A Case-Based Architecture for a Dialogue Manager for Information-Seeking Processes

Anne Tißen
GMD
Integrated Publication and Information Systems Institute (IPSI)
Dolivostr. 15, D-6100 Darmstadt, FRG
e-mail: tissen@darmstadt.gmd.de

Abstract
In this paper, we propose a case-based architecture for a dialogue manager. The dialogue manager is one of the main components of the cognitive layer of an interface system for information-seeking processes. Information-seeking is a highly exploratory and navigational process and needs therefore elaborated interaction functionality. In our approach, this functionality will be provided by the dialogue manager operating on a set of case-based dialogue plans. In a case-based planning system a new plan will be generated by retrieving the plan which is most appropriate to the user's goals and adapting it dynamically during the ongoing dialogue. We propose a case-based architecture for two reasons. First, operating on old solutions provides a coherent framework which prevents the user from being 'lost in hyperspace'. Second, it allows flexible adaptations, domain dependent ones, using perspectives on domain objects, and domain independent ones, that change the sequence of dialogue steps.

Keywords: case-based reasoning, human-computer interaction, information-seeking

1 Introduction

How to build user interfaces to information systems can be examined on different layers of interaction: the physical, and the cognitive layer, both consisting of components for presentation and for dialogue.

Belkin and Marchetti [Belkin/Marchetti90] suggest a multi-layered task model and stress that the process of information retrieval is inherently interactive and that intelligent interfaces need more interaction functionality.
In our approach we concentrate on the interaction functionality and present a cognitive dialogue model where tasks are implicitly integrated as parts of dialogue actions. We break away from the classic information retrieval paradigm and the problem of query formulation and propose a navigational and exploratory information-seeking process [Bates86].

In a navigational process the user starts the interaction and controls the process. In the beginning she has only vague ideas in mind about her information need and how to explore in the information space. To prevent the user from being 'lost in hyperspace' the system uses a dialogue model for guiding the user and focuses the dialogue by means of object-oriented perspectives. The dialogue model is represented by dialogue plans which can be dynamically adapted to the ongoing dialogue.

We present an example system which at the physical layer uses a hypermedia interface, implemented in HyperNeWS [HyperNeWS89]. At the cognitive layer, knowledge-based dialogue and presentation components are realized in CommonLisp and a frame representation language. The system provides the basic functionality for the exploratory information access allowing the user to navigate between different presentations of retrieval results.
In order to achieve a flexible, yet coherent dialogue we propose an elaborated architecture for the dialogue component based on a case-based design. A case-based dialogue manager

(CADI) generates new dialogue plans from a set of old plans by system-driven modifications of dialogue steps and by initiating subdialogues for the correction of misunderstandings during the dialogue between the user and the system.

A case-based dialogue manager can be seen as a special adaptation of a case-based reasoner for the generation of dialogue plans. The general idea of case-based reasoning (CBR) is described in [Riesbeck/Schank89] as: "*A case-based reasoner solves new problems by adapting solutions that were used to solve old problems.*" or very similar in [Kolodner/Riesbeck89]: "*Case-based reasoning is reasoning from precedents, adapting old solutions to solve new problems, or retrieving old cases to illuminate aspects of the current situation.*" These definitions implicitly indicate two main components of a case-based system architecture: We need a "retriever" that will find the 'best' plan in the planning memory according to the user's goals and we need a "modifier" that will adapt this plan using modification rules and object restrictions. In accordance with this approach we describe information-seeking dialogues by a small set of alternative dialogue plans that get modified during the dialogue depending on specific dialogue knowledge and the preceding discourse.

Riesbeck and Schank [Riesbeck/Schank89, p.15] stated in their theory of reminding that "*Human experts are not systems of rules, they are libraries of experiences. Further, these libraries are adaptable.*" They mentioned that it is difficult to remember abstraction, but it is easy to remember a good coherent story. They illustrate *cases* as follows [Riesbeck/Schank89, p. 8]:

> "*If we go up to a librarian and describe our interests well enough, the librarian may do one of three things. The librarian might tell us where in the library to look for what we want, or she might be reminded of something that she read that suits us exactly, or she might be able to find a book that she has no idea of the actual contents of, but which fits what we have described perfectly.*"

Cases of information seeking processes are a semantic and therefore a domain-dependent means for reducing and focusing the navigation space in the interaction. They describe the sequence of action as scripts, and in this way provide a better framework for a coherent discourse. Additionally, the case-based framework offers domain independent mechanisms for flexible modifications and corrections of plans.

In section 2 we describe a *hypermedia system for information access* to the domain of conference information. Details about the dialogue manager containing the dialogue plan and perspectives will be given. Chapter 3 presents a *cased-based architecture* for the dialogue manager and in chapter 4 we discuss one submodule in detail, the plan modifier, which is responsible for a flexible adaptation of dialogue plans to user's goals. The last chapter contains our conclusions.

2 A Hypermedia System for Information Access

As a first prototype we developed a *System for Information about Conferences* (SIC). SIC is an interface system to a frame-oriented knowledge base which represents the information included in 'Calls for Participation/Paper' for workshop and tutorial announcements.

2.1 SIC dialogue

This prototype demonstrates the integration of dialogue and presentation components on the cognitive layer. The functionality of the interface system is tailored to the generation of multiple presentations for the complex information structure of the retrieved data.
The physical layer of the SIC user interface is implemented in HyperNeWS, an object-oriented hypermedia system, controlled by a CommonLisp Client with direct access to domain and dialogue knowledge bases via a Dialogue Manager[1]. A dialogue plan models the dialogue between the user and the system.

The concept of 'perspective' is used as a means to focus the dialogue in an information-seeking process. In contrast to task-oriented dialogues which strictly follow a well-structured, mostly hierarchical, and complete task structure, an information-seeking dialogue is a navigation process characterized by an 'open end' missing a distinct goal description, by topic shifts, by the necessity of subdialogues, e.g. for explanations, for commitments, or rejections. Therefore we describe the interaction between user and information system by a dialogue model, rather than taking a task-oriented approach. Task

1. The Dialogue Manager is implemented in Common Lisp, extended by CRL, the frame representation language of Knowledge Craft. Knowledge Craft is a trademark of Carnegie Group Inc.

elements for retrieval and presentation of information only exist implicitly, represented as actions of dialogue steps.

At the beginning of a SIC dialogue the user has to choose from a set of predefined perspectives to determine the main concept she is interested in. For instance, she wants to get information about the tutorials of the conference: the main concept is 'tutorial' and related concepts, such as 'speaker' or 'topics of the content', are determined by the perspective.
Starting at the main concept she can navigate through the knowledge base by posing a query and browsing through various graphical presentations. According to the selected perspective the generic, domain-independent dialogue plan will be instantiated and relevant attributes and related concepts will be determined. This notion of an 'object-oriented perspective' is rather similar to the ideas of McCoy [McCoy87]. She uses object perspective as a mechanism for modeling the highlighting of certain aspects of the domain influenced by the preceding discourse. While she uses salience values to describe a metric of object similarity, we have chosen a fuzzy set approach for mapping user terminology to the represented domain knowledge, described in more detail in [Kracker91]. In addition to the object-oriented description of a perspective we define a navigation path where the navigation space is restricted depending on the selected perspective.The navigation space is defined by the semantic distance between the initial concept and other related concepts. Instead of moving to further concepts directly, the user can either start a new dialogue or she can interrupt the current dialogue and start a subdialogue.

2.2 Interaction Style and Architecture

For the architecture of our SIC system we have chosen a multilayered design (see Fig. 1).

User
Physical layer: HyperMedia Interface
Cognitive layer: dialogue and presentation model
Semantic layer: conceptual domain model
Information System

Fig. 1 : Layers of an intelligent interface system

On the *physical layer* a user interface management system controls multimodal I/O between user and the interface. In the context of intelligent information systems the *semantic layer* can be optionally related to the information system or to the interface. It contains knowledge about the structure of the underlying data, abstract specification of concepts and attributes and interrelations between them. Defining a *cognitive layer* aims at a more user-oriented information system. It aims to bridge the gap between the model the user has in mind, the system's model of the domain and the information seeking process. Especially in information retrieval there are a lot of cognitive task models, e.g. Belkin and Marchetti suggest a multi-layer (effective, knowledge-based, intelligent) support for the end user and elaborate a cognitive task model. They stress that the process of information retrieval is inherently interactive and intelligent interfaces need more interaction functionality [Belkin/Marchetti90].

Information Retrieval using a computer system can be described as a process of requesting information from an information source by formulating queries and presenting the information, using various user support facilities, e.g. online thesauri. This process seems to be clearly structured into two main tasks of query formulation, respectively (re-)formulation, and presentation of the response, e.g. in form of listings of data. Belkin and Marchetti view information retrieval as progressive development or refinement of a search formulation [Belkin/Marchetti90, p.154]. In addition, new techniques for domain modeling with knowledge-based representation languages, multimodal I/O allow more sophisticated interaction techniques which require the development of an elaborated framework of Information Retrieval terminology. Waterworth and Chignell [Waterworth/Chignell90] present a model of 'information exploration'. In their model they differentiate three dimensions of information exploration:

- *Structural responsibility*, navigational or mediated:
  In a navigational process either the user himself/herself is responsible, or the user is not responsible and the process is mediated by a (human or system) information provider.
- *Target orientation*, querying or browsing:
  If the target is strongly defined we will have a query, or, if the target is less defined, the user will browse through the information space.
- *The interaction method*, referential or descriptive:
  The interaction method can either be referential, that means pointing to something, or descriptive, i.e. formulating a query in natural language, either as sentence or text.

Using the Hypermedia metaphor and related tools, like Hypercard or HyperNeWS, the interaction style of the retrieval process will become mainly navigational and browsing. That means, that the interaction is controlled by the user, without a clear target in mind. In the navigational and querying style the user is still responsible for controlling the search process, but a query can be formulated.

In our SIC prototype we use a navigational interaction style where the user is responsible for the dialogue control in combination with both forms of target orientation: starting with querying and then continuing with browsing interaction. The user is always interacting with the information system through a hypermedia interface, implemented in HyperNeWS (see Fig. 2). A HyperNeWS interface can be controlled low-level by scripts consisting of HyperNeWS commands, if necessary extended by postscript procedures, or it can be controlled by high-level systems, so-called 'clients'. Using a client provides a lot of flexibility that we take advantage of for dialogue control and presentation generation modelled on the cognitive layer. In the SIC system we implemented the client in CommonLisp extended by a frame-based knowledge representation language for the related knowledge bases.

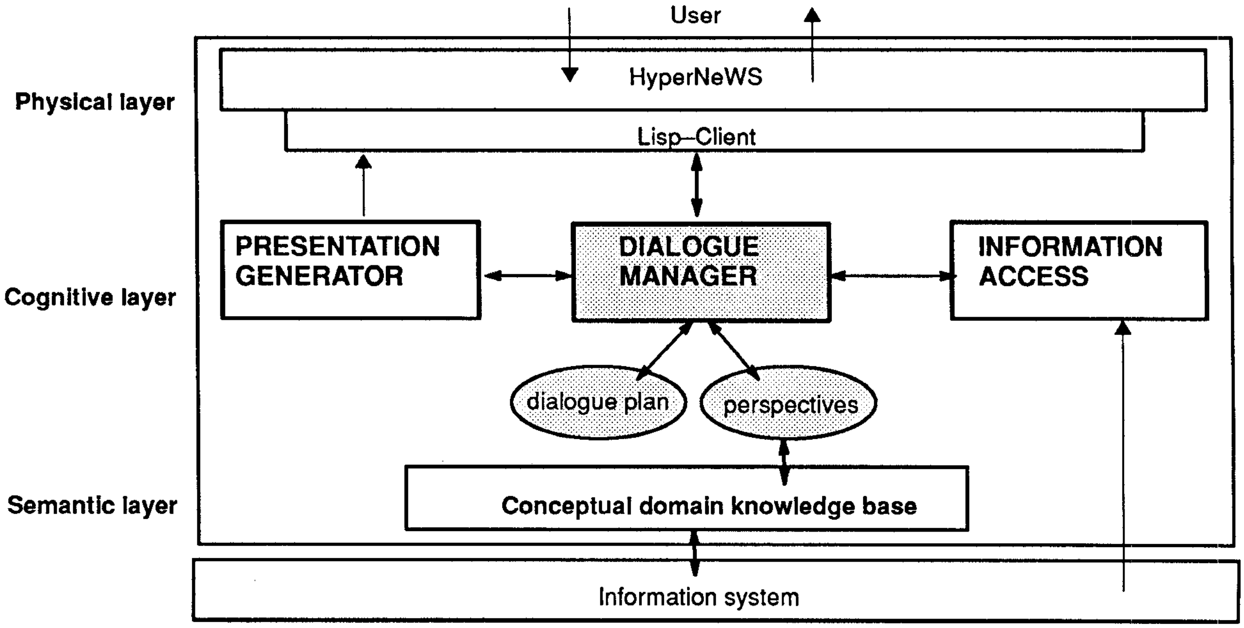

Fig. 2 : Architecture of the SIC prototype

2.3 Functionality of the Cognitive Layer

Now we describe the functionality of the components of the cognitive layer from the perspective of the dialogue manager. The complete interaction between user and system is controlled by the *dialogue manager*. It operates on the dialogue knowledge base consisting of a *dialogue plan* and a set of domain specific *perspectives*, which will be instantiated during a dialogue session. The dialogue plan consists of dialogue steps and states which are represented as frames. A perspective is represented as a frame with slots for the main concept, relevant attributes, related concepts, and the navigation path. Frames are used in combination with CRL[2] relations and CRL path descriptions.

A dialogue is roughly divided into four phases:

- the *introduction* phase where both participants have to introduce each other,
- the *queryformulation* process where the problem, here in particular the hyperthema of the dialogue, will be specified;
- *presentation of information and navigation* as a kind of problem solution,
- and at last *the closing of the dialogue.*

At present, in SIC we concentrate on the navigational aspects of the information seeking process. Therefore, the dialogue plan was especially elaborated for the third phase of information presentation and navigation and the second phase of

2. CRL is the frame representation language of Knowledge Craft.

query formulation is reduced to a single dialogue step, realized by a query form. On this query form there are a lot of possible actions controlled by the information access module, but they are represented in the dialogue model only as one global request.

In order to support the navigational and browsing interaction style, we have modelled a number of dialogue steps for the navigation through presentations, generated dynamically by the *presentation generator* [cf. Kerner/Thiel91]. Possible navigational dialogue steps are e.g. expanding a presentation in order to show additional or to show more concrete information, or reducing the presentation to a subset, which is then presented in another presentation form. Using them in a concrete dialogue situation is influenced by the selected perspective and the input of the user in the preceding dialogue.

3 A Case-based Architecture for the Dialogue Manager: CADI

The dialogue manager is the central component within the SIC architecture (see Fig. 2). Powerful dialogue functionality is required to provide flexible dialogue control reflecting the user's intentions seeking information in a hypermedia environment. In order to achieve this, we extend the dialogue manager of the interface system by a cognitive model.

How can a case-based design be used for cognitive dialogue modeling? Riesbeck and Schank [Riesbeck/Schank89] describe the basic cycle of a case-based reasoner in general as "input a problem, find a relevant old solution, adapt it". We use this general problem solving method as the basic idea for generating dialogue plans which support a flexible control of the interaction between a user and an information system.

3.1 Components of CADI

A case-based dialogue manager – CADI – (see Fig. 3) consists of several components: a *retriever* selecting the most appropriate dialogue plan according to the user's goals; a *modifier* that adapts the chosen plan to the user's goals by replacing objects and adding or deleting dialogue steps and, as a further component, a *corrector*, which handles subdialogues about misconceptions. The *storer* is responsible for updating the plan memory.

We can distinguish between planning components (retriever, modifier and corrector) and updating, or in a more elaborated version, learning components (storer). Each of the planning components operates on its own *knowledge base*: the retriever has access to the dialogue plans indexed by the generic goals they support; the modifier uses modification rules and object perspectives and the corrector has a set of misconceptions and correction strategies.

Object perspectives (see chapter 4) and misconceptions are built on top of the conceptual domain knowledge and have to be adapted when the conceptual knowledge changes, e.g. creating new perspectives for new (dialogue relevant) domain concepts or extending/ reducing the perspective definition when concepts get more or less attributes. The two knowledge bases for perspectives and misconceptions are domain-dependent and have to be defined before a dialogue is running. During the dialogue a dialogue history is built up automatically. This trace of the dialogue incorporates the user's inputs as well as actions of the system. The dynamic adaptation of dialogue plans during a dialogue session will be realized by rules.

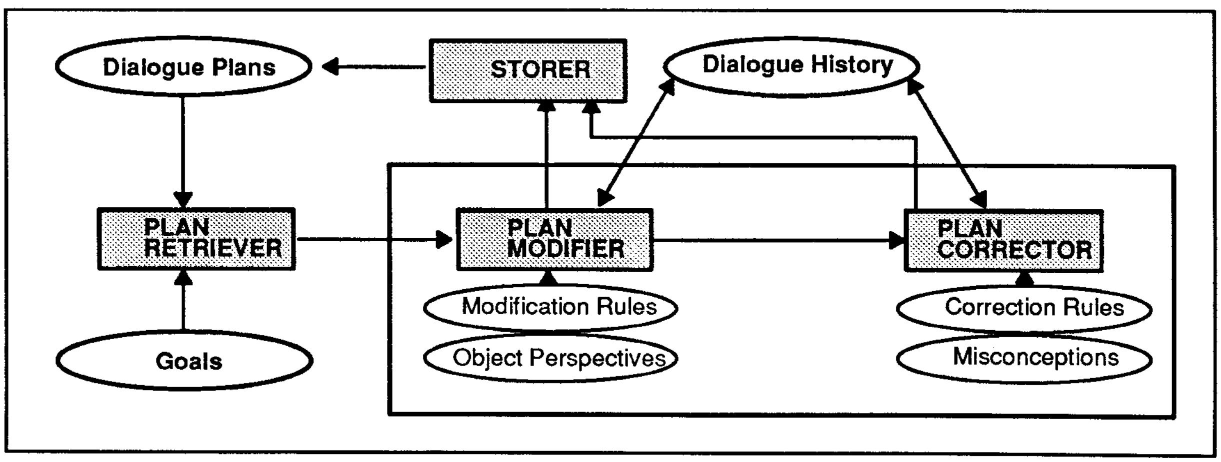

Fig. 3 : A case-based architecture for the dialogue manager (CADI)

Modification rules use object perspectives and correction rules access definitions of misconceptions. In addition, the actual entries of the dialogue history are relevant for a context-specific adaptation of a dialogue situation.

3.2 Dialogue Steps

Plans consist of a sequence of plan operators. In our case these plan operators are *dialogue steps*. Dialogue steps are the central concept for modeling and controlling the interaction between user and system in an information-seeking process. Since a detailed specification of the representation language is beyond the scope of this paper, we will limit our discussion to the different kinds of dialogue steps and their general structure.

Most of the dialogue steps are defined in the context of case-based plans, as *local dialogue steps*. In addition, there are various kinds of *global dialogue steps*, which can be executed independently of a concrete plan or plan sequence.
A first kind of global dialogue steps refer to the interaction style, defined for the whole dialogue sequence. The interaction style can be *mediated* or *navigational* (see chapter 2.2), or a mixture of both. The mediated style is realized through a case-based plan, which guides the user through a fixed sequence of dialogue steps. In the navigational style the systems offers all steps of the current plan (as a menu) of global steps to be selected by the user. Of course, only those steps can be offered whose preconditions are true.
Other global steps operate on the dialogue history, e.g. go back to a previous step in order to make changes, or jump ahead to a later step instead of following the actual plan.
Another, third kind of global steps is used to leave the information system or to suspend the information-seeking process, in order to use other software tools like editors, calendar tools, dictionaries, etc. They are to be realized by graphical input objects which are normally hidden and become visible only by additional steps.

A *dialogue step* is defined by basic components *preconditions, action* and *side-effects* (like plan operators in traditional approaches to plan representation as in STRIPS [Fikes/Nilsson71] or NOAH [Sacerdoti77]), extended in two dimensions: (1) *Input/Output descriptions* are added to generate input/output objects like menus, buttons, text input, icons, etc., at the system's interface and (2), *goal descriptions* and *sets of modification and correction rules* are added to modify the step according to the case-based approach. The components of a dialogue step are as follows:

Preconditions: have to be true before an action can be activated,

Example: Before a presentation of retrieved data can be generated, the search providing the data required by the selected presentation form must be completed;

Action: may be divided into two parts: a *main* action, which is always required, and one or more *additional* actions, which can be optional or required;

Example: Domain actions like 'formulate a query concerning workshops' may have an additional action like 'ask for terminology support for the workshop topic "xyz"'; a domain action like 'present the retrieved workshops and their related topics' as the main action can be elaborated by an additional action like 'explain the difference between the presentation forms table and list', which explicitly describes the relation between two dialogue steps as a kind of discourse relation.

Side-effects: changing the dialogue state or modify the dialogue history,

Example: Changing the state of the dialogue by a perspective shift: the user has started a dialogue under the workshop–perspective 'submitting a paper'; after workshops concerning her research topics have been presented, the user has navigated to topics similar to the topics of the retrieved workshops, e.g. advanced topics. Now she is interested in 'tutorials' about the advanced topics.
This perspective shift will trigger other side-effects like suspending the actual dialogue history, starting a new history for the subdialogue, initiating the subdialogue, etc.

Input–description: setting input mode, defining input objects;

Example: asking for a presentation of workshops and their organizers: in standard input mode input objects can be generated by the dialogue manager automatically (text input, menu selection, standard buttons for a simple 'goto-next-step' operation, ...), i.e. the main perspective of the dialogue is 'workshop' and the user selects the second concept 'organizers' from a menu; the input mode can be set to graphical input, in order to generate complex graphical input objects for making discourse relations like offer–accept, offer–withdraw, or offer–reject explicit to the user; this needs additional interface functionality;

Output–description: specifying constraints for the presentation generator [Kerner/Thiel91] concerning 'what' to present; the presentation generator has to decide 'how' to present.

Example: presenting 'workshops' and their 'topics': according to the goals the user defined in the beginning of the dialogue, the presentation should present complete information or in form of an overview.

Goals and perspectives: indexing the dialogue step according to goal descriptions;

Example: The user has asked for the maximum of information on workshops (by defining user goals in the first dialogue phase, see chapter 3.3) and the current step has the effect that the presented information will be expanded in order to reach the global goal;

Set of modification rules: modifying the dialogue step;

Example: changing the perspective:

if the global goal is 'topical information about workshops', i.e. "What is the workshop about?";

then change the output definition to a workshop-perspective focusing on topical information e.g. title, topics, related publications and hide organizational information, like the date of acceptance notification, number of copies for submitting a paper, etc.;

Set of correction rules: initiating a subdialogue:

Example: subdialogue about relevant attributes:

if there is a mismatch of presented and requested attributes of the current research topic, e.g. 'attributes of workshop topics',

and there is no automatic plan modification possible,

or the user has rejected the system's offer:

then start a subdialogue and ask the user to specify relevant attributes for workshops.

This representation of steps is a discourse- and interface-oriented extension of the traditional representation of plan operators found in STRIPS [Fikes/Nilsson71]. For text generation, as a specific kind of discourse, Moore and Paris [Moore/Paris89] suggest a plan language and plan structure that builds *detailed* text plans consisting of intentional, attentional and rhetorical structures. They apply their text generation system to the generation of responses in explanation dialogues.

We have chosen the plan representation with the same motivation: to model discourse aspects of an interaction. In contrast we distinguish two levels of dialogue modeling, a *strategical* level, covered by the case-based approach presented in this paper, and a *tactical* level, modeled by the 'pragmatic dialogue model', described in detail in [Sitter/Stein91] Their model represents intentional and rhetorical structure of a dialogue and is closer related to the *detailed* plan representation of Moore and Paris.

The strategical representation of the dialogue plan, as presented in this paper, stresses the interface-oriented extensions and takes care of the attentional structure of a discourse by representing user's goals and object perspectives explicitly.

Both levels, the strategical and the tactical level, are complementary. Therefore, each step on the strategical level is linked to the tactical level, so that it can be expanded to a sub–dialogue. This is initiated on the strategical level using correction rules, but controlled by the pragmatic dialogue model on the tactical level. Subdialogues can be started to achieve different kinds of goals, e.g. for explanation (paraphrasing or presenting background information), in order to reject or to withdraw actions, or for discourse goals, e.g. to distinguish between dialogue sequences either initiated by a system's *offer* to the user followed by a user's *accept* or initiated by a user's *request* followed by a system's *assert*. Both sequences lead to the same action which is described on the strategical level.

3.3 Phases of a Dialogue

In chapter 2 we have structured the dialogue of the SIC–system into four phases. Dialogues, in general, can be structured in the same way: introduction, query formulation, presenting the information and navigation, and closing the dialogue. The operations of the case-based dialogue manager in each of these phases can be described as follows:

During the *introduction* phase the user determines his/her goals interacting with the system and the system will retrieve the 'most appropriate' dialogue plan from the knowledge base of old and already represented plans. One can compare this to the task of an information provider who has a standard set of initial questions to an information seeker in order to give him/her instructions on how to find the information. The retrieved plan is case-based, and therefore instantiated with concrete data and not generalized. It is structured into dialogue steps and states where dialogue steps (see chapter 3.2) contain actions like query formulation or presentation generation parameterized by object and attribute specifications.

In our first version, we use a set of heuristic rules to select a plan from the plan memory. The retrieved plan will be adapted to the user's goals by the modifier component, to the extent it is possible in the beginning of the dialogue, e.g. domain objects of a dialogue step have to be exchanged by objects relevant to the goals; attribute lists have to be replaced, according to the perspective. Now, an initial plan is generated and the dialogue manager can start the main phases.

The second phase of *query formulation* and the third phase of *information presentation and navigation* are the parts of the

dialogue plan which have to be handled in a context-specific manner. In the cased-based approach we do not decompose both tasks into separate phases. Instead, case-based dialogue plans allow a permanent alternation of query (re-)formulation and presentation of information. These are task-oriented classifications of interaction steps. Taking a dialogue-oriented perspective on the interaction process both tasks can be treated in the same way, only with changing initiatives for an action between user and system.
The dialogue manager has a global and a local level of dialogue control. The global level, presented in Fig. 3 by the outer box, deals with the whole dialogue, the local level, presented in Fig. 3 by the inner box, handles the selection and adaptation of the single dialogue step. On the global level we have the knowledge base of old dialogue plans, the user's goals, and the dialogue history. When the dialogue manager has to determine the next dialogue step, it tests the preconditions of all steps and modification takes place depending on the context of the discourse. Only in case of misconceptions the corrector starts a correction subdialogue.

The modification of dialogue plans using perspectives and modification rules is discussed more closely in the following chapter. Especially the concept of *object perspectives* and how they influence the ongoing dialogue is described in detail.

Our system architecture is similar to the architecture of the CHEF-system which is the most well-known case-based reasoning system for planning [Hammond89]. In the domain of cooking recipes CHEF builds new plans from old plans on the basis of user's requests. In CHEF there is an additional component for anticipating goals in order to control the updating of the goal memory. The theoretical impact of the system lies in the development of domain independent strategies for failure repair and explanation followed by goal anticipation, motivated by a theory of 'learning from planning'.

In contrast to CHEF we handle the correction interactively by allowing subdialogues. A preliminary class of misconceptions occurs when mapping user's terminology onto the system's vocabulary is not possible. The correction rules handle a subdialogue clarifying the misconception and offer corrections. At present, the concept of 'misconception' is not yet worked out in detail.
Modifier and corrector both adapt the chosen plan to the user's goals and to the preceding dialogue. What is the difference between the two components? The modifier adapts the chosen plan to the user's goals automatically and the new plan or new plan step will be executed by the dialogue manager. Therefore it is not an interactive component. In the contrary, the corrector refers to a set of misconceptions and causal representations, why a misconception appears and how it can be corrected. This allows a kind of meta-dialogue, a 'dialogue on the dialogue'. In case of misconception a 'pragmatic dialogue model' is responsible for the dialogue control and the correction will be made interactively. This model [Sitter/Stein91] regulates the sequence of dialogue acts like request (for information), accept–reject, offer, etc. and controls embedded subdialogues, e.g. meta-dialogues.

Closing the dialogue in the case-based model will be executed by the storer. The knowledge base of dialogue plans need to be extended by adding the new adapted plans at the end of the dialogue sequence. Starting a new dialogue, the retriever can access the added plan which is again a complete and instantiated plan labeled with goal descriptions.

4 Object Perspectives for Case Modification

Perspectives on a domain can be used as an adaequate means for reducing the information space to a relevant subset and subsequently focusing the dialogue during the navigation process.

4.1 Object Perspectives: Definition and Examples

We use an *object-oriented* definition of 'perspectives': Perspectives can be defined for any domain object; normally one will only take those domain objects that are relevant for information seeking. For instance, in the conference domain relevant objects are: conferences, workshops, tutorials, topics, and persons in different roles (participant, organizer, author, speaker). Irrelevant objects would be, e.g. addresses.
A perspective of an object contains a list of relevant attributes and related objects, eventually but not necessarily a situation or condition, and a list of goals they support.
Each perspective is embedded into a hierarchy of perspectives. There are operations on perspectives for zooming in or out, and for changing to another topic in discourse. Although the definition of perspectives relates to objects, the hierarchy of perspectives is orthogonal to the generalization hierarchy of objects represented in the conceptual domain knowledge base.
A perspective shift may take place whenever the dialogue manager has to decide what to say or to do next.

Consider the following set of examples each of which illustrates different kinds of focusing for the same object in the domain of conference information:

- *workshop*-perspective 'organizing–before–the–conference'
  The organizer of a workshop is interested in the title, time and date, the participants and their addresses, and eventually what paper or what topics they present, but he is not interested, e.g. in the final schedule.
- *workshop*-perspective 'participation–schedule'
  A participant of the same workshop, who is already at the conference and schedules his/her timetable, is primarily interested in the topics, who will present a paper for relevant topics, and when papers will be presented.
- *workshop*-perspective 'submission–of–a–paper'
  A person who is looking for a workshop to submit a paper, has yet another perspective on workshops. Here, for example, the deadline is the focus of his/her interests, in addition to title and topic-list.

All three perspectives concern the same object, *workshop*, but they highlight different attributes of this object, like topics or deadline, and attributes of other related objects, like the address of a participant. In an intelligent interface to a conference system we expect that the information will be presented appropriately. These examples are similar to examples of stereotyped user models [Rich88]. But we determine perspectives according to the dialogue plan, because we stress the importance of the ongoing discourse.
During an information-seeking dialogue between user and information system the perspective will change continuously, according to the user's information need. As a consequence the thematical, or topical, structure can change as well as the amount of data, or the granularity of information, depending on whether the user expects an overview or a detailed retrieval result.

4.2 Case Modification and Perspective Shifts

The perspective knowledge base is organized in an object-oriented way with a link to the conceptual knowledge base of the domain. It contains a structured representation of perspectives relevant in the domain. The modifier uses these perspectives to make object-related adaptations inside a single dialogue step, but doesn't change the sequence of dialogue steps. I.e., the object(s), relevant in the actual step, or the attributes of an object can be changed according to the user's intentions. For each relevant domain object type there is a set of perspectives, related to possible goals they can support in an information seeking discourse. Goals differ e.g. with respect to the amount of data the user expects and the granularity of the presented information, e.g. when the user wants an overview about all workshops concerning a global topic, then an attribute which restricts the amount of data, should be presented, but it should not be used for restricting the query to the database. The modifier may expand the perspective to include more information, it may zoom to a subset of the original content, or it may shift to a perspective with a new thematic context.

How perspective shifts can support a coherent dialogue or discourse structure will be explained by shifting the last perspective example 'submission-of-a-paper', mentioned above:

- 'submission-of-a-paper' – zooming in:
  add all attributes concerning the submission of a paper, like the amount of copies, the address the paper should be submitted to, format requirements, ...
  or: add all attribute in order to clarify the topics, like super/-subtopics, reference to relevant publications, ...
- 'submission-of-a-paper' – zooming out:
  reduce the attributes to title and a list of the topics of the workshop which are relevant for the information seeker and the paper should focus on (e.g. because the deadline is o.k., and the paper can stress different topics depending on the chosen workshop), ...
- 'submission-of-a-paper' – semantic shift:
  a workshop is selected and the user wants information about organizational aspects, like registration fee, location, hotel arrangements, ...
  or: ask for researchers who have already published to similar topics; ...

Using the second knowledge base, the *modification rules*, the modifier changes the dialogue structure 'around the actual state'. That means that the dialogue plan will be adapted by *deleting, replacing and adding dialogue steps* according to the ongoing discourse which is represented in the dialogue history. Some examples should demonstrate what kind of modifications are possible: the actual step can be *deleted*, e.g. because the requested information is already presented on the screen; it can be *replaced*, e.g. because the presentation form is not adaequate for the information that has to be given to the user, and another presentation has to be generated; integrating *additional* steps can become necessary, e.g. when an example can help the user to understand the semantics of the nodes and edges of a net presentation, or when an additional step can offer a menu of relevant topics for the query before a query form can be filled out.

4.3 'Perspectives' in Discourse Theory

The idea of using perspectives for coherent dialogue modeling draws on work in related fields of discourse theory, especially attentional state and focus of attention [Grosz/Sidner86], moving in context spaces [Reichman85], and traversing focus trees [McCoy/Cheng90]. Most of this research is of theoretical impact on discourse theory. Most of the computational systems, which have been realized on the basis of this theoretical work, apply natural language as their primary interaction mode.

Our work differs from this research in two major ways. First, we stress the development of a computational model. Therefore we put it on the basis of a flexible system architecture using a case-based approach. Second, instead of regarding natural language interfaces we regard multimodal interaction modes.

5 Conclusions

Using Case-Based Reasoning (CBR) for controlling information seeking processes is a rather new idea. In the information seeking domain there is no generally accepted theory of problem solving. In particular, navigational and exploratory interaction styles do not lend themselves to a strong domain model such as a goal-oriented task hierarchy. From the user's perspective, information systems mostly contain incomplete domain knowledge that, in addition, changes continuously. These characteristics of the domain make CBR a sensible method for problem solving in information seeking.
Because information seeking is an inherently interactive process we focussed on the development of a dialogue manager and the dynamic modification of dialogue plans.

Apart from the plan modification a number of questions remain. One of the main problems is to find a set of basic cases of information seeking strategies. Furthermore, it is important to find a balance between domain independent information-seeking strategies and domain dependent though more powerful strategies. The idea of using object-oriented perspectives is one possibility to make use of domain knowledge for dialogue support.

References

Bates, M.J. (1986). An Exploratory Paradigm for Online Information Retrieval. In: Brookes, B.C. (Ed.), *Intelligent Information Systems for the Information Society*, Elsevier, 91–99.

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

McCoy, K.F. (1987). Contextual Effects on Responses to Misconceptions. In: Kempen, G (Ed.). *Natural Language Generation*. Dortrecht/NL, 43–54.

McCoy, K.F., & Cheng, J. (1990). Focus on 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.

Fikes, R., & Nilsson, N.J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2, 189–208.

Grosz, B.J., & Sidner, C.L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics (12)*, 175-204.

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

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

Kerner, A., & Thiel, U. (1991). Inferencing within Graphical Retrieval Dialogues. to be presented at: IEEE Workshop on Visual Languages, Kobe/Japan, Oct. 1991

Kolodner, J.L., & Riesbeck, C.K. (1989). *Case-Based Reasoning: Introduction, Overview, and Algorithms*. Tutorial presented at the 11th International Joint Conference on Artificial Intelligence (IJCAI '89), Detroit, Michigan.

Kracker, M. (1991). *Unscharfes assoziatives Begriffswissen zur Unterstützung der Formulierung von Datenbankanfragen*, Dissertation, TU Wien, April 1991, 167 Seiten. title in english: Fuzzy Associative Conceptual Knowledge for Supporting Query Formulation.

Moore, J.D., & Paris, C.L. (1989). Planning Text for Advisory Dialogues. In: *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics, ACL '89*, Vancouver/ Canada.

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

Rich, E. (1988). Stereotypes and User Modeling. In: Kobsa, A., & Wahlster, W. (Eds.), *User Modeling in Dialog Systems*, pp. 35–51, Springer-Verlag, Berlin–New York.

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

Sacerdoti, E.D. (1977). *A structure for plans and behavior*. New York. American Elsevier.

Sitter, S., & Stein, A. (Febr. 1991). *Modeling the illocutionary aspects of information-seeking dialogues*. Arbeitspapiere der GMD, No 515, Sankt Augustin: GMD; to be published in: *Information Processing & Management*, (forthcoming);

Waterworth J.A., & Chignell, M.H. (1990). *A Model of Information Exploration*. (submitted)