User Modeling in Expert Man–Machine Interfaces: A Case Study in Intelligent Information Retrieval

GIORGIO BRAJNIK, GIOVANNI GUIDA, AND CARLO TASSO

Abstract —**The user modeling issue is analyzed in the broader context of expert interfaces: a general architecture is proposed and the role of user modeling is illustrated. The requirements of a user modeling component for an expert interface are analyzed, and the main points of the proposed approach to user modeling are stated. Then the authors focus on a knowledge-based system, called UM-tool, devoted to creating, maintaining, and using explicit user models within an expert interface. UM-tool supports a novel approach to user modeling, which is based both on the use of stereotypes and on a dynamic reclassification scheme. The architecture of the system is described, the organization and content of its knowledge bases are illustrated, and the modeling mechanisms utilized are presented in detail. An example of the use of UM-tool in the frame of the IR-NLI II expert interface devoted to support end-users in accessing on-line information retrieval services is then discussed. This focuses on the specific role of the user modeling component. Finally, an evaluation of the proposed approach and a critical comparison with related works are presented. An outline of future research directions concludes the paper.**

I. Introduction

THE DESIGN of graceful and cooperative man–machine systems that can help a user in effectively interacting with a complex artificial system has recently been widely recognized as a very promising and challenging topic both for basic research and applications. In this framework we have proposed in recent years the concept of expert interface as a new approach to man–machine system design [1], [16], [17]. Intuitively an *expert interface* is an intelligent intermediary that can support cooperation between man (the *user*) and machine (the *target system*) in an interactive problem-solving environment. It includes three main capabilities:

- supporting man–machine communication both from the linguistic and conceptual points of view, to make mutual understanding easy and effective;
- assisting the user in the correct, effective and efficient use of the target system, taking an active part in the problem-solving process in which he is engaged;
- incrementally training the user in the operation and use of the target system.

The benefits of an expert interface are not only to extend the direct usability of a complex artificial system by an end-user who no longer needs to resort to the assistance of a professional intermediary, but also—and more importantly—to improve (1) the quality of interaction, (2) the performance obtained from the machine, and (3) the degree of satisfaction of the user.

Of all the main characteristics that an expert interface should include, the capability to model individual users of the target system has been recognized as a fundamental one [25]. In fact for an expert interface to be effective and natural, it is important to tailor interaction style and content according to the principal features of each individual user. In most cases it is not enough for an expert interface to incorporate a generic—often implicit—user model, based on an analysis of the main common characteristics and requirements of a typical user in the expected user population. If this capability to manage individual and explicit user models is not available, the interface will generally have a rigid behavior and the interaction will be long, boring, poorly focused, ineffective, and, sometimes, misleading. An expert interface must be capable of modeling and remodeling dynamically in a nonobtrusive way the users it is interacting with.

These issues have been explored by many authors, and some approaches to their solution have been proposed in the literature of recent years—see, for example [12], [18], [21], [23], [27], [29]. Of all these approaches, some focus more on the issue of modeling user characteristics (e.g., [23]) and others on the issue of investigating the mental models of a user (e.g., [29]). In this paper we are concerned only with the former—the main features of the most significant of such approaches are described and commented in Section V-B. Although significantly contributing to the user modeling issue, these approaches leave important issues unresolved. Among these are the following.

1) User stereotypes embodying knowledge about typical characteristics of the user population are poorly exploited. In fact they are used only for initially classifying the user and constructing a first user mode, but not in the subsequent model refinement activity.
2) Stereotypes are generally organized into a single classification scheme. This is, however, often inade-

Manuscript received December 22, 1988; revised July 28, 1989.
The authors are with the Dipartimento di Matematica e Informatica, Università di Udine, Via Zanon, 6I-33100 UDINE, Italy.
IEEE Log Number 8931003.

0018-9472/90/0100-0166$01.00 ©1990 IEEE

quate for user modeling since the extremely broad variety of user characteristics makes it very difficult to define *a priori* a complete and appropriate taxonomy of user stereotypes.

3) User classification is used only to provide information about typical user characteristics. However knowing that a user belongs to a certain class may help in several other ways, including: a) tuning the interaction and dialogue with the user; b) determining what information has to be acquired next; c) guiding the classification process itself by deciding which is the most interesting user class to investigate next.
4) Most user modeling systems are designed as stand-alone and so they do not face the problem of actually including a user modeling component into an expert interface. How should a user modeling component interact with the other components of an expert interface, what is an effective granularity of such an interaction, how control must be distributed among the components, what contribution the user modeling component is expected to bring to the behavior of the expert interface, are just a few of the still open questions.

In this paper we tackle these open issues and propose a novel approach to user modeling that contributes to overcome the aforementioned conceptual limitations by providing an answer to the following questions.

- What is a user model, what knowledge does it involve, and how can it be appropriately represented?
- How can a user model be constructed and maintained?
- Where and how can a user model be integrated into the architecture of an expert interface?
- How can a user model be utilized to support and improve man–machine interaction?

This work is part of a larger project. In fact we have been involved in the design and experimentation of expert interfaces for more than five years [5], [6], [16], [17], and, more recently, have focused our attention on the user modeling issue [7], [8], [9]. Our project on expert interfaces and user modeling has been centered around a test-bed application in the domain of on-line information retrieval. The three main reasons for the choice of this domain have been:

- the difficulty and variety of the problems involved, which cover the whole spectrum of issues related to the design of expert interfaces, including user modeling;
- the fact that several research and application issues in this area are still open or only partially and inappropriately solved;
- the great importance of this domain from the application point of view, and the availability of several publications and extensive practical experience in man–machine interaction in information retrieval.

Within this framework, we have developed an original approach to user modeling in expert interfaces. The main contributions of this approach are the following.

- User modeling is a dynamic process based on a repeated-classification scheme. It dynamically combines several user taxonomies, and throughout the session the system tries to reuse the available stereotypes in order to gather the maximum of information about its current user. Furthermore, the information accumulated in the user model is always kept in a consistent and well-founded state.
- The knowledge supporting the user modeling function of the expert interface, which consists of both user and modeling knowledge, is represented either declaratively or procedurally. It is carefully structured and organized in such a way to extend the usability of the stereotypes for several tasks: tuning the dialogue, simplifying the acquisition of facts concerning the user, etc.
- The definition of a general framework to organize and use the modeling and user knowledge within an expert interface. This should be general enough to be successfully applied to other application domains besides information retrieval.

This approach has been concretely experimented in a prototype tool, called UM-tool (user modeling tool), for creating, maintaining and using individual and explicit user models within an expert interface. UM-tool has been used in the development of the IR-NLI II (information retrieval-natural language interface) system, which can support the access of end-users to on-line information retrieval services.

The purpose of this paper is to present and discuss the results of this research and to comment on their relevance to the general issue of man–machine system design. The paper is organized in the following way. Section II introduces our view on the role and benefits of user modeling in expert interfaces and develops a generic architecture for an expert interface including user modeling. In Section III we present UM-tool, a generic, application-independent tool for user modeling, focusing on design standpoints, system architecture, representation of both user and modeling knowledge, structure and content of the user model, and the modeling process. In Section IV we describe and discuss in detail an example of use of UM-tool in the IR-NLI II expert interface. Section V closes the paper, and includes a discussion of benefits and drawbacks of our approach, comparisons with related proposals, and an illustration of directions for future research.

II. The Role of User Modeling in Expert Interfaces

This section first introduces the concept of user model, and illustrates its role and use within an expert interface and the main benefits that can be expected. Further on, the architecture of a generic expert interface including a user modeling component is presented.

A. The Concept of User Model: Role and Use Within an Expert Interface

In a very broad sense a *model* is a (partial) representation of a real object that describes its characteristics relevant to a given task or goal. A model is an abstraction taken from reality: it must explain only the patterns of interest in terms of a set of easily understood elements. By using a model one is able to concentrate only on the important characteristics and properties of the situation at hand and to ignore the irrelevant ones. So the use of models is a way to handle the complexity of problem-solving tasks that involve reasoning about real objects, and implements a fundamental principle of economy. In the specific case of user modeling in an expert interface, a user model is a representation of the characteristics of a user that are relevant to the goal of supporting an effective and graceful man–machine interaction. This should dynamically adapt to the individual characteristics of the users the interface is interacting with.

User models may be of several types and may be classified according to different criteria. Four dimensions are specially important to our purpose.

1) Implicit versus Explicit: A user model is called *implicit* if it has been devised at system design time, has been used for making technical choices about system structure and organization, but has not been represented explicitly in the implemented system. An implicit model is hard-wired in a system so as it cannot be recognized in a specific piece of code or collection of data: in a sense, it is dispersed in several points and hidden under the system code. Therefore an implicit model cannot be modified or updated without deeply modifying the structure and organization of the system in which it is embedded. A user model is called *explicit* if it is directly represented in the system as a separate piece of knowledge, available for use during system operation, and accessible from the outside for modification and updating. Note that this concept of explicitness is different from the one found in most user modeling literature [23], where "explicit versus implicit" refers to what we have called here "given versus inferred."

2) Given versus Inferred: An explicit user model is called *given* if it has been defined and coded at system design time and later stored for use during system operation. A given model is in a sense a read-only piece of knowledge, which can be modified or updated only off-line from system operation through specific intervention by the system designer. In a given model only very limited updates, such as parameter adjusting, may be allowed to occur automatically during system operation. An explicit user model is called *inferred* if it is automatically constructed during system operation, without any external intervention by the system designer.

3) Static versus Dynamic: A user model is called *static* if it is supposed to remain unchanged and fixed over the long-term operation of the system it belongs to. A user model is called *dynamic* if it is continuously and incrementally refined, extended, and updated during system operation in order to cope with new facts and evidence about the user. Sparck Jones [26] defines static versus dynamic models differently: static models represent permanent characteristics of the user that are independent of his interaction with the target system, while dynamic models cover those specific features of the user that result from his interaction with the target system.

4) Canonical versus Individual: A user model is called *canonical* if it is aimed at capturing common characteristics of a whole class of users, focusing on the most frequent and shared features and overlooking individual peculiarities. A user model is called *individual* if it tries to represent the characteristics of each individual user in a given population, including both those common to other users and those unique to each specific user. This distinction is well accepted in user modeling literature [23].

Clearly, these dimensions of user modeling are not independent of each other: for example, implicit implies static and canonical, given implies explicit and static, dynamic implies explicit and inferred, individual implies explicit, etc. In the design of an expert interface the type of user model adopted plays a crucial role, since it influences limits and capabilities of the system (e.g., adapting its behavior to a wide range of casual users, dynamically changing interaction mode with a specific user from session to session, etc.). In our approach we will assume a concept of explicit, inferred, dynamic, individual user model, as will be discussed later, in Section III-A.

B. Benefits of User Modeling

Inserting a user modeling component into the architecture of an expert interface may offer several practical advantages, which are shortly to be described. Of course depending on the specific application domain where the interface is applied (e.g., process control, information retrieval, advice-giving systems, etc.) these benefits may have a different relative importance and other domain-dependent advantages may come into play that are not visible in the general case.

1) Economy of the Interaction: The dialogue between the user and the expert interface may be shorter, more concise, and better focused. In fact the expert interface needs to ask the user fewer questions, since it can rely on default knowledge already stored in the user model, thus speeding up interaction.

2) User Acceptability: The dialogue between the user and the expert interface may be individually tailored, thus becoming more acceptable to the user. In fact generation of questions and formulation of answers to the user takes into account the specific features of each individual user, thus making the dialogue clearer and more comprehensible and the communication more effective. Furthermore availability of a user model may have a crucial role in supporting explanation and justification capabilities, which in-

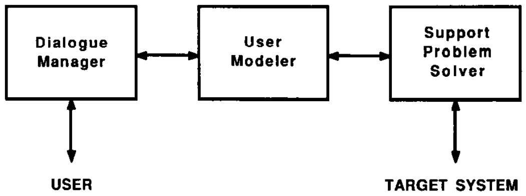


Fig. 1. General architecture of expert interface.

clude such important issues as defining concepts the user is not knowledgeable about, illustrating the meaning of actions taken by the target system, supporting the user in understanding what the role of the machine and also his own role in the problem-solving process should be, etc. [3].

3) Effectiveness and Efficiency of the Use of the Target System: The access to the target system and its use may turn out to be more effective and efficient, in terms of both quality and cost of the performance that may be obtained by the user. In fact the interface can support each individual user taking into account his specific characteristics and traits, his normal needs and goals, and also his known deficiencies, mistaken tendencies and typical errors.

C. An Architecture for an Expert Interface Including a User Modeling Component

The general architecture we propose for an expert interface that includes a user modeling component is illustrated in Fig. 1. It is made up of three modules.

1) The *dialogue manager*, which constitutes the front-end towards the user and is devoted to managing a mixed-initiative dialogue between the interface and the user. Dialogue is assumed to be based on the use of natural language, but it can also include the use of icons, menus, and flexible interactive tools, as well as the use of more rigid interaction protocols based on formal languages.
2) The *support problem solver*, which constitutes the back-end towards the target system and is devoted to emulating the skill and experience of a human expert competent in accessing and utilizing the target system.
3) The *user modeler*, which is connected to both of the other two modules, without having any direct link to either the user or the target system, and which is devoted to the management of user models.

Going further in the illustration of this architecture, we now examine in more detail the links between the user modeler, dialogue manager and support problem solver modules. These connections play a different role in the two processes of constructing the user model and using it during the activity of the expert interface; so we shall deal with each of these two aspects separately.

During construction of the user model, the principal task is, of course, the acquisition of information bout the user and its organization in the model. Such information may be of two types.

- *Linguistic information* concerns the features of the language utilized by the user for communicating with the target system, and including knowledge about the structure of typical user utterances, his favorite lexicon, phrases, and idioms, the specific meaning he attaches to certain words and phrases, his style of organizing dialogue, etc.
- *Conceptual information* refers to the concepts that the user utilizes during interaction with the target system, and including general world knowledge, domain-specific concepts, user goals, beliefs, assumptions, etc.

Information about the user may be obtained through two different mechanisms.

1) External Acquisition: Information about the user is obtained by analyzing utterances coming from the user, usually answers to questions posed to him by the expert interface during interaction. These questions can originate from the user modeler itself (e.g., when some very specific information about the user is needed), from the support problem solver (e.g., when user intervention is requested during the problem-solving process), and also from the dialogue manager (e.g., in the case of a clarification dialogue aimed at resolving possible linguistic ambiguities in the user utterances). External acquisition includes two more specific techniques for acquiring information:

- *dialogue inspection*—based on the observation and analysis of dialogues between the user and the expert interface, but not initiated by the user modeler with the explicit purpose of acquiring new information about the user;
- *direct questioning*—based on the analysis of user answers to questions specifically posed by the user modeler that engages him in focused dialogue with the explicit aim of acquiring new information.

External acquisition is suitable for collecting both conceptual and linguistic information.

2) Internal Derivation: Information about the user is obtained through an inference activity performed by the user modeler utilizing only its internal knowledge and the information gathered in current or previous sessions, and therefore already present in the current user model. Internal derivation is especially appropriate for collecting conceptual information. Internal derivation includes two more specific techniques for deriving information:

- *inference*—based only on the use of knowledge internal to the user modeler and of information already present in the current user model;
- *retrospection*—based on the processing of information gathered in past sessions, devoted for example, to recognizing repeating patterns and characteristics typical of the user.

These two ways of obtaining user information are illustrated in Fig. 2.

In order to conclude this illustration about how information on the user is obtained, it is worth mentioning that

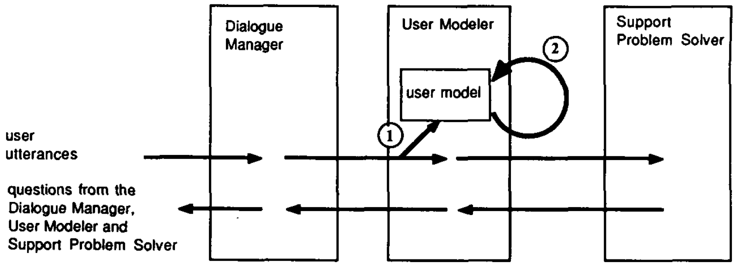


Fig. 2. Ways of obtaining user information, (1) External acquisition. (2) Internal derivation.

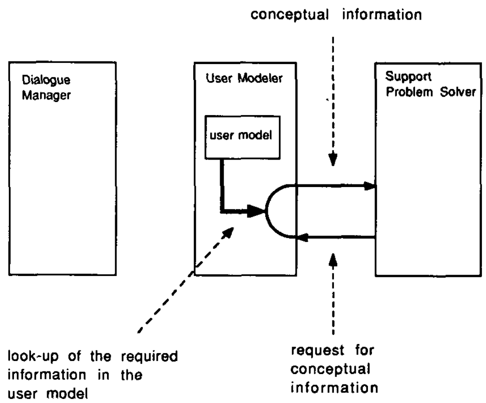


Fig. 3. Request for conceptual information from support problem solver.

such information may be acquired in two different ways, according to the way this information is organized. They are:

- acquisition of a *single* information item at a time, which represents a specific fact about the current user;
- acquisition of a *cluster* of information items at a time, consisting of a collection of facts about the user that are in some way interrelated; cluster organization is justified by the assumption that several user characteristics are not uniformly distributed over the user population, but occur most frequently together.

During operation of the expert interface, the user model is accessed whenever specific information about the current user is needed to support the operation of the expert interface. In all such cases the support problem solver and dialogue manager can call the user modeler and ask for the needed information. Generally they need different types of information: the support problem solver mostly utilizes conceptual information, whereas the dialogue manager mostly uses linguistic information. The role of the user modeler in the two cases of interaction with the support problem solver and dialogue manager is illustrated in Figs. 3 and 4, respectively.

Whenever the support problem solver (respectively, dialogue manager) needs conceptual (respectively, linguistic) information about the user, the user modeler is called upon. Through a simple look-up operation in the user model, it checks whether the needed information is already available. If this is the case the request coming from the support problem solver (respectively, dialogue manager) is immediately answered. Otherwise if the look-up in the current user model fails, the needed information is collected through internal derivation or external acquisition, the user model is updated, and finally the request of the support problem solver (respectively, dialogue manager) is answered. The user model is therefore the only source of information about the user for both the support problem solver and dialogue manager, and it is extended and updated whenever a need arises for more complete or accurate information about the user. Model construction and operation of the expert interface (dialogue manager and support problem solver modules) are therefore separate but cooperating processes, which can run concurrently.

III. UM-Tool: A Generic Tool for User Modeling

This section presents an experimental system, called UM-tool (user modeling tool), which has been developed to serve as a generic, application-independent tool for user modeling. In particular, assumptions and requirements of our approach concerning both the users to be modeled and the modeling process are illustrated first. Further on we focus on system architecture, on representation of user and modeling knowledge, on structure and content of the user model, and on the modeling process.

A. Design Standpoints

Our approach to user modeling is based on a set of assumptions and requirements about the users to be modeled and the modeling process, which motivate the design principles of our user modeling system UM-tool. These standpoints are listed and briefly commented below.

1) Classifiability of the User Population: In the potential user population, classes of users that feature common characteristics may be identified. Such classes of users are not supposed to be disjoint, i.e., each individual user may belong to several classes.

2) Insufficiency of the Classification Criterion: In general knowing the list of all classes a user belongs to provides only a partial characterization of the user, since he may possess some very specific and individual characteristics which are not represented by any class. So a user model should include characteristics derived from class member-

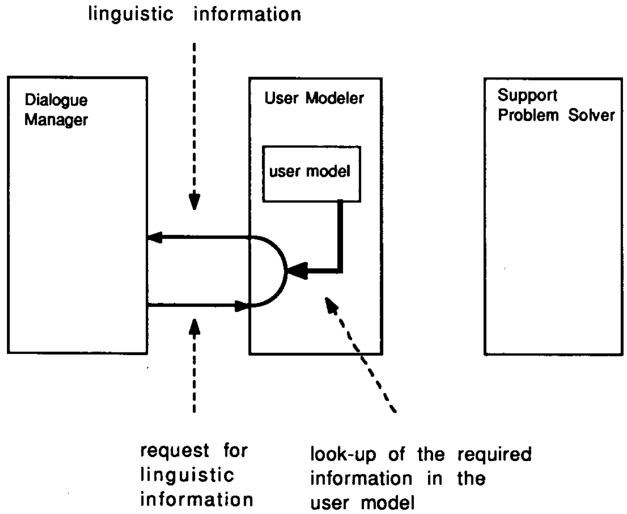


Fig. 4. Request for linguistic information from dialogue manager.

ship and also characteristics pertaining only to each individual user.

3) Stability of the User Population: Users in the potential user population do not change their characteristics during the modeling process.

4) Incompleteness of User Information: Information available about a user at any moment during the modeling process is inherently incomplete. In fact, a user may be considered as an unlimited source of information, which can be progressively elicited over time, but never completely acquired.

5) Incrementality of User Modeling: As a consequence of incompleteness of user information, user modeling is an incremental process, based on iterative extension, refinement and revision activities, which can take into account any new piece of user information as soon as it becomes available during the modeling process. Therefore modeling can be considered as an approximation process which tends in the limit to capture and represent all the characteristics of a user relevant to a given goal.

6) Incompleteness of User Models: As a consequence of incompleteness of user information and of incrementality of user modeling, user models are inherently incomplete.

7) Plausibility of User Models: As a consequence of incompleteness of user information and of incrementality of user modeling, information contained in a user model is generally not certain, since its validity may be contradicted by new user information. However at each time instant, a user model is supposed to be consistent with all user information acquired so far. Therefore information contained in a user model is generally assumed to be plausible.

8) Nonmonotonicity of User Modeling: In order to preserve consistency of a user model over time, the modeling process is basically nonmonotonic, since it must be capable of incrementally detecting and solving inconsistencies, by, if necessary, retracting plausible information already stored in the user model

Moreover a general requirement about the behavior of an expert interface with user modeling capabilities is that interaction with the user should not suffer from being overloaded by the modeling process. This should take place in a nonobtrusive way, without bothering the user and so deterring him from his primary problem-solving task and without conflicting with the primary goals of the expert interface.

On the basis of these standpoints we assume a concept of user modeling that according to the types of models introduced in Section II-A, can be classified as:

- inferred, and hence also explicit, since it is impossible to model all possible users *a priori*;
- individual, to integrate specific information pertaining only to an individual user with characteristics derived from class membership;
- dynamic, in order to cope with the incrementality and nonmonotonicity of the modeling process.

B. UM-Tool Architecture

We here describe the architecture of the user modeling tool (UM-tool), a tool specifically devoted to the development of the user modeler module for expert interfaces, designed taking into account the assumptions and requirements presented in the previous section. The mechanism proposed for performing the modeling task follows a knowledge-based paradigm and is centered around an original repeated classification scheme based on the use of stereotypes [23].

The main goal of UM-tool is the construction, maintenance and exploitation of user models, specialized data structures devoted to storing information about individual users currently accessing an expert interface. UM-tool activity is governed by the model manager, which is the core module of the system that includes the specific reasoning mechanism used for implementing the modeling

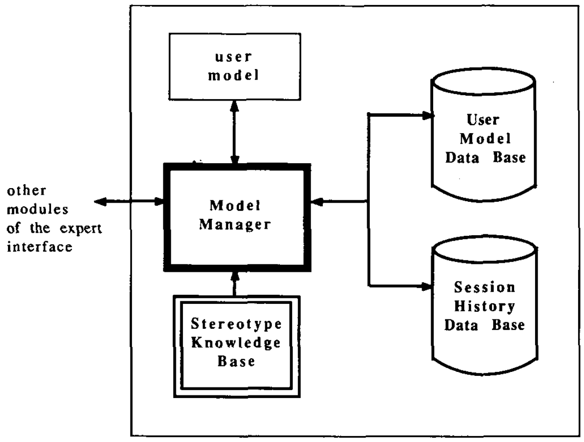


Fig. 5. Architecture of UM-tool.

process. The model manager is responsible for:

- the interaction with the other modules of the expert interface;
- the repeated classification activity on which construction and maintenance of user models is based;
- all read and write operations performed on user models for construction, maintenance and exploitation activities.

The model manager utilizes in its operation the following knowledge and data bases:

- *stereotype knowledge base*—devoted to storing knowledge necessary for the modeling activity, including both knowledge about the user population and knowledge about the modeling process;
- *user model data base*—containing the most recent model of each user who has accessed the expert interface so far;
- *session history data base*—devoted to storing records of past sessions held by users with the expert interface.

The architecture of UM-tool is shown in Fig. 5. A detailed illustration of the mode of operation of UM-tool, with special emphasis on the repeated classification scheme employed, is presented in the following sections.

C. Representation of User and Modeling Knowledge

UM-tool utilizes two types of knowledge, namely: knowledge about the population of potential users, called *user knowledge*; and knowledge about how to build and maintain a user model, called *modeling knowledge*. User knowledge takes a declarative form, and is mainly based on a taxonomic description of the population of possible users, classified according to appropriate sets of characteristics. Modeling knowledge encompasses a structured collection of procedures used to build and maintain a user model: acquiring information about the user, inferring user features, correlating different user classes, resolving possible conflicts and inconsistencies arising during the modeling process, etc. The representation of user and modeling knowledge is based upon frame-like structures called stereotypes, which are dealt with in detail in the following sections.

1) Stereotype Structure and Organization: A *stereotype* [23] is a description of a class of users sharing common characteristics and it specifies information regarding all the members of that class. A relation, called IS-A, is defined among stereotypes, and is derived from the inclusion relationship between the classes of users denoted by the stereotypes.[1] Thus the intuitive meaning of the IS-A relation is that of specialization (or generalization, depending on the direction considered) among stereotypes. There exists one stereotype, called *generic user stereotype*, which contains the most general characterization of all possible users of the expert interface in the user population considered. The other stereotypes called *class stereotypes* represent specific classes of users and are specializations of the generic user stereotype.[2] Both kinds of stereotypes contain default knowledge about user characteristics: this knowledge concerns the whole user population in the case of the generic user stereotype, and a specific user class in the case of a class stereotype.

A stereotype is represented through a *frame* that includes a collection of *slots*. Each slot is identified by a *name* and represents a characteristic or a property of the relevant class of users. The content of a slot encompasses both declarative knowledge represented through symbolic or numeric *values* and procedural knowledge represented through *methods*. Values and methods may be inherited by child stereotypes through the IS-A hierarchy. In general slots may be multivalued, i.e., they may contain several different values and methods.

A value may be atomic or structured. An *atomic value* consists of an elementary information structure (e.g., a number, a symbol, etc.). A *structured value* is a subframe that exploits the same frame organization. In this paper we use the term atomic (structured) slot to denote a slot with an atomic (structured) value.

Four methods are attached to each slot of each stereotype. These methods defined for the generic user stereotype may be either inherited or overridden and specialized in the class stereotypes. An intuitive description of such methods is given next.

a) Acquisition method: is used to acquire the value of a slot, according to the external acquisition mechanism described in Section II-C.

b) Inference method: is used to obtain the value of a slot on the basis of values of other slots in the user model, according to the inference mechanism described in Section II-C.

[1]Given two stereotypes $S1$ and $S2$ and denoting by $C(S1)$ and $C(S2)$ the classes of users represented by $S1$ and $S2$ respectively, we say that $S1$ IS-A $S2$ if $C(S1) \subseteq C(S2)$. If $S1$ IS-A $S2$, we call $S1$ a *child* of $S2$ and $S2$ a *parent* of $S1$. IS-A is clearly a partial order relation (a hierarchy).

[2]The generic user stereotype is the least upper-bound of the partially ordered set of stereotypes (under the IS-A relation).

c) Retrospective method: is used to compute the value of a slot through a retrospective analysis of the sessions recorded in the session history data base, according to the retrospection mechanism described in Section II-C.

d) Inconsistency method: is used to identify possible inconsistencies in the user model and to solve them. Each inconsistency method deals with one specific inconsistency and refers to several atomic slots. Therefore an atomic slot usually contains many different inconsistency methods: one for each kind of semantic constraint that is to be satisfied to maintain consistency. An inconsistency method encompasses two parts: the *inconsistency detector*, which is a predicate devoted to detecting an inconsistency, and the *consistency restorer*, which can take specific action to restore consistency.

2) The Generic User Stereotype: Generic user stereotype is a stereotype representing characteristics shared by the entire user population. The names of the slots of the generic user stereotype make up the dictionary used to specify the user features that are considered important in a given application domain. Of course the generic user stereotype generally contains only a few default values in its slots, while it is used as a support for a large number of methods that may later be inherited by all its parent stereotypes in the IS-A hierarchy.

3) Class Stereotypes: Class stereotypes represent specific classes of users and are obtained through a classification of the user population according to various classification criteria which, in general, are not independent or mutually exclusive. Thus a user may belong to several classes and may be described by more than one stereotype.

Class stereotypes are subclasses of the generic user stereotype and therefore they inherit its structure, values and methods. Class stereotypes may, however, override these slots by redefining their contents (values and methods). In this case they provide more specific information that is used when the user is believed to belong to the class denoted by the stereotype. In such a way methods can be tailored to each specific user class.

Class stereotypes include four types of information: a) characteristics shared by the users belonging to the class represented by the stereotype (described through slot values), b) conditions that suggest classifying a user as belonging to the class represented by the stereotype, c) pointers to other stereotypes that might be used to classify a user already belonging to the class represented by the stereotype, and d) pointers to parent stereotypes in the IS-A hierarchy. While a) is default knowledge about the users belonging to the class described by the stereotype, b), c), and d) are control knowledge used to exploit the stereotype itself. Such control information is stored in specific slots defined for any class stereotype as

- *triggering slot*—containing a method that is used to decide whether there is sufficient evidence that a user does belong to the class described by the stereotype;
- *correlation slot*—containing pointers to other stereotypes that might also appropriately describe a user; correlation pointers induce a partial order relationship among stereotypes, called *COULD-BE-A*[3]; and
- *superclasses slot*—containing pointers to the parent stereotypes in the IS-A hierarchy.

We would emphasize that information contained in the triggering, correlation and superclass slots is, in general, independent from each other. In fact these methods may relate different viewpoints for representing knowledge and reasoning about the user population.

D. The User Model

In order to introduce the concept of user model, we need a preliminary definition. As illustrated in the previous sections, stereotypes are static knowledge, provided at system design time and concerning permanent features and properties of a given user population. During operation of UM-tool, stereotypes are used to build and maintain a model of the current user by exploiting the default knowledge they contain about specific user classes. To this purpose it is necessary to identify which stereotypes, called *active stereotypes*, may describe the current user. A stereotype is active if at least one of the following conditions holds:

- its triggering method is satisfied;
- it is the successor of another active stereotype in the COULD-BE-A hierarchy;
- it is the parent of another active stereotype in the IS-A hierarchy.

We can now introduce our definition of user model. The *user model* is an instance of the generic user stereotype, appropriately augmented with dynamic individual knowledge about the current user and with default knowledge obtained through a multiple inheritance procedure from all active stereotypes. Therefore the user model contains: 1) default knowledge concerning a generic user inherited from the generic user stereotype, 2) specific default knowledge concerning particular classes of users inherited from active class stereotypes, and 3) individual information pertaining exclusively to the current user obtained dynamically through interaction with the expert interface.

The procedure utilized to identify active stereotypes and to handle multiple inheritance in order to construct and maintain a user model is the subject of Section III-E.

E. The Modeling Process

The top-level algorithm used by UM-tool (more precisely, by the model manager) for governing the overall modeling process is reported next.

[3]Given two stereotypes $S1$ and $S2$ and denoting by $C(S1)$ and $C(S2)$ the classes of users represented by $S1$ and $S2$ respectively, we say that $S1$ COULD-BE-A $S2$ if, for any user x, if $x \in C(S1)$ then it may be that $x \in C(S2)$. If $S1$ COULD-BE-A $S2$, we call $S1$ an *ancestor* of $S2$, and $S2$ a *successor* of $S1$. COULD-BE-A is clearly a partial order relation (a hierarchy).

```
BEGIN
perform user identification
IF user is known
  THEN  retrieve relevant user model in User Model Data
        Base;
        process historical information from Session His-
        tory Data Base;
        insert obtained information into the user model
  ELSE  perform preliminary interview;
        insert collected information into the user model
END-IF
determine active stereotypes;
insert default information into the user model
REPEAT
  WHEN request for information about the user DO
       look-up for requested information in the user
       model
  END-WHEN
  IF  requested information is not found
      THEN  acquire requested information;
            insert obtained values into the user model;
            determine active stereotypes;
            insert default information into the user
            model
  END-IF
  answer request for information about the user
UNTIL end of session
store updated user model in User Model Data Base
END.
```

After user identification, if the user is already known to the expert interface, the model manager retrieves his model from the user model data base, and performs a synthesis of historical information from the records stored in the session history data base. This operation is accomplished by executing retrospective methods attached to the slots of the retrieved user model. In this way the initial values of such slots for the current session are obtained and then inserted in the user model. Otherwise if it is the first time the user interacts with the expert interface, a preliminary interview is carried out devoted to collecting basic individual information about the user directly from him. The information collected is then inserted in the user model, which then yields its initial version for the current session. After this the set of active stereotypes is determined. The user model is updated by inserting default information in the propriate slots.

After this initial phase the model manager operates on the user model only when a request of information about the user comes from the dialogue manager or from the support problem solver. In such a case if the information requested is already available in the user model, it is then passed to the requesting module and the model manager returns to a waiting state. Otherwise if the information requested is not available in the user model, an important modeling phase is entered, which comprises the following steps: 1) the requested information about the user is first obtained through external acquisition or internal derivation, and the information obtained is inserted in the user model; 2) the set of active stereotypes is then redetermined, and the user model is updated by inserting default information in the appropriate slots.

Finally the original request for information about the user is answered and the model manager returns to a waiting state. At the end of the session with the expert interface the (updated) user model is stored in the User Model Data Base for future use.

We now focus in more detail on the most important steps carried out by the model manager in the algorithm previously illustrated, namely: the acquisition of information about the user, the insertion of information in the user model, and the determination of active stereotypes.

1) Acquisition of information about the user: The acquisition of information about the user is performed both in the course of the preliminary interview and when the model manager has to answer a request for information not yet available in the user model. It encompasses the following two steps:

- selecting the appropriate acquisition method in the slot of the user model for which a value has to be produced; and
- executing the selected method and generating a value for the slot at hand.

Note that the first step is complicated by the fact that several acquisition methods may be bound to a single slot in the user model. Choice of the appropriate method is made according to a set of simple selection criteria (applicability of the method, generality of the method, cost of its application, plausibility of the result, etc.).

2) Insertion of information in the user model: Inserting a value in a slot of the user model is performed several times during the modeling process, and may concern several types of information: information obtained through external acquisition or internal derivation, information obtained through retrospection, and also default information derived from active stereotypes. The insertion of information comprises the following four steps:

- checking the consistency of the value by evaluating the inconsistency detector method bound to the slot where it should be inserted;
- if some inconsistency is found, resolving it by applying the consistency restorer method bound to the slot;
- writing the value provided by the consistency restorer method, or the original one if no inconsistency has been found, into the appropriate slot of the model (if no value is returned, the slot is emptied); and
- eventually executing all the inference methods of the updated slot.

Note that the last step hides a major point: execution of inference methods may require the insertion of new results into other slots of the user model, therefore causing recursive call of the insertion procedure previously defined.

3) Determination of active stereotypes: The determination of active stereotypes aims at the exploitation of default information about the user population stored in the stereotype knowledge base. This is done by determining the current set of active stereotypes, called the *active set*, and by inserting their default knowledge in the user model. Determination of the active set is performed in two cases during the modeling process: a) at the beginning of a work session, when a first inital version of the user model has been constructed; and b) whenever the current user model has been updated through acquisition of new information about the user.

Determination of the active set is based on execution of an *activation procedure*, which is applied to each stereotype in the stereotype knowledge base in order to select those that are active and include them in the active set (assumed to be empty at the beginning of a work session). The activation procedure examines the stereotypes in the IS-A hierarchy bottom-up (breadth-first, from the leaves of the hierarchy towards the root, i.e., the generic-user stereotype), disregarding stereotypes already on the active set. For each stereotype, the following five steps are performed.

a) The triggering method of the stereotype is evaluated: if it is not satisfied then the activation procedure proceeds to the next stereotype.
b) If the triggering method is satisfied, the stereotype is included in the active set.
c) The default knowledge contained in the stereotype is inserted in the user model. This is accomplished in two steps.
    - Each value of the stereotype is inserted into the user model in the appropriate slot, using the insertion procedure previously described.
    - All the methods of the stereotype are appropriately installed in the user model in the appropriate slots (should there be conflict, this is overcome using a set of simple selection criteria).
d) The correlation slot of the stereotype is examined, and Steps b), c), d), and e) are recursively applied to all successor stereotypes in the COULD-BE-A hierarchy;
e) The superclass slot of the stereotype is examined, and steps b), c), d), and e) are recursively applied to all parent stereotypes in the IS-A hierarchy.

Note that since the user model is updated at each step of the activation procedure, the order in which stereotypes are considered for activation is not immaterial to the determination of the active set. The choice of the bottom-up traversal of the IS-A hierarchy, modified through Steps d) and e), is inspired by a cognitive conjecture about how humans utilize classification knowledge proceeding incrementally from specific to general through a basically bottom-up traversal of the knowledge base, which may develop locally in a depth-first style following IS-A and COULD-BE-A links.

From the previous illustration of the activation procedure, one could infer that the active set may only be extended and never decreased. In fact, it is not so, since stereotypes are removed from the active set through a *deactivation procedure*, as soon as the reasons that supported their inclusion no longer hold true. Each time the procedure for the insertion of information in the user model is executed and a value is updated, the deactivation procedure is invoked. The active set is then revised in the following way: all stereotypes in which the activation depends on an updated value in the user model are checked. Those stereotypes that no longer satisfy the definition of active stereotype are removed using a truth-maintenance procedure [14], together with all their child stereotypes in the IS-A hierarchy. Removal of an active stereotype from the active set causes deletion of all the values in the user model which depend on it. Clearly this operation may cause recursive execution of the deactivation procedure.

Let us point out that while both the IS-A and COULD-BE-A relationships are dealt with similarly during activation, they are differentiated by the deactivation procedure. In fact the deactivation of a stereotype necessarily implies the deactivation of all child stereotypes in the IS-A hierarchy, whereas it has no effect on COULD-BE-A ancestors.

The combined effect of the activation and deactivation of stereotypes results in a dynamic *classification* of the user. This type of classification is characterized by the fact that during a work session the active set always tends to be maximal, i.e., to include all stereotypes that appropriately describe the current user, discarding those that are inappropriate as soon as they turn out to be inconsistent with a user characteristic.

IV. Using UM-Tool: A Case Study in the Intelligent Information Retrieval Field

This section illustrates an application of UM-tool in the field of intelligent information retrieval. In particular it focuses on an experimental expert interface, called IR-NLI II, which is devoted to assisting nontechnical users in the access to on-line data bases. A brief introduction and overview of the IR-NLI project is first presented, and the general architecture of the IR-NLI II system is illustrated with particular attention to the user modeling subsystem. Finally a sample session with IR-NLI II is described, and its main features relevant to the user modeling issue are discussed.

A. The IR-NLI Project

The IR-NLI project is devoted to the design and experimentation of intelligent interfaces towards on-line information retrieval systems. The project has been the basis for several specific investigations, and has been developed through three main phases. Initially attention was focused on the conceptual aspects of man–machine interaction, and the novel concept of expert interface was proposed [1], [5], [6], [16], [17]. The outcome of this phase of the research was the prototype system IR-NLI (information retrieval-natural language interface). IR-NLI is devoted to supporting end-users in the access to on-line bibliographic systems in the field of computer science and is based on an implicit and canonical model of the potential users of an on-line

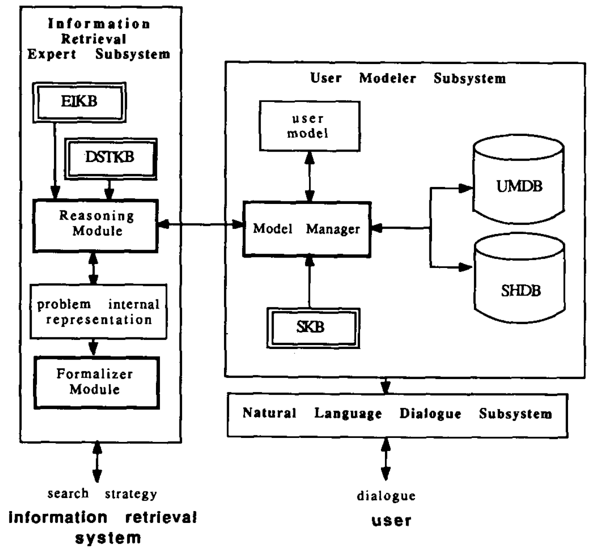


Fig. 6. Overall architecture of IR-NLI II.

bibliographic system. Later on the project centered on the role of user modeling as a basic component of an expert interface [7]. As a result of these activities UM-tool has been developed and a second version of the IR-NLI prototype, called IR-NLI II has been designed and implemented. This includes full user modeling capabilities [8], [9]. Recently a new research direction has been started, focusing on the linguistic aspects of man–machine interaction, and aimed at developing a natural language dialogue module for IR-NLI II. This issue, which has received only minor attention in our project in past years [15], is however beyond the scope of this paper.

B. Overall Architecture of the IR-NLI II System

The overall architecture of IR-NLI II, shown in Fig. 6, mirrors the general organization of an expert interface already illustrated in Section II-C. The support problem solver utilized in IR-NLI II is a specialized knowledge-based system, called information retrieval expert subsystem (IRES), competent in the domain of information retrieval. It is devoted to supporting a casual user in solving an information need through access to an on-line information retrieval system. Its specific skills include the analysis of the information problem of the user, the elicitation of his information needs, and the design of a suitable search strategy to be used for accessing the target information retrieval system. IRES operates interactively. The user first provides a statement of his information problem. IRES analyzes this statement and then engages the user in a dialogue aimed at gathering more detailed specifications of his information need and a deeper comprehension of his request. At the same time the data bases to be utilized in the search are selected, a suitable approach [20] is chosen, and appropriate tactics [2] are adopted in order to design the search strategy. As soon as the analysis of the information problem has been developed to a satisfactory point, the formalizer module (FM) produces a search strategy coded in an appropriate query language, which is submitted to the information retrieval system. The documents retrieved are shown to the user, and if his evaluation is not completely satisfactory, the search strategy is refined and the search is repeated until his information needs are fully satisfied.

IRES operation is driven by a reasoning module (RM), which constructs and maintains an information structure, called problem internal representation (PIR), devoted to storing the representation of the information problem of the user and all the information needed to build the search strategy. The reasoning module is supported in its activity by two knowledge bases, namely: 1) the expert intermediary knowledge base (EIKB), which contains knowledge devoted to modeling the competence and skill of a human intermediary, represented by means of production rules; and 2) the domain specific terminological knowledge base (DSTKB), which contains terminological knowledge about the specific subject domain at hand, represented through a semantic network.
The reader may refer to [6], [17] for a detailed illustration of the IRES module and of its operation.

The specific dialogue manager utilized in the present version of IR-NLI II, called natural language dialogue subsystem (NLDS), can support simple user-system dialogues based on canned sentences and menus. Extension of this module to include natural language understanding and generation capabilities is underway, but it is outside the scope of this paper.

Finally the user modeler subsystem (UMS) of IR-NLI II strictly adheres to the organization presented in Section III-B, being implemented by means of UM-tool. So it comprises a model manager (MM), a stereotype knowledge base (SKB), a user model data base (UMDB), and a session history data base (SHDB). More specifically a knowledge-engineering activity has been performed in order to acquire all the knowledge necessary to define some stereotypes describing the typical classes of users of a university information center. This knowledge has been later inserted in the stereotype knowledge base of UM-tool, and the resulting system has been appropriately linked to the two modules of IR-NLI II.

The running prototype of IR-NLI II is oriented towards bibliographic data bases in the domain of computer science. EIKB contains knowledge about three approaches (namely: building block, citation pearl growing, and most specific first) and 13 tactics (namely: parallel, pinpoint, super, sub, sibling, superordinate, subordinate, respell, truncate, combine, split, exhaust, reduce) [2], [20]. DSTKB contains terminological knowledge concerning about 300 terms (both free and controlled), related to each other through several terminological relationships (related term, narrower term, broader term, used-for, etc.). DSTKB also contains information about the posting count and the level of generality of each term. Most of this knowledge has been automatically acquired from on-line searching referral aids. The SKB utilized by the MM contains 24 class stereotypes obtained through several interviews with real human intermediaries by focusing on the following main attributes: professional position (researcher, graduate, undergraduate, teacher, technical manager, sales engineer, technical consultant, etc.), background (chemist, physicist, computer scientist, industrial engineer, physician, etc.), experience in computer science (novice, knowledgeable, expert), experience in librarianship (novice, knowledgeable, expert), experience in the use of information retrieval systems (novice, knowledgeable, expert).

C. A Sample Session with IR-NLI II

In this section we illustrate a sample session with IR-NLI II. The purpose of this example is twofold: first, to illustrate how user modeling is performed, and, second, to show how user modeling can substantially improve the performance of an expert interface. This example shows fragments of dialogues between the user and IR-NLI II, explains the actions taken by the system, and illustrates how the user model is incrementally constructed and used to support an effective interaction. Little attention is given here to the IRES and NLDS modules and to their operation.

The sample session presented next is based upon the following scenario. The user of IR-NLI II is already known to the system since he has already held a few search sessions in the past. He has been classified as a researcher and computer-science-expert (these are the active stereotypes on which his model is based). He is knowledgeable in information retrieval; he has made several requests in the past in the computer science domain, and a few others in other domains; in domains different from computer science he is generally imprecise and too specific in stating his information problem.

At the beginning of the search session, after user identification, the relevant user model is retrieved from the UMDB, and, after the processing of historical information, the initial version of the user model to be utilized in the current session is generated. This is reported next.

```
GENERAL PROFILE
    EDUCATIONAL BACKGROUND
        FIELD: computer science
            DEGREE: Ph.D.
            DATE: 1978
    PROFESSIONAL BACKGROUND
        FIELD: computer science
            POSITION: researcher
            SINCE: 1982
    INFORMATION RETRIEVAL BACKGROUND
        EDUCATION: medium
        TRAINING: medium
        EXPERIENCE
            TYPE: user
            MODE: assisted
            SINCE: 1987
    PERSONAL TRAITS
        COMMUNICATION
            LEVEL: concise
            QUALITY: precise
        ATTITUDE: cooperative
INFORMATION RETRIEVAL PROFILE
    NORMAL EXPRESSION OF NEEDS
        DOMAIN: computer science
            ACCURACY: high
            COMPLETENESS: average
            ATTITUDE: browsing-oriented
        DOMAIN: others
            ACCURACY: too specific
            COMPLETENESS: low
            ATTITUDE: browsing-oriented
    NORMAL REQUEST PATTERN
        DOMAIN: computer science
            SOURCE OF PROBLEM: write paper
            GOALS LEADING TO SEARCH:
                        discover something,
                        fill gaps in knowledge,
                        review literature
    NORMAL SEARCH REQUIREMENTS
        DOMAIN: computer science
            SEARCH OBJECTIVES: high precision
            LIMITS
                DATE: 1975
                LANGUAGE: English
                TREATMENT: theoretical, technical
            OUTPUT FORMAT
                FIELDS: title, authors,
                affiliation, date, abstract
                MODE: off-line
            PREFERRED DATA
                BASES: INSPEC, NTIS,
                       COMPENDEX
USER KNOWLEDGE
    SUBJECT DOMAINS
        DOMAIN: computer science
            COVERAGE: high
            DEPTH: high
    DATA BASES
        NAME: INSPEC
            TOPIC: known
            TERMINOLOGY: well-known
        NAME: NTIS
            TOPIC: known
            TERMINOLOGY: not known
        NAME: COMPENDEX
            TOPIC: known
            TERMINOLOGY: well-known
    INFORMATION RETRIEVAL SYSTEMS
        NAME: ESA-IRS
            FUNCTIONS: known
            LANGUAGE: well-known
        NAME: DIALOG
            FUNCTIONS: known
            LANGUAGE: unknown
    INFORMATION RETRIEVAL ACTIVITY
        SEARCH SESSION STRUCTURE: known
        APPROACHES: citation pearl growing, most
                    specific first
        TACTICS: exhaust, reduce, pinpoint,
                 respell, parallel
```

Initially the user is engaged in the following dialogue[4] from the system.

S: Please, state your information problem.
U: I am interested in papers dealing with computer assisted NMR medical devices.
S: Ok. I understand the following concept: C1: Computer-assisted NMR Medical Devices. Is that correct?
U: Yes.
S: Please, select any of the following term combinations if you think they express the same meaning as concept C1:
Computer and NMR Medical Devices
Computer Devices
Computer-assisted and NMR Medical Devices
Computer-aided and NMR Medical Devices
Computer-based and NMR Medical Devices
U: Computer and NMR Medical Devices is the best combination, but also the others are good ones, except for Computer Devices.
S: Now please choose the discipline(s) related with the topic(s) of your query and rank them (1-5). The scale is:
5 very relevant
4 relevant
3 not known
2 hardly relevant
1 not relevant
The disciplines are:
Administration
Aeronautics
Agriculture
...
U: Computer Science (5) for Computer and Medicine (4) for NMR Medical Devices.
S: Ok. Please, tell me why you need this information (to write a report, to prepare a thesis, ...).
U: I am going to teach a course for operators of specialized medical machines.
S: I understand the following:
Your information need derives from the fact that you have to teach a course. Its topic is about operators of specialized medical machines.
Is this correct?
U: No. The topic is specialized medical machines.
U: I understand the following:
The course topic is specialized medical machines.
Is this correct?
U: Yes.
S: Considering your previous search sessions and the disciplines involved in the current search, I would suggest the following data bases:
INSPEC, NTIS, PASCAL
Please, add any entry you consider relevant and remove those you consider not relevant. State also the order in which data bases are to be searched.
U: INSPEC, NASA, COMPENDEX, in this order.

This first dialogue is completely driven by IR-NLI II. In particular, all the questions are originated by IRES that is trying to build up the first version of the PIR. Most of the questions are addressed by IRES to UMS that, being unable to answer using the information contained in the user model, forwards them to NLDS that, in turn, translates them into natural language utterances. The last question follows a slightly different path and deserves more detailed attention. Initially a request for information concerning the data bases to be utilized in the search is submitted to UMS by IRES. UMS can, however, only find a tentative answer in the user model (consider the slots: INFORMATION RETRIEVAL PROFILE, NORMAL SEARCH REQUIREMENTS, DOMAIN: computer science, PREFERRED DATA BASES: INSPEC, NTIS, COMPENDEX). So, since the current query includes both the computer science and medical domains, it decides to look for a better answer through a direct question to the user. UMS therefore formulates a specific query to the user taking into account the already-known tentative answer: it supplies a tentative list of data bases taking into account default information about the user, and allows the user to make corrections to this list since he is believed to be knowledgeable in information retrieval (consider the slots: GENERAL PROFILE—INFORMATION RETRIEVAL BACKGROUND, EDUCATION: medium, TRAINING: medium, EXPERIENCE, TYPE: user, MODE: assisted, SINCE: 1987). Eventually NLDS generates from the query constructed by UMS the actual natural language utterance. Note that if the user were not knowledgeable in information retrieval, UMS would not have requested him to check and possibly modify the proposed list of data bases, and also the dialogue illustrating the list of data bases proposed by the system would have been less concise (e.g., the main characteristics of the selected data bases would have been described). All information gathered by IRES through this initial dialogue is used to construct the first version of the PIR.

After this first step IRES enters the central part of the search session devoted to refining and deepening the initial formulation of the information problem stated by the user. This is accomplished by extending the terminology about concept C1: first acquiring keywords directly from the user, and later refining and expanding the formulation of the concept by retrieving other terms from DSTKB. The user is therefore engaged in a new dialogue.

S: Please, give me some keywords for the term NMR Medical Devices.
Please, rank them according to their relevance (1-5).
U: Magnetic Resonance Imaging (4)
Tomography (4)
Heart Imaging (3)
Ramses (4)
Tomotron (4)
S: I think some of these terms are too specific for your current search motivations and objectives. I suggest

[4]The free, natural language dialogues reported next will be fully supported by the extended version of NLDS currently under development.

you also consider the following broader terms. Please select those you like and rank them (1-5):
Computer-assisted Radiology
Nuclear Medicine
Radiotherapy
Radiology
Radiography

U: Computer-assisted Radiology (4)
Nuclear Medicine (2)
Radiotherapy (2)
Radiology (2)
Radiography (2)

S: Your formulation of the information need is too narrow in scope and incomplete for your search motivations and objectives. I suggest you also consider the following related or used-for terms. Please select those you like and rank them (1-5):
Magnetic Resonance Spectroscopy
Digital Radiography
Computer-Assisted Radiation Therapy
Digital Radiotherapy
Digital Subtraction Myelography
Digital Subtraction Angiography
Nuclear Magnetic Heart Imaging
Emission Computed Tomography
Digital Tomosynthesis.

U: Magnetic Resonance Spectroscopy (2)
Digital Radiography (3)
Computer-Assisted Radiation Therapy (3)
Digital Subtraction Myelography (2)
Digital Subtraction Angiography (2)
Nuclear Magnetic Heart Imaging (3)
Emission Computed Tomography (2)
Digital Tomosynthesis (3)

S: Ok. I have now a fairly good formulation of the problem.
I will search the INSPEC data base first.

This dialogue, too, is mainly driven by IRES and is influenced by the user model by several points.

1) System-generated dialogues are tailored to the user experience in the information retrieval domain (consider the slots: GENERAL PROFILE—INFORMATION RETRIEVAL BACKGROUND and USER KNOWLEDGE, INFORMATION RETRIEVAL ACTIVITY). Technical terms, such as "keyword," "broader term," "related term," "used-for term" are not explained in the dialogue because the user is believed to be fairly expert in the information retrieval field.
2) Furthermore IRES tries to remedy a (supposed) limitation of the user in expressing his information needs in the medical domain: he is in fact believed to be too specific and scarcely complete (consider the slots: INFORMATION RETRIEVAL PROFILE, NORMAL EXPRESSION OF NEEDS, DOMAIN: others, ACCURACY: too specific, COMPLETENESS: low). Thus it limits the importance of most of the terms provided by the user, since they are considered too specific and narrow (some of them are even unknown to IRES), and prompts the user to also consider broader, related, and used-for terms (extracted from DSTKB).
3) Finally IRES generates a search strategy, where the terms suggested by the system and selected by the user receive much greater attention than those directly supplied by the user (supposed to be too specific and narrow). In fact as can be seen next, in the final search strategy some of the terms suggested by the user are totally discarded (e.g., Ramses), and some of the terms that have been suggested by the system appear in the strategy although the user did not judge them very relevant (e.g., Digital Radiography).

After this dialogue, IRES generates an initial version of the search strategy, which is shown next (expressed in an intermediate query language, independent of any specific information retrieval system).

SET TREATMENT = (practical OR theoretical)
SELECT INSPEC
1. FIND/CT Computer
2. FIND Computer.assist?
3. FIND Computer.based
4. FIND Computer.aided
5. FIND COMPUTERIZ?
6. OR 1 2 3 4 5
7. FIND NMR.Medical.DEVIC?
8. FIND Nuclear.Magnetic.Resonance.Medical.DEVIC?
9. OR 7 8
10. FIND Magnetic.Resonance.IMAG?
11. FIND/CT TOMOGRAPH?
12. FIND Heart.IMAG?
13. FIND/CT Computer.Assist?.RADIOLOG?
14. FIND Nuclear.Medicine
15. FIND/CT RADIOTHERAP?
16. FIND RADIOLOG?
17. FIND RADIOGRAPH?
18. OR 10 11 12 13 14 15 16 17
19. FIND/CT Magnetic.Resonance.SPECTROSCOP?
20. FIND/CT Digital.RADIOGRAPH?
21. FIND Computer.Assisted.Radiation.THERAP?
22. FIND/CT Digital.RADIOTHERAP?
23. FIND/CT Digital.TOMOSYNTH?
24. OR 19 20 21 22 23
25. OR 9 18 24
26. AND 6 25
27. FIND Computer.Assist?.NMR.Medical.DEVIC?
28. OR 26 27

CHOOSE FIELDS TITLE, AUTHOR, AFFILIATION, DATE, ABSTRACT,
SET MODE = ONLINE
OUTPUT 28.

This strategy is submitted to INSPEC, which yields 26 document citations. These papers are first ranked according to their relevance to the current search (on the basis of simple keyword matching), and then shown to the user. Unexpectedly he considers only a few of them relevant to his task. The system therefore starts an evaluation dialogue.

S: I would like to know why most of the retrieved citations do not satisfy your needs. Are they not relevant at all, too specific, too general, ...?
Please, explain.

U: Since I have to teach a course, I need information on issues related to human–computer interaction with complex medical devices.

S: Ok.
I will consider the following new concepts:
C2: Human–Computer Interaction
C3: Complex Medical Devices
Is this correct?

U: Yes.

S: Ok.
Please enter the discipline(s) to which concepts C2 and C3 refer.

U: C2 refers to Computer Science, while C3 to Computer Science and Medicine.

S: Ok.
Is there any relation between one (or more) of these concepts (C2 C3) and the previous one (C1)?

U: Yes. C3 is a related term of C1.

S: Ok.

At this point the user takes a look at the search strategy that has been submitted to INSPEC, and makes the following comment:

U: Although very specific, a few terms I specified before are good ones; they were however excluded from the search strategy. I am referring to Ramses and to Tomotron, which are specific systems I am interested in. Why did you decide to exclude them from the search strategy?

S: I am assuming that you are not knowledgeable in medicine, and that, therefore, you tend to be too specific in the formulation of your problem. Are these assumptions correct?

U: No. I am knowledgeable in medicine.

S: Okay. Please answer the following question:
Have you had any significant education in medicine?

U: Not an extensive one. I have attended in the past two years a few basic university courses in medicine and many specialized courses in specific medical topics, such as magnetic resonance imaging and radiography.

S: Have you any professional background in medicine?

U: Yes. For two years I have been cooperating with a medical team involved in the analysis and design of a computer-assisted heart-monitoring system.

S: Okay.
Do you want to review the formulation concerning search concept C1?

U: No.

S: Please, give me some keywords for the concept C2.
Please, rank them according to their relevance (1-5).

U: Man–Machine Interaction (4).
Human–Computer Interface (5).

S: I suggest you also consider the following terms.
Please select those you like and rank them (1-5):
Human Factors
User Modeling
Man–Machine Systems
Interaction Modes
Intelligent Interfaces

U: Human Factors (4).
Interaction Modes (4).

S: Okay.

This dialogue shows several interesting points. The first questions in the dialogue, generated by IRES, are aimed at obtaining some feedback from the user about the results obtained through the execution of the first search strategy. Since the user mentions new concepts, their disciplines and their relationships with previous concepts are checked. Later the initiative in the dialogue is taken by the user, who examines the search strategy utilized for the search, comments on it and asks for explanations. At this point IR-NLI II starts a phase of analysis of its own operation. IRES identifies first the reasons why the two terms addressed by the user (namely, Tomotron and Ramses) were not included into the search strategy. It discovers that they were discarded since: 1) IR-NLI II believes that the user is not knowledgeable in medicine and, hence, he is not reliable as far as very specific terms are concerned, 2) the terms Tomotron and Ramses are unknown to the system, i.e., there is no entry for them in DSTKB. Since the former reason depends on the user model, IRES tries to figure out why UMS believes that the user is not knowledgeable in medicine. UMS discovers that this depends on the user not having been classified as knowledgeable in medicine (the slots GENERAL PROFILE—EDUCATIONAL BACKGROUND, PROFESSIONAL BACKGROUND do not show any values related to medicine). After the user, on the request of IRES, denies the correctness of these assumptions, UMS is faced with an inconsistency that leads to a reclassification process. The stereotype knowledgeable in medicine is activated and initiates the part of the dialogue, shown above, concerning details about his background in medicine. Finally UMS signals to IRES that one of the assumptions it made is no longer valid, and that a significant modification of the model has occurred. Therefore IRES revises its operation, and reevaluates the importance and relevance of all the terms that have been collected so far and which depend on the user model. So, the user is once again engaged in a dialogue focused on obtaining a reformulation of the new concepts which are still unrelated to the others (human–computer interaction). The new, updated version of the user model after the aforementioned dialogue is shown next (only updated slots are reported).

```
GENERAL PROFILE
      EDUCATIONAL BACKGROUND
            ...
            FIELD: medicine
                  DEGREE: basic education
                  DATE: –
      PROFESSIONAL BACKGROUND
            ...
            FIELD: medicine
                  POSITION: cooperator
                  SINCE: 1987
            ...
INFORMATION RETRIEVAL PROFILE
            ...
      NORMAL EXPRESSION OF NEEDS
            ...
            DOMAIN: medicine
                  ACCURACY: average
                  COMPLETENESS: average
                  ATTITUDE: browsing-oriented
            ...
USER KNOWLEDGE
      SUBJECT DOMAINS
            ...
            DOMAIN: medicine
                  COVERAGE: low
                  DEPTH: low
            ...
```

Now IRES revised the search strategy previously produced and submits it to INSPEC. The resulting document set is collected, ranked and then shown to the user. Now the user is satisfied with the documents retrieved and exits the work session. Finally a log of the session is generated and stored in SHDB for further processing and the updated user model is stored in UMDB.

V. Evaluation and Perspectives

This section presents an informal evaluation of the approach to user modeling proposed in the paper, and develops some comparisons with related proposals. Openings for future activity are discussed as well.

A. An Informal Evaluation

The approach to user modeling for expert man–machine interfaces presented in this paper features several advantages. From the point of view of the system designer, three main points are worth mentioning:

1) the representation of user and modeling knowledge through fine-grained stereotypes allows knowledge acquisition and debugging to occur in a highly modular and incremental way, thus facilitating the job of the knowledge engineer (which turns out to be especially hard in the particular domain of user modeling);
2) the organization of stereotypes into a hierarchy supports a top–down discipline for the analysis of domain knowledge that turns out to be natural and effective;
3) the classification of the user population according to non-necessarily independent concepts (which in turn produces a collection of non-necessarily disjoint user classes) implies that stereotypes may be partially overlapping and are not required to be designed in such a way as to represent separate views of user characteristics: this feature is of the greatest importance for knowledge acquisition activity that can then focus on one single view at a time without worrying too much about the integration problem.

From the point of view of how the modeling activity is carried out, the main feature of our approach is constituted by the concept of repeated classification: default knowledge about user classes is used several times and in several different contexts during the modeling process. The most evident advantage of this choice is that the user model is updated and refined according to the actual user traits as soon as they are manifested during a work session. Modifications to the model are performed in an event-driven way, following a procedure that allows a fast-response to new evidence about the user and guarantees internal coherence and consistency with all available evidence. As a consequence, a more flexible and adaptive behavior of the expert interface is obtained. Moreover an effective exploitation of user and modeling knowledge about the user population (coded in the stereotypes) is obtained, since it is used several times and with several different purposes during a modeling session. Finally our approach based on repeated classification, also features a good level of cognitive adequacy, since it quite closely mirrors some of the mental processes of human modeling activity (at least in the specific case considered here, concerning a professional intermediary that interacts with a user of an information retrieval system).

The proposed approach to user modeling does involve a few problems. First of all, the underlying classification mechanism requires that the knowledge engineer grant a lot of attention to the definition of stereotypes and their methods, in order to prevent, in certain situations, the system from entering infinite loops involving insertion of information and classification. Secondly, during the definition of stereotypes the knowledge engineer is in no way supported in defining the ways for recognizing and (more importantly) resolving inconsistencies. Therefore stereotype definition requires strong discipline. If this is not so, a stereotype base that will generate confusing and odd behavior will ensue. A third critical aspect of the proposed approach is its efficiency: since a classification is started after any inclusion of new information into the model, and because classification requires complex processing through many stereotypes, it may slow down the performance of the expert interface.

B. Related Works

Many research projects dealing with the issue of user modeling in man–machine systems are reported in the relevant literature. Daniels [11] provides a thorough and deep analysis of most of the known approaches to the topic of user modeling. Sparck Jones [26] analyzes the problem of building and utilizing a user model within an expert system. In this section we present a critical outline of some of the most significant approaches to modeling user characteristics.

In the pioneering work by Rich [22], [23] an advice-giving system, called Grundy, is described. It simulates the performance of a good librarian at a public library when a reader looks for something to read. It recommends books to people according to a user model built by the system and a set of descriptions of available books. Grundy initially asks the reader for some personal information, and later it expands and refines the user model (called user synopsis) using predefined stereotypes, and finally searches in a data base for books that are likely to be of interest to the reader. Grundy also incorporates a mechanism (based on confidence factors) that allows the taking into account of user feedback about the level of satisfaction with the books suggested, in such a way as to refine the current user model. Stereotypes contain a set of triples ⟨attribute, value, certainty factor⟩ and are organized in a multiple hierarchy (a direct a-cyclic graph). To each stereotype one or more triggers are associated that are used to activate the stereotype. When a stereotype is activated also its parents in the hierarchy are activated. If more than one stereotype is active, their effects are combined; conflicts are dealt with by appropriately modifying certainty factors and propagating the change according to two simple rules. A learning mechanism is also illustrated that, on the basis of a parameter setting technique, allows the set of stereotypes to adapt to the user population to which the system is exposed.

With reference to UM-tool we presented here, Grundy shows the following main communalities and differences.

1) Both approaches feature a long-term user modeling. Besides of reusing the user model of the last session of that user, Grundy learns in order to modify its stereotypes, while UM-tool provides the possibility to analyze the detailed records of the interaction, in order to discover new facts about the user.
2) In both approaches stereotypes are defined, organized and used similarly, and the user model is the result of an operation of combining active stereotypes.
3) Grundy and UM-tool differ for the way in which conflicting information is dealt with: Grundy deals with it by modifying the certainty factors of the information, while UM-tool provides a framework in which more structured procedures can be defined (e.g., inconsistency methods). They offer therefore a different degree of flexibility.
4) In Grundy user classification is performed mainly at the beginning, against a set of facts directly given by the user. Therefore there is no complex activity of acquisition of user information, and even the dialogue with the user is extremely simplified.
5) In Grundy there is no way to tell whether an information derived from a stereotype is well-founded. In fact Grundy does not care about the deactivation of a stereotype: once a stereotype becomes active, it is no longer deactivated, even if its trigger becomes false.

Another advice-giving system that includes a user modeling component is described in [21]: the system, called Real Estate Agent, assumes the role of a real estate agent, where the user is an apartment-seeker. The system can propose suitable apartments chosen from a list on offer that suit the requirements and characteristics of the user. The user must provide some personal information about himself and his family (e.g., general requirements about the apartment, number of occupants, price range, etc.) and the system responds with ranked recommendations for some of the apartments in the offer. The system, given the initial set of facts about the user requirements, uses a set of stereotypes to categorize a set of choice criteria as very important, important and unimportant and to select, for each useful criterium, a set of evaluation rules used to evaluate the criterium itself ("large apartment" may mean different things to different users). This classification of criteria and the selected evaluation rules form the model of the preferences of the user. Advantages and drawbacks of each apartment are then evaluated and used to formulate an appropriate recommendation to the user. In the Real Estate Agent only short term user modeling is performed, as the user model is not retained through several sessions. The user model contains a description of the current preferences of the user, and not a description of his characteristics. Further on, these stereotypes (as they are called by the authors) are production rules used to rank the choice criteria, but do not provide any user characteristics, and neither are place-holders for auxiliary procedures (for acquisition, conflict resolution, etc.). By comparing this approach to UM-tool, the following points can be mentioned.

1) No reclassification of the user is performed, and the feedback of the user to system recommendations is not taken into account: therefore, no model revision takes place.
2) The user model is used only to store user preferences: the dialogue is not user-tailored, and the acquisition of new information and conflict resolution are independent of the user-model.

Over the last few years a general conceptual model of the activity of an intermediary mechanism (either human or machine) to information retrieval systems, called Monstrat, has been developed by Belkin and his colleagues

[4]. The Monstrat model, obtained by analyzing a large collection of discourses between intermediaries and users, specifies ten main functions performed by the intermediary during a session. One of these, the user modeling function, encompasses five subfunctions [12] aimed at identifying five different classes of information about the user. They are: 1) *user*—determining whether the user is academic or nonacademic, whether he is acting on behalf of somebody else or he is an end-user; 2) *goals*—determining current user goals; 3) *knowledge*—determining the user state of knowledge in a given field (how deep and wide is the user's knowledge); 4) *information retrieval system*—determining user's familiarity with information retrieval systems and whether he has done some previous search; 5) *background*—determining whether user's academic background is relevant for the current search, and determining other auxiliary information (his place of residence, his current employment, etc.).

In a more recent paper Belkin [3] concentrates on the explanation function in an expert intermediary system. The goal of this function is to bring the conceptual model to a level sufficient for effective interaction. His analysis indicates that the explanation given by the intermediary to a particular user is triggered by a comparison of what the user ought to know about the information retrieval scenario (the conceptual model) and what he actually believes. Therefore the intermediary must have a model of the conceptual model of the user. The model of the intermediary seems to be built very early in the interaction (therefore suggesting that stereotypes are appropriate to emulate that performance) and is subsequently incrementally modified as required.

Monstrat proposes only a conceptual approach to user modeling and it does not investigates how it could be implemented on a computer. However Belkin and his colleagues suggest that a blackboard architecture could be effectively used to implement the system, where each expert is responsible for one of the specified functions. For what concerns the user modeling function, Daniels [12] suggests that a frame-like representation should be adequate for the task. Since Monstrat is not an implemented system but rather a conceptual proposal and it is deeply bound to the information retrieval application, direct comparison with UM-tool seems inappropriate.

The I3R (Intelligent Interface for Information Retrieval) system described in [4], [10] is an expert assistant that provides information and tools to help a user formulate a query that specifies his information need. The interaction of the user with the system features a mixed initiative, and the system follows a basic plan of action comprising the following principal tasks: constructing the user model, determining the model of the request of the user, searching in the data base, and evaluating the retrieved documents.

The user model is mainly used for offering explanations at a level appropriate to the user (e.g., more help is given to the user on selecting terms rather than offering assistance on system use). The user model includes information about the domain knowledge of the user, information derived from user stereotypes and an interaction summary. User stereotypes describe classes of users according to two simple classification criteria: how familiar the user is with the subject domain and how familiar he is with the system I3R. I3R is implemented on a blackboard architecture, comprising seven cooperating experts. A scheduler controls them (by modifying priorities associated with their operations) according to the basic hierarchical plan. Information concerning the user is acquired by directly asking him. An interaction summary derived from a system journal contains a description of what facilities the user utilizes, how often, etc.

In comparison with UM-tool the following may be noticed.

1) What the stereotypes contain and the way in which they are used to construct the user model is not clearly illustrated. However it seems that, differently from UM-tool, most of the information concerning the user are directly acquired from him since there are no inference mechanisms capable of producing such information.
2) The user model has a much more limited content than that we use in the IR-NLI II application.
3) Once a user model has been constructed, no model revision activity takes place.

Kok and Botman [19] describe an intelligent interface to a data base management system, called Impact. The main assumption underlying their work is that it is usually difficult for a human to exactly describe what his interests are, while, on the other hand, a data base system requires a precise and exhaustive query. They propose therefore an interaction system that helps the user to obtain the information that turns out to be the most interesting for him. Such an interaction system must be *active* in the sense that: 1) it should provide extra information not explicitly asked, but that might be of some interest for the user, 2) it should remember preferences from previous sessions, and 3) it should suggest and answer, when appropriate, follow-up questions. The interaction system must also be *impertinent* in the sense that it warns the user when new information, deemed interesting for the user, enters the data base system.

To perform in this way the interaction system utilizes a user model and a data model. The user model captures the user current interests, while the data model describes how users perceive the data. The authors propose to use *profiles* to describe the current interests of a user in a certain field: these, appropriately combined, form the *focus*, i.e., the model of all the interests of a user. The focus is then used for deciding what kind of data to retrieve and how to present them.

Comparing impact with UM-tool the following observations can be made.

1) Both approaches lead to a dynamic user modeling, although the user models and modeling procedures are very different.

2) The type of user model utilized in Impact is more a model of the interests of the user, rather than a model of permanent user traits. This simplifies somewhat the modeling process, but limits the variety of reasoning activities that can be performed using the models.
3) In this approach there are no stereotypes as they are generally understood in the literature. A profile does not denote a class of users, but just the temporary interest of a single user in a certain field. For this reason it is possible for the system to generate completely new profiles during the interaction with the user.
4) There is no trigger for profiles, and they are used for the generation of the focus according to a quite rigid procedure.
5) Moreover the problem of how to resolve conflicts is not considered.

C. Openings and Future Activity

The practical experience had with IR-NLI II, and especially with UM-tool, as reported in the previous sections, has disclosed several directions for future activity. Two sets of activities, currently ongoing and deserving further attention in the near future, have been identified.

The former concerns the default reasoning approach outlined. In particular it seems that there are some general principles that specify how a general class of inconsistencies can be treated, instead of operating on a case-by-case basis. This kind of behavior could be obtained by an ATMS-like module [13] to keep track of the belief status of each information in the model, coupled with an inference engine that determines the stereotypes to process, the links to follow, and those to inhibit.

The latter set of activities concerns practical, application-oriented issues focussed on an extended experimentation of the IR-NLI II system with an appropriate sample of casual users with the purpose of testing, among others, the following points: 1) long-term effects of the modeling process, including convergence and stability of the user model, incremental improvement, saturation effects, etc. 2) adequacy of the stereotype knowledge base from the knowledge engineering point of view, and effects of its content on the overall modeling process; and 3) evaluation of the effects of user model refinement on the overall performance of the expert interface.

In addition to this experimental activity, the completion and improvement of the natural language dialogue subsystem and the design and implementation of a specific explanation and tutoring module are also foreseen.

References

[1] J. L. Alty and G. Guida, "The use of rule-based system technology for the design of man–machine systems," in *Analysis, Design and Evaluation of Man–Machine Systems, Proc. 2nd IFAC/IFIP/IFORS/IEA Conf.*, G. Johannsen, G. Mancini, and L. Mårtensson, Eds., London: Pergamon Press, 1985.

[2] M. J. Bates, "Information search tactics," *J. Amer. Soc. Informat. Sci.*, vol. 30, pp. 205–214, 1979.

[3] N. J. Belkin, "On the nature and function of explanation in intelligent information retrieval," in *Proc. 11th Int. Conf. Research Development Informat. Retrieval*, Grenoble, France, 1988, pp. 135–145.

[4] N. J. Belkin *et al.*, "Distributed expert based information systems: An interdisciplinary approach," *Informat. Process. Management*, vol. 23, no. 5, 1987.

[5] G. Brajnik, G. Guida, and C. Tasso, "An expert interface for effective man–machine interaction," in *Cooperative Interfaces to Information Systems*, L. Bolc and M. Jarke, Eds., Berlin, FRG: Springer-Verlag, pp. 259–308, 1986.

[6] ____, "Design and experimentation of IR-NLI: An intelligent user interface to bibliographic data bases," in *Expert Data Base Systems*, L. Kerschbergb, Ed., Menlo Park, CA: The Benjamin/Cummings Publishing Company, pp. 151–162, 1987.

[7] ____, "User modeling in intelligent information retrieval," *Informat. Process. Management*, vol. 23, no 4, pp. 305–320, 1987.

[8] ____, "Modeling individual users in the IR-NLI II system for intelligent information retrieval," in *Analysis, Design and Evaluation of Man–Machine Systems, Proc. 3rd IFAC/IFIP/IFORS/IEA Conf.*, J. Ranta and B. Wahlström, Eds., London: Pergamon Press, 1988.

[9] ____, "IR-NLI II: Applying man–machine interaction and artificial intelligence concepts to information retrieval," in *Proc. 11th Int. Conf. Res. Development Informat. Retrieval*, Grenoble, France, 1988, pp. 387–399.

[10] W. B. Croft and R. H. Thompson, "An expert assistant for document retrieval," Report COINS-85-05, Dept. of Computer and Information Science, Univ. of Massachusetts, Amherst, MA, 1985.

[11] P. J. Daniels, "Cognitive models in information retrieval—An evaluative review," Final report to British Library Research and Development Department, Project No. SI/G/753, The City Univ., London, UK, 1986.

[12] ____, "The user modelling function of an intelligent interface for document retrieval systems," in *Proc. IRFIS 6: Intelligent Informat. Syst. Informat. Soc.* Amsterdam, NL: North-Holland, 1986, pp. 162–176.

[13] J. De Kleer, "An assumption-based TMS," *Artificial Intell.*, vol. 28, pp. 127–162, 1986.

[14] J. Doyle, "A truth maintenance system," *Artificial Intell.*, vol. 12, pp. 231–272, 1979.

[15] G. Guida and C. Tasso, "NLI: A robust interface for natural language person–machine communication," *Int. J. Man–Machine Studies*, vol., 17, pp. 417–433, 1982.

[16] ____, "IR-NLI: An expert natural language interface to online data bases," in *Proc. ACL Conf. Appl. Natural Language Process.*, Santa Monica, CA, 1983, pp. 31–38.

[17] ____, "An expert intermediary system for interactive document retrieval," *Automatica*, vol. 19, no. 6, pp. 759–766, 1983.

[18] W. Hoeppner, K. Morik, and H. Marburgber, "Talking it over: The natural language dialog system HAM-ANS," in *Cooperative Interfaces to Information Systems*, L. Bolc and M. Jarke, Eds., Berlin, FRG: Springer-Verlag, pp. 189–258, 1986.

[19] A. J. Kok and A. M. Botman, "Retrieval based on user behaviour," in *Proc. 11th Int. Conf. Res. Development Informat. Retrieval*, Grenoble, France, 1988, pp. 343–357.

[20] F. W. Lancaster, *Information Retrieval Systems.* New York: John Wiley, 1979.

[21] K. Morik and C.R. Rollinger, "The real estate agent—Modeling users by uncertain reasoning," *AI Mag.*, vol. 6, no. 2, pp. 44–52, 1985.

[22] E. Rich, "User modeling via stereotypes," *Cognitive Sci.*, vol. 3, pp. 329–354, 1979.

[23] ____, "Users are individuals: Individualizing user models," *Int. J. Man–Machine Studies*, vol. 18, pp. 199–214, 1983.

[24] P. Shoval, "Principles, procedures, and rules in an expert system for information retrieval," *Informat. Process. Management*, vol. 21, no. 6, pp. 475–487, 1985.

[25] D. Sleeman, D. Appelt, K. Konolige, E. Rich, N. S. Sridharan, and B. Swartout, "User modeling panel," in *Proc. 9th Int. Joint Conf. Artificial Intell.*, Los Angeles, CA, 1985, pp. 1298–1302.

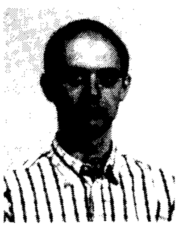
[26] K. Sparck-Jones, "Issues in user modelling for expert systems," in *Artificial Intelligence and Its Applications*, A. G. Cohn and J. R. Thomas, Eds. New York: John Wiley, 183–195, 1986.

[27] K. Sparck-Jones, "Realism about user modelling," Technical Re-

port 111, Computer Laboratory, Univ. of Cambridge, Cambridge, UK, 1987.

[28] A. Vickery, E. Brooks, and B. Robinson, "A reference and referral system using expert systems techniques," *J. Documentation*, vol. 43, no. 1, pp. 1–23, 1987.

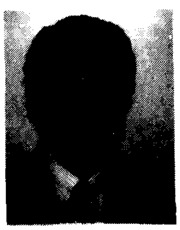
[29] W. Wahlster and A. Kobsa, "Dialog-based user models," Report No. 3, Univ. of Saarbrücken, Saarbrücken, FRG, 1985.



Giorgio Brajnik received the doctoral degree in computer science in 1984 from the University of Udine, Italy.

Since 1984 he is with the Department of Mathematics and Computer Science of the University of Udine where, since 1986, he is a Research Associate.

Dr. Brajnik's research interests are within artificial intelligence, and particularly in intelligent man–machine interfaces, user-modeling, multiple modeling of physical systems and distributed artificial intelligence systems. He is member of AAAI.



Giovanni Guida received the doctoral degree in electronic engineering (computer science) from the Politecnico di Milano, Italy in 1975.

Since 1979 he has been Associate Professor of Computer Science at the University of Udine, Italy where he co-founded the AI Laboratory of the Department of Mathematics and Computer Science. Since 1986 he is full Professor of Computer Science at the University of Udine and he also teaches a course at the Politecnico di Milano.

Dr. Guida's major research interests are in the fields of natural language text understanding, dialogue systems, intelligent interfaces, user modeling, distributed problem solving, physical systems modeling, diagnostic knowledge-based systems, knowledge-based scheduling and planning, evaluation of intelligent systems. He is author of several scientific papers and co-editor of the volumes "Computational models of natural language processing," (North-Holland, 1984) and "Expert system design: methodologies and tools," (North-Holland, 1989). He is also active as a scientific and technical advisor for research and development of industrial applications in the fields listed previously. He is representative of AI*IA (the Italian Association for Artificial Intelligence) in ECCAI (the European Co-ordinating Committee for Artificial Intelligence). He is member of IFAC Working Group on "Man–machine systems," AAAI, and ACL.

Carlo Tasso received the doctoral degree in electronic engineering (computer science) from the University of Trieste, Italy in 1980.

He has been at the University of Udine, Italy, since 1981. He is currently Associate Professor of Programming Languages. He has been a co-founder of the AI Laboratory of the Department of Mathematics and Computer Science of the University of Udine.

His major research interests are in the areas of knowledge-based systems and natural language processing. More specifically his research projects have included intelligent interfaces, natural language interfaces, knowledge representation methods for natural language texts, user modeling, multiple modeling of physical systems, distributed problem solving, and intelligent tutoring systems. He is author of several scientific papers and recently co-edited the volume "Expert system design: methodologies and tools," (North-Holland, 1989). He is a member of AAAI, ACL, and AI*IA (the Italian Association for Artificial Intelligence).