USER MODEL ACQUISITION HEURISTICS BASED ON DIALOGUE ACTS

Wolfgang Pohl, Alfred Kobsa and Oliver Kutter
Working Group Knowledge-Based Information Systems
Information Science Department, University of Konstanz
P.O. Box 5560-D73, D-78434 Konstanz, Germany
Tel.: +49-7531-882613, Fax: +49-7531-883065
{pohl,kobsa,kutter}@inf-wiss.uni-konstanz.de

Abstract

A wide-spread technique for user model acquisition is the use of acquisition heuristics, which are normally employed for inferring assumptions about the user's beliefs or goals from observed user actions. These beliefs or goals can often be characterized as presuppositions to communicative actions that the user performs. In the area of natural-language systems, presupposition analysis techniques have been applied for making assumptions about the dialogue partner based on the types of speech acts that he or she employs. In this paper, we will generalize this approach and investigate the analysis of so-called 'dialogue acts', i.e. communicative actions on the user interface whose execution entails user beliefs or goals as presuppositions of the action.

Dialogue act types with schematic presuppositions will be proposed as a means for formulating and generalizing user model acquisition heuristics. Several dialogue act types, both general ones applicable to any interactive system and specialized ones for an adaptive hypertext, are presented. The BGP-MS user modeling shell system contains a dialogue act analysis component that allows the developer of an adaptive application to define relevant dialogue act types and associated presupposition patterns. During run-time, the application can then inform BGP-MS about observed dialogue acts. BGP-MS will instantiate the presupposition patterns of the corresponding dialogue act type and enter them into the current user model.

L'aquisition d'un modèle d'utilisateur se fait communément par des heuristiques qui permettent à travers l'observation d'actions de l'utilisateur d'inférer des hypothèses sur les croyances ou les buts de celui-ci. Souvent ces croyances ou ces buts peuvent être considérés comme des présuppositions nécessaires à l'action communicative que l'utilisateur effectue. Dans le domaine du traitement du langage naturel, des techniques d'analyse de présuppositions ont été mises en œuvre pour établir des hypothèses sur l'interlocuteur sur la base du type d'acte de la parole qu'il utilise. Dans cet article, nous généralisons cette approche et étudions le traitement de ce que nous avons appelé 'actes du dialogue', c'est-à-dire d'actions communicatives effectuées par l'intermédiaire de l'interface homme-machine et dont l'exécution entraîne des croyances ou des buts de l'utilisateur en tant que présupposition à cette action.

Nous proposons l'utilisation de types d'actes du dialogue qui contiennent les schemas de présuppositions pour aider à la formulation et à la généralisation d'heuristiques pour l'aquisition du modèle d'utilisateur. Nous montrons plusieurs types d'acte, tant généraux en ce qu'ils s'appliquent à n'importe quel système interactif que spécialisés, ici pour un système hypertexte. Le système d'aide à la modélisation d'utilisateurs BGP-MS contient une composante de traitement d'actes du dialogue qui permet au développeur d'un logiciel adaptif de définir les types d'actes du dialogue utiles ainsi que la forme des présuppositions associées. Pendant l'exécution, le logiciel peut informer BGP-MS de l'observation de ces actes. BGP-MS instancie alors le schema des présuppositions concernées et incorpore celles-ci dans le modèle d'utilisateur courant.

Keywords: user modeling, user model acquisition, user modeling shell systems, adaptive hypertext, dialogue acts

International Workshop on the Design of Cooperative Systems, Antibes-Juan-les-Pins, France, 471-486, 1995.

1 Introduction: Making Assumptions Based on User Actions

As is the case with knowledge-based systems in general, acquiring and representing knowledge is crucial for user modeling in interactive software systems. In addition to representation and management mechanisms, user modeling components therefore must include suitable user model acquisition mechanisms (see [Wahlster and Kobsa, 1989; Chin, 1993]). The developed methods can be divided into two groups: those that extract *primary assumptions* about the user from his/her system input, and those that extract *secondary* (or *derivative*) *assumptions* from primary and secondary assumptions (like forward inferences or stereotype activation).

During the past few years, a number of tool systems for user modeling have been developed (the so-called user modeling shell systems; see [Finin, 1989; Kobsa, 1990; Brajnik and Tasso, 1992; Kay, 1994; Kobsa and Pohl, 1994]). They must provide acquisition, inference and retrieval mechanisms that are often used in user modeling components, and serve as the basis for the development of user modeling components in application systems. To date, however, none of the developed user modeling shell systems has included mechanisms that extract primary assumptions about the user from his/her system input. At first this is surprising, since the acquisition of a user model plays an important role in a user modeling component and therefore should be supported by a shell system. The omission is understandable, though, if one considers that a user modeling shell system must be domain-independent while heuristics for acquiring primary assumptions concerning the user are mostly domain-dependent. For example, if the user asks the system the following question:

> When is the next train to Montreal? [Allen, 1979]

then one would most likely assume that the user wants to go to Montreal on the next train. But this is only true in travel domains. The assumption is no longer valid in rail shipping domains (for example in [Allen and Schubert, 1993]), where it is more likely that the user may just want to ship a container or a freight car to Montreal.

However, there are also domain-independent heuristics that may lead from user input to new primary assumptions. The following ones can be found in the literature:

- *Correct use*: if the user employs objects correctly (e.g. operating system commands, mathematical operations, concepts), then the user is familiar with these objects [Chin, 1989; Nwana, 1991; Sukaviriya and Foley, 1993].
- *Incorrect use*: if the user uses objects incorrectly, then he/she is not familiar with them [Quilici, 1989; Hirschmann, 1990].
- *Request for explanation*: if the user requests an explanation for concepts, then he/she is not familiar with them [Chin, 1989; Boyle and Encarnacion, 1994].

- *Request for detail information*: if the user wants to be informed about objects in more detail, then he/she is familiar with them [Boyle and Encarnacion, 1994].
- *Feedback*: if user feedback concerning a system output that was based on certain assumptions in the user model is positive/negative, then the plausibility of these assumptions should be increased/decreased [Rich, 1979a; Rich, 1979b].

It seems to be common to at least the first four heuristics that assumptions about the user are derived from observed user actions, and that the assumptions can be understood as prerequisites to the actions. For example, the correct use of an object presumes that the user knows the object. It seems that a wide variety of domain-independent user model acquisition heuristics follows a common scheme, namely deriving the prerequisites of observed user actions.

This reminds one of the *presupposition analysis* technique that has been applied in natural-language dialogue systems to support the acquisition of a dialogue partner model [Kaplan, 1979; Kobsa, 1983; Kobsa, 1985]: a user utterance is analyzed with respect to the speech acts it verbalizes, and from each speech act the presuppositions are derived that must have been valid for the speaker in order to perform the act correctly. The method is particularly interesting if these derivations can be made without regard to the contents of the speech act, i.e. if they are only determined by its type (like 'question' or 'information').

We generalize the notion of natural-language speech acts to *dialogue acts* that may occur in human-computer interaction, following other speech-act based approaches in this area [Winograd, 1988; Sitter and Stein, 1992]. A dialogue act is independent of any specific user interface, i.e. it may be performed in a command interface, a direct-manipulative interface, a natural-language interface, etc. A *dialogue act type* comprises all dialogue acts with structurally equal presuppositions, only differing in the objects of the acts. A dialogue act is then an instance of a dialogue act type. A dialogue act type is normally parametrized and can be associated with a set of *presupposition patterns*, which schematically describe the presuppositions of all instances of the dialogue act type. We already saw two examples of dialogue act types above, namely a request for explanation and a request for detail information. A kind of *dialogue act analysis* can be applied in interactive systems if a set of such types along with their presupposition patterns has been defined: the presuppositions of an observed dialogue act can be computed by suitably instantiating the presupposition patterns of its type.

The user modeling shell system BGP-MS [Kobsa and Pohl, 1994] has been equipped with a *dialogue act analysis component* that supports the formation of primary assumptions about the user. The application system can inform BGP-MS about the dialogue act(s) that underlie an input operation of the user. BGP-MS then automatically enters all relevant user presuppositions of this dialogue act in a suitably instantiated form into the user model. This component saves the application system that utilizes BGP-MS for user modeling of having to derive possible assumptions about the user's knowledge or goals itself. A prerequisite is that the user model developer must introduce all dialogue act types that are relevant in the application to the dialogue act analysis component, along with their presupposition patterns. For this purpose, he can take advantage of the set of pre-defined and application-independent dialogue act types that is offered by BGP-MS. In most cases however, the developer will have to define additional dialogue act types that may occur in the specific application.

This paper describes how dialogue acts and dialogue act analysis can be used as a general mechanism to support the formation of primary assumptions from observed user input in a user modeling shell system. The principles involved in dialogue act analysis will be explained in the next section. Subsequently, we will show in more detail how dialogue act types as generalizations of speech act types can represent domain-independent user model acquisition heuristics. Examples of dialogue act types will be presented, which were identified by analyzing user interfaces in general as well as specifically an adaptive hypertext (for a detailed description of this analysis and all identified dialogue act types see [Kutter, 1994]). Afterwards, we will describe the dialogue act analysis component of the user modeling shell system BGP-MS, and discuss related work and future developments.

2 An Introduction to Dialogue Act Types

2.1 Dialogue Act Types Generalize Speech Act Types

According to Searle [Searle, 1969], every utterance in an interaction can be considered as a speech act. One of the aspects of a speech act is its function or role in the interaction, which is called the illocutive act and is characteristic of each utterance. In general, the illocutive aspect of an utterance can be described by natural-language verbs. Since verbs often have similar meanings, verb classifications should offer a good starting point for categorizing speech acts. A well-known example for English is [Wierzbicka, 1987], who distinguishes the following verb categories, among many others: command/request/order; question; consent/accept; information. These categories can be regarded as speech act types.

A basic and important feature of speech acts is that inferences can be made when they occur. On the one hand, the content of a speech act may bear logical consequences. For instance, if somebody asserts “The cat eats the mouse” the asserted proposition logically entails “In a while the mouse will be no more”. On the other hand, conclusions may be drawn based solely on the type of a speech act. For instance, the above assertion implies “I think the cat eats the mouse” by virtue of the fact that the person used a specific type of speech act (namely an assertion).

The latter inference may be generalized to the rule “If a dialogue partner asserts P, then it can be inferred that he thinks that P is the case”. Note that this inference is independent of the meaning of

P and has the form of a scheme in which P can be suitably instantiated for drawing inferences based on a concrete assertion. Since the propositions that can be inferred when a speech act occurs must be true in order that the speech act can be used correctly, we will call them 'presuppositions' of the speech act. Speech act types then have schematic 'presupposition patterns' associated.

Since our primary interest is not natural-language dialogue but human-computer interaction (HCI) in general, we will henceforth use the more general term 'dialogue act' instead of 'speech act' for referring to communicative actions in man-machine dialogue. One goal of this work then is to find dialogue act types that, like speech act types, have presupposition patterns and therefore allow content-independent derivations from the dialogue acts they subsume. They shall be used as heuristics for drawing useful assumptions about the beliefs or goals of the user of an interactive computer system when he is observed to perform a dialogue act. The assumptions then are made by appropriately instantiating the presupposition patterns of the respective dialogue act type. This application of dialogue act types for user model acquisition will be referred to as 'dialogue act analysis'.

Before showing examples of dialogue act types, we first discuss other work in the field of HCI that makes use of speech/dialogue act types.

2.2 Dialogue Acts in Other Work

Winograd and Flores [Winograd and Flores, 1986] use dialogue act types (i.e., speech act types in their terms) such as "request", "promise", "reject" and "accept" to model larger conversation structures in transition networks. An example for a conversation structure is a request for an action ("conversation for action"; CfA). Sitter and Stein [Sitter and Stein, 1992] start with a CfA model and develop a considerably complexer model of information-seeking dialogues. Their dialogue act types are very similar to those of Winograd and Flores, although they can be further developed and extended to complex dialogue contributions. For example, a "request" act can recursively include a complete dialogue for establishing information about the context of the request. The models developed in each of these studies can be used as a basis for implementing conversation [Winograd, 1988] or dialogue systems [Stein *et al.*, 1991]. User or partner models were either not mentioned or even indirectly considered unimportant in both studies. Occurring dialogue acts merely cause a state change in the dialogue model. Unfortunately, the dialogue act types used in all these papers appear to be too general to allow precise conclusions regarding the dialogue partner's knowledge, goals, etc., and are therefore not appropriate for our purposes.

The VIE-DPM system [Kobsa, 1985] uses presupposition analysis to form assumptions about the user in a natural-language dialogue system. It distinguishes the following dialogue act types: assertion, "yes/no" question, wh-question (which starts with "who", "what", etc.), and command. It is interesting to note that these dialogue act types are very similar to the speech act types mentioned in Section 2.1, one difference being that the category "question" becomes subclassified. The reason for this subclassification is that "yes/no" questions and wh-questions have different presupposition patterns.

3 Dialogue Act Types as User Model Acquisition Heuristics

Since we want to build a user model acquisition mechanism for a generally-usable user modeling shell system, we are particularly interested in dialogue act types that are not specific for a single application system or application domain only. This section presents examples of such dialogue act types and their presupposition patterns. They may be employed for acquiring assumptions about users of general-purpose interactive computer systems and of hypertext-like systems, respectively.

First, however, we have to explain the formal notation for presupposition patterns that will be used in this paper (with minor modifications, it has also been used in the implementation of dialogue act analysis). In section 2 we interpreted the statement "The cat eats the mouse" as an "assert" dialogue act with regard to the content *eat(cat,mouse)*. In our notation for presupposition patterns the symbol P (or $P(x)$ if a variable is included) refers to the content of a dialogue act. $P(x)$ may be preceded by the operators $\forall x$ and $\exists x$ (with their standard meanings) as well as ιx (with the meaning *that x*). In order to formalize beliefs and goals of the user we employ the modal operators B for *believes* and W for *wants*, and index them for modal subject identification with U, S and M for *User*, *System* and *Mutually*. $B_S W_U\ P$ is a simple example of a presupposition pattern and would roughly mean: The system "privately" assumes that the user wants to achieve P. In contrast, $B_M W_U\ P$ means that both S and U mutually believe $W_U\ P$ (meaning that not only $B_S W_U\ P$ holds true, but also $B_S B_U W_U\ P$, $B_S B_U B_S W_U\ P$, etc.).

3.1 Dialogue Act Types for Interactive Systems

In a first study, various interactive systems were examined for dialogue acts that are generally applicable. Natural-language speech acts were again the first reference point. Below are a few examples of dialogue act types that seem to occur in many kinds of interactive systems. Their presupposition patterns will be first described in English and then in our formal notation.

WH-QUESTION

Description: A user request for information that can be expressed in English with the interrogatives "who", "which", etc.

Example: Invoking a find command in the Apple Macintosh operating system would correspond to the following question: “Where is the file named XY?”.

Presupposition Patterns: It will be mutually assumed that …

1. …the user believes that what he wants to know exists;
2. …the user is not familiar with the desired information;
3. …the user believes that the system possesses the information;
4. …the user wants that he and the system both know the desired information.

Formalized Presupposition Patterns:

1. $B_M B_U \; \exists x \, P(x)$;
2. $B_M \neg B_U \; \iota x \, P(x)$;
3. $B_M B_U B_S \; \iota x \, P(x)$;
4. $B_M W_U B_M \; \iota x \, P(x)$

YN-ANSWER-YES

Description: A positive answer to a “yes/no” question of the system about the user’s intentions.

Example: After invoking the formatting command the system asks: “All data will be erased if the disk is formatted. Format the disk anyway [y/n]?”. The user enters “y”.

Presupposition Pattern: It will be mutually assumed that the user would like to reach the described state of affairs.

Formalized Presupposition Pattern: $B_M W_U \; P$

AGREE

Description: The user consents to an action announced by the system.

Example: After entering a false command the user is asked whether he/she would like to get help. The user clicks the OK button.

Presupposition Pattern: It will be mutually assumed that the user agrees to reach the described state of affairs – in no way does he desire the opposite.

Formalized Presupposition Pattern: $B_M \neg W_U \; \neg P$

Remark: It is important to note that the dialogue act types YN-ANSWER-YES and AGREE belong both to the verb category “consent/accept” listed above. However, they differ in their presupposition patterns and therefore have to be separated.

3.2 Dialogue Act Types for Adaptive Hypertext Systems

Unfortunately, general and application-independent dialogue act types do not seem to be very helpful as user model acquisition heuristics since the assumptions that can be inferred from them using dialogue act analysis are likewise general. Assumptions about the user that are more profitable for the purposes of an adaptive system must be more expressive than those derivable from the AGREE act and less obvious than those following from the YN-ANSWER-YES act. In the latter case an assumption is made about a user goal that immediately afterwards is met by the system, thus rendering the assumption irrelevant.

We therefore decided to analyze dialogue act types that are no longer completely domain-independent, but still apply to all interactive systems of a particular application class. In this subsection we will present some dialogue act types representing acquisition heuristics for adaptive hypertexts as described in [Boyle and Encarnacion, 1994; Kobsa *et al.*, 1994].[1] Since we concentrated on the interactive behavior of the hypertext (namely the possible user actions and the presentation of the document), these dialogue act types can also be found in other hypertexts showing similar behavior.

REQUEST-FOR-EXPLANATION (cf. section 1)

Description: The user can ask the system to explain a hotword of the hypertext[2] by clicking on it.

Example: In a hypertext node describing operating systems, a mouse click on the hotword "UNIX" opens a pop-up menu that offers "explanation" as one choice. If this item is selected, a hypertext node is shown that contains explanatory information about UNIX.

Presupposition Pattern: It can be mutually believed that the user does not know the concept denoted by the hotword or keyword.

Formalized Presupposition Pattern: $B_M \neg B_U \; concept(P)$[3]

Another dialogue act type that implements one of the user model acquisition heuristics mentioned in section 1 is REQUEST-FOR-DETAILED-INFORMATION. It occurs for example when the "more info" item of a hotword pop-up menu is selected and implies a mutual belief that the user already knows the concept under consideration. Other cases are more difficult: What can be concluded if the user ignores a hotword in the current hypertext node? Does the user know the corresponding concept? Or is he/she just not interested in a deeper understanding of the text, and therefore skipped the hotword? In our system, a good heuristic might be that if it is currently assumed that the user

[1] We assume that the reader is familiar with the basic concepts of hypertexts like node, link, hotword, and glossary.

[2] In our hypertext, only concepts of the domain become explained, while in other systems also complete propositions may be explained. In this case, the presuppositions below would not be restricted to concepts.

[3] $concept(P)$ is an expression on the lexical level. It refers to the concept named P.

does not know the concept (since it was e.g. derived from a REQUEST-FOR-EXPLANATION), the contrary should be concluded from now on. In order to represent such a heuristic, an **if** … **then** … construct had to be introduced:

IGNORE-HOTWORD

Description: The user does not perform any action on a hotword in a hypertext node.

Example: A node contains the hotword "UNIX". The user does not use it as a starting point for further navigation.

Presupposition Pattern: If the user is currently assumed not to know the concept denoted by the hotword, then it can be mutually believed from now on that he/she knows it.

Formalized Presupposition Pattern: **if** $B_S \neg B_U\ concept(P)$ **then** $B_M B_U\ concept(P)$

Dialogue acts of this and similar types that correspond to "non-actions" of the user are difficult to detect. IGNORE-HOTWORD dialogue acts can be reported by the application system, when the user leaves the current hypertext node – all hotwords that no action was performed upon may be regarded as "ignored". In general, sophisticated observation techniques may be required for a decision about reporting "non-action" dialogue acts (cf. [Kutter, 1994]).

4 Dialogue Act Analysis in BGP-MS

The task of the dialogue act analysis component in BGP-MS is to convert the dialogue acts observed in the user's interaction with an application system into their presuppositions by instantiating the presupposition patterns of their dialogue act types. This component operates in the following way:

1. A library of pre-defined dialogue act types with domain-independent presupposition patterns for each of them has been made available to the developer of the user modeling component of the application system.
2. The developer can both add further application-specific dialogue acts and complement the presupposition patterns of the predefined dialogue acts in an application-specific way.
3. The application system can report observed dialogue acts to BGP-MS.
4. The reported dialogue acts will be converted into their presuppositions by instantiating the presupposition patterns of the corresponding type definition.

Examples of possible pre-defined dialogue acts were given in the previous section. The following subsections will explain items (2) – (4) in more detail.

4.1 Defining Dialogue Act Types

The range of possible presupposition patterns for dialogue act types is strongly determined by the available knowledge representation language. The most powerful formalism available in BGP-MS is multimodal first-order predicate logic (MM-FOL), which includes first-order logic and allows MM-FOL expressions to be preceded by indexed modal operators, or combined with other expressions by the standard logical connectives. Using multimodal predicate logic means that most of the presupposition patterns listed in section 3 can remain unchanged in the definition of dialogue act types, and that instantiations of them can be entered as presuppositions at run-time. Only few descriptive elements used in section 3 must be disregarded, like e.g. the ι-operator.

Let us take the dialogue act types YN-ANSWER-YES, AGREE, REQUEST-FOR-EXPLANATION, and IGNORE-HOTWORD from section 3 as examples. When defining a dialogue act type in BGP-MS, its name (`:name`) and its parameters (`:parameters`)[4] must be given, and its presupposition patterns (`:presupp`) must be declared as a list containing Lisp notations of MM-FOL patterns. The pattern variables (P and $P(x)$ in the notation of section 3) are replaced by parameter symbols.

```
(define-d-act :name YN-ANSWER-YES :parameters (queried-item)
              :presupp ((B M (W U queried-item))))

(define-d-act :name AGREE :parameters (topic)
              :presupp ((B M (not (W U (not topic))))))

(define-d-act :name REQUEST-FOR-EXPLANATION :parameters (hotword)
              :presupp ((B M (not (B U (:concept hotword))))))

(define-d-act :name IGNORE-HOTWORD :parameters (hotword)
              :presupp
                ((if (belief S (not (belief U (:concept hotword))))
                     (belief M (belief U (:concept hotword))))))
```

4.2 Reporting and Processing Observed Dialogue Acts

The application system reports occurring dialogue acts with a message to the dialogue act analysis component, indicating the name of the corresponding dialogue act type along with a list of the concerned facts (which must be ground atoms of predicate logic). The dialogue act analysis component of BGP-MS will then generate the presuppositions of the dialogue act by instantiating the presupposition patterns in the definition of its dialogue act type with the reported facts. In this

[4] In BGP-MS, reported dialogue acts may have one or more parameters that specify the facts concerned.

way, new assumptions about the user are derived from observed user actions. These assumptions then become entered into the individual user model via `bgp-ms-tell`, the general input interface of BGP-MS.

In order to illustrate this method, we will now analyze possible messages to the dialogue act analysis component using the dialogue act type descriptions of section 3.

1. Observation: `(d-act YN-ANSWER-YES ((formatted disk1)))`

   Presupposition: `(bgp-ms-tell '(B M (W U (formatted disk1))))`

2. Observation: `(d-act AGREE ((displayed help-item-5)))`

   Presupposition:
   ```
   (bgp-ms-tell
      '(B M (not (W U (not (displayed help-item-5))))))
   ```

Now a possible sequence of two dialogue acts is shown, as can be observed in the adaptive hypertext system (note that the conditional expression in the presupposition of IGNORE-HOTWORD is satisfied after the presupposition of the REQUEST-FOR-EXPLANATION act has been entered by the dialogue act analysis component):

3. Observation: `(d-act REQUEST-FOR-EXPLANATION (UNIX))`

   Presupposition: `(bgp-ms-tell '(B M (not (B U (:concept UNIX)))))`

4. Observation: `(d-act IGNORE-HOTWORD (UNIX))`

   Presupposition: `(bgp-ms-tell '(B M (B U (:concept UNIX))))`

Figure 1 summarizes the dialogue act analysis of BGP-MS using the example given in item 3 above. The user interface recognizes a mouse click and reports it to the application system, which itself informs the dialogue act analysis component of BGP-MS about the dialogue act that took place, along with all necessary parameters. Then the presuppositions of this dialogue act are determined (using the defined dialogue act types and their presupposition patterns) and entered into the individual user model via `bgp-ms-tell`.

In this specific example, the user of our hypertext system wants an explanation of one of the hotwords of the current node, "UNIX". The hypertext system reports a REQUEST-FOR-EXPLANATION to BGP-MS, and the dialogue act analysis component derives the assumption that it is mutually believed that the user does not know the UNIX operating system. Beyond its immediate reaction (e.g., displaying explanatory text), the application could consider this assumption later and provide explanatory information about UNIX again when it displays the contents of another hypertext

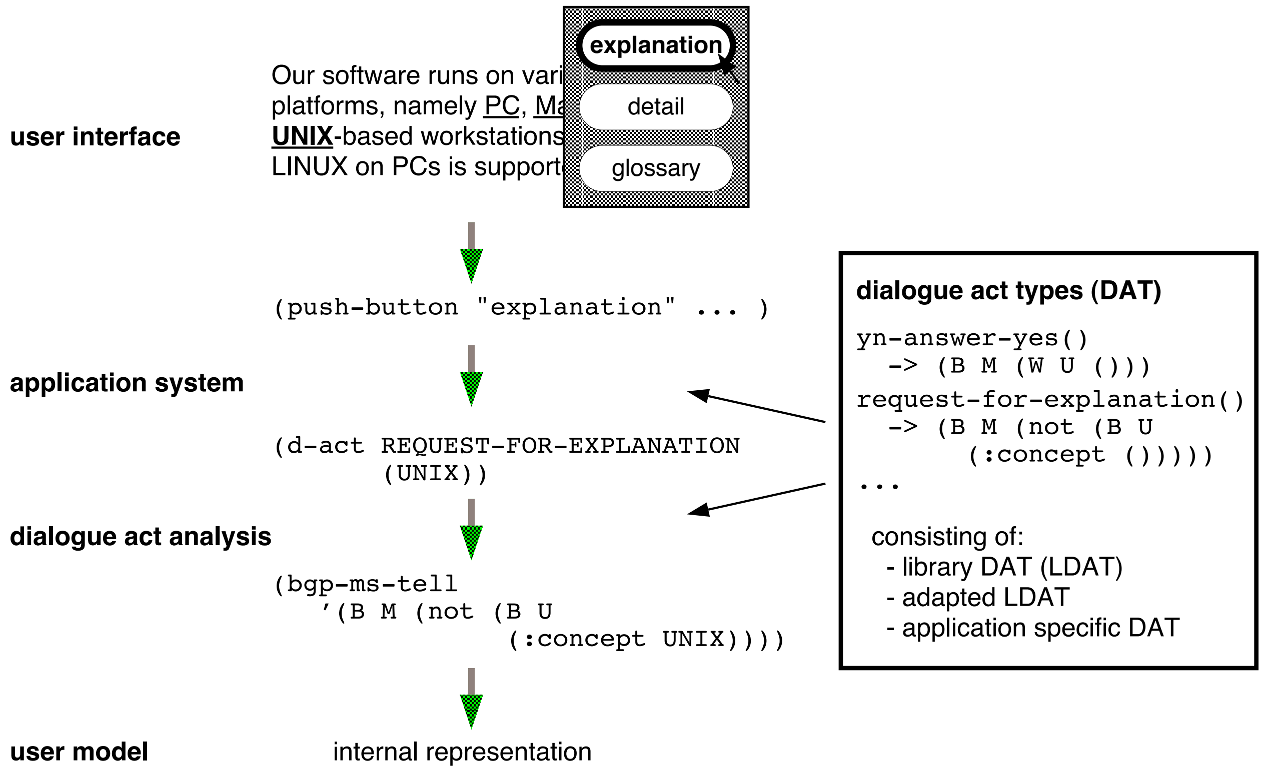

Figure 1: Dialogue Act Analysis in BGP-MS

node that contains "UNIX" or a related hotword. In this or similar ways, dialogue act types like REQUEST-FOR-EXPLANATION can be useful for adaptive information systems, particularly adaptive hypertext systems like those described in [Beaumont, 1994; Boyle and Encarnacion, 1994; Kobsa *et al*., 1994].

5 Related Research and Discussion

The aim of the work described here was to offer user model developers the possibility to define heuristics for the acquisition of primary assumptions about the user in a declarative manner. The definition of dialogue act types together with their associated presupposition patterns allows one to generalize many "local" acquisition rules into a single general heuristic: if an instance of a defined dialogue act type occurs, then the presuppositions of performing this dialogue act should be entered into the individual user model.

The idea of interpreting user input in a dialogue system as dialogue acts was first researched by Allen, Cohen, and Perrault (see e.g. [Allen and Perrault, 1980; Allen, 1983]). These authors also defined knowledge and goal prerequisites for speech acts. They did not use them for acquiring user models, however, but rather for planning dialogues and resolving ambiguities in utterances. In BGP-MS, the kind and the number of pre-defined dialogue act types are different from that and related work. In addition, the set of dialogue act types and the dialogue act types themselves are not fixed, but can be changed and augmented by the user model developer.

In comparison to more recent research on dialogue act analysis especially within natural-language systems, the presupposition analysis in BGP-MS is limited to a specific level of dialogue acts, namely the "core speech acts" [Traum and Hinkelman, 1992]. Lower level speech acts (such as turn-taking, turn-keeping) or higher level speech acts (such as elaborate, summarize, clarify, convince) are beyond the scope of our work since either they do not contain interesting knowledge and goal presuppositions, or they are too strongly connected to natural-language interaction.

A general observation in our work on dialogue act analysis in BGP-MS was that linguistic research on presupposition analysis can only be a starting point for the definition of dialogue act types and associated presupposition patterns in interactive computer systems. Examples of well-accepted presuppositions of standard dialogue acts can be found that do not apply any more when the dialogue act occurs in an interaction with a computer system. Human-computer interaction creates a background whose specific characteristics can and must be taken into account in the definition of dialogue act types.

Another observation was that a dialogue-act-based analysis of an interactive application might offer new insights into the consistency and usability of its interface. For example, one of the interaction possibilities associated with hotwords in the adaptive hypertext system that we analyzed is to request "more information". Quite different kinds of information nodes can be accessed by clicking on hotwords: explanations, graphics, detail information, examples, and even justifications for the whole sentence containing the hotword. So the user cannot have precise beliefs about what the system will present. Consequently there are no interesting presuppositions to such a request and hence no interesting dialogue act types can be defined. If there were a better correspondence between possible user actions on the one side and kinds of available information on the other side, the user could construct a preciser model of the system behavior and the system could construct a better user model. Thus a dialogue-act-based analysis of an interactive system might help discover possible ambiguities in the user's expectations concerning the system's behavior. However, these thoughts are based on few observations only, and considerably more research must be done in this regard.

Current work in BGP-MS includes the analysis of dialogue acts of the system, which also have presuppositions associated with them. When the system performs these acts, the user may make assumptions about the system based on their presuppositions. We will investigate to what extent the dialogue act types that we defined for the analysis of user actions can also be used for anticipating these assumptions of the user about the system, and whether these assumptions are interesting enough in user-adaptive systems that they should be entered into the individual user model.

Another research topic will be the analysis of presupposition patterns that contain conditions, like "if

$B_S \neg B_U\ concept(P)$ then $B_M B_U\ concept(P)$" for the dialogue act type "IGNORE-HOTWORD" in section 3.2. For these dialogue acts, the new assumptions that will be made about the user do not only depend on the presupposition pattern of the dialogue act type, but also on the current entry in the user model. Strictly speaking this already goes beyond the acquisition of primary assumptions about the user. It seems however, that quite a few dialogue act types contain conditions in their presupposition patterns (some of them are quite complex). We therefore plan to examine them for underlying general principles that might be supported by the dialogue analysis component of BGP-MS in the future.

Acknowledgements

This paper is a result of our work in the BGP-MS project, which has been supported by the German Society for the Advancement of Scientific Research (Grant number Ko 1044/4-2) and the University of Konstanz (Grant number AFF 17/92). We thank Jörg-Cyril Höhle for translating the abstract into French.

References

[Allen and Perrault, 1980] J. F. Allen and C. R. Perrault. Analyzing Intentions in Utterances. *Artificial Intelligence*, 15(3):143–178, 1980.

[Allen and Schubert, 1993] J. F. Allen and L. K. Schubert. The TRAINS Project. TRAINS Technical Note 91-1, Dept. of Computer Science, University of Rochester, 1993.

[Allen, 1979] J. F. Allen. A Plan-Based Approach to Speech Act Recognition. Technical Report 131/79, Dept. of Computer Science, University of Toronto, 1979.

[Allen, 1983] J. F. Allen. Recognizing Intentions from Natural Language Utterances. In M. B. Brady and R. C. Berwick, editors, *Computational Models of Discourse*. MIT Press, Cambridge, MA, 1983.

[Beaumont, 1994] Ian Beaumont. User Modeling in the Interactive Anatomy Tutoring System ANATOM-TUTOR. *User Modeling and User-Adapted Interaction*, 4(1):21–45, 1994.

[Boyle and Encarnacion, 1994] Craig Boyle and Antonio O. Encarnacion. MetaDoc: An Adaptive Hypertext Reading System. *User Modeling and User-Adapted Interaction*, 4(1):1–20, 1994.

[Brajnik and Tasso, 1992] G. Brajnik and C. Tasso. A Flexible Tool for Developing User Modeling Applications with Nonmonotonic Reasoning Capabilities. In *Proc. of the Third International Workshop on User Modeling*, pages 42–66, Dagstuhl, Germany, 1992.

[Chin, 1989] D. N. Chin. KNOME: Modeling what the User Knows in UC. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 74–107. Springer, Berlin, Heidelberg, 1989.

[Chin, 1993] D. N. Chin. Acquiring User Models. *Artificial Intelligence Review*, 7:185–197, 1993.

[Finin, 1989] T. W. Finin. GUMS: A General User Modeling Shell. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 411–430. Springer, Berlin, Heidelberg, 1989.

[Hirschmann, 1990] A. Hirschmann. *Das Hilfesystem MATHILDE*. PhD thesis, Philosophische Fakultät, University of Regensburg, 1990.

[Kaplan, 1979] S. J. Kaplan. *Cooperative Responses from a Portable Natural Language Data Base Query System*. PhD thesis, Department of Computer and Information Sciences, University of Pennsylvania, Philadelphia, PA, 1979.

[Kay, 1994] J. Kay. The um Toolkit for Reusable, Long Term User Models. To appear in *User Modeling and User-Adapted Interaction*, 1994.

[Kobsa and Pohl, 1994] A. Kobsa and W. Pohl. The BGP-MS User Modeling Shell System. To appear in *User Modeling and User-Adapted Interaction*, 1994.

[Kobsa *et al.*, 1994] A. Kobsa, D. Müller, and A. Nill. KN-AHS: An Adaptive Hypertext Client of the User Modeling System BGP-MS. In *Proc. of the Fourth International Conference on User Modeling*, pages 99–105, Hyannis, MA, 1994.

[Kobsa, 1983] A. Kobsa. Präsuppositionsanalyse zum Aufbau von Dialogpartnermodellen. *Conceptus*, 17(40/41):165–179, 1983.

[Kobsa, 1985] A. Kobsa. *Benutzermodellierung in Dialogsystemen*. Springer-Verlag, Berlin, Heidelberg, 1985.

[Kobsa, 1990] A. Kobsa. Modeling The User's Conceptual Knowledge in BGP-MS, a User Modeling Shell System. *Computational Intelligence*, 6:193–208, 1990.

[Kutter, 1994] O. Kutter. Dialogakte als Akquisitionsheuristiken für die Benutzermodellierung in Hypertexten. WIS Memo 11, WG Knowledge-Based Information Systems, Department of Information Science, University of Konstanz, Germany, 1994.

[Nwana, 1991] Hyacinth S. Nwana. User Modelling and User Adapted Interaction in an Intelligent Tutoring System. *User Modeling and User-Adapted Interaction*, 1(1):1–32, 1991.

[Quilici, 1989] A. Quilici. AQUA: A System that Detects and Responds to User Misconceptions. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*. Springer, Berlin, Heidelberg, 1989.

[Rich, 1979a] E. Rich. *Building and Exploiting User Models*. PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1979.

[Rich, 1979b] E. Rich. User Modeling via Stereotypes. *Cognitive Science*, 3:329–354, 1979.

[Searle, 1969] J. R. Searle. *Speech Acts*. Cambridge University Press, 1969.

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

[Stein *et al.*, 1991] A. Stein, U. Thiel, and A. Tißen. Towards Coherent Hypermedia Navigation by Pragmatic Dialogue Modeling. Arbeitspapiere der GMD 580, Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, 1991.

[Sukaviriya and Foley, 1993] P. Sukaviriya and D. Foley. A Built-in Provision for Collecting Individual Task Usage Information in UIDE: the User Interface Design Environment. In M. Schneider-Hufschmidt, T. Kühme, and U. Malinowski, editors, *Adaptive User Interfaces: Principles and Practise*, pages 197–221. North Holland Elsevier, Amsterdam, 1993.

[Traum and Hinkelman, 1992] D. R. Traum and E. A. Hinkelman. Conversation Acts in Task-Oriented Spoken Dialogue. *Computational Intelligence*, 8(3):575–599, 1992.

[Wahlster and Kobsa, 1989] W. Wahlster and A. Kobsa. User Models in Dialog Systems. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 4–34. Springer, Berlin, Heidelberg, 1989.

[Wierzbicka, 1987] A. Wierzbicka. *English Speech Act Verbs*. Academic Press, Australia, 1987.

[Winograd and Flores, 1986] T. Winograd and F. Flores. *Understanding Computers and Cognition*. Ablex, Norwood NJ, 1986.

[Winograd, 1988] T. Winograd. A Language/Action Perspective on the Design of Cooperative Work. *Human Computer Interaction*, 3(1):3–30, 1988.