Information Processing & Management Vol. 23, No 4, pp. 305–320, 1987
Printed in Great Britain

0306-4573/87 $3.00 + 00

USER MODELING IN INTELLIGENT INFORMATION RETRIEVAL

GIORGIO BRAJNIK, GIOVANNI GUIDA, and CARLO TASSO
Laboratorio di Intelligenza Artificiale, Dipartimento di Matematica e Informatica, Università di Udine, Udine, Italy

(*Received* 25 *February* 1987)

Abstract—The issue of exploiting user modeling techniques in the framework of cooperative interfaces to complex artificial systems has recently received increasing attention. In this paper we present the IR-NLI II system, an expert interface that allows casual users to access online information retrieval systems and encompasses user modeling capabilities. More specifically, an illustration of the user modeling subsystem is given by describing the organization of the user model proposed for the particular application area, together with its use during system operation. The techniques utilized for the construction of the model are presented as well. They are based on the use of stereotypes, which are descriptions of typical classes of users. More specifically, they include both declarative and procedural knowledge for describing the features of the class to which the stereotype is related, for assigning a user to that class, and for acquiring and validating the necessary information during system operation.

1. INTRODUCTION

The development of cooperative user interfaces for supporting information retrieval systems has become a well-defined research and application field. It comprises both traditional systems, including menu-driven interaction, extensive online help, and keyword recognition and extraction—consider, for example, the work of Marcus [1,2] and Doszkos and Rapp [3]—and more advanced interfaces based on artificial intelligence techniques. This class includes, among others, the work of Pollitt [4–6], the systems developed by Croft and Thompson [7] and by Defude [8,9], and the IR-NLI interface designed and implemented by the authors [10–12].

From a general viewpoint, the design of an expert interface to an information retrieval system encompasses two major tasks:

- How to overcome the linguistic gap between the user and the system.
- How to support the user at the conceptual level in the analysis of his information needs, in the formulation of an appropriate search strategy, and in the evaluation of the obtained results.

These issues, in turn, pose several technical problems, which include:

- Natural language understanding and dialogue management.
- Representation of subject knowledge in the domain of the search (including available data bases, their content, terminology and organization).
- Representation of technical knowledge about information retrieval (session structure, query language, techniques for strategy construction).
- Elicitation and representation of the intermediary's skill and expertise.
- Design of appropriate problem-solving methods for knowledge processing and inference management.

Giovanni Guida is also affiliated with Progetto di Intelligenza Artificiale, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy.
Carlo Tasso is also affiliated with CISM (International Center for Mechanical Sciences), Udine, Italy.

All these topics have been dealt with by the authors in the last years in the frame of the IR-NLI (Information Retrieval–Natural Language Interface) project, which has produced a prototype system (written in LISP and running on a SUN-2 workstation) devoted to support end users in accessing a data base on computer science (operating systems). During experimentation with this prototype, it became apparent that a major bottleneck of IR-NLI is the lack of capability of adapting its behavior to different types of users. IR-NLI embodies a fixed implicit model of a typical user of an information retrieval system, and it is unable to recognize and take into account the specific characteristics of each individual user. Naive users, users with specific background in the search domain, newcomers, and very experienced users are all treated the same way.

This motivated the extension of IR-NLI with specific user modeling capabilities. A new version of the prototype, called IR-NLI II, has been designed and is currently being developed.

The present paper is devoted to a discussion of the user modeling issue and its application to the information retrieval task. General knowledge about information retrieval is assumed; the reader may refer to refs. 13–16. Also, the general architecture and the mode of operation of IR-NLI are only briefly surveyed; further details may be found in the studies by Guida and Tasso [10] and Brajnik, Guida, and Tasso [11,12].

The paper is organized in the following way. Section 2 discusses the general problem of user modeling and illustrates the major approaches relevant to information retrieval. Section 3 presents the overall organization of IR-NLI II. In Section 4 the structure and content of the user model adopted in IR-NLI II are described, and the specific roles of user modeling within IR-NLI II are discussed. Section 5 is devoted to illustrate the techniques used for constructing and refining user models. Finally, Section 6 presents some conclusive remarks and outlines the directions of future activity.

2. SURVEY OF USER MODELING

In this section we first present a general definition of user modeling, followed by a short analysis of the main taxonomies of user modeling proposed in the literature. Later, attention is focused on the description of some experimental systems that encompass user modeling techniques, with particular attention to the domain of intelligent information retrieval.

2.1. Frameworks for user modeling classification

By "user modeling" we refer to the consideration of any kind of information that a program has about its users, to be utilized in order to increase, in a general sense, the level of man-machine interaction [17,18]. More specifically, user modeling is aimed at improving the performance of the system, for what concerns both tuning the system external behavior to the interaction (e.g., dialogue with the user, information displayed or explanations given, corrections of user's errors and possible suggestions) and adjustment of system internal operation to user's characteristics.

The issue of user modeling has been raised in many areas of artificial intelligence, from man-machine interfaces to expert systems and intelligent tutoring systems [18], causing in such a way a proliferation of different approaches and techniques. Several classification schemes have been proposed in the literature, which can be employed in order to provide an analytical understanding of user modeling. These schemes are helpful in characterizing both the common features and the peculiarities of each approach, and are used to clarify how user modeling can be taken into consideration in the design of intelligent information retrieval systems. In the following, we will survey two major approaches, one proposed by Carbonell [19] and one proposed by Rich [20].

The scheme proposed by Carbonell identifies two broad categories: empirical quantitative models and analytical cognitive models. Empirical quantitative models entail information derived from an abstract formalization of general classes of users. The model is defined through parameters compiled from empirical data, encoding quantitative relations between primitive operations carried out by the user during the interaction with the system in solving a specific task, and a measurement of the performance shown by the user

in solving such a task. Classical examples of this class are the keystroke model developed by Card, Moran, and Newell [21] and the ZOG system [22]. These models contain only surface knowledge about the user (or a specific class of users), and no internal reasoning takes place. Furthermore, this knowledge is usually taken into consideration explicitly only during the design of the system, and then it is hardwired into its implementation. The designer considers a defined class of users without trying to conform to the peculiarities of each individual user, adapting in such a way the system behavior to the common features of the whole class. Therefore, the resulting system does not contain any separate knowledge base devoted to represent user modeling information.

Analytical cognitive models, on the other hand, are aimed at simulating aspects of user cognitive processes taking place during interaction with the system. These models are based on explicit representation of user knowledge, of a rather qualitative nature compared to the quantitative one of the preceding case. Implementation strongly utilizes artificial intelligence techniques, differing in such a way from the former class, which is well suited for traditional (non-knowledge-based) methodologies. The consideration of a knowledge base devoted to store user modeling information allows the specific traits of each single user in a given class to be followed.

Rich, in her fundamental effort toward classifying user models [20], finds three dimensions to be useful in building up a taxonomy. The first dimension is canonical versus individual models, that is, one single model of a typical user versus a collection of many individual models. The first category proposed by Carbonell conforms to the canonical model approach, whereas any system that has to be capable to tailor its behavior to a heterogeneous variety of users has to conform to the individual model paradigm.

The second dimension of the space of user models is explicit versus implicit modeling, that is, models provided explicitly by the user (either under user control or system control) versus models built up by the system. From the point of view of an increase of the level of man-machine interaction and of the usability of the system, the latter approach is much more desirable because of its unobtrusive nature and higher degree of reliability. In fact, the model can be built without user intervention, and the knowledge included in the model, though uncertain because it results from an inference process, is generally more reliable than that directly provided by the user, which is usually a bad source of information about himself [23].

The last dimension to consider in classifying user models is short-term versus long-term modeling, the former focusing on information that changes in the short term (e.g., during a single session) and the latter on characteristics that change more slowly, possibly over a long period of time (e.g., a whole series of sessions). This last dimension is useful to point out some of the differences between user modeling techniques adopted in intelligent tutoring systems and those utilized in man-machine interfaces. In fact, the rapid change of user characteristics during a session is a distinguishing feature of tutoring systems, which indeed have the goal of changing (possibly in the short term) the knowledge level of the student. This goal of educating or training the user is only secondary in man-machine interfaces, where more emphasis is placed on the accomplishment of some very specific task (typically using an artificial system, rather than learning how to use it). From this discussion it follows that a whole class of modeling techniques developed in the framework of intelligent tutoring systems do not seem adequate to cope with man-machine interaction tasks that are not directly related to teaching situations. Independently of short-term or long-term modeling, a dynamic evolution of the model can, however, be present both in intelligent tutoring systems and in man-machine interfaces, whenever the user modeling technique employed includes also the capability of updating the model during a single session for refinement purposes. In this case, the model will change in the short term as the system continuously improves it, making it conform more to the real situation of the user, but independently of any possible evolution of the user himself.

2.2 User modeling in intelligent information retrieval

We turn now to a brief survey of some applications of user modeling techniques in the information retrieval domain. A few different proposals will be analyzed utilizing as much as possible the classification frameworks presented in the previous section. As a gen-

eral remark, all information retrieval applications are based on analytical cognitive models conforming to the individual modeling paradigm.

Limited examples of user modeling can be found in the THOMAS [24] and ASK [25, 26] systems. Although the two projects differ for the specific applications considered and the implementation techniques utilized, they both employ modeling knowledge containing short-term information about the user's specific requests. No long-term modeling is present in either system. ASK utilizes an explicit method for acquiring the model from the user: it identifies the needed information from a written statement or a transcribed interview provided by the user. No refinement of the model takes place during interaction. On the other hand, THOMAS shows a rather more implicit approach to user modeling: a model updating is driven by information provided by the user upon system request.

GRUNDY [20,27] exploits user modeling more extensively. Its main objective is to tailor interaction to the individual user. The system utilizes a hierarchy of frames, called stereotypes, for storing knowledge about possible users; each frame comprises several slots, called facets. At the beginning of the interaction, GRUNDY prompts the user with some specific questions and, from the answers obtained, it collects the information needed to select the stereotypes appropriate for that user. These will constitute the initial user model. The model contains long-term knowledge preserved from session to session, which includes the user's background and characteristics, and a record of past interactions with the system. An (implicit) refinement of the model is carried out by GRUNDY when the user expresses his judgment on the quality of the results produced so far by the system: The refinement is performed by changing (increasing or lowering) confidence factors attached to the facets of the stereotypes belonging to the current user model; it is supported by information gathered through a bounded scope dialogue driven by the system.

Finally, the work of Croft and Thompson [7] presents an extended proposal for utilizing user modeling to increase the effectiveness of an intelligent information retrieval system. The model comprises both long-term knowledge for a general characterization of the user (such as user's domain knowledge and summary of previous interactions and requests) and short-term information concerning specific user needs submitted in the current session. Both explicit and implicit strategies are used to acquire knowledge to be inserted in the model. Frame-like structures, called stereotypes, are used for encoding descriptions of classes of users. They can be activated by means of explicit information contained in the answers that the user gives to questions posed by the system. The model of a user built during the initial session is later refined during subsequent interactions with the system, in order to improve the accuracy of the description of user's characteristics.

As a conclusion of the above discussion, we point out some specifications that seem appropriate for approaching user modeling in intelligent information retrieval. The general framework is that of analytical cognitive models, taking into account individual descriptions. An implicit approach to the acquisition of model information is desirable, but not always possible; in several cases, a mixed strategy has to be considered. The kind of knowledge to be included in the model should pertain to both short-term and long-term information. More specifically, the latter should comprise the general characteristics of the user, his cultural and educational level, the specific knowledge about the subject matter considered in the information requests, his knowledge about information retrieval systems, and the like. This information should be stored from session to session, and it must be considered as slowly varying in time. A refinement activity of this part of the model can be considered as a long-term objective of user modeling. On the other hand, short-term information consists, first of all, of data concerning the current session, including specific information needs, user plans and intentions, objectives of the search, and points of view to be considered in the search.

3. OVERVIEW OF IR-NLI II

3.1. Overall architecture

To meet the general requirements outlined in the previous section, the architecture of IR-NLI [10–12] has been extended to include new capabilities devoted to user modeling.

The architecture of the new system, called IR-NLI II, is shown in Fig. 1. Two major subsystems can be identified: the information retrieval expert subsystem and the user modeling subsystem. The information retrieval expert subsystem is designed to conduct the search session, interact with the user, and interrogate the information retrieval system. More specifically, it is responsible for performing three main functions: (1) natural language dialogue handling; (2) assistance in the elicitation of the information needs of the user and in the refinement of the search formulation; and (3) construction of the search strategy and its submission to the information retrieval system.

The user modeling subsystem, on the other hand, is designed to carry out the user modeling activity, both within a search session and over several sessions. It performs two basic tasks: (1) extraction of information relevant to user modeling from the dialogue between the user and the system and (2) construction and updating of the user model.

The information flow between the two subsystems is fixed in content, form, and direction. The subsystems share a common data base, the user model, and utilize two communication channels. The user model is constructed and updated by the user modeling subsystem, while it can only be accessed in read mode by the information retrieval expert subsystem. It contains information that characterizes the user currently interacting with IR-NLI II. The two communication channels between the information retrieval expert subsystem and the user modeling subsystem allow, respectively, the transfer of information regarding the current user (flowing from the information retrieval expert subsystem to the user modeling subsystem) and the requests of further knowledge issued by the user modeling subsystem and directed to the user and hence flowing from the user modeling subsystem to the information retrieval expert subsystem.

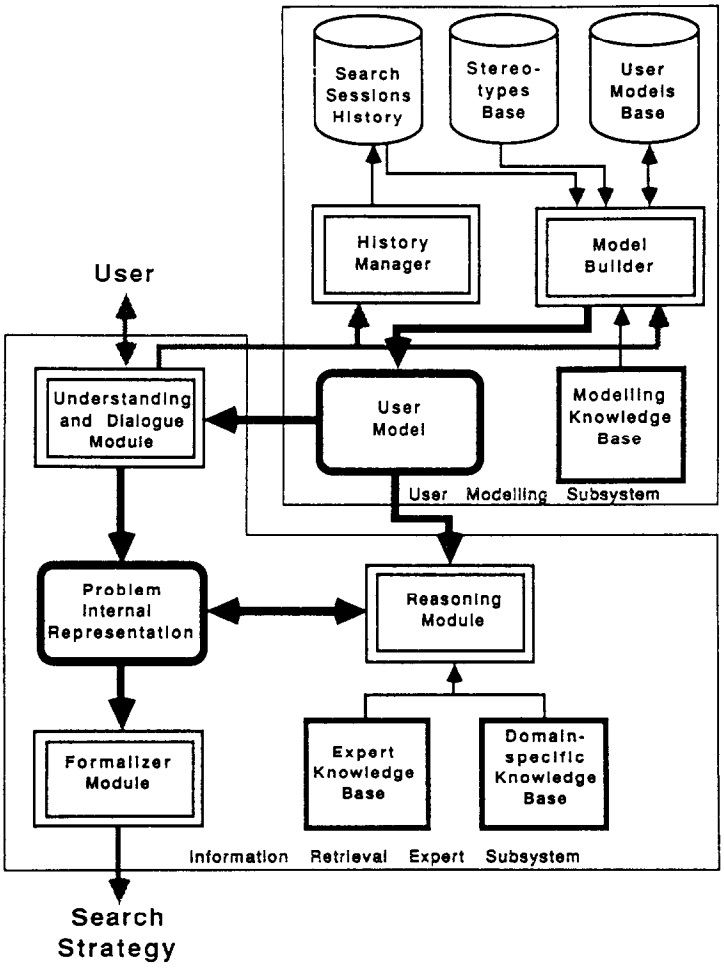

Fig. 1. Architecture of IR-NLI II.

Since the exchange of information is fixed and the operation of the two subsystems is quite independent, both subsystems can work in parallel, synchronizing just for information exchange. Furthermore, since the user modeling subsystem works on data gathered by the information retrieval expert subsystem, it can be activated by the information retrieval expert subsystem only when needed.

3.2. Information retrieval expert subsystem

This part of IR-NLI II comprises basically the original IR-NLI system extended in such a way as to support the interface with the user modeling subsystem. The information retrieval expert subsystem comprises three modules:

1. The understanding and dialogue module, which manages the interaction with the user, including input comprehension and question generation.
2. The reasoning module, which analyzes, incrementally refines, and completes the initial description of the information problem given by the user and produces a search formulation suitable for constructing the appropriate search strategy (the formal program to be submitted to the information retrieval system).
3. The formalizer module, which constructs the search strategy and actually connects with the information retrieval system.

The kernel of the information retrieval expert subsystem is the reasoning module, which uses two knowledge bases:

1. The expert knowledge base, which contains knowledge (represented through production rules) devoted to modeling the competence and skill of the human intermediaries [13–15].
2. The domain-specific knowledge base, which contains a representation of the specific subject domains to which the information problems considered refer (represented through a semantic network) and which mirrors the structure and content of usual searching referral aids, such as thesauri and directories.

The working memory of the information retrieval expert subsystem is the problem internal representation, which contains the information (provided by the user, acquired from the user model, or extracted from the domain-specific knowledge base) that describes the current information problem.

3.3. User modeling subsystem: basic structure

The user modeling subsystem is composed of two modules:

1. The model builder, which constructs and updates the user model.
2. The history manager, which records a summary of each search session into a long-term data base.

The history manager takes in input information related to the current user-system interaction and collects it into a data base of session summaries, called search sessions history, to be used as a source for statistical processing. The input to the history manager consists of answers to specific questions posed by the system to the user (e.g., his background, experience level in some domain) or information directly provided by the user and describing some aspects of his current needs (e.g., search objectives and limitations).

The search sessions history contains data referring to past sessions, organized according to user names and dates of sessions. It is used by the model builder to construct or refine a user model, through statistical processing of stored records, devoted to extracting the most common features and traits of each individual user (e.g., his preferences and communication attitude). The search sessions history is updated at the end of each search session with the data gathered by the history manager.

The model builder produces or updates the model of the current user. The model is successively utilized by the understanding and dialogue module for tuning the dialogue and by the reasoning module directly for its operation.

The model builder is connected to the search sessions history and to two other data bases:

1. The stereotypes base, which contains a collection of stereotypes, that is, canonical user models describing several classes of users in terms of common traits and typical features. The stereotypes base is accessed whenever a new user connects to the system, and it supplies a first tentative and partial model of the current user.
2. The user models base, which stores the model of each user known thus far to the system. The user models base is accessed in the first part of a search session, and it supplies the model of the current user constructed thus far (if the user is already known to the system). At the end of the session the (possibly refined) user model is stored back in the user models base.

The model builder is supported by a specific knowledge base, called the modeling knowledge base, devoted to encoding knowledge (represented through production rules) about the modeling process.

4. ORGANIZATION AND USE OF THE USER MODEL

4.1. Content and structure of the model

At a conceptual level, the user model is represented by a frame structure divided into two parts: user profile and user knowledge. In addition to these subframes, a user model includes a user identification name, which identifies the user to whom it relates, and a model history, which contains information about the history of the model (e.g., creation date and successive refinements and updating). The resulting structure of the user model is therefore:

MODEL < user identification name >
< model history >
< user profile >
< user knowledge >

The user profile subframe encompasses knowledge about specific features, attitudes, and traits of an individual user of an information retrieval system. The user profile referring to a hypothetical user is shown below as an example:

USER PROFILE
EDUCATION
FIELD: computer science
DEGREE: PhD
DATE: 1980
FIELD: medicine
DEGREE: master
DATE: 1985

PROFESSIONAL BACKGROUND
FIELD: computer science
KIND: academic
EXTENT: 4 years

INFORMATION RETRIEVAL BACKGROUND
EDUCATION: medium
TRAINING: medium

```
EXPERIENCE
        TYPE: user
        MODE: assisted
        EXTENT: 2 years

        TYPE: user
        MODE: through IR-NLI
        EXTENT: 6 months

PERSONAL TRAITS
    COMMUNICATION
        LEVEL: concise
        QUALITY: precise
    ATTITUDE: confident, cooperative

USUAL SEARCH REQUIREMENTS
    DOMAIN: computer science
        SEARCH OBJECTIVES: high precision
        OPERATION MODE: off-line preparation
        LIMITS
            DATE: 2 years
            LANGUAGE: English
            TREATMENT: technical
        OUTPUT FORMAT
            FIELDS: title, author, affiliation, abstract,
                    date, references
            MODE: off-line
    DOMAIN: medicine
        SEARCH OBJECTIVES: high recall
        OPERATION MODE: browsing
        LIMITS
            DATE: 5 years
            LANGUAGE: English, French
        OUTPUT FORMAT
            FIELDS: title, abstract, date
            MODE: on-line
```

This user profile describes a user with an extensive educational background in computer science, with four years of academic experience, and a fairly good knowledge of medicine, without practical experience. The user also has a good background in information retrieval: He has had specific training in information retrieval, considerable experience (2 years) in intermediary assisted searching, and some experience (6 months) in information retrieval using the IR-NLI interface. The user communicates in a precise and concise manner, and feels confident with the system (IR-NLI II). Finally, when searching in computer science, he usually specifies a high-precision objective, prefers the formulation of the search strategy to be done before actually searching the data base, and specifies that the documents to be retrieved must not be older than two years, must be written in English, and should be printed off-line according to the requested format (title, author, affiliation, abstract, date, references). When searching in medicine, however, the search requirements are quite different, since the user is much less knowledgeable in and familiar with this field.

The user knowledge subframe is devoted to storing information describing what the user knows about the environment of IR-NLI II operation. The user knowledge referring to a hypothetical user is given below as an example:

USER KNOWLEDGE

SUBJECT DOMAINS
DOMAIN: computer science
COVERAGE: high
DEPTH: very high
DOMAIN: medicine
COVERAGE: medium
DEPTH: low
DOMAIN: internal medicine
COVERAGE: high
DEPTH: medium

DATA BASES
FILE: inspec
FEATURES: technical
UPDATING RATE: 1 month
TOPIC: computer science
TERMINOLOGY: well known
COST: known
FILE: medline
FEATURES: technical
UPDATING RATE: 1 month
TOPICS: medicine
TERMINOLOGY: low
COST: not known

INFORMATION RETRIEVAL SYSTEMS
HOST: dialog
FUNCTIONS: high
LANGUAGE
SYNTAX: high
USE: medium

INFORMATION RETRIEVAL ACTIVITY
SEARCH SESSION STRUCTURE: low
APPROACHES: building block, citation pearl growing
TACTICS: pinpoint, respell, sibling, truncate

This example refers to a user with specific knowledge in three subject domains—computer science, medicine, and internal medicine—in which he has different competence levels (both for coverage and depth of knowledge). The user knows about two data bases—INSPEC and MEDLINE—and for each of them he has knowledge about the type of stored information, the updating rate, the topics covered, the terminology used, and the cost of accessing the data base. Moreover, the user only knows about one information retrieval system, namely DIALOG; he has good knowledge of the functions of DIALOG and of the syntax of the query language, but only limited experience with its use. Finally, the user is not fully acquainted with the structure and organization of a typical search session, and only some of the most usual approaches (building block and citation pearl growing) and tactics (pinpoint, respell, sibling, truncate) are known to him.

It is worthwhile noting that some information present in the model represents an evaluation made by the system about user knowledge (e.g., COVERAGE: high), while other information refers to specific knowledge possessed by the user (e.g., UPDATING RATE: 1 month). Moreover, even if some information in the user model is static (e.g., EDUCATION) and other is typically dynamic (e.g., INFORMATION RETRIEVAL ACTIVITY),

we assume, according to the hypothesis made at the end of Section 2, that the user and hence his model do not change over the time period of interest for IR-NLI II operation.

4.2. How to use the model

As seen from the architecture of IR-NLI II, the user model can be used by the reasoning module and by the understanding and dialogue module. In general, it supplies further information that supports the system when performing some operations, the most important of which are:

1. Tuning the user-system dialogue. An example is the generation of explanations and justifications about system operation: knowledge contained in the slots INFORMATION RETRIEVAL BACKGROUND and INFORMATION RETRIEVAL ACTIVITY can be useful to tailor the level and content of system-generated utterances to each individual user.
2. Interpreting user utterances and answers to system queries. For example, during the presearch interview, knowledge provided by the COMMUNICATION slot may be useful to support the system when interpreting user's utterances.
3. Completing the current problem internal representation using default information extracted from the user model. For example, the slot USUAL SEARCH REQUIREMENTS provides default information that can be used when more specific values are missing.

5. CONSTRUCTING USER MODELS

5.1. Model building techniques

The techniques utilized by the model builder for collecting or producing the information necessary to construct the user model can be classified according to two major criteria: (1) elicitation mode, that is, the way information is organized at acquisition time, and (2) acquisition procedure, that is, the way information is actually collected or produced.

According to the first criterion, two basic modes of information elicitation can be mentioned:

- Acquisition of a single information item at a time, which represents a specific fact about the user currently interacting with the system and corresponds to the content of just one slot of the model.
- Acquisition of a cluster of information items in one shot, which represents a collection of facts about the current user that are in some way interrelated and correspond to the content of several slots of the model. The rationale behind this mode of information acquisition is that generally the several possible aspects of each individual user are not uniformly distributed over a user population, but come in clusters: Some features are always present together with some specific others and exclude the presence of some different ones.

According to the second criterion, three main procedures for information acquisition can be identified:

- Observation includes dialogue inspection, that is, observation of free dialogues between the user and the system, and answer analysis, that is, observation of the answers provided by the user to specific direct questions posed by the system. Observation refers to the fact that the information items to be used for building a user model are already present in the information source considered; they only have to be identified and inserted in the relevant slots of the model.
- Inference from observed facts encompasses two steps: (1) acquisition through observation of facts that are not appropriate to be directly inserted in the user model but can serve as raw data from which some useful information items can be derived and

(2) inference from the observed facts of the appropriate information items to be inserted in the model.

- Inference from known facts is the expansion and refining of the model without using new information items acquired from the user-system dialogue.

Of course, single and cluster modes of information elicitation can be combined in various ways with the above-mentioned procedures for information acquisition in order to produce a full range of possibilities. Not all combinations are meaningful. For example, observation and cluster mode do not fit together well; one should not confuse possibly repeated observation of several facts with the global one-shot acquisition of a set of interrelated facts. One particularly useful combination is instead the cluster mode in conjunction with inference from observed facts or inference from known facts. The implementation of such a technique, however, requires that the model builder have the ability to store clusters of facts together and access them when needed through a sophisticated associative search. Each cluster, called stereotype, describes a class of users sharing a collection of common traits. Thus, given some criterion for classifying a community of users, the common features of each class can be grouped together to form a stereotype. Therefore, this provides a partial description of each member of the class.

Let us note finally that each technique (mode/procedure) for model building may be appropriate for filling some slots of the model but unsuitable for others, so that a combination of several techniques is needed for the global task of the model builder.

5.2. *Structure and content of stereotypes*

As it has been illustrated in the previous section, stereotypes play a major role in building user models, particularly in the case where the system acquires user information in clusters. A stereotype is a description of a class of users and specifies their most important aspects. Usually a stereotype does not describe all known user traits but is restricted to just a subset of them, and therefore it provides only a partial description of the class.

A stereotype has a frame-like structure that includes both declarative and procedural knowledge, organized, respectively, in slots and procedures attached to the slots. From a structural viewpoint, a stereotype has basically the same organization of the user model, where the ⟨model history⟩ part is omitted. Moreover, each stereotype is identified by a name. Thus, the overall structure of a stereotype is:

STEREOTYPE ⟨name⟩

⟨user profile⟩
⟨user knowledge⟩

The slot structure is fixed for all stereotypes, and some slots can be multiple-valued. Furthermore, each slot may belong to one of the following three types:

- Identification slots, whose values, once acquired, must satisfy a given predicate in order for a user to be classified as a member of the class described by the stereotype. These slots are suitable to include information that plays a critical role in the decision process devoted to assigning a user to the class related to a stereotype. Consequently all the identification slots of a stereotype are considered for this selection.
- Default slots, which contain information relating to the typical features and traits of all users of the class described by the stereotype. The values of these slots are fixed and defined a priori.
- Generic slots, whose content is empty, since they are not restricted to any specified value. These slots will possibly include information that does not characterize the class of users related to the stereotype, and therefore any value may be acceptable.

Moreover, to the slots of the stereotype the following types of procedural knowledge may be attached:

- Acquisition methods, which are devoted to acquiring the relevant user information through the techniques for acquisition illustrated in the previous section.
- Validation methods, which are devoted to checking the acquired information against semantic constraints aimed at ensuring the local correctness and the global consistency of the user model.
- Identification methods, which are always associated with identification slots and are responsible for evaluating the relevant predicates that guarantee that some necessary condition is met by the current user.

Finally, to each stereotype as a whole, an activation method is attached, which takes the form of a predicate and, if satisfied by a user, ensures that there is suggestive evidence that he belongs to the class described by the stereotype. This predicate may refer to identification slots, but it does not coincide, in general, with the conjunction of all the predicates contained in the identification methods. In fact, the activation method only suggests a possible, reasonable membership of a user to a class, but it does not assert it definitely.

A partial order relationship (called is_a) is defined between stereotypes, which reflects the relation of inclusion between the classes of users they identify. We say that stereotype A is_a B if the class of users described by A is included in that described by B. Of course, if A is_a B, A represents a further specialization of B, that is, it identifies a more specific class of users. Therefore, A will inherit from B all default and identification slots, along with all acquisition, validation, and identification methods; in addition, it will have its own specific slots and methods. Clearly A will have, in general, fewer generic slots than B. This relation is very useful both for structuring the stereotype base and for supporting the elicitation of new stereotypes through specialization of already existing ones. If A is_a B, the slots inherited from B are not duplicated in the description of A, but they are globally referred by mentioning B in a special slot of A called IS_A.

Since we have assumed that the user model has a fixed structure (see Section 4.1), we can also assume that all stereotypes are specializations of a universal stereotype, called generic user; it has only generic slots, and its activation method is always satisfied, but it contains the definition of all relevant acquisition and validation methods, which are thus inherited by all other stereotypes. A fragment of the generic user stereotype is shown below:

```
STEREOTYPE generic user
        ACT_M: (true)
        USER PROFILE
            EDUCATION
                ACQ_M: (Activate Dialogue-edu-1)
            PROFESSIONAL BACKGROUND
                ACQ_M: (Activate Dialogue-prof-1)

                .....
```

ACT_M and ACQ_M are labels used to introduce activation and acquisition methods, respectively. All slots are generic, as this stereotype does not carry any value. The activation method is always satisfied, and no identification slot is present in this stereotype, as it applies to any user. The IS_A slot is omitted, since it is empty. The two acquisition methods presented above constitute an example of the acquisition procedure based on dialogue inspection. For such a purpose two specific dialogues (Dialogue-edu-1 and Dialogue-prof-1) are activated in order to acquire the needed information through explicit answers given by the user.

An example of a more specific stereotype, which is of course a specialization of the generic user, is shown below (not all generic slots are shown):

```
STEREOTYPE expert user
        IS_A: generic user
        ACT_M: (INFORMATION RETRIEVAL BACKGROUND.EXPERIENCE.TYPE = user
                and
                INFORMATION RETRIEVAL BACKGROUND.EXPERIENCE.EXTENT >= 3 years)
        USER PROFILE
            EDUCATION
                FIELD: -
                    DEGREE: -
                    DATE: -
            INFORMATION RETRIEVAL BACKGROUND
                EDUCATION
                    ACQ_M: (IF EDUCATION.FIELD = information science
                              THEN set "value" to >= medium
                              ..... )
                TRAINING
                    ID_M: ( "value" >= medium)
                EXPERIENCE
                    TYPE
                        ID_M: ( "value" = user)
                    MODE
                        ID_M: ( "value" = non-assisted)
                    EXTENT
                        ID_M: ( "value" >= 3 years)
            PERSONAL TRAITS
                COMMUNICATION
                    LEVEL: -
                    QUALITY:
                        VAL_M: (IF
                              INFORMATION RETRIEVAL BACKGROUND.EDUCATION >= medium
                              THEN "value" >= precise )
                ATTITUDE: cooperative

        USER KNOWLEDGE
            INFORMATION RETRIEVAL ACTIVITY
              SEARCH SESSION STRUCTURE: good

          .....
```

In the above example, ID_M and VAL_M indicate identification and validation methods, respectively. The symbol "value" denotes the value of the corresponding slot. Generic slots are identified by the "-" symbol as value. Identification slots can be recognized by the presence of the identification methods attached to them. Moreover, in the above example the slot INFORMATION RETRIEVAL BACKGROUND.EDUCATION contains an acquisition procedure that is implemented by means of an inference rule from known facts. Finally, the validation method shown is an example of a consistency check between the values of two slots.

5.3. Basic operation of the model builder

Operation of the model builder starts by identification of the user accessing IR-NLI II. After this preliminary phase, the model builder looks in the user models base for the model: If the user is new, no model exists and model building has to start from scratch. Later, the operation of the model builder proceeds through the five phases described below:

1. *Preliminary interview.* This phase is devoted to acquiring basic information about the user, through a bounded scope system-driven dialogue. For example, preliminary information about EDUCATION and PROFESSIONAL BACKGROUND is

collected in this phase. The model building techniques used generally conform to the answer analysis acquisition procedure combined with the single mode of information elicitation.

2. *Stereotype activation.* The preliminary user information gathered in the previous phase is used here to test the activation methods of the stereotypes available in the stereotypes base. All stereotypes whose activation method is satisfied become active stereotypes and are candidates for further consideration as possible starting points for constructing the individual model of the current user.
3. *Stereotype discrimination.* This phase considers the set of active stereotypes and aims at identifying the one that will be used as the kernel of the model construction. Each active stereotype is considered in turn: If all its identification methods evaluate to true, the stereotype is kept; otherwise, it is discarded (note that testing an identification method usually requires that the related acquisition method has already been executed in order to acquire the values of the identification slot to which the method refers). The set of active stereotypes, pruned through the above described procedure, now contains all the stereotypes that correctly apply to the current user (note that this set is never empty since the generic user stereotype applies to any user). The model builder chooses from this set the stereotype that is expected to best fit the features so far known of the current user and to represent the best kernel for starting the construction of the individual model of the current user. The criterion currently utilized for the choice selects the most specialized stereotype.
4. *Model refinement.* This phase is aimed at incrementally extending, tuning, and refining the individual model of the current user during a single search session. Model refinement starts working on the stereotype discriminated in the previous phase, and continually iterates over two main activities: (a) information acquisition, which is aimed at collecting or producing new information about the user through the appropriate acquisition methods, and (b) information validation, which is devoted to checking the correctness and consistency of the new information just acquired by means of the relevant validation methods. This model refinement phase ends when the current session is terminated.
5. *Closing operations.* At the end of the search session, the individual model of the current user is first completed with user identification name and model history slots and then stored in the user models base. Moreover, a summary of the current search session, collected by the history manager working in parallel with the model builder during the whole session, is stored in the search sessions history.

The operation of the model builder is only slightly different when the current user is already known to the system and therefore an individual model is available in the user model base. In such a case, phases 1, 2, and 3 above become meaningless and are substituted by two new ones directed at initializing the activity of the model builder before model refinement is started. Thus, the operation of the model builder encompasses in this case the following four phases:

1. *Model retrieval.* Once the current user has been identified, his model is retrieved from the user models base and made available for further processing.
2. *Historical information processing.* This phase is concerned with processing the summaries of the past search sessions carried out by the current user in order to derive or refine typical values of some slots of the model (e.g., PERSONAL TRAITS, USUAL SEARCH REQUIREMENTS, and INFORMATION RETRIEVAL ACTIVITY). This is done through a statistical processing aimed at identifying meaningful patterns denoting specific features of the user. The historical information necessary for this activity is extracted from the search sessions history, which contains records of all the sessions. The output of this phase is an updated user model that will be used and further refined throughout the current search session.

3'. *Model refinement.* See phase 4 above.
4'. *Closing operations.* See phase 5 above.

5.4. Improved concept of model builder: a preliminary discussion

The mode of operation of the model builder illustrated in the previous section is quite clear and effective, but it is still a little simplistic. In fact, it is based on the assumption that whenever the model builder has to make a critical decision, all the information necessary to decide in the most informed way is available. This occurs, for example, in the stereotype discrimination phase, as well as in the model refinement phase whenever the model builder has to select between alternative values for some slot (e.g., one provided directly by the user and another inferred by the system). Moreover, all decisions taken by the model builder are definite; no backtracking is possible. Unfortunately, information necessary to make a decision is often lacking at decision moment (it will be possibly available only later), and not all actions taken by the model builder on the user model are reversible or monotone [28].

A way to overcome these problems is to delay decision making as much as possible, until they can be really resolved in the most informed and reliable way. This implies that the operation of the model builder is revised in two major points:

1. Each time an alternative arises and a selection should be made, the model builder adopts a generate and test approach: A branching point is instantiated and all alternatives are carried out and tested according to an appropriate search strategy [28].
2. To each slot of the user model a confidence factor is attached, which represents the degree of reliability and accuracy of the corresponding value. Confidence factors are updated dynamically during system operation and serve to compute a global confidence factor for each alternative user model. At each instant, the model with the highest confidence factor is used by the system (namely the reasoning module and the understanding and dialogue module) as the current user model.

Of course, in order to implement such a more refined version of the model builder, the breakdown of its operation into phases presented in the previous section has to be revised and corrected accordingly. This new approach is clearly appealing and should guarantee, at least in principle, a better modeling of each individual user. However, the research carried out so far has shown that a number of the advantages of this improved version of the model builder can also be obtained through the previous simpler version, provided that the modeling knowledge base is skillfully structured and contains rich and detailed knowledge on the model building process. Therefore, there is, in a sense, a trade-off between system architecture and quality of the knowledge base: Most of what is lost with simpler and less general architecture can be recovered with a richer and better structured knowledge base. Taking into account the dramatic negative effect on system performance that results from structuring the operation of the model builder according to a general search paradigm like that outlined in the above discussion, we believe that, as far as we have experimented up to now, the approach proposed in the previous sections should be preferred.

6. CONCLUSION

In the paper we have presented the IR-NLI II system, an expert interface for accessing online information retrieval systems. More specifically, we have focused on the new user modeling capabilities present in IR-NLI II. In particular, the general architecture of the system, encompassing a module specifically devoted to building and managing individual user models, has been described in detail. The construction and refinement of the models rely heavily on a base of stereotypes, which represent typical classes of users. A description of the knowledge contained in the stereotypes and of the way of utilizing them for building individual models has been given.

The IR-NLI II system evolves from the experience acquired by the authors through the development and experimentation of the IR-NLI system. From an experimental point

of view, IR-NLI II is currently being developed on a SUN 2/170 using Franz LISP at the Laboratorio di Intelligenza Artificiale of the University of Udine.

An important problem that deserves further investigation is that of the efficient implementation of the algorithm for improving stereotype management with backtracking and approximate reasoning techniques.

REFERENCES

1. Marcus, R. S. A translating computer interface for end-user operation of heterogeneous retrieval systems: I. design; II evaluations. Journal of the American Society for Information Science, 32: 287–317; 1981.
2. Marcus, R. S. An automated expert assistant for the information community: an alliance for progress. Proceedings of the forty-fourth ASIS annual meeting, Washington, DC, October 25–30. Published by Knowledge Industry Publications, Inc., White Plains, NY, 1981: 270–273.
3. Doszkos, T. E.; Rapp, B. A. Searching Medline in English, a prototype user interface with natural language query, ranked output, and relevance feedback. Proceedings of the forty-second ASIS annual meeting, Minneapolis, MN, October 14–18. Published by Knowledge Industry Publications, Inc., White Plains, NY; 1979: 131–137.
4. Pollitt, A. S. An expert system as an online search intermediary. Proceedings of the fifth international online meeting, London, December 8–10. Published by Learned Information, Oxford; 1981: 25–32.
5. Pollitt, A. S. A search statement generator for cancer therapy related information retrieval. Proceedings of the sixth international online information meeting, December 7–9. Published by Learned Information, Oxford; London; 1982: 405–413.
6. Pollitt, A. S. Reducing complexity by rejecting the consultation model as a basis for the design of expert systems. Expert Systems, 3: 234–238; 1986.
7. Croft, W. B.; Thompson, R. H. An expert assistant for document retrieval. Report COINS-85-05. University of Massachusetts, Department of Computer and Information Science, Amherst, 1985.
8. Defude, B. Knowledge based systems versus thesaurus: an architecture problem about expert systems design. In: Van Rijsbergen, C. J., ed. Research and developments in information retrieval. Cambridge, England: Cambridge University Press; 1984: 267–280.
9. Defude, B. Different levels of expertise for an expert system in information retrieval. Proceedings of the eighth ACM SIGIR conference on research and development in information retrieval; 1985; Montreal; June 1985.
10. Guida, G.; Tasso, C. An expert intermediary system for interactive document retrieval. Automatica, 19: 759–766; 1983.
11. Brajnik, G., Guida, G.; Tasso, C. Design and experimentation of IR-NLI: an intelligent user interface to bibliographic data bases. Proceedings of the first international conference on expert database systems; 1986; Charleston, SC, April 1–4.
12. Brajnik, G.; Guida, G.; Tasso, C. An expert interface for effective man-machine interaction. In: Bolc, L.; Jarke, M., ed. Cooperative interfaces to information systems. Berlin: Springer-Verlag; 1986: 259–308.
13. Bates, M. J. Information search tactics. Journal of the American Society for Information Sciences, 30: 205–214; 1979.
14. Lancaster, F. W. Information retrieval systems. New York: Wiley; 1979.
15. Meadow, C. T.; Cochrane, P. A. Basics of online searching. New York: Wiley; 1981.
16. Salton, G.; McGill, M. J. Introduction to modern information retrieval. New York: McGraw-Hill; 1983.
17. O'Shea, T.; Self, J. Learning and teaching with computers: artificial intelligence in education. Englewood Cliffs, NJ: Prentice-Hall; 1983.
18. Sleeman, D.; Appelt, D.; Konolige, K.; Rich, E.; Sridharan, N. S.; Swartout, B. User modelling panel. Proceedings of the ninth international joint conference on artificial intelligence. Los Angeles, CA: Morgan Kaufmann; 1985, August 18–23, 1298–1302.
19. Carbonell, J. G. The role of user modelling in natural language interface design. Report CMU-CS-83-115. Carnegie-Mellon University, Department of Computer Science, Pittsburgh, 1983.
20. Rich, E. Users are individuals: individualizing user models. International Journal of Man-Machine Studies, 18: 199–214; 1983.
21. Card, S. K.; Moran, T. P.; Newell, A. The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum; 1983.
22. Robertson, G.; Newell, A.; Ramakrishna, K. The ZOG approach to man-machine communication. International Journal of Man-Machine Studies, 14: 461–488; 1981.
23. Nisbett, R. E.; Wilson, T. D. Telling more than we can know: verbal reports on mental processes. Psychological Review, 84: 231–259; 1977.
24. Oddy, R. N. Information retrieval through man-machine dialogue. Journal of Documentation, 33: 1–14; 1977.
25. Belkin, N. J.; Oddy, R. N., Brooks, H. M. ASK for information retrieval: I. background and theory. Journal of Documentation, 38: 61–71; 1982.
26. Belkin, N.J.; Oddy, R. N.; Brooks, H. M. ASK for information retrieval. II. results of a design study. Journal of Documentation, 38: 145–164, 1982.
27. Rich, E. Building and exploiting user models. Report CMU-CS-79-119. Carnegie-Mellon University, Department of Computer Science, Pittsburgh; 1979.
28. Nilsson, N. Principles of artificial intelligence. Palo Alto, CA: Tioga; 1980.