

SCRIPTS FOR INFORMATION SEEKING STRATEGIES

From: AAAI Technical Report SS-93-07. Compilation copyright © 1993, AAAI (www.aaai.org). All rights reserved.

*Nicholas Belkin**, *Colleen Cool**, *Adelheit Stein#*, *Ulrich Thiel#*

* *SCILS, Rutgers University, 4 Huntington St, New Brunswick, New Jersey 08903, U.S.A.*

GMD-IPSI, Dolivostr. 15, W-6100 Darmstadt, FRG

Abstract. In this paper, we propose a model of information retrieval (IR) 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. These concepts are integrated within a framework of IR as interaction.

1. Introduction: Information Retrieval as Interaction

As the processes of query formulation and of inspection and judgement of retrieved data items are inherently interactive, we will understand IR itself as interaction, and in particular, as human-computer interaction. This notion involves two aspects. The more static aspect can be cast into the following question:

- *Given a certain situation in the dialogue session, what are the relevant documents in the database?*

Traditionally, researchers in information retrieval (as well as in the database field) assumed that it suffices to look at the current query and to identify the alleged relevant documents by a procedure that computed those documents that matched the query according to a given retrieval model. However, as the user is in most cases unable to formulate a query expressing his or her information need precisely, the second aspect of the information retrieval process is extremely important:

- *How can the interaction be structured in a way that supports the user in inspection, judgement and query formulation?*

This question has also resulted in a number of approaches, ranging from simple "browsing the database" approaches to sophisticated dialogue and user models. Apart from the formal differences, most of the approaches capture the overall structure of a given dialogue by assuming that it is governed by the user's information seeking strategy.

A space of information-seeking strategies (ISSs) might be characterized by dimensions such as: *goal* of interaction; *method* of interaction; *mode* of retrieval; and, *level* of information interacted with (Belkin, Marchetti & Cool, 1993, Belkin & Cool 1993). If these dimensions were orthogonal, with dichotomous values, we could identify sixteen distinct ISSs. If they took multiple discrete, or continuous values, we could identify regions of the space near each of these sixteen extremes.

We suggest that for each such region, there might be some general 'optimal' pattern of interaction, which 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. Such a plan would be specified in any particular interaction according to such factors as the task which brought the user to the system, the domain of discourse, the contents and structure of the information resources, and various characteristics of the user.

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 how 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 sense of case-based reasoning (CBR). 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 – called *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

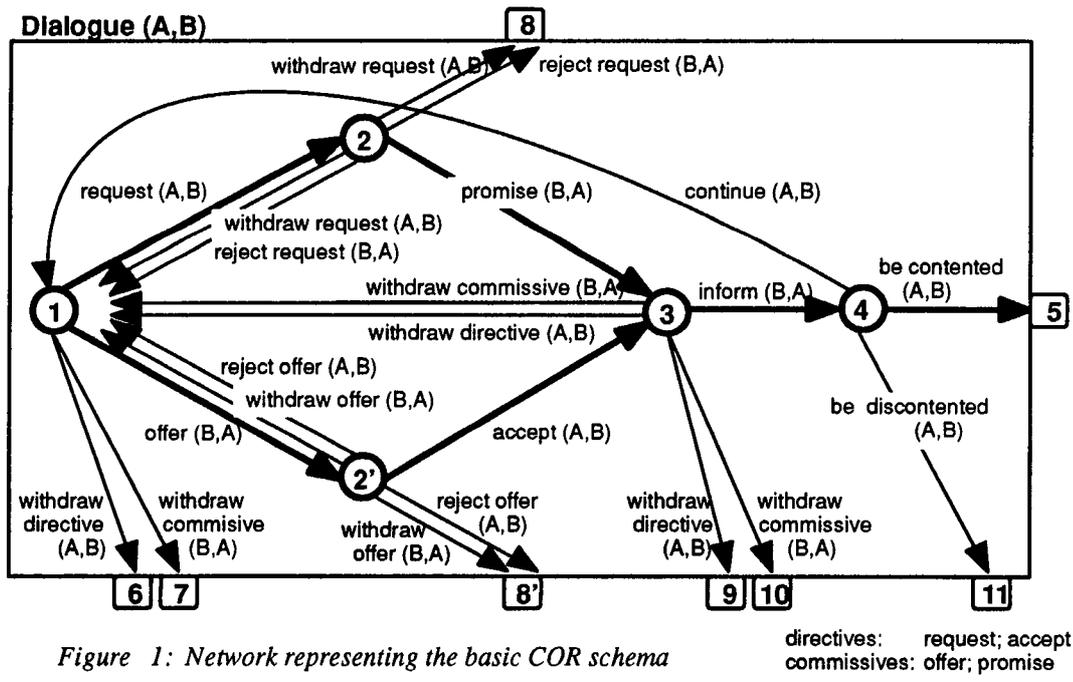


Figure 1: Network representing the basic COR schema

to solve the problem. However, this may not suffice, since from the user's behaviour we might conclude that she is actually preferring some other strategy. Then we have to switch to some other class of cases.

The concept of cases of information seeking dialogues was employed in the MERIT IR system interface (cf Stein, Thiel & Tißen, 1992, Tißen 1993a,b). Here, we show how the case-based approach of MERIT can be given a systematic foundation by incorporating within it the concept of ISSs, and demonstrate the validity of our general approach to IR system design by examples taken from that enhanced system.

2. Dialogue Structures for Information Seeking

A formal model of information seeking dialogues (the "Conversational Roles Model" – COR, cf Sitter & Stein, 1992; Maier & Sitter, 1992) gives the possibility of describing such dialogues at the discourse act level. Basically, this model defines the kinds of dialogue acts available and their possible relationships. 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 clarifying

subdialogue where the conditions for this decision are negotiated or backing information is exchanged. The 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) and Rhetorical Structure Theory (Mann & Thompson, 1987).

Figure 1 represents a recursive transition network as the basic schema of COR: 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. 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. The dialogue contributions (transitions) are themselves transition networks which may contain (sub-)dialogues of the type of the basic schema. The presentation in figure 1 displays a variety of underlying categorizations of the dialogue

contributions by graphical means (eg, orientation of arrows and placement of circles and squares). The main ideas can be described as follows: The bold arrows between the states <1> and <5> represent two quite 'idealized' straightforward courses of a dialogue: A utters 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. B offers 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.

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

The COR network covers the illocutionary structure of dialogues, but does not supply means for a specification on the thematic level. Since the thematic 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 et al. 1973).¹ This means they are idealized discourses, which are easy to understand etc. because of a certain well-formedness. The real dialogues which are

1. Note that although a script is a dialogue, it is abstract in the following sense: It reflects idealized defaults concerning the dialog partners and the task. In a realistic dialogue which evolves in a non-idealized situation these defaults may be replaced, eg for a specific query there may be too much or no relevant data. The structure of the dialogue will then be changed accordingly.

instances of these entities will be more complex (with cycles, subdialogues, branches etc.).

The classes comprise dialogues that show a similar basic pattern. Usually these patterns are closely related to certain strategies which are pursued by the user during her interaction. A collection of such dialogue plans is the basis for selecting an appropriate plan for a given information need. Once a plan has been chosen, it provides suggestions to the user how to continue in a given dialogue situation as well as specifications of cooperative system reactions.

On the implementation level, we represent dialogues as sequences of *dialogue steps*. The internal structure of a dialogue step is given by two parameters: The *perspective* of the step, and its *implementation*. The perspective determines the topical spectrum that can be addressed in this step without destroying the thematical coherence of the dialogue in general. Similar notions have been proposed by McCoy (1986) in the area of natural language interfaces and Reichman (1986, 1989) who takes a discourse analytical approach to multimodal dialogues. The second component of a step describes the possible and actual ways to implement the corresponding dialogue step. It may be implemented by a single dialogue act. In this case the variety is given by the different forms the utterance may have. For instance, the presentation of a certain set of data may be realized as a list of the data records, as a table, or as a graphical presentation. However, the step may also be realized by a certain sequence of dialogue acts which then build a subdialogue that may – according to the COR model – replace the single act. Thus, we have a means to prescribe a certain act as appropriate in the given situation, but allowing the user to realize this in a way she prefers, e.g. by requesting context or help information.

The selection of a certain dialogue plan in the starting phase of the dialogue can be based on a specific user's preference, or on a formal criterion, eg. minimal number of exchanges. However, going beyond purely subjective or more or less reductionistic criteria, we think that this is a matter of experience. The decision should be backed by the experience gathered in previous dialogues, with the same user or others, which have been successful in fulfilling their information needs.

An approach to problem solving based on past experiences is pursued in the area of case-based

reasoning (CBR). In our experimental work, we adapted the ideas of CBR to the requirements of a user-guidance component (for details cf Tißen 1991, 1993a,b; Stein, Thiel & Tißen 1992), which was developed as part of a prototypical multimedia information system.

3. 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. The COR model provides the general scheme for interpreting menu selections etc. as dialogue acts of the user, and, allows identifying appropriate continuations of an ongoing dialogue. While this aims at a coherent dialog 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 behaviour of a certain persons 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. 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".

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).

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 subset 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, 1993).

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 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.

4. The Concept of a Space of Information Seeking Strategies

The concept of a multi-dimensional space of information seeking strategies does seem to offer a means for identification of some general types of information-seeking dialogues. In the four-dimensional space suggested by Belkin, Marchetti & Cool (1993), for instance, each 'quadrant' of the space could have associated with it an interaction structure specific to that region. This would, of course, be a rather general dialogue, but it could form the basis for case specification at more detailed levels. Each general interaction structure, or script, could also serve as a substructure within other scripts, thus defining means for movement from one region of the information seeking strategy space to another. And individual cases could use sub-sequences of these scripts as models for parts of cases. This concept could lead us to a

poly-hierarchical classification of scripts and cases.

The idea of ISSs is meant to respond to and formalize the observation that when searching for information in some knowledge resource such as a textual database, people engage in a variety of different behaviors. Not only do individuals differ from one another in their interactions with such databases, but any one individual might do different things at different times. Indeed, even within the course of a single information-seeking episode, an individual will often change from one form of information-seeking to another. For instance, a person might begin an information-seeking episode by searching for some specified item; having found it (or not), the person might continue by browsing among other items that are conceptually (or physically) close to the original one; having found something of potential interest, the person might look more closely at it, in order to see if it will be useful; and so on.

We would like to be able to characterize this kind of behavior in such a way that would suggest means for supporting users in their various ISSs, and especially in their movement from one ISS to another. A means to accomplish this is to identify underlying dimensions of these behaviors. Doing this leads directly to the concept of a space of ISSs, in which any one particular kind of ISS is associated with some region in that space. Belkin, Marchetti and Cool (1993) proposed four such dimensions (Figure 2), which we use in our example system design here.

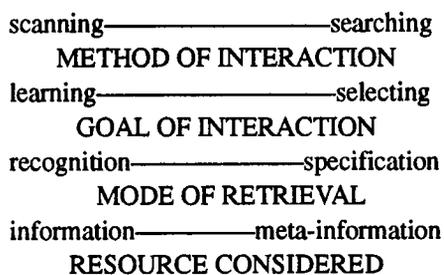


Figure 2. Dimensions of information-seeking strategies (after Belkin, Marchetti & Cool, 1993)

Using this model, any information-seeking episode can be described as activity within one region, characterized by the values along the dimensions, and by sequences of movement

from one region to another. Such a model suggests that specific support for the ISSs can be based upon knowledge of where the users lie on these dimensions, and upon prototypical, or possible, sequences of moves. The latter could constitute a grammar of ISSs.

5. The Relationships Among ISSs, Dialogue Scripts, and Cases

In general, we can construe the underlying model of the MERIT system as being based on the identification of cases of information seeking interactions, which are specific to particular goals, tasks, domains and users. These cases are instantiated by using a general, formal model of information seeking dialogues as the underlying representation which defines the structure of the case, with the surface level being based on an analysis of the requirements of the specific task being engaged in.

To illustrate how this might work, consider the following example. Any information-seeking episode begins with a common sequence, as follows:

- 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).
(note that this sequence can be iterated several times, as necessary.)

On the formal level these steps are seen as a well-formed sequence of discrete discourse acts (A: offer → B: accept → A: inform) which build a complete dialogue cycle according to the COR-model. 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 items of interest. This sequence may get extended or structured without changing this overall function, e.g. when the user revises his decision and makes another choice, or when he initiates a subdialogue in order to get more information on the system's offer or the suggested plan. An elaborate dialogue interface should account for such modifications.

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. Interaction in this space would be based on the following prototypical dialogue:

4 **user** Find me something that corresponds to this (specification of kinds of items to be retrieved).

5 **sys** a. Here's something that you'll like (presents the one item found, in specific 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.

6 **user** a. 1.I like this.--> 7
 2.I don't like this.
 --> 6c

user b. 1. Let's look at this one (chooses one from list)
sys 2. Here (presents item in specific detail).

user 3. I like this. --> 7
 4. I don't like this.

sys 5. How about one of these (presents list)?; or, if nothing left in list --> 6c

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). --> 3
 2. Let's quit. --> 9

7 **sys** Shall we save this and continue?

8 **user** a. Yes. --> 6b.2. or 6c
 b. No, let's continue.
 --> 6b.2. or 6c
 c. No, let's just quit.
 d. No, let's just save this and quit.

9 **sys** What would you like to do with what's been found (presents list of choices of action)? [conditional on something having been found]

10 **user** This (chooses something).

11 **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 formal COR-model this sequence can be interpreted as follows:

- The steps 4-6.a build a second dialogue cycle (A: request -> B: inform -> A: evaluate; or: A: request -> B: regret). Here, the main topic or hypertheme is the specification of items to be retrieved from the database.
- The 6.b1-4 steps are interpreted as a subdialogue (in 5.b) initiated by a user's request for more detailed information on one of the retrieved items. 6.b5 is a system's offer which also initiates a subdialogue with a similar function, i.e. to elaborate on information already given in overview.
- With 6.c the system turns again to the top-level of the dialogue offering a set of tactics for continuing search or for proceeding with a new dialogue plan. The sequences 7-8 and 9-10 can also be seen as negotiations on a meta-level where the system requests advice 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. This dialogue begins with the initial three steps common to all information-seeking episodes (see the previous example), 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:

4 **sys** a. You can look at the overall structure of the [name of meta-information resource] --> 5a, or
 b. I can suggest a possible starting point from which to view the structure --> 5b, or
 c. You can specify a starting point from which to view the structure --> 5c

5 **user** a. OK. --> 6a
 b. Let's start from this one (selects a starting point from the display) --> 6a
 c. Here's a starting point --> 6a, 6b

6 **sys** a. Here's the part of the structure that you wanted to see --> 7
 b. I couldn't find what you wanted in the structure. Here are some things that you could do --> 4

7 **user** a. Show me the structure around this item (selects from display) --> 6a
 b. I like this (selects from display) --> 8
 c. Let's quit --> 10

8 **sys** Shall we save this and continue?

9 **user** a. Yes --> 7a
 b. No, just continue. --> 7a
 c. Yes, but quit. --> 10
 d. No, and quit. --> 10

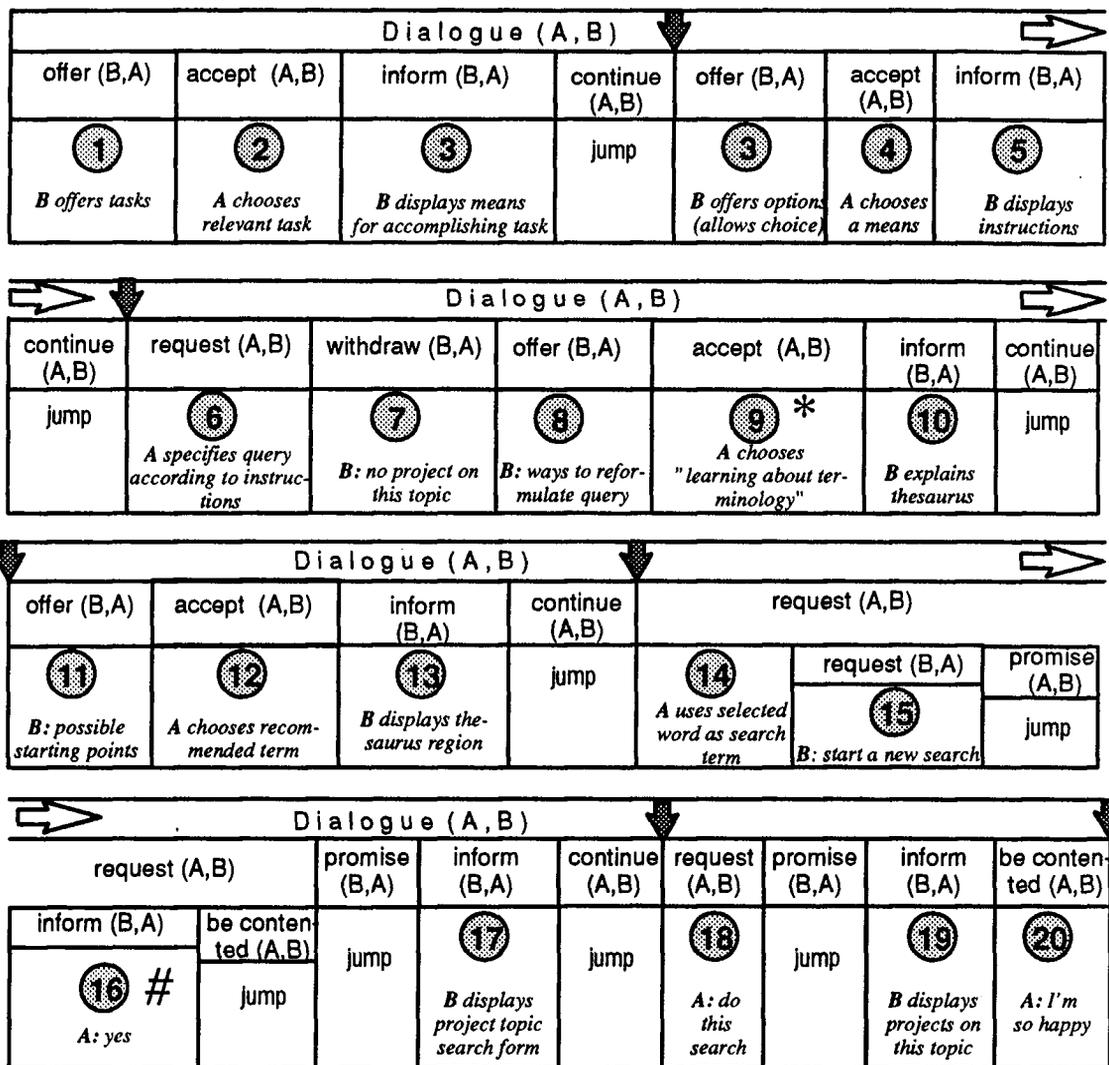
10 **sys** Goodbye.

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.

Next, we consider a single information-seeking episode, and describe it as instantiations of

aspects of the two general scripts we have outlined above; that is, 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. 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 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. Briefly, this case goes as follows (the numbers refer to the corresponding steps in figure 3).

The system offers the user the range of tasks it can support (1); the user chooses the task: Finding Partners on a Topic in a Program (2). The system displays the ways in which this task can be accomplished, which include starting with: a desired topic statement; a known project; an institution; or, a person (3). The user chooses to start by specifying a topic (4). This instantiates the topic specification case. The system displays the procedure for specifying a project topic (5), and the user inputs a topic description (search term) (6). In this example, the system informs the user that there is no project in the database indexed by this topic description (if something had been found, it, or they, would have been displayed) (7). 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 (8). The user decides to learn about topic terminology (9). This decision leads the user to the second script described above. The system then explains to the user that learning about topic terminology can be done using a thesaurus (10). The system then shows the user some recommended starting points for browsing in the thesaurus. These starting points are based on the user's initial search term, compared to previous cases in which that term has appeared, although not in the topic description field. The suggested terms will be topic terms that have appeared in similar cases. The possibilities are also given for the user to look at the structure of the thesaurus as a whole, or to enter some other



- ↓ Start/end of COR cycle
- * Note that this leads the user to a new script
- # Note that this choice returns the user to previous script

Figure 3. Schematic diagram of the example information seeking episode (case).

specific term as a starting point (11). This user chooses one of the recommended terms (12); the system then displays the thesaural structure for that term (13). The user recognizes one of the displayed terms as being relevant to her interests, and selects it as such (14). The system asks the user if she wants to search for projects on this topic (15), and the user says yes (16). The system then displays the project topic search form (with

the selected term filling the appropriate slot) for user confirmation. This returns the user to the first script of this information-seeking episode, selecting by searching with specification in an information resource (17). The user confirms that the query is correct by selecting the "search" button (18), and the system retrieves and displays the projects in the database on the specified topic (19). The user recognizes the

name of an institution associated with one of the projects that she is familiar with, and goes away happy (20).

Our example case shows how the interaction follows prespecified, 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 subcases. 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.

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.

The performance of the system as a cooperative retrieval interface depends on the appropriateness of the given cases and scripts. Since this can only be decided by experiments, the provision of scripts should be based on an evaluation. Such an evaluation has been designed for the MERIT system, and will be the next step in the development of this approach to information retrieval system design.

6. Conclusions

We have suggested a model of IR system design that provides a means for supporting users in their various information-seeking strategies, especially in their movement from one to another, through the specification of relationships among ISSs, dialogue structures and cases of specific information-seeking dialogues. This model is cast within a larger framework which views information retrieval as interaction, and for which the central problems of IR system design become problems of supporting users in their various information interactions.

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.

Despite these advantages, several issues remain to be addressed before we can properly make claims about the effectiveness of this model. First, it is necessary to do more empirical specification and validation of the dimensions of ISSs, which until now have been based primarily on informal reanalysis of existing data. And, of course, it is necessary to evaluate the performance of such a system. Both of these are now ongoing projects, the former being investigated at Rutgers University, and the latter at the GMD-IPSI, where an evaluation of the MERIT system using this model will be carried out.

References

- Belkin, N.J. & Cool, C. (1993)** The Concept of Information Seeking Strategies and its Use in the Design of Information Retrieval Systems. *This volume*.
- Belkin, N.J., Marchetti, P.G. & Cool, C. (1993)** BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval. In: *Information Processing and Management*, vol. 29 (4) (in press).
- Belkin, N.J. & Marchetti, P.G. (1990)** Determining the Functionality and Features of an Intelligent Interface to an Information Retrieval System. In: Vidick, J.-L. (ed.): *Proceedings of the 13th Annual International Conference on Research and Development in Information Retrieval (SIGIR '90)*, Brussels, 1990, pp. 151-178.

- Fawcett, R.P., van der Mije, A. & van Wissen, C. (1988)** Towards a Systemic Flowchart Model for Discourse. In: R.P. Fawcett & D. Young (eds.): *New Developments in Systemic Linguistics. Vol. 2: Theory and Application*. London: Pinter, 1988, pp. 116-143.
- Kerner, A. & Thiel, U. (1991)**. Graphical Support for Users' Inferences within Retrieval Dialogues. In: *Proceedings of IEEE Workshop on Visual Languages, Kobe/Japan, 1991*, pp. 211-216.
- MacNeil, R. (1991)** Generating Multimedia Presentations Automatically using TYRO, the Constraint, Case-Based Designer's Apprentice. In: *Proceedings of IEEE Workshop on Visual Languages, Kobe/Japan, 1991*, pp. 74-79.
- Maier, E. & Sitter, S. (1992)** An Extension of Rhetorical Structure Theory for the Treatment of Retrieval Dialogues. In: *CogSci '92, Proc. of the 14th Annual Conference of the Cognitive Science Society, Bloomington, Indiana, July 1992*. Hillsdale, NJ: Lawrence Erlbaum, 1992, pp. 968-973.
- McCoy, K.F. (1986)** The ROMPER system: Responding to object-related Misconceptions using Perspective. In: *Proc. 24th Ann. Meeting of the Association for Computational Linguistics, New York*.
- Mann, W.C. & Thompson, S.A. (1987)** Rhetorical Structure Theory: A Theory of Text Organization. In: L. Polanyi (ed.): *Discourse Structure*. Norwood, NJ: Ablex, 1987.
- Reichman, R. (1986)** Communication Paradigms for a Window System. In: Norman, D.A. & Draper, S.A. (eds): *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ & London, England: Lawrence Erlbaum, 1986, pp. 285-313.
- 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 et al.: North-Holland, 1989, pp. 209-228.
- Riesbeck, C.K. & Schank R.C. (1989)** *Inside Case-Based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum, 1989.
- 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, 1976, pp. 382-439.
- Sitter, S., & Stein, A. (1992)** Modeling the Illocutionary Aspects of Information-Seeking Dialogues. *Information Processing & Management*, Vol. 28 (2), 1992, pp. 165-180.
- Schank, R. & Abelson, R. (1977)** *Scripts, Plans, Goals and Understanding*. Hillsdale, NJ: Lawrence Erlbaum, 1977.
- Stein, A., Thiel, U., & Tißen, A. (1992)** Knowledge-Based Control of Visual Dialogues in Information Systems. In: *Proceedings International Workshop on Advanced Visual Interfaces (AVI '92), Rome, Italy, May 27-29, 1992*. Singapore: World Scientific Press, 1992. (in press)
- 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.): *SIGIR '91, Proc. of the 14th Annual International Conference on Research and Development in Information Retrieval, Chicago, October 1991*. New York: ACM Press, 1991, pp. 152-161.
- Tißen, A. (1993a)** Knowledge Bases for User Guidance in Information Seeking Dialogues. In: *Proceedings of the 1993 Int. Workshop on Intelligent User Interfaces, January 4-7, 1993, Orlando, FLA, 1993*
- Tißen, A. (1993b)** How to use Cases in Information Seeking Processes. *This volume*.
- Winograd, T. & Flores, F. (1986)** *Understanding Computers and Cognition*. Norwood/NJ: Ablex, 1986.