT shows a considerable flexibility in its responses and seems to adopt an appropriate ISS in order to guide the user cooperatively. Seen from the system's perspective, the interaction is governed by some background processes which are not visible to the (naive) user: Once a certain task has been selected as relevant to the user's information problem, MERIT accesses its internal repertoire of stored plans, either to fetch a script that has been associated with this task explicitly by a previous user, or to retrieve a case from the case library. The second option which is, of course, potentially more apt for customizing the system, requires a detour via the "case retriever" tool which allows querying the case library. In both cases, however, a dialogue plan is identified that implements the appropriate ISS. The execution of this plan leads to specific system offers and responses which enable the user to pursue a useful strategy without deliberately choosing or planning it. The system's dialogue manager keeps track of the interaction and determines the subsequent dialogue contribution according to the user's behavior. In the normal course of events, the form of the system reaction is governed by the current plan, while its content is derived from the current situation, e.g., according to a query. On the other hand, the user is free to deviate from the straight path and can use common conversational moves to change the flow of interaction, e.g., change the topic or return to a previous dialogue context. We will discuss these options to a greater extent in the following section.

In general, the MERIT system addresses the user's information problem on two levels, employing scripts or cases and traditional query processing, respectively. The strategic problem solving dimension of information-seeking behavior is tackled by the plan-based or CBR part, whereas IR techniques are employed on the local level of identifying appropriate data items, which fit into the larger puzzle. The interactive (conversational) problem solving approach to information seeking allows the integration of both techniques.

3.2 Script-Based Conversational Interaction in MERIT

In this section we show how scripts and cases affect interaction in MERIT. We do this by considering a single information-seeking episode, describing it as an instantiation of the two general scripts outlined above; that is, as a *case*. Consider the task of preparing a proposal for a research project to be submitted to a European funding agency, and especially consider the sub-task of finding good partners for a proposal within a special funding program. The information resource that is available is a database of previously funded projects, which has descriptions of the projects and the names and addresses of institutions and people who worked on these projects. The user in this situation is a person in a research institution, who has already done some work on this issue. This case has been analyzed using both the COR model and the example scripts 1 and 2 introduced in section 2.1. The resulting structure is outlined schematically in figure 3. There, we indicate the dialogue act associated with each case step, and the specific action performed by the user or the system. This case goes as follows (the prefix numbers refer to the corresponding circles in figure 3; the step numbers refer to the steps defined by the example scripts described in section 2.1).

(1) The system offers the user the range of tasks it can support *(step 1 of the <u>standard introductory sequence</u> – first offer)*;

(2) the user chooses the task: Finding partners on a topic in a program *(step 2)*.

(3) The system describes the ways in which this task can be accomplished *(step 3)*,

(4) and offers some options which include starting with: a desired topic statement; a known project; an institution; or, a person *(*return to *step 1 – second, more specific offer)*.

(5) The user chooses to start by specifying a topic *(step 2)*.

(6) This instantiates the topic specification case. The system displays instructions for specifying a project topic *(step 3)*,

(7) and the user specifies a query (inputs a topic description) *(step 5 of <u>example script 1, ISS15</u>)*.

(8) In this example, the system informs the user that there is no project in the database indexed by this topic description (*step 6c*) (if something had been found, it would have been displayed).

(9) The system then suggests to the user ways to try to get something, which include: modify the topic description; begin again with a known project title or a known person; or learn about topic terminology used by the database *(step 7c)*.

(10) The user decides to learn about topic terminology *(step 7c.1)*.

(11) This decision leads the user to the second example script. The system then explains to the user that learning about topic terminology can be done using a tool, called "Knowledge Explorer" (KX), and offers the user to specify a term as a starting point *(step 5c of <u>example script 2, ISS2</u>)*.

(12) The user activates the KX-tool specifying the starting point, that is, her initial search term or another search term she wants to learn about *(step 6c of script 2)*.

(13) The system (KX) indicates that the start term has been accepted and that it is searching (actually 'computing') a list of similar terms from its terminology network,

(14) then shows the terms in the form of a ranked list *(step 7a)*.

(15) The user recognizes one of the displayed terms as being relevant to her interests, and selects it as such *(step 8b)*.

(16) As the system is not able to infer from that selection what to do next (e.g., either to offer further browsing in the terms network, or to take the selected term as a search term), it asks the user if she wants to search for projects on this topic *(step 9)*,

(17) and the user says "yes", that is, decides to start a query *(step 10g)*. This returns the user to the first script of this information-seeking episode: selecting by searching with specification in an information resource.

(18) The system then displays the project topic search form (with the selected term filling the appropriate slot) for user confirmation.

(19) The user confirms that the query is correct, e.g., by pressing a button *(step 5 of <u>example script 1, ISS15</u>)*;

(20) the system indicates that it is searching the information base

(21) and displays the retrieved projects on the specified topic *(step 6b)*.

(22) The user recognizes the name of an institution associated with one of the projects that she is familiar with, and goes away happy *(step 7b.2)*.

Our example case shows how the interaction follows pre-specified, yet highly flexible dialogue structures for specific regions of the ISS space. The case described above illustrates how the different sequences are responsive to the preexisting knowledge state and goals of the user, and how these conditions lead to specific case sequences, branchings, and sub-cases. In particular, it shows how scripts are associated with regions of the ISS space, how scripts can provide prototypical structures for guiding individual interactions, how movement is possible from script to script to support changes in ISSs, how scripts can index cases, and how cases can be used to guide interaction and to suggest new patterns in the current case.

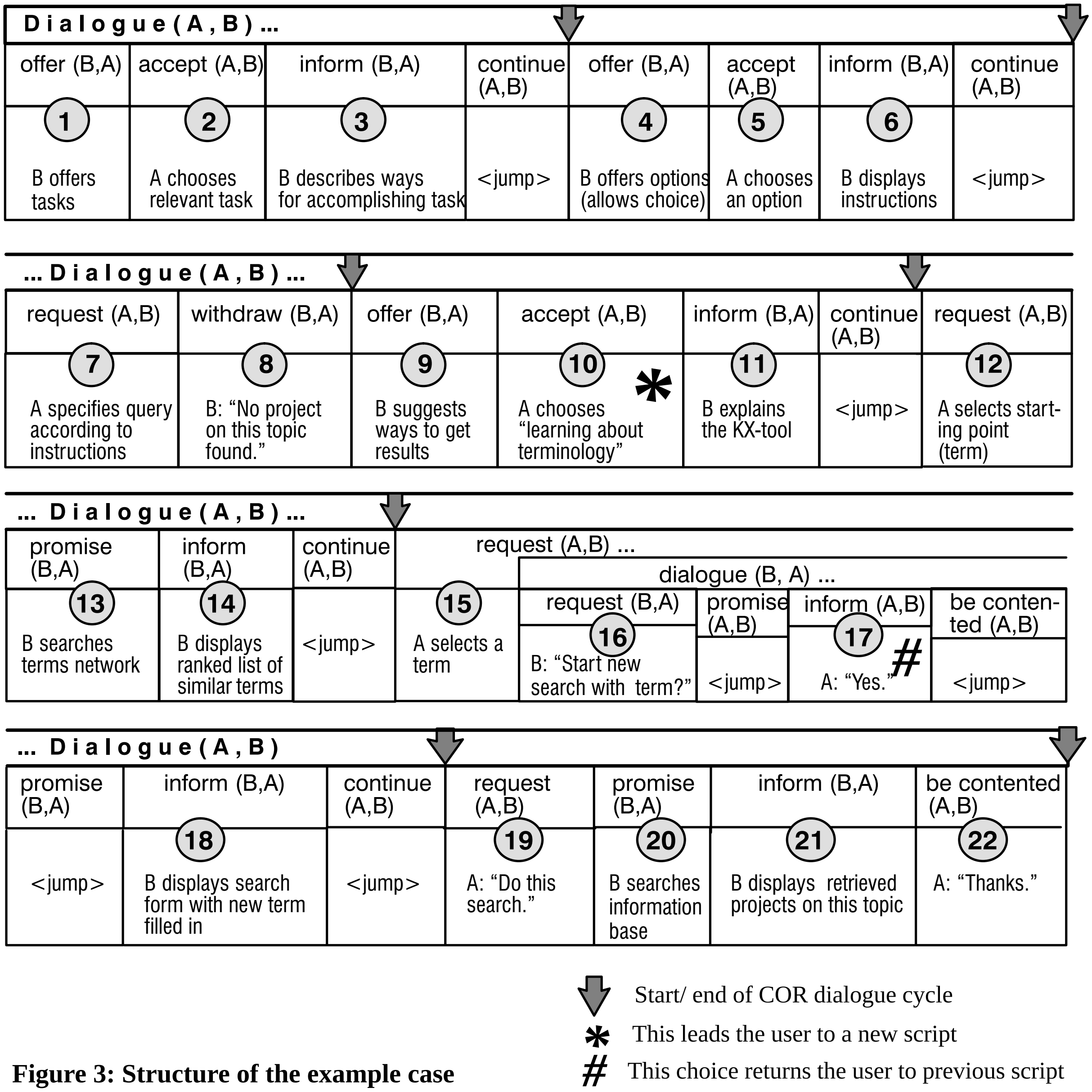

Figure 3: Structure of the example case

Notice that associating cases with ISSs automatically indexes them for subsequent use, according to the choice of ISS by the user. And if no cases are available for a particular region, then the default interaction will be based on the general script for that ISS, which thereby provides the structure for the development of a new case.

To illustrate how this example was performed in MERIT we show a snapshot of the system. Figure 4 displays a situation which corresponds to step 15 of the case, where the user is about to select a term ("analogical reasoning" which is the second one in the ranked list of the pop-up menu). As the case and its structure are the same as given in natural language above, we are now in the position to discuss the impact of the interaction style. While natural language examples suggest sequential order of utterances, the graphical interface of MERIT allows an object-oriented presentation. The user performs a dialogue act by clicking on special dialogue icons,

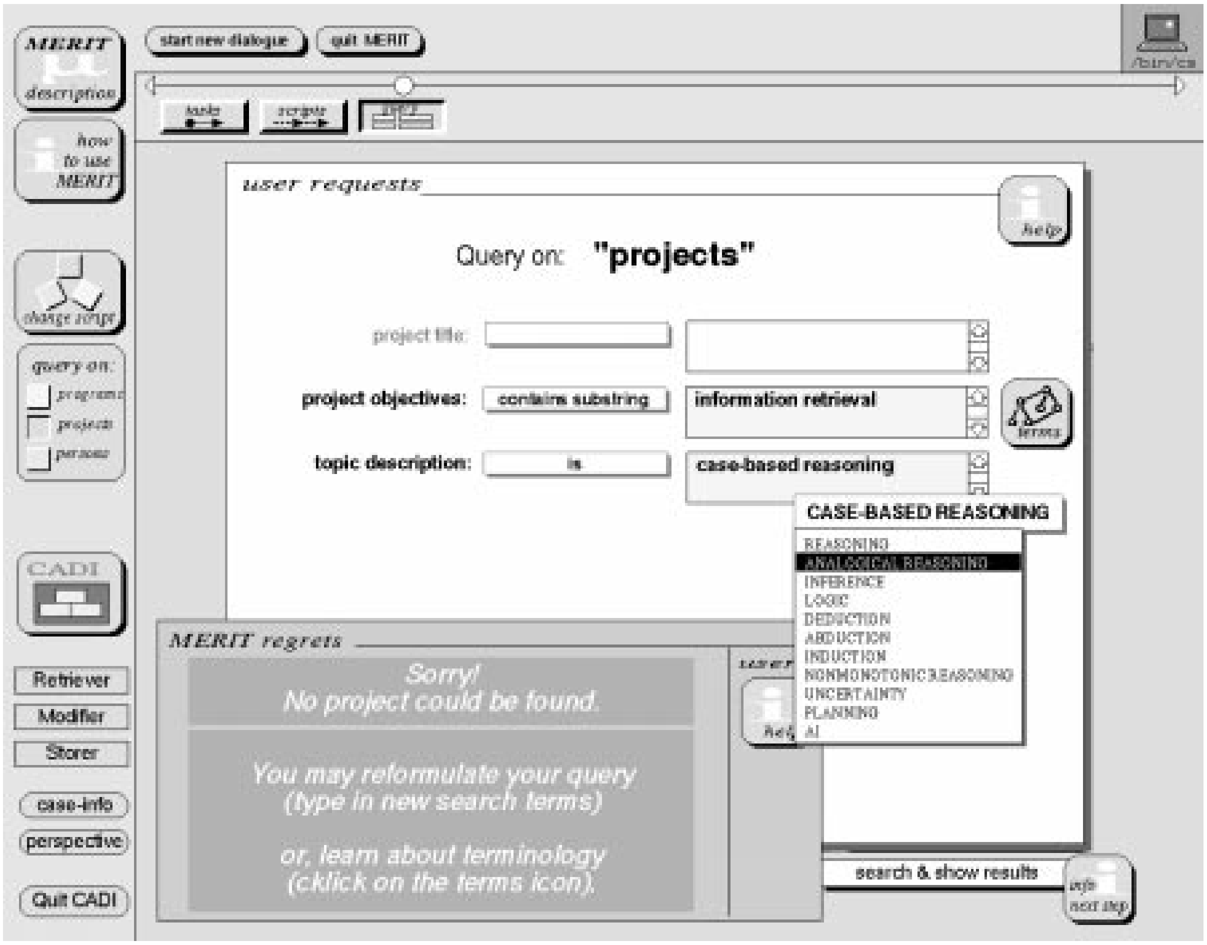

Figure 4: Example screen of MERIT (query form and display of a ranked list of search terms)

selecting items from a menu, or filling out forms. The system responds by dynamically generating graphical objects (dialogue boxes, menus, etc.) dependent on the current situation/ context. By visual means such as positioning and grouping of the objects a situation dependent perception of the dialogue course is facilitated. The focus of attention is always directed to the current (local) options to proceed. For example, active objects pop up in the middle area of the screen, whereas previous states are in the back and alternative (global) options represented by icons are located in the left and upper bars.

The snapshot displays some prominent states of the dialogue history (before step 15 was reached). The history icons appearing in the upper horizontal bar ("tasks", "scripts", "query", etc.) are representatives of the dialogue *cycles*; objects in the middle area represent special states which build the visual context for the current move. In this example the user had filled in "information retrieval" and "case based reasoning" in slots of the query form and then pressed "search & show results" (step 7). The dialogue box "MERIT regrets ..." combines steps 8 and 9. For selecting a starting point for the KX the user marked "case based reasoning" and clicked on the "terms" icon (12); the system popped up the menu (14) and the user selects "analogical reasoning" (15).

Although the abstract representation of the dialogue structure (figure 3) is the same as in the natural language example, the graphical interface facilitates the interaction. The MERIT example demonstrates how we combined graphical manipulation with additional textual information to construct a "multimodal conversation" (cf. Stein & Thiel 1993).

4 Conclusions

In this paper, we have suggested a model of IR system design that provides a means for supporting users in their various information-seeking behaviors. This model is based on the following premises:

- that information retrieval is most properly considered as *interaction*, specifically in the IR system case, as human-computer interaction;
- that such interaction can be considered as information-seeking strategies, characterized on a limited set of dimensions;
- that the human-computer interaction in information-seeking strategies can be modeled as a dialogue, and should be implemented as such;
- that particular dialogue structures (scripts) can be associated with different information-seeking strategies;
- that the techniques of CBR can be used to structure patterns of interactions involving combinations of information-seeking strategies; and,
- that the interplay of information-seeking strategies, dialogue structures, scripts, and cases can be used in a system design strategy which uses the advantages of each to ameliorate the disadvantages of each.

We have presented a set of proposals related to each of these premises, which both support their general validity, and indicate how they can be implemented in a system design strategy. We have described a prototype IR system, MERIT, which incorporates these ideas in its design, and shown, through an example interaction, how our proposals are implemented, and how their incorporation in the system can lead to effective support for complex interactive information seeking. In particular, we have shown how ideas from IR and CBR can be effectively combined in an integrated system design.

Among the advantages to this approach to IR system design are that it is based on a principled argument for the construction and use of cases of information-seeking dialogues, and that it allows a natural means of indexing such cases according to general dialogue structures or scripts, which are associated with the goals, knowledge and behavior of the user of the system. In addition, it offers a natural and effective means for involving the user in the interaction, through offering the user choices for interaction strategies at appropriate points, thus reducing the case retrieval problem substantially.

Although these advantages seem substantial, and although we have demonstrated by example and prototype how they can be implemented, several issues remain to be resolved before we can properly make strong claims about the effectiveness of our model. First, it is necessary to do more empirical specification and validation of the dimensions of the ISSs, which until now have been based primarily on relatively informal reanalysis of existing data. Secondly, we need to develop a stronger empirical basis for the explicit scripts associated with the different ISSs, and to specify formally the scripts for each region. And, of course, it is necessary to evaluate the performance of the MERIT system, or some other system designed on these principles, in proper experimental conditions. These are all long-term and difficult research problems. Each of them is currently being addressed, in ongoing research projects at both Rutgers University and GMD-IPSI.

5 References

Bates, M. (**1989**) The Design of Browsing and Berrypicking Techniques for the Online Search Interface. *Online Review, Vol. 13 (5)*, pp. 407-424.

Belkin, N.J. (**1982**) Models of Dialogue for Information Retrieval. In: Friberg, I. (ed.): *Proceedings of the 4th International Research Forum in Information Science.* Boras, Sweden: Skrifter fran Högskolan i Boras, pp. 15-36.

Belkin, N.J. et al. **(1990**) Taking Account of User Tasks, Goals and Behavior for the Design of Online Public Access Catalogs. In: *Proceedings of the 53rd ASIS Annual Meeting,* Medford, NJ, Learned Information, pp. 69-79.

Belkin, N.J., **Marchetti**, P.G. & **Cool**, C. (**1993**) BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval. *Information Processing and Management, Special Issue on Hypertext, Vol. 29 (3)*, pp. 325-344.

Belkin, N.J. & **Vickery**, A. (**1985**) *Interaction in Information Systems.* London: The British Library.

Croft, W.B. & **Thompson**, R.H. (**1987**) I3R: A New Approach to the Design of Document Retrieval Systems. *Journal of the American Society for Information Science*, *Vol. 38 (6)*, pp. 389-404.

Ellis, D. (**1989**) A Behavioral Approach to Information Retrieval System Design. *Journal of Documentation*, *Vol. 45 (3)*, pp. 171-212.

Fawcett, R.P., **van der Mije**, A. & **van Wissen**, C. (**1988**) Towards a Systemic Flowchart Model for Discourse. In: Fawcett, R.P. & Young, D. (eds.): *New Developments in Systemic Linguistics. Vol. 2: Theory and Application.* London: Pinter, pp. 116-143.

Fischer, M., **Maier**, E. & **Stein**, A. (**1994**) Generating Cooperative System Responses in Information Retrieval Dialogues. In: *Proceedings of the International Workshop on Natural Language Generation (INLGW '94)*, Kennebunkport, Maine, pp. 207-216.

Hancock-Beaulieu, M. (**1990**) Evaluating the Impact of an Online Library Catalogue in Subject Searching Behaviour at the Catalogue and at the Shelves. *Journal of Documentation*, *Vol. 46 (4)*, pp. 318-338.

Ingwersen, P. (**1992**) *Information Retrieval Interaction.* London: Taylor Graham.

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

Kobsa, A. (**1985**) *Benutzermodellierung in Dialogsystemen*. Berlin: Springer.

Kracker, M. (**1992**) A Fuzzy Concept Network Model and its Applications. In: *Proceedings of the FUZZ-IEEE '92*, San Diego. pp. 760-768.

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

Mann, W.C. & **Thompson**, S.A. (**1987**) Rhetorical Structure Theory: A Theory of Text Organization. In: Polanyi, L. (ed.): *Discourse Structure*. Norwood, NJ: Ablex, pp. 85-96.

Moore, J.D. & **Paris**, C.L. (**1993**) Planning Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information. *Computational Linguistics, Vol. 19 (4)*, pp. 651-694.

Moore, J.D. & **Swartout**, W.R. (**1990**) Pointing: A Way Toward Explanation Dialogue. In: *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI '90) (Vol. I)*, Menlo Park: AAAI Press/ The MIT Press, pp. 457-464.

Oddy, R.N. (**1977**) Retrieving References by Dialogue Rather than by Query-Formulation. *Journal of Informatics, Vol. 18*, pp. 37-53.

Reichman, R. (**1985**) *Getting Computers to Talk Like You and Me*. Cambridge, MA: MIT Press.

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

Riesbeck, C.K. & **Schank** R.C. (**1989**) *Inside Case-Based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum.

Rosch, E. H., **Mervis**, C.B., **Gray**, W.B., **Johnson**, D.M. & **Boyes-Braem**, P. (**1976**) Basic Objects in Natural Categories. *Cognitive Psychology, Vol. 8*, pp. 382-439.

Salton, G. & **Buckley**, C. (**1990**) Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science, Vol. 41 (4)*, pp. 288-297.

Schank, R. & **Abelson**, R. (**1977**) *Scripts, Plans, Goals and Understanding*. Hillsdale, NJ: Lawrence Erlbaum.

Sitter, S. & **Maier**, E. (**1992**) Rhetorical Relations in a Model of Information-Seeking Dialogues. In: Neumann, B. (ed.): *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI '92)*, Vienna, Austria. Chichester: John Wiley, 1992, pp. 179-180.

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

Stein, A. & **Maier**, E. (**1993**) Modeling and Guiding Cooperative Multimodal Dialogues. In: *Human-Computer Collaboration: Reconciling Theory, Synthesing Practice*. Papers from the AAAI Fall Symposium Series at Raleigh, NC, AAAI Technical Report FS-93-05, 1993, pp. 107-112.

Stein, A. & **Maier**, E. (**1995**)
Structuring Collaborative Information-Seeking Dialogues. To be published in: *Knowledge-Based Systems. Special Issue on Human-Computer Collaboration, Vol. 8, March 1995* (in press). Preliminary version available as: *Arbeitspapiere der GMD No 853*, Sankt Augustin: GMD, June 1994.

Stein, A. & **Thiel**, U. (**1993**) A Conversational Model of Multimodal Interaction in Information Systems. In: *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI '93)*, Washington DC. Menlo Park: AAAI Press/ The MIT Press, pp. 283-288.

Stein, A., **Thiel**, U., & **Tissen**, A. (**1992**) Knowledge-Based Control of Visual Dialogues in Information Systems. In: Catarci et al. (eds.): *Proceedings of the International Workshop on Advanced Visual Interfaces (AVI ’92)*, Rome, Italy. Singapore: World Scientific Press, pp. 138-155.

Thiel, U. (**1990**) *Konversationale graphische Interaktion mit Informationssystemen: Ein sprechakttheoretischer Ansatz.* Ph.D. Dissertation, University of Konstanz, Germany.

Thiel, U., **Kracker**, M. & **Stein**, A. (**1993**) MERIT Motifs. In: *Proceedings of the 5th IEEE International Conference on Tools with Artificial Intelligence (TAI ’93), Boston, MA.* Washington: IEEE Computer Society Press, 1993, pp. 482-483.

Tissen, A. (**1991**) A Case-Based Architecture for a Dialogue Manager for Information Seeking Processes. In: A. Bookstein, Y. Chiaramella, G. Salton, & V.V. Raghavan (eds.): *Proceedings of the 14th Annual International Conference on Research and Development in Information Retrieval (SIGIR ’91)*, Chicago. New York: ACM Press, pp. 152-161.

Tissen, A. (**1993a**) Knowledge Bases for User Guidance in Information Seeking Dialogues. In: Wayne, D.G. et al. (eds.): *Proceedings of the International Workshop on Intelligent User Interfaces (IWIUI ’93)*, Orlando. New York: ACM Press, pp. 149-156.

Tissen, A. (**1993b**) How to use Cases for Information Seeking Processes. In: *Case-Based Reasoning and Information Retrieval – Exploring the Opportunities for Technology Sharing.* Papers from the AAAI Spring Symposium Series at Stanford University, AAAI Technical Report SS-93-07, pp. 120-127.

Walker, D.E. (**1971**) *Interactive Bibliographic Search: The User/Computer Interface.* Montvale, NJ, AFIPS Press.

Wersig, G. (**1979**) The Problematic Situation as a Basic Concept of Information Science in the Framework of the Social Sciences – a Reply to N. Belkin. In: *New Trends in Informatics and its Terminology.* Moscow, VINITI, pp. 48-57.

Winograd, T. & **Flores**, F. (**1986**) *Understanding Computers and Cognition.* Norwood, NJ: Ablex.