How to Use Cases for Information Seeking Processes

Anne Tißen
Integrated Publication and Information Systems Institute (IPSI)
Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD)
Dolivostr. 15, D – 6100 Darmstadt
tissen@darmstadt.gmd.de

Abstract

Information seeking activities, whether in databases, knowledge bases, or any kind of information system are highly interactive processes. User interfaces to information systems should enable the end user, to retrieve the data without help from an intermediary expert. Knowledge about information seeking tasks and strategies of the user, as well as the related system's functionality are starting points to build up cases of information seeking dialogues and to adapt cases to a new information need. We use a case for two purposes: first, for controlling the interaction, that means going forward and backward in the dialogue plan or history, and second, as a means to guide the user through the information space and through the interaction with the system. In contrast to most domains of case-based reasoners, our domain is underconstrained and highly interactive. Therefore, the adaptation of cases can be done only with mixed initiative of user and system. Our approach is implemented in a prototype of a case-based dialogue manager (CADI), applied in the MERIT interface system to the CORDIS databases. Currently, we are implementing CAIRO (Case-Based Dialogues for Information Retrieval Objectives), a further interface system to CORDIS data, now focusing on case-based user guidance according to information seeking strategies.

Keywords: case-based dialogues; information retrieval tasks and strategies; intelligent user interfaces;

Introduction

Information seeking activities, whether in databases, knowledge bases, or any kind of information system are highly interactive processes. Mostly, in the beginning of the interaction the information seeker does not have a strong plan of how to find the information she is looking for. Instead, the response to each single request for data may influence the next or future actions in her search. Traditionally, this process is a human-human interaction between an information seeker and an information provider, who offers support and access to the information system. User interfaces to information systems should enable the end user, to retrieve the data without help from an intermediary expert.

Modeling information seeking processes come across typical problems in knowledge-based systems: the user's knowledge about the system and the stored data is incomplete, it is vaguely formulated, and it is neither consistent nor persistent during a dialogue session. The best match between the user's information needs and the system's capabilities to find and to present information according to the user's intention and according to the dialogue situation is often not obvious. For this reason, a top-down dialogue, starting with a global goal and solving all subgoals, is not realistic in information seeking.

Case-based Reasoning techniques are supposed to overcome those modeling problems [cf. Kolodner/ Simpson 86], if the problem solving domain, here information seeking, is adequate to a data-driven, not a goal-driven, technique. Instead of solving problems using a general task model, case-based reasoners start with a set of previous cases, and adapt a selected case to the new problem [cf. Riesbeck/ Schank 89]. The basic idea is to use positive experience, rather than solving a problem from scratch. Most important aspects in adaptive planning [cf. Alterman 86/89, Hammond 89] are the treatment of plans specific to the application and situation matching to use an old plan to interpret the current situation. Cases in our context are instantiations of dialogue plans [cf. Carberry 88], describing a sequence of dialogue steps, which are related to an information seeking task. In an interactive system, case adaptation takes place with mixed initiative of user and system. In order to develop cases to start with in our application, we extracted information needs by guided interviews with potential users.

We use a case for two purposes: first, for controlling the interaction, that means going forward and backward in the dialogue plan or history, and second, as a means to guide the user through the information space and through the interaction with the system. For guiding the user in an information seeking dialogue with an information system we need different kinds of background knowledge [cf. Tißen 93]. The partition into several knowledge sources is necessary to make the interface system portable into multiple dimensions: according to the task in the domain, according to the user, or to other

database systems. During interaction, we combine all kinds of knowledge in order to adapt to current situations: (1) knowledge about the dialogue structure, represented as a dialogue plan; the expected steps, represented in the selected case, as well as the executed steps, represented as states in the history; (2) knowledge about the domain, making the structure of the information system transparent and supporting query formulation; (3) knowledge about the user, in particular, the possibility of making dialogue knowledge user-specific.

We implemented the case-based dialogue manager CADI [cf. Tißen 91; Stein/ Thiel/ Tißen 92] as a submodule of the MERIT[1] interface system to information systems. Both were tested in the domain of the CORDIS database about research programs, projects, and project partners. This version of CADI already contained the main functionality for dialogue control, and retrieving/storing of cases in the case library. Case indexing and adaptation was mainly determined by object perspectives. The concept of *object perspectives* is derived from research on the focus of attention in coherent conversation or discourse [cf. McCoy/ Cheng 90, see also Reichman 89] and adapted for controlling dialogues containing information seeking and information presentation actions. Perspectives are used as means to build cases from a thematically structured point of view. However, it was difficult to make the system's behavior, suggesting a next step, transparent to the user. modeling a case as a discourse of themes needs means to make the discourse coherent. In our domain, in which the information need of the user is vague and may often change during the interaction, thematical progression on the basis of object perspectives as the only means for case adaptation, seems to be too weak. We need additional knowledge about the user's goals.

To overcome this problem in the new version, called CAIRO[2] ("Case-Based Dialogues for Information Retrieval Objectives"), we make use of information seeking tasks for modeling the functionality of the system and we make use of information seeking strategies for describing user goals. This way, user goals are used in the case-based dialogue manager for indexing and adapting cases.

In this paper, we start with a catalogue of information seeking tasks and strategies and a short overview on a case-based dialogue manager. Then, we describe how information seeking tasks and strategies can be mapped to dialogue structures and be used for case indexing, and how we control the interaction and guide the user through the information space. Before the paper finishes with conclusions and an outline to future work, we present the current state of the implementations of the new prototype, the CAIRO system.

Information Seeking Tasks and Strategies

Information seeking tasks

The cognitive view of information retrieval as well as the demand for intelligent, interactive interfaces of retrieval systems are presented comprehensively in [cf. Ingwersen 92]. Belkin and Marchetti [cf. Belkin/ Marchetti 90] examined the information seeking (IS) process and presented a cognitive task analysis, containing tasks as *searching*, *specifying*, and *selecting* in information sources. Although they suggest a task model, they even stress that the information seeking process is inherently interactive and that intelligent interfaces need more interaction functionality. For that reason, in the BRAQUE system [cf. Belkin/ Marchetti/ Cool 93] they extend their task analysis and add *browsing*, *recognizing*, and *learning* in *information* and *meta information sources* to the set of information seeking tasks.

Information seeking strategies

In order to relate IS tasks and strategies, in BRAQUE they arrange these cognitive tasks in four dimensions: the method of interaction (browsing/ searching), the goal of interaction (learning/ selecting), the mode of retrieval (recognition/ specification), and the resource considered (information items/ meta information). They term combinations of those factors as 'information seeking strategies', describing behavior like, e.g. "searching for items similar to some known item; ...; looking around for something interesting...; inspecting items and their content; ...".

Belkin[3] took the strategy determined by 'searching', 'selecting', and 'specifying' in information sources and developed a script describing the interaction process [cf. Belkin et al. 93b]. He suggested scripts of interaction behavior containing conversational roles of speaker and hearer. This approach is influenced by the research on a conversational dialogue model for information seeking done by Sitter and Stein [cf. Sitter/ Stein 92]. A script contains dialogue steps related to IS functions for the user's request, for the system's display of the results, and a step for feedback of the user. Then, we

1. MERIT ("Multimedia Extensions of Retrieval Interaction Tools") has a SYBASE interface to the CORDIS data. The Cordis databases are offered by ECHO, the official host organization of the Commission of the European Community (CEC). For MERIT, we use subsets of the original databases covering about 180 programs with ca. 900 projects funded by different research programs. About 6000 organizations are involved in project consortia.

2. CAIRO (as well as CADI and MERIT) runs on SUN color workstations (SPARC stations) and is written in CommonLISP. Its knowledge bases are based on CRL, a frame representation language of Knowledge Craft [cf. KC 89]. The Cordis data were transferred into a LISP intermediate representation and then automatically converted to a netlike frame structure in CRL. For the graphical interface, CAIRO uses HyperNeWS, a NeWS-based graphics environment developed at the Turing Institute in Glasgow [cf. vanHoff 89].

3. Personal discussions with Nick Belkin during a collaboration at IPSI in July 1992.

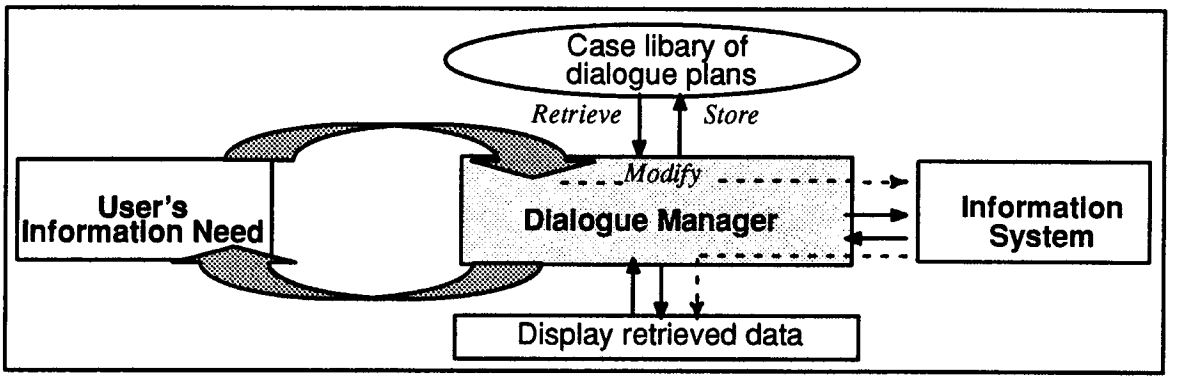

Fig. 1 : Schema of Case-based Dialogues for Information Seeking

took the results of the interview with potential users of our application system, and instantiated the scripts with domain knowledge to solve specific information seeking tasks. In this paper, I suggest that a script describing the interaction following a specific strategy can be mapped to parts of cases of dialogue plans, the dialogue sequences and cases can be build from a loosely ordered set of dialogue sequences.

Let me give an example of the first two dialogue sequences of a case about 'Looking for proposed projects on a known topic'. Here, the user may start with a search, specifying a list of known, relevant research programs. In a second dialogue sequence she may change to another strategy in order to browse through the projects of the specified programs. Here, she can choose between again specification, e.g. specify the relevant projects by attributes like the duration of a project, project funding, substrings in the project title, etc., or, in an interaction mode of browsing and recognition, she may select project acronyms from the display of retrieved data and ask for a display with more detailed information.
This example demonstrates two important requirements for interfaces to information systems. First, they should have a flexible set for information seeking functionality according to cognitive tasks of the user. Combining the four dimensions BRAQUE distinguishes up to 16 strategies of information seeking. In CAIRO, they can be parameterized by domain specific data or perspectives and listed or aligned. Second, interface systems should be able to offer guidance through the space of IS strategies and their instantiation. Therefore, we suggest a dialogue model and choose a case-based approach for dialogue control.

Information seeking functionality

Now, cognitive tasks and strategies of the user have to be mapped to the functionality of the interface system to the information sources. Belkin, Marchetti, and Cool present a set of functions, influenced by the functionality of the Hyperline system [cf. Agosti/ Gradiengo/ Marchetti 1992]. They list a set of system capabilities for displaying information, selection and storage of items, search formulation, selection of initial points from displays for browsing activities, etc. Movements on function sets are a first step to come to a process model. By the next step, the formulation of scripts [cf. Schank/ Abelson 77], the process has to become interactive with mixed initiative of both, user and system.

In the new CAIRO system, we realized relevant subset of those functions. See chapter 5 for the description of the CAIRO implementations in order to realize a dialogue control in two different control modes: in the non-guided, user-controlled mode, the IS functions are called by the user directly, or in the guided control mode functions are parts of cases of dialogues and they will be suggested to the user as dialogue steps. For the second mode CAIRO needs a dialogue manager component which is able to operate on cases.

The Case-Based Dialogue Manager

Cases are represented as instantiations of dialogue plans, describing a typical working context in the domain, see fig.1. A dialogue plan consists of a sequence of dialogue steps, extended by a description of user inputs and system's outputs. This explicit representation of the dialogue and various knowledge bases about dialogue, domain, and user knowledge enable the interface system for adaptation [cf. Tißen 93]. In order to develop cases for our application to start with, we extracted information needs by guided interviews with potential users. Cases in our domain about research programs and projects describe dialogues to solve typical information needs of the user like e.g. 'Looking for partners for a new project about a known topic', 'Looking for contact persons for an ongoing project', 'Overview on proposed research programs', etc.

To operate on cases, the case-based dialogue manager CADI [for more detail see Tißen 91] has functions like *retrieve* a dialogue plan from the case library, *store* the ongoing dialogue as a plan in the case library, and *modify* a plan to the current goals of the user. In a first dialogue phase, the

introduction phase, a case will be selected according to the current information need. Then, during the whole dialogue, the system offers the user the selected case as a means for guiding her in multiple dimensions: (1) navigating through the information space, (2) focusing information, (3) switching the information seeking strategies, or (4) switching to alternative presentation forms [cf. Kerner/ Thiel 91]. In a final phase, the user may permanently store the dialogue with all the modifications. This will be represented as a new case and stored in the case library.

As well as the dialogue model, the case library is known before starting an interaction. The case library may increase from session to session, and needs an own management for retrieving and storing cases. The content is domain dependent, and in addition, it may be user specific.

Integrating Information Seeking Tasks and Strategies into Cases of Dialogues

Mapping tasks and strategies onto the dialogue model

In the dialogue model, a dialogue plan is described as sequences of dialogue steps. In MERIT, cases of dialogues are modelled according to a thematical progression, that means that each dialogue sequence is related to a theme, described by a perspective, and that there are links between those themes, which make the dialogue coherent. During tests with users unfamiliar with the system, we got the experience that the topical coherence is not always obvious to the user. To overcome this problem, we make use of results of the cognitive tasks analysis and started a new implementation. In the new CAIRO system, a dialogue step describes perspectives to the database and the interaction behavior for a specific task. There are two kinds of dialogue steps: requests (of the user to the system) and inform steps (of the system to the user) according to dialogue acts at the main dialogue level of the conversational roles model of [cf. Sitter/ Stein 92]. We can enrich the interaction with IS functionality and adapt it to the working context of information seeking.

Strategies will be mapped to sequences of dialogue steps of request and inform acts. The dimensions of the selected strategy are mapped to user goals which are used to index the sequence, and consequently they are used to index the case.

Modes for dialogue control

In order to offer the IS functions directly at the interface, we are implementing two different modes for dialogue control: a guided, case-based mode and a user controlled, non-guided mode. In the case-based, *guided mode* the system guides the user through an information seeking dialogue, that means that the interaction takes place according to a previously stored successful dialogue, a case, going forward step by step in this case. We provide different instantiations of 'continue' and 'go back' operations in dialogue control and make them transparent to the user at the interface level. 'Continue' and 'go back' can be specified according to characteristic points in the dialogue plan (for continue) or in the dialogue history (for go back). During an ongoing dialogue CADI creates a history of the interaction in form a stack of dialogue states. From this history the CADI storer can generate a new case automatically. That means, that the new case is developed from (the) old case(s) and IS tasks, the user has inserted in the non-guided mode. In addition to the control of dialogue steps initiating related information seeking functions, the system controls in the guided mode the whole interface layout (pop up/ hide, positioning of windows).
In the user-oriented, *non-guided mode* the user chooses the information seeking functions directly at the system's interface. There are pulldown menus where the user can select search, browse, display and feedback activities and controls specification of dialogue steps, e.g. choosing perspectives for a search formulation, switching between information seeking strategies, and information displays. The control of the whole interface has to be done by herself, including the control of the layout of the screen with multiple simultaneous opened windows. Switching between both interaction modes at the interface provide the feasibility to evaluate comparatively the guided dialogue control vs. a direct manipulative, user controlled interaction and to examine and improve the modeling of the dialogue and the cases.

CAIRO proposes the guided mode as the major dialogue mode. The non-guided mode will be handled as the minor mode, which should only be used if the cases are too restricted to a specific task, or if a single information seeking task should be inserted into the ongoing case. That is possible at any time during the dialogue.

Adaptation of cases

Let us reflect on user goals in information seeking processes. Examples can be formulated informally, like: I know <x> and want to know more about <y> (which implies that there is a relation between x and y!); I want to see data similar to <x>; I want to specify what kind of data I'm looking for; I want to look around, I want to look around on a specific subset; I want to learn about the database; I want to select data for later use; etc. It is obvious that they are not independent from each other and that a hierarchical classification of user goals is not possible. Therefore, we follow the approach of spaces of information seeking strategies, build up by four dimensions of information seeking behavior. We take these dimensions, to index dialogue sequences according to information seeking behavior.

If we look again to the examples of user goals given above, we can differentiate kinds of adaptation. First, as we already did in MERIT's version of CADI, the perspectives, focusing on a partition of the database, can be flexibly modified. Perspectives are organized in a hierarchical structure, so that a shift, e.g. from organizational aspects to a thematic search of projects ('Looking for projects about ...'), as well as zooming into more detail, or zooming out to an overview, can be realized easily by a single dialogue step. Now, in the

Information seeking requests	**Find known data**	by specification or recognition of the name
	Search	by specification on attributes
	Browse	use relations between concepts
Displays and display requests	**Show 1 item**	in detail
	Show n items	in a list or table
	Show overview	a set of lists
	Show next/ pevious item	of the list in detail
	Compare items	in detail
Feedback	**Select item(s)**	for subset
	Delete item(s)	from list
	Store item(s)	on history

Fig. 2 : information seeking tasks in CAIRO

new CAIRO version of CADI, we use some more adaptations on top of that. Another type of adaptation can reflect to what the user knows and what the user wants to know by selecting adequate information seeking strategies or by replacing strategies that are suggested in a case according to the user's goals. In addition, the user modifies the suggested case during the ongoing dialogue: she can replace search terms, attributes or perspectives, skip suggested sequences of dialogue steps, mix cases to a new case, etc. The new generated case is different from any other case. In order to find this case for later use, it has to be indexed with user goals about the strategy how to find information and goals about the requested information in terms of object perspectives.

The CAIRO System

Implementing a new version of an interface system to CORDIS data aims at investigating the concept of a case-based dialogue manager for information seeking dialogues and to improve the CADI system. The new system focuses on the integration of information seeking tasks and strategies into the interface system. Therefore, we studied results from cognitive tasks analysis of information seeking processes, as done by Belkin, Marchetti, and Cool, and implemented most functions they proposed. We modified MERIT's interface features and designed more flexible input forms for user requests and interactive displays for the presentation of retrieved data, see an overview in fig. 2.

The users' request for information

Find known data is a special form of a search. Because the user knows the data item she can recognize it on a display, or she can specify it by a name or a unique attribute specification. Usually one data item will be found, that can directly be presented by the detailed presentation.

Search. If the user wants to search for data, she usually knows something about the data she is interested in, but the specification may be vague and/ or incomplete. For this kind of user request we implemented a flexible input form. The input form is titled by the main concept of the search, e.g. 'searching for projects', and is restricted to a specific perspective onto the main concept, e.g. the project partners. This perspective determines two lists of search attributes which will be suggested to the user, a first one with attributes that are closely to the theme of the perspective, e.g. in the project partner perspective 'name of partner organization', and a second list with attributes to limit the dataset, e.g. the country of a partner organization, or a program, the partner should have experience with. Therefore, the input form is divided into two areas, on the left side for gathering data (internally that is translated to an 'or' combination of search terms), the other one on the right side for reducing this data (combined by 'and'). This is an attempt to hide the logical operators 'and' and 'or' at the system's interface and to simplify the reformulation of search specifications.
When the user has accepted the suggested search attributes, she can formulate one or more search terms in a text field, or, she selects search terms from a list of suggestions offered by the system. Then, a query will be generated and a simple information retriever component will start a substring search in the information space. This kind of search form is flexible in that sense, that it presents a well prepared partition of the information space by the concept of perspectives. However, all suggestions can be modified by the user interactively.

Browse. If the user wants to browse through an information space the information need is only vague, is hard to formulate as a search term, or the user explores the information space around a specific known data item. In this case, the goal of interaction is a kind of 'learning'. At first, the user needs a specification of the information space and then a starting point for browsing. The information space can be the complete data, or it can be restricted to a specific subset like the current retrieved data, or a subset stored previously in the

ongoing dialogue. The starting point is always a known item, that really exists in the database. The interaction can be done without input forms. Browsing is a more presentation oriented activity than other types of requests for information. The system presents information to the user and the user only selects an item to ask for more detailed information. Consequently, browsing functionality is determined by the functionality of interactive displays of an interface system.
On the implementation level, browsing makes use of the semantic net of domain concepts in the domain knowledge base, which is implemented in the frame representation language CRL [cf. KC 89]. Using 'CRL relations', navigational steps from one data item to related data items were easy to realize. Browsing is hard to handle at the system's interface. Each browsing step may create a new window and the screen will soon be overloaded, the well-known problem of 'being lost in hypertext'. To avoid this, we have only one overview and one list display, one display of details for each concept. An additional display will only be generated for comparing two items of the same kind.

The presentation of retrieved information

Retrieved information can be displayed in three different kinds of interactive displays: a list display presenting all retrieved data, e.g. all retrieved projects; an overview display, with requested data and all related data presented simultaneously; a detailed display with detailed information about a single data item. Interactive displays need some specification of the interaction taken place directly on the presented items. In CAIRO, on overview and detailed displays the selection of an items means a request for detailed information. On lists we have various additional functions, so that the user can manipulate the list for later (re-)use during the following interaction. Now, let us have a closer look at each kind of display.

The *list display* presents all data of the concept, the user is looking for, in a list of interactive items. For example, if the user has specified a search for projects, she will get a list of project names. For the list, the system offers special functionality: to sort the list, to delete items from the list, to mark or unmark items on the list to create a subset for further operations like storing in a subset or on the history of data. The default for selecting an item with the mouse is to present the detailed display for the chosen data item.
This kind of display can be easily extended to a table presentation, presenting the name of the data concept in the first column, and attributes in further columns. Using tables data items can be presented according to object perspectives which describe i.e. situation specific sets of attributes of the concept, e.g. the thematical perspective on programs contains attributes like title, general information, and subprograms, describing what the program is about. A modification of a perspective can easily be done and presented to the user. The main problem of presenting tables is the optimization of the table layout. For this reason, at the moment we present perspectives without tables and use lists of structured items instead.

The overview display presents all data sorted by the concept class they belong to. In our domain concepts are e.g. project, program, organization, person. Each retrieved data is presented by a short name which is interactive, that means that it can be selected by the user to see details. It can be used as a starting point for browsing in the information space. The overview display can be used to present a specific set of data, as the set of current retrieved data, the previous data set, the complete data set, or any other data set which can be specified by the user. If the overview is used to display the retrieved data, it can be seen as an extension of the list display, where the retrieved data of the requested concept, e.g. projects, are presented together with all the related data, the program of the project, the project partner organizations, and the contact person.

Detailed displays. For each relevant concept of the domain there is one *detailed display.* It can be filled automatically with information about a single data item. Some examples: for the domain concept 'organization' CAIRO presents name and address, and country, the projects in which the organization participates in as a partner or prime contractor. For the domain concept 'program' the system presents the acronym, the title, the number of projects in the program, a list of all projects that exist in the database with more detailed information, a contact person, and a description of the program and its objectives.
This kind of display provides several starting points for browsing activities. Interactive display items are visualized by specific boxes so that the user can easily recognize possible starting points for browsing steps. To enable browsing through the list of retrieved data, the system offer buttons to go forward or backward directly without leaving the display. In addition to the browsing functionality, there are two new features, one to compare items and a second to read long text. To compare items CAIRO presents two detailed displays in two simultaneously visible windows.

Asking the user for feedback: Do you like it?

CAIRO offers some choices to the user to formulate her feedback, 'I like it!' or 'I don't like it!', according to the two polarities selecting and learning of the IS dimension 'goal of interaction'. In each polarity the user's overall goal is quite different.

In the 'selection mode', during the dialogue and, in particular, at the end of the dialogue, the user wants to have a kind of a result, usually in form of a set of data items. Different modi for the presentation of the result are conceivable: directly on the screen, or as an electronic or printed document. We will support the 'selection' mode at the user interface by an interactive history of retrieved data, the user has selected for intermediate storing or storing as a result.

In the 'learning mode', we assume that the user is mainly interested, to get a feeling what kind of information is in the database, for the granularity of stored information, how to

find relevant information, etc. From my point of view, feedback of the user makes sense mainly in the selecting mode. At the moment, there is no active user support for 'learning'. Therefore, in this mode we only hide the data history to avoid a cognitive overloading during the learning process.

I like it! Let us assume that the user wants to 'select' data from the database, and that the user has answered to a system's request for feedback that 'she likes it'; 'it' means the list of – eventually manipulated – retrieved data. For this case, CAIRO places those items, which are marked for selection, on a history of selected data. Although the data history is part of a history of dialogue states [cf. Tißen 91], it is visualized at the system's interface separately on a data stack. There, the concept name is interactive and a click means that the item will be presented in detail again. Later on, this history display should be manipulatable as well as the list display for retrieved data.

I don't like it! If the user doesn't like the retrieved data, the system provides some suggestions for modification, like a reformulation of the search formulation, skipping the current dialogue phase, switching to the 'learn mode' to inspect the database, then formulate a new query, etc.

Conclusions and Future Work

I have presented CAIRO, an interface system to CORDIS data, that is designed and developed as a dialogue system. Information seeking tasks and strategies are integrated into the dialogue model and related to sequences of dialogue steps. Using knowledge representation techniques, sequences can be arranged in a flexible way and adapted to user goals. For controlling the interaction, guiding the user through the information space, and for adaptation to user goals I suggested a case-based approach. The new implementation provides a lot of flexibility for the interface design and interaction behavior.

Most implementations for the new system are already done. Storer and retriever of the CADI system have to be adapted to new case indexes. Basic cases which describe very typical tasks in our domain can be taken from MERIT and extended with IS tasks and new indexes. Individual cases can be generated automatically by the CADI storer. It is still an open question, how to handle the interaction about user goals at the system's interface. This is a kind of meta dialogue, which is concerned with the dialogue itself and not with data or the retrieval process. In particular, this is an important issue for modeling the introduction dialogue. This is the dialogue phase, general to all dialogues, to retrieve and select a case from the case library which may help to solve the current information seeking problem.

Acknowledgements

I would like to thank Nick Belkin for fruitful discussions during his stay at IPSI, Adelheit Stein for valuable comments on a previous draft of this paper.

References

Agosti, M., Gradiengo, G., & Marchetti, P. G. (1992).
A hypertext environment for interacting with large textual databases. *Information Processing & Management, Vol. 28, No 3, 1992, p. .*

Alterman, R. (1986).
An Adaptive Planner. in: *Proc. of the 5th National Conference on Artificial Intelligence,* Menlo Park, California. American Association of AI., 1986, pp. 65-95.

Alterman, R. (1988).
Adaptive Planning. *Cognitive Science,* 12, 1988, pp. 393-421.

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

Belkin, N.J., Marchetti, P. G., Cool, C. (1993a).
BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval. *Information Processing & Management, Special Issue on Hypertext (in press), 1993.*

Belkin, N.J., Cool, C., Stein, A., & Thiel, U. (Belkin et al. 1993b)
Scripts on Information Seeking Strategies. to be presented at: CBR-IR workshop '93.

Carberry S., (1988).
Modeling the User's Plans and Goals. *Computational Linguistics (14/3), 23–37.*

McCoy, K. F., & Cheng, J. (1990).
Focus of Attention: Constraining what can be said next. In: *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Paris, Swartout, Mann (Eds.). Kluwer Academic Press, Norwell, MA.

Hammond, K. (1989).
Case-Based Planning. Viewing Planning as a Memory Task. (Perspectives in Artificial Intelligence; Vol. 1), San Diego, CA: Academic Press.

Hoff, A. van (ed.) (1989).
HyperNeWS User's Guide. The Turing Institute, Glasgow, UK, 1989

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

Kerner, A., and Thiel, U. (1991).
Graphical Support for Users' Inferences within Retrieval Dialogues. In: *Proc. of IEEE Workshop on Visual Languages, Kobe/Japan.* Los Alamitos, CA: IEEE Computer Society Press, 1991, pp. 211-216

KC (1989).
KC – *Knowledge Craft Manual*, version 4.0, Carnegie Group Inc., 1989

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

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

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

Schank, R, and Abelson R. (1977).
Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Associates, Hillsdale, N.J.

Sitter, S., and Stein, A. (1992).
Modeling the Illocutionary Aspects of Information-Seeking Dialogues. *Information Processing & Management, Vol. 28,* No 2, 1992, pp. 165-180

Stein, A., Thiel, U., and Tißen, A. (1992).
Knowledge Based Control of Visual Dialogues in Information Systems. In: *Proc. International Workshop on Advanced Visual Interfaces,* Rome, Italy, May 27–29, 1992.

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

Tißen, A. (1993).
Knowledge Bases for User Guidance in Information Seeking Dialogues. in: Wayne, D. G., et al. (eds.): *Proc. of the 1993 International Workshop on Intelligent User Interfaces (IWUI '93).* Orlando/ Florida, U.S.A., New York: ACM Press, 1993, pp. 149-156.