# Strategic Help in User Interfaces for Information Retrieval

**Giorgio Brajnik, Stefano Mizzaro, Carlo Tasso**
*Department of Mathematics and Computer Science, University of Udine, Via delle Scienze, 206, Loc. Rizzi, I33100 Udine, Italy. E-mail: {giorgio|mizzaro|tasso}@dimi.uniud.it*

**Fabio Venuti**
*Department of Information Science, City University, Northampton Square, London EC1V OHB, United Kingdom. E-mail: fabio@soi.city.ac.uk*

**Although no unified definition of the concept of search strategy in Information Retrieval (IR) exists so far, its importance is manifest: nonexpert users, directly interacting with an IR system, apply a limited portfolio of simple actions; they do not know how to react in critical situations; and they often do not even realize that their difficulties are due to strategic problems. A user interface to an IR system should therefore provide some strategic help, focusing user's attention on strategic issues and providing tools to generate better strategies. Because neither the user nor the system can autonomously solve the information problem, but they complement each other, we propose a collaborative coaching approach, in which the two partners cooperate: the user retains the control of the session and the system provides suggestions. The effectiveness of the approach is demonstrated by a conceptual analysis, a prototype knowledge-based system named FIRE, and its evaluation through informal laboratory experiments.**

## Introduction

Today many information retrieval (IR) systems are directly used by end users. Because of the absence of a human intermediary, users face a difficult task, because they have to know how to interact directly with the system and cope with various kinds of problems: learning interface commands, using boolean logic, choosing the more effective terms, and designing and applying effective search strategies.

Many researchers are focusing on how to relieve the cognitive load on users by visually presenting the information (terms, documents, results, and so on) in more effective ways: see Baeza Yates and Ribeiro Neto (1999, chap. 10), Marchionini (1995, chap. 6), Korfhage (1997, chap. 7) for some reviews. There is considerably less effort spent at the deeper cognitive and reasoning level of how to help users to design and apply effective search strategies. This article focuses on this issue.

Our general aim is to investigate the role played by strategic reasoning during IR sessions, and to learn design principles for user interfaces for IR. The literature presents evidence of the importance of a strategic support given by the interface to the user (Brajnik, Mizzaro, & Tasso, 1996; Chen & Dhar, 1991; Fenichel, 1981; Hsieh-Yee, 1993; Larson, 1991; Manglano, Beaulieu, & Robertson, 1998; Marchionini, 1989; Sullivan, Borgman, & Wippern, 1990; Sutcliffe, Ennis, & Watkinson, 2000; Twidale, Nichols, Smith, & Trevor, 1995; Vollaro & Hawkins, 1986): nonexpert users, directly interacting with an IR system, apply only simple and noneffective search strategies; they do not know how to react in critical situations; even when users are not stuck in a critical situation, they often find themselves in what we call an enhanceable situation (that holds when the actions done by the users are not the most effective ones); and users often do not even realize that their difficulties are due to strategic problems. A user interface to an IR system should therefore provide some strategic help, aimed at both (1) focusing user attention on strategic issues of the current search process, and (2) providing the user with tools and concepts that will enable her to generate better strategies.

However, deciding how strategic help has to be provided to end users by a user interface is not a trivial design decision at all. Because neither the user nor the system has enough knowledge to autonomously solve the information problem, but they complement each other, we propose a collaborative coaching approach, in which the two partners cooperate: the user retains the control of the session and the system provides suggestions to her.

Received January 9, 2001; revised September 4, 2001; accepted September 4, 2001.

We present a conceptual analysis, the knowledge-based FIRE prototype, and the results of its evaluation through informal laboratory experiments. The results give evidence that the collaborative coaching approach and the prototype are effective, and that it is feasible to include a module giving strategic help in a user interface to an IR system.

The outline of this article is the following. In the next section various views of the concept of search strategy in information retrieval are recalled, as well as the evidence supporting the thesis that some form of strategic help should be provided to the user by a user interface to an IR system. The Strategic Help section justifies the choice of collaborative coaching as the modality to provide strategic help. The Conceptual Model section proposes a conceptual model of a system providing strategic support. The Strategic Help in the Fire System presents the prototype implementing such a conceptual model. The Evaluation section describes the evaluation of the prototype. The Conclusion section summarizes the article and sketches some future developments.

## Search Strategies in Information Retrieval

Evidence of the importance of the concept of *search strategy* in IR is shown by many researchers, even if no unified and standard definition exists so far.

In the classical context of IR performed by professional intermediaries, a search strategy is often considered as the sequence of search statements that identify, restrict, limit a set of retrieved documents (Lancaster, 1979; Meadow & Cochrane, 1981), whereas a *search approach* is the specification of a set of well-structured steps exploited for building up a search strategy.

On the basis of a protocol analysis of actual behaviors shown by users interacting with either librarians or IR systems, Chen and Dhar explain users actions as a *problem solving* context (Chen & Dhar, 1991), where each action is performed to move in the problem space, from one solution state to the next one. Moreover, they define a search strategy as an approach adopted to traverse the problem space, and identify five different search strategies (*known-item instantiation, search-option heuristics, thesaurus-browsing, screen-browsing,* and *trial and error*).

Ellis and his colleagues (Ellis, 1989a, 1989b; Ellis, Cox, & Hall, 1993; Ellis & Haugan, 1997), follow a similar approach in discussing information-seeking patterns. On the basis of systematic observations, they classify user's behavior into six categories (*starting, chaining, browsing, differentiating, monitoring,* and *extracting*).

Belkin and his colleagues (Belkin, Cool, Stein, & Thiel, 1995; Belkin, Marchetti, & Cool, 1993) define an *information-seeking strategy* as a behavior pattern of the information seeker. They describe the pattern by specifying binary values for four independent dimensions, in this way enumerating 16 different categories of behavior patterns, each one constituting a specific information-seeking strategy that can be used to exhaustively cover all possible behaviors.

A rather different approach is followed by Bates (1990), who defines four concepts useful for discussing searchers behaviors: the *move,* i.e., an identifiable thought or action that is a part of information searching; the *tactic,* i.e., one or a handful of moves made to further a search; the *stratagem,* i.e., multiple tactics designed to exploit the file structure of a search domain; and the *strategy,* a plan containing tactics and stratagems for the whole information search. Strategies, tactics, and stratagems are opportunistically selected, applied, and monitored. Bates (1989) also proposes the *berry-picking* model of information seeking, stating that a single final query retrieving all and only the relevant documents is unrealistic and ineffective; rather, the documents are incrementally collected along all the search process. This opportunistic approach has been confirmed by recent studies (Belkin, 1996; Xie, 2000).

Several different information-seeking models and definitions of search strategy have been proposed, and only recently efforts have been made in an attempt to unify them (Wilson, 1999). In this article, a *search strategy* is defined as the possibly partial and implicit plan that a searcher adopts to solve an information problem. Therefore, following Bates (1989), a strategy is usually developed incrementally and in a data-driven fashion, depending on the specific situations the searcher faces.

Turning our attention to the issue of strategic help, several studies have shown that users often get in trouble because of their poor ability to implement effective search strategies.

Novices seem to apply only simple strategies (Fenichel, 1981; Hsieh-Yee, 1993; Larson, 1991; Sullivan et al., 1990; Sutcliffe et al., 2000): they build queries using only terms they have in mind, without consulting available thesauri and not taking advantage of the interaction with the database; they perform only little query modifications and refinements. Moreover (Chen & Dhar, 1991), they overwhelmingly apply *screen-browsing* and *trial-and-error* strategies (the two least effective strategies), whereas the majority of the reference librarians and the sophisticated searchers adopt the *known-item instantiation,* the *search-option heuristics,* and the *thesaurus-browsing* strategies. But even experienced users may find it difficult to apply effective strategies and, although they construct more complex queries and make use of thesauri, their performance may be as poor as that of novices (Sutcliffe et al., 2000).

Some users react in ineffective ways to critical situations: for instance, with a boolean system where no documents have been retrieved, users may add *and*-ed terms, an operation that can only lead to a lower number of retrieved documents (Marchionini, 1989). Even if the search progresses, some users do not exploit in full potential the IR system. For instance, after retrieving documents, users do not look carefully at them, missing in such a way the opportunity to improve the search process. Even when users read the retrieved documents, they are not always able to exploit the information they gathered: for instance, they judge a document as relevant without realizing that they could extract some terms to better reformulate their query (Marchionini, 1989; Sullivan et al., 1990; Vollaro & Hawkins, 1986). Many users adopt a single search strategy for the whole session (Manglano et al., 1998), with negative consequences when this is not well suited for the particular situation at hand. Even when a valid but not familiar strategy is adopted, minor errors (e.g., typing errors) can cause its abandonment (Twidale et al., 1995).

Many of the finding reported in the literature were also confirmed by an experiment we carried out on FIRE, a knowledge-based user interface to a boolean IR system (Brajnik et al., 1996) (in the following, we will refer to this experiment as the FIRE 1995 experiment). FIRE allows the user to submit a query graphically organized in concepts (called *facets*). Each facet contains some *terms* (single words or sequences of words). Facets are *and*-ed together, terms in one facet are *or*-ed together. If the query does not lead to satisfactory results, the user can start a semiautomatic reformulation process: FIRE suggests terms to modify the query, exploiting morphological knowledge provided by a stemming algorithm and terminological knowledge extracted from a thesaurus. Experimentation confirmed that users lack an overall strategic view of the search. Less successful users are bound to a wrong and unchanging conceptualization of the information need and use a limited portfolio of actions: they keep adding, removing, and modifying terms that do not derive from the information resources—database or searching referral aids—but are found by thinking on the query; they never do field restrictions to search on authors, journals, classification codes, or controlled terms; when they get stuck in some difficult situation (for instance, when they keep getting no documents), they apply previously tried actions even though these were clearly ineffective; finally, they do not realize that the cause of their difficulties is a strategic one, and insist in asking for terminological help.

Modern IR systems often provide some terminological help, by means of query expansion (Efthimiadis, 1996; Greenberg, 2001a, 2001b; Mandala, Tokunaga, & Tanaka, 2000) and/or relevance feedback (Belkin et al., 2001). However, the literature shows, and the FIRE experiment confirms, that, besides terminological knowledge and knowledge about basic query manipulation, a successful support to end users has to be given also at a strategic level. Such a *strategic help* has to be provided autonomously by the system (i.e., without user's request), with the aims of (1) making users aware of the strategic aspects of their searches, and (2) enlarging the tool box of actions that the users might want to try, providing them with tools and concepts that will enable them to generate better strategies.

## Strategic Help Based on Collaborative Coaching

Deciding how strategic help has to be provided to end users by a user interface to an IR system is not trivial, as it involves figuring out how to represent and monitor the solution process and determining the best way to approach and solve a search problem.

Providing on-line help in the form of manual-like screens given on demand is a possible solution, but it usually suffers from focusing at a too low abstraction level and, moreover, is basically fixed and not contextualized. What users need is instead help keyed on the specific current problem-solving process, and strategic help that is to be provided on the basis of autonomous system decisions.

A different approach is to implement user help via tips selected more or less randomly from a collection of homogeneous ones: the system plays in this case a more active role by providing unprompted suggestions that are, however, only little contextualized to relevant features of the problem state.

*Coaching* systems (Wenger, 1987) have many features that are appropriate for providing this kind of help. They work in the background, monitoring users' activities and tracking the level of performance that users are expected to reach. When this does not happen, a coaching system decides to intervene by interrupting the user, providing help and, on demand, justifying it. The main positive consequences of a well-designed coaching system are: help is contextual with respect to the problem-solving state and the evolving information problem; it is not obtrusive; control of the search session remains in the hands of the user (a basic requirement for a user interface to an IR system; Bates, 1990; Brajnik et al., 1996); and the system fosters user's learning of more effective behaviors for the future.

Unfortunately, in the context of IR there are two fundamental obstacles to the development of a coaching system. First, in practice, there is not an optimal search strategy against which the coaching system could compare user behavior. The large spectrum of different routes that the problem solving could follow depends on the complexity of the information resource being searched and on the cognitive state of the user, making impossible the generation and evaluation of specific strategies, and the identification of optimal ones. Second, the coaching system has to work upon very incomplete data. The correct solution of the information problem is out of its reach, because only the user is capable of correctly judging the usefulness of retrieved documents. In many cases the user—the only source of the information problem—is not even in a position to provide all the relevant data describing the input of the problem, due to her *anomalous state of knowledge* (Belkin, 1980).

To overcome these hurdles, an effective way to provide strategic help is to frame a coaching system within a collaborative work model (Cummings & Self, 1989), where the interface can be viewed as a problem-solving partner of the user. Neither of the partners has enough knowledge to solve autonomously the information problem, but they complement each other: the user provides the problem (input and output specification) and controls the problem-solving process; the interface accesses the database, provides partial results, suggests alternative solution steps, and presents the current problem solution state. According to this view the search strategy is not explicitly represented. It is implicitly and opportunistically revised and extended by the user who decides which action to take next. In turn, such decisions can be influenced by system suggestions showing: (1) actions potentially useful and novel for the actual situation, and (2) partial results of actions that can be carried out autonomously (and non obtrusively) by the system. We call such an approach *collaborative coaching.* Because system suggestions can momentarily distract user attention from an original goal, the resulting strategy will be coherent with Bates's berrypicking model.

A collaborative coaching system has thus a twofold goal: improving the currently adopted strategy, and raising user awareness on planning the session. To develop such a system, there are three fundamental issues that need to be dealt with: to know what to say, when, and how.

1. *What.* The content of the communication between user and system. To achieve the twofold system goal, the communication can be organized in the two levels defined in (Cummings & Self, 1989). The *task level* concerns suggestions aiming at improving the specific situation. Suggestions of this level can thus be both partial results that directly show which are the nearby solution states, that the user can select, and actions that can be applied to the current problem state to generate new states, possibly closer to the solution. The *discussion level* concerns more general suggestions aiming at fostering user's learning of reusable effective strategies. Suggestions of this level are usually both general principles and actions that can be applied in states with certain properties, that hold for the current state.

   More specifically, the information provided by the system should include, at both the above mentioned levels: (a) *Descriptions* of actions that in the current situation are plausible for achieving progress. The current situation can be characterized in terms of history of query modifications and sets of retrieved, displayed, or selected documents. Actions to be suggested are constrained by the database, the IR system, its user interface, and include, for instance, moves for reformulating the query, search tactics, stratagems, browsing search referral aids, zooming on a set of documents, providing relevance feedback, and classifying displayed documents. (b) *Applicability* of proposed actions. Each action should be described in terms of set of objects it applies to (terms, facets, sets of documents, and so on), how such objects are to be identified, set of operators that need to be applied, and how these are to be identified. (c) *Motivations* for the proposed actions, that is why the system "believes" they are appropriate to the current situation. Novelty of the action, its applicability, and its emphasis on a not-yet followed search route are simple criteria adequate for the selection of candidate actions to be suggested to the user. Motivation should also be described in terms of the kind of results that the action leads to, and how they are to be used to improve the current search problem state. (d) *Action results,* for those actions that can be safely carried out in the background by the system, unknown to the user. The system could then not only suggest a route to be followed, but also present the actual results that the action yields. For example, instead of simply suggesting to zoom on the documents selected so far, the system could actually do it in the background and present the results in addition. In this way the user is provided not only with a description of the action, but with immediate feedback on its effectiveness in a certain situation, leading to a faster learning curve of the IR skill and improved effectiveness.
2. *When.* The system needs to recognize when the situation requires (or is adequate for) providing some strategic help relevant to the search. Many problematic situations are characterized in terms of objective data: too many or no documents being retrieved, the same query submitted again and again, wrong usage of query operators (boolean ones, truncation, phrase searching, and field restrictions), ineffective usage of such operators (e.g., performing a field restriction with a query that already leads to a high precision search), nonusage of applicable actions (e.g., relevance feedback, zoom). Other situations are more complicate, because they also require consideration of the (implicit) user's current goal (e.g., the user has a precise plan to relax the query, and her first step happens to be in the opposite direction of doing a field restriction). In this case, the system could provide wrong or noisy suggestions to the user, and therefore, it has to be designed with special care.

   Moreover, the purpose of a suggestion may be to improve the search strategy for subsequent steps both in critical situations where the user got stuck (e.g., suggesting thesaurus browsing and a search on controlled terms to a user stuck because too many irrelevant documents have been retrieved) and in enhanceable situations where the user could get better results than what she is achieving (e.g., suggesting an author search to a user mainly involved in reformulating the query).
3. *How.* The system needs to identify the most appropriate way to convey its suggestions to the user in the most comprehensible way. An advanced interface that satisfies these requirements is likely to have both a detailed model of the user and knowledge about interaction modalities. These are interaction problems that involve advanced human–computer interaction issues (Dix, Finlay, Abowd, & Beale, 1998), such as user modeling (Kay, 1999), intrusion level control (Bailey, Konstan, & Carlis, 2000), and so on. Our efforts in the study of strategic help have been concentrated mainly on the previous "what" and "when," and we have not dealt with these problems yet.

On the basis of the above analysis, we have designed an improved version of the FIRE prototype, which is described in the following with particular emphasis on the modules that provide strategic help. The complexity of the problems involved in designing a system able to provide strategic help induced us to concentrate our attention on some aspects of such problems, leaving to future developments the analysis of the remaining ones.

## A Conceptual Model for Strategic Help

A *conceptual model* (Guida & Tasso, 1994) of the activity aimed at providing strategic help has been developed, and includes several kinds of knowledge used by the reasoning processes. It is described, at a conceptual level, in this section, together with some examples illustrating how the concepts can be implemented.

### *Preliminary Definitions*

For the sake of clarity, we introduce the following (informal) definitions. A *user action* is any command that the user gives to the system: adding (or removing) a term or facet to (from) the query, performing a search, reading or classifying a document, and so on. A *system action* is any action that the system performs: a search in the database, the visualization of some data to the user, and so on. A particular kind of system action is the *suggestion,* i.e., a system action aimed at giving strategic help for reaching the two-fold system goal (see earlier). A *situation* is a state reached by the search session, after a sequence of user and system actions, and is expressed by means of the query, the retrieved, read, and classified (as relevant or useful) documents, and the last user and system actions.

### *Kinds of Knowledge*

Coaching systems are characterized by the presence of three types of knowledge, about the domain, the user, and the system–user interaction modalities. In this work we take into account mainly the knowledge about the domain (the domain of IR). It can be divided into two classes, related respectively to:

1. *The document database:* knowledge about the database structure (areas covered by the databases, type of documents classification, and organization), terminological knowledge (terms of the databases and relations among them), and morphological knowledge (to find term's morphological root). In practice, the first two types of knowledge can be represented by means of classification schemata and thesauri, and the latter can be obtained by using specific algorithms (Porter, 1980).
2. *Strategic help:* knowledge about the situations that can occur during the searching process, about the suggestions, about the criteria for the selection of proper suggestions, about the criteria for the ranking of the selected suggestions, and about the execution of the suggestions that require some system actions.

The first class is well known and widely used in IR, while there is little work in literature about the second one. In the following we discuss both kinds of knowledge.

*Situations.* This kind of knowledge is necessary to understand in which situations a strategic support is desirable. Such situations are of two types:

1. *Critical situations,* in which the search does not progress either because the user keeps on retrieving documents that are not useful or because she repeatedly fails to find documents at all. A typical critical situation happens when a search retrieves few documents. If the user adds a new facet or deletes a term appearing in a facet and then performs a new search, the situation cannot improve: she will retrieve few or no documents again, and the new situation will be still critical. A repetition of such situations (referred to as *stalling situations* in the following) can lead to confusion, disappointment, and loss of trust in the system.
2. *Enhanceable situations,* in which the user could obtain better results by following a different route. The help given in these situations should also help to prevent the critical ones. A typical enhanceable situation is the insertion of a low posting count term. It is not a critical situation by itself, but it can lead to a critical one if the user adds new facets; on the contrary, inserting the term stem might improve the rest of the search and avoid subsequent problems.

*Suggestions.* We distinguish two kinds of suggestions:

1. *Hints,* i.e., suggestions that signal the existence of a particular situation and invite the user to undertake some activity, whose execution is on total user's charge (for instance, it is normally useful to notify the insertion of a low posting count term, or, even more, the insertion of a term not contained in any document of the database).
2. *Advices,* i.e., suggestions related to the application of some activity whose execution is carried on in a collaborative way by the user and the system (for instance, to browse the thesaurus in a guided way).

According to the classification proposed by Bates (Bates, 1990, p. 577), hints are recommendations given by the system whenever it identifies a need during the monitoring of the search process (level 3-b in the Bates proposal), whereas advices are actions performed in a (partially) automatic way by the system (level 4-a).

The system activities executed to propose an advice to the user can be selected by instantiating and implementing some of the tactics and stratagems proposed in (Bates, 1989, 1990). The *tactics* provide terms to be used for query reformulation (modifying the query by adding terms or substituting existing ones). Some relevant tactics are:

1. *Fix,* to truncate a term using morphological knowledge.
2. *Parallel,* to find terms related to the query terms. It can be executed by means of: *morphological analysis,* which exploits morphological knowledge to find thesaurus terms that are similar to a particular query term; *thesaurus browsing,* which can be carried on either manually by the user or automatically by the system; and *classification schema browsing,* which is similar to thesaurus browsing and can be carried on, either automatically by the system or manually by the user, on the classification schema.
3. *Relevance feedback,* to find the most frequent terms

appearing in the documents retrieved or in those classified as useful.

*Stratagems* allow the user to perform activities different from query reformulation. They are normally and naturally applied by users during manual searches in a library, and allow them to obtain other documents related to the area of interest. Some relevant stratagems are:

1. *Area scanning,* to explore the area of interest or, in a physical library, to browse through the shelves near an interesting area. This stratagem is effective because it exploits the positions of the documents on the shelves, which reflects the subject classification. Our implementation is an attempt to use the same ideas in a digital database: the system tries to find the classification codes (i.e., to find the area) that best represent the topics contained in the query. This stratagem can be performed either by finding the codes morphologically similar to the query terms and then searching for semantically related codes in the classification schema, or by finding the most frequent codes appearing in the documents retrieved or classified as useful.
2. *Journal run,* to find papers, in the same journal, concerning the same topic. It is a sort of area scanning restricted to a particular journal, because the issues of a journal are stored (in a library) in the same place.
3. *Author search,* to find papers written by the same authors of the documents classified as useful.

At a more abstract level, these tactics and stratagems can also be seen as an implementation of some of the activities proposed by Ellis (Ellis, 1989a, 1989b): *extracting* can be carried out by means of relevance feedback, journal run, and area scanning from a set of documents; *differentiating* can be carried out by means of area scanning and journal run; and *chaining* can be carried out by means of an author search, though in a very limited way (proper "backward chaining" is done through the references at the end of the paper, and proper "forward chaining" is done through some citation index to find the citations to the paper).

*Criteria for selecting the suggestions.* This kind of knowledge is used to decide when it is sensible to provide some particular suggestion.

At the beginning of the search, the query normally needs some processing; thus, tactics such as parallel or respell (trying morphological variants of a term) might be suggested. When the user adds a low posting count term, a sensible advice is the fix tactic.

The analysis of the search results is another crucial moment. If the search has retrieved no documents, a reformulation of the query is sensible, but sometimes (e.g., if the query topics are not well covered by the thesaurus) the system could rather suggest an area scanning, or a journal run. If the search has retrieved some documents, they can be exploited to refine the query using the relevance feedback tactic, as well as to undertake the activity of differentiating (by area scanning and journal run stratagems) within the set of retrieved documents.

Another important milestone during a search is the judgment of documents: a relevant document can be used to retrieve other similar documents. This can be done applying the activities of chaining, perhaps by an author search, or the activity of extracting, by relevance feedback, journal run, and area scanning.

*Criteria for ranking the suggestions.* In any single situation there are, in general, different possible suggestions. These can be ranked in order of importance before being shown to the user, according to simple criteria like:

1. The suggestions less demanding for the user are preferred. For instance, the advice involving a query reformulation by an automatic thesaurus browsing is preferred to the hint of manually browsing the thesaurus.
2. After having classified useful documents, the suggestions involving chaining, extracting, and differentiating are preferred to query reformulation, for two reasons: (1) the user on his own is less incline to execute these activities than to reformulate, and, (2) because some useful documents have been retrieved, probably the query is a good one, further reformulation might be not effective, and it seems sensible to exploit the information contained in the documents classified as useful.

*Execution of the advices.* This knowledge is procedural and concerns the execution modalities of tactics and stratagems. Their execution almost always implies the use of other kinds of knowledge regarding the document database (which is more likely to be possessed by the system), the user, and her information problem (which is more likely to be possessed by the user herself); therefore, the execution is done in a collaborative way.

The tactics are executed in two steps, the first one in charge of the system, the second one of the user: (1) search of a set of possible candidates to be inserted into the query; and (2) insertion or substitution of one or more terms into the proper facets.

The execution of stratagems is carried on in four steps; the second one is on user's charge (and the evaluation of the retrieved documents as well), the other three are carried on by the system: (1) search of a set of potentially interesting authors, classification codes, or journals; (2) choice of particular authors, codes, or journal taken from the previous set; (3) formulation of a query that contains the user's query terms in one facet (or-ed together) and the selected classification codes, authors, or journal name in another facet; and (4) search the database.

### *The Reasoning Process*

The problem solving activity aimed at providing strategic help is represented by means of a cycle of single

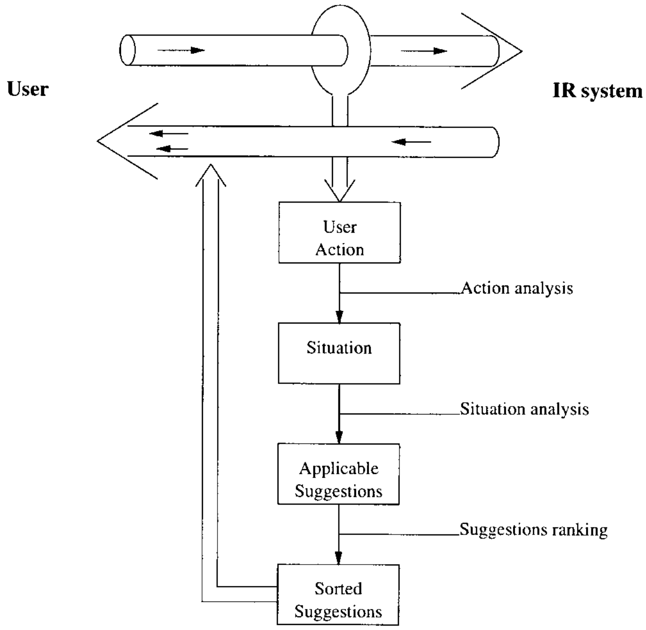


FIG. 1. The reasoning process.

reasoning steps, that begins every time a user action is performed (Fig. 1).

The first reasoning step is called *Action analysis,* and concerns the analysis of the user's activity to build a model of the state of the search. It takes in input data about the current situation (e.g., actual query, documents retrieved, documents classified, etc.) and data about the user action in this situation (e.g., term insertion, document classification, etc.). Using the knowledge about situations, the model of the new situation is built. This model is then input to the second step, called *Situation analysis,* which uses the knowledge about suggestions selections to find hints and advices that apply to the situation and are hopefully useful. The third step (*Suggestions ranking*) ranks the list of the selected suggestions (and those possibly already selected during previous executions of the cycle).

The first suggestion is then presented to the user, who can accept and carry on an advice, accept a hint, examine the following suggestions, or ignore them, performing some action autonomously. In the latter three cases, the user actions simply cause the reasoning cycle to start again. In the first case, the execution of the selected advice is started by the system using the knowledge about suggestions and the execution of advices. The results of this phase are terms (in the case of tactics), classification codes, authors names, or journals names (in the case of stratagems) that are proposed to the user. The terms selected by the user are inserted into the query, whereas the codes, authors, or journals selected are used by the system to perform a search and present the set of retrieved documents. The user can then evaluate the obtained results and the cycle starts again.

## Strategic Help in the FIRE System

On the basis of the conceptual model, the prototype described in this section has been implemented.

### *Overall Architecture*

The overall architecture of FIRE is shown in Figure 2. The system is composed of four main modules, discussed in more detail in the following subsections:

1. *User Interface* (*UI*), which allows the users to access the system.
2. *Information Retrieval Server* (*IRServer*), which stores the documents and allows searches by means of a traditional boolean system.
3. *Terminological Aid Module* (*TAM*), which provides UI, SAM, and the knowledge engineers with tools to access and modify the terminological knowledge bases.
4. *Strategic Aid Module* (*SAM*), which provides the user with strategic help. The inner working of this rule-based module recalls the conceptual model presented in the last section. SAM monitors user's activity and, using a first set of rules, adds to the working memory some facts representing user actions; on the basis of these facts and using another set of rules, other facts are added, representing the candidate suggestions for the user; using a third set of rules, these suggestions are ranked, and the first one is presented to the user. SAM can also autonomously search the database and present partial results to the user.

*The user interface.* This module allows the user to: insert and modify a boolean query; perform a search; classify a document by inserting it into a *useful, topical,* or *trash* folder; view the titles list of the documents retrieved by the user's query, the documents retrieved by the system, the documents classified as useful or topical by the user; view a particular document selected from one of the above lists; view and accept SAM's suggestions; view and select one of

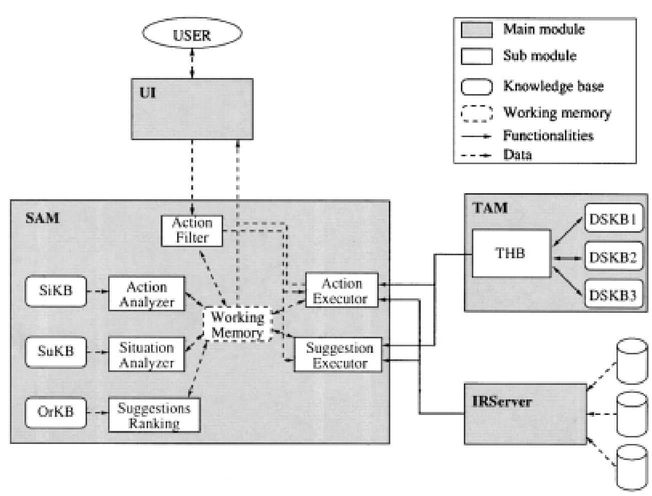


FIG. 2. Overall architecture.

**Rule:** Bottleneck

**Comment:** In a situation where the search has retrieved few documents, the user inserts a term into a facet that is not the bottleneck

**Hypotheses:**

**IF** the working memory contains the fact "few documents retrieved by the last search"

**AND** the working memory contains the fact "term added in facet $F_i$"

**AND** the working memory contains the fact "$F_j$ is the bottleneck"

**AND** $i \neq j$

**Actions:** Add to the working memory the following two facts:

- "term insertion in a facet which is not the bottleneck"
- "action aiming at enlarging the set of documents retrieved" (the action is not logically incorrect, even if it is not the best one)

**Activation level:** 2

FIG. 3. A rule in SiKB.

the available search activities (reformulation, author search, etc.).

We define as *useful* a document that the user wants to retain at the end of the search, and *topical* a document that deals with the topics of interest, although the user does not want to retain it for various reasons (e.g., it is already known to the user). A topical document can anyway be exploited during the search to extract some information (e.g., to extract terms).

*The information retrieval server.* This module provides the functions to access the databases of documents, abstracting from the specific IR system used. FIRE can access four different databases, but in this article we will actually refer to one of them only, called INSPEC20K, consisting of 20,000 documents taken from the INSPEC database and dealing with expert systems and artificial intelligence.

*The terminological aid module.* This module consists of the following submodules:

1. The terminological knowledge bases: three thesauri, concerning the INSPEC20K database, called *DSKBs* (*Domain Specific Knowledge Bases*). DSKB1 represents the classification codes and the hierarchical relations among them (narrower or broader codes), as stated in the INSPEC classification schema (See http://www.iee.org.uk/publish/inspec/classif.html). The other two DSKBs represent the terms in the database domain and some semantical relations among them: DSKB2 has been manually built by the knowledge engineers, and contains the classical Narrower Term, Broader Term, Related Term, Used For Term, and Use Term relations; DSKB3 has been built automatically, using a statistical approach, and it contains only cooccurrence relation.
2. The *THB* (*Terminological Help Builder*): procedures and functions that allow the other modules in the system and the knowledge engineers to access the DSKBs. In particular, this module allows to access different DSKBs at the same time; it contains stemming algorithms and morphological knowledge for finding the thesauri terms similar to a given starting term; and it contains also a spreading activation algorithm that finds, starting from one or more terms, the thesauri terms that are in relation with them.

*The strategic aid module.* SAM is devoted to the monitoring of the user's activity, to the selection of suitable suggestions, and, after the user's choice, to the execution of the involved tactics and stratagems. SAM is composed of three knowledge bases containing 94 production rules, six submodules, and a working memory.

Each rule in the knowledge bases consists of five parts: a *name;* a *comment,* used by the knowledge engineers to clarify the meaning of the rule; some *hypotheses,* that must be satisfied to activate the rule; some *actions,* executed if the hypotheses are satisfied; and an *activation level,* that is used for inhibiting, on the basis of user's degree of expertise, the rules leading to suggestions that would be useful for less expert users only.

The submodules are:

1. *Action Filter,* which filters the user actions to be monitored. Some of these actions are simply passed on to the Action Executor and executed; other ones also cause some facts to be added to the working memory, for representing user's activity.
2. *Action Executor,* which executes the user actions, with the exception of the suggestions proposed by the system and accepted by the user, which are executed by the Suggestion Executor.
3. *Action Analyzer,* which uses the facts added by the Action Filter and facts regarding previously identified situations to identify, on the basis of the rules in the *Situations Knowledge Base* (*SiKB*), the current situation. The SiKB contains 30 rules that monitor user's activity and identify situations that are critical (or can easily become critical) or enhanceable. As mentioned earlier, the monitoring is difficult because of the complex nature of the problem solving activity in the field of IR; thus, SiKB considers only the actions since last search. Other situations, that would need a long, complex, and probably sometimes wrong reasoning to be detected, are not considered. Notwithstanding this limitation, the system can anyway be effective, because such situations are often prevented. Figure 3 shows, in

**Rule:** Hint for bottleneck

**Comment:** Hint given responding to the situation described by the rule "Bottleneck"

**Hypotheses:**

**IF** the working memory contains the fact named "term insertion in a facet which is not the bottleneck"

**Actions:** Add a fact representing the hint: "You have added some term to the facet $F_i$. This is correct, but you should try to expand the other facets, which seem the bottleneck"

**Activation level:** 2

FIG. 4. A rule in SuKB producing a hint.

**Rule:** Advices to increase the number of retrieved documents

**Comment:** Advices given in a situation where few documents have been retrieved by the search

**Hypotheses:**

**IF** the working memory contains the fact named "few documents retrieved by the last search"

**AND** there are no documents classified so far as useful or topical

**Actions:** Add two facts representing the following advices:

- "area scanning by spreading activation on the classification codes thesaurus, starting from query terms"
- "journal run: look for journal titles containing query terms"

**Activation level:** 2

FIG. 5. A rule in SuKB producing an advice.

**Rule:** Viewed suggestions

**Comment:** Weight modification for viewed suggestions

**Hypotheses:**

**IF** the working memory contains the fact "suggestion displayed to the user"

**AND** the working memory contains the fact "the suggestion has not been accepted"

**Actions:** decrease the suggestion's weight by a constant value

**Activation level:** 1

FIG. 6. A rule in OrKB.

pseudocode, one of the rules in SiKB.

4. *Situation Analyzer,* which, on the basis of the current situation, finds the suggestions contained in the *Suggestions Knowledge Base* (*SuKB*) that can be applied. In general, more than one suggestion can be selected at this time. The 48 rules belonging to SuKB have as hypotheses the situations stored in the working memory and as actions some suggestions (see Fig. 4 for an example of a rule producing a hint and Fig. 5 for an example of a rule producing an advice).
5. *Suggestions Ranking,* which uses the rules contained in the *Order Knowledge Base* (*OrKB*) to sort the available suggestions according to their importance: only the suggestion with the highest weight is immediately visible to the user (the other ones can be examined by scrolling a list). OrKB contains 16 rules based on simple heuristic considerations, aimed at assigning and modifying each suggestion weight (an integer value ranging from 0 to 100). Some *OrKB* rules modify the weight assigned to previously selected suggestion, so that the weight of the suggestions displayed to the user, but not accepted, will be decreased until zero, when the suggestion is deleted from the list. If the suggestion has not yet been displayed to the user, its weight will be lowered by a smaller value (it will remain in the list for a longer time). Figure 6 shows an example of a rule that modifies the suggestion weight.
6. *Suggestions Executor,* which executes the suggestion chosen by the user.

The most important facts in the working memory concern:

7. Type of the user's last action: query modifications (insertion and removal of terms and facets), searches performed, reading of a document, and classifications of documents as useful, topical, or trash.
8. Parameters of the action: the terms or facets (in the case of a query modification) or the document identifier (in the case of a document classification).

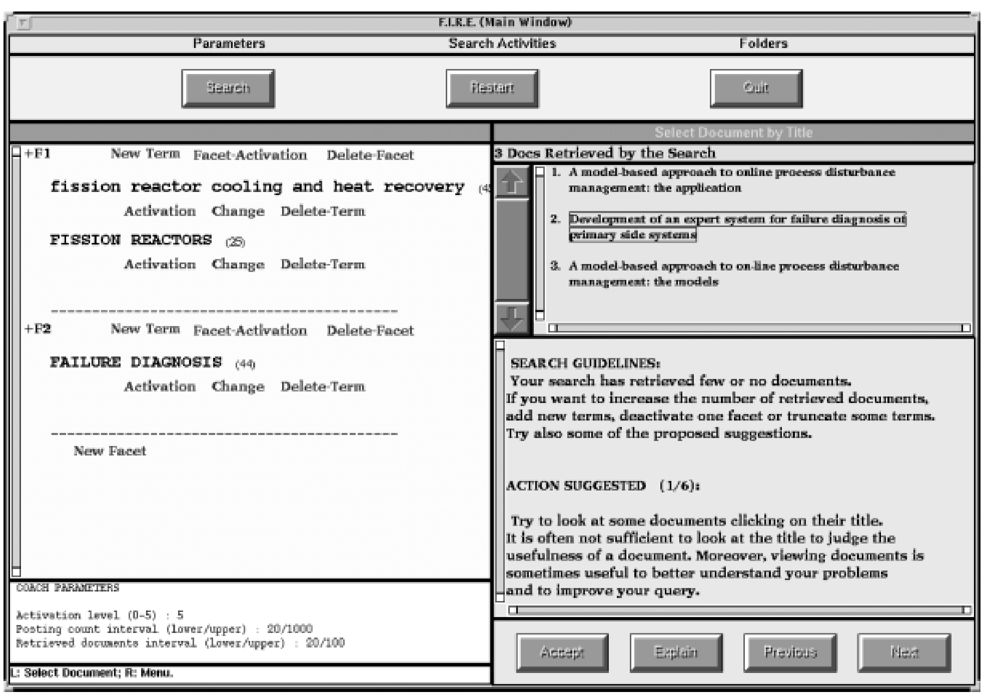


FIG. 7. FIRE main window.

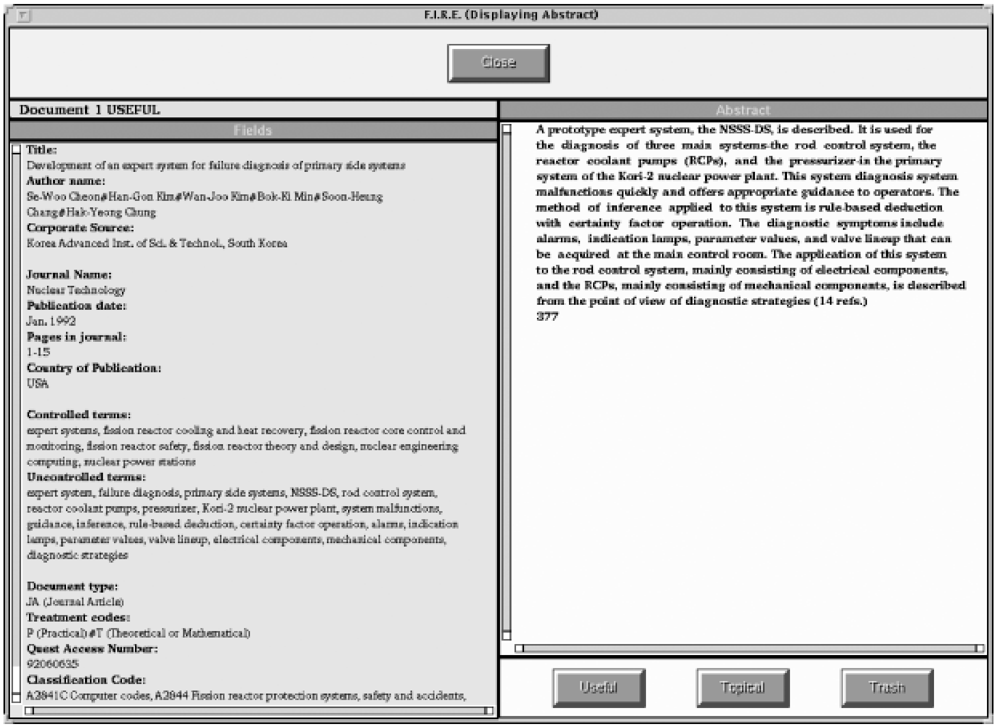


FIG. 8. Abstract window for reading and classifying the surrogates.

9. Current query.
10. Folders content: the lists of documents contained in each folder.
11. Facts about the current situation, by the Action Analyzer.
12. List of suggestions, by the Situation Analyzer and Suggestions Sorter.
13. Current suggestion: the one shown to the user.

*User Interface*

Figure 7 shows the main window of FIRE user interface. It displays to the user the terms (with their posting count) and facets in the boolean query, the titles of some documents, the suggestions proposed by the system, and the values of some system parameters. By means of buttons and menus, the user can modify the query (adding, deleting, changing, activating, and deactivating terms and facets), perform a search in the database on the basis of the current query (Search button in the upper left corner), directly undertake some tactic or stratagem (Search Activities menu), examine the documents in a separate window (by clicking on a document title, a surrogate, i.e., some bibliographic data and the abstract, of the document is displayed in a separate window, see Fig. 8), classify the displayed documents putting them into relevant, useful, or trash folders (buttons in lower right corner in the Abstract window), view the list of the titles of the documents retrieved by the last search or contained in one folder (Folders menu in the main window), accept an advice, ask for explanations about a suggestion, and examine the not displayed suggestions (buttons in the lower right corner in the main window), restart the session, quit the system, and change its parameters (buttons Restart and Quit and menu Parameters in the main window).

*A Sample Session*

For demonstrating the effectiveness of the system, we describe a sample session in which many system capabilities are exploited. Let us assume that a user is interested in designing an expert system able to diagnose failures in a fission reactor cooling system and needs to find papers

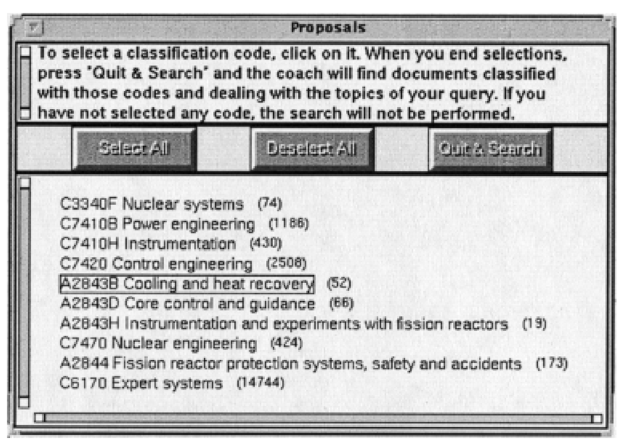


FIG. 9. The window for choosing a classification code for an area scanning.

about this subject. The user starts the interaction with FIRE entering the two-facets query "fission reactors *and* failure diagnosis." SAM immediately gives the suggestion of examining a list of terms related to those inserted into the query. The user accepts the advice and a list is displayed, from which the user chooses the term "fission reactor cooling and heat recovery" to be added to the first facet. The user performs a search into the database, and the results are those shown in Figure 7: three documents are retrieved, and SAM suggests to examine them. The user reads the second document and classifies it as useful (Fig. 8). SAM suggests an author search, that leads to retrieve one, not-yet retrieved, document, and SAM gives the hint of reading it. The user accepts the hint and classifies as useful the document.

After that, the first suggestion proposed by SAM is an author search again. The user decides to examine the second ranked suggestion, which is an area scanning, and to accept it. A list of classification codes is displayed, the user chooses the "A2843B Cooling and heat recovery" classification code (Fig. 9), and the subsequent search finds 51 documents, among which many are classified as relevant. The search goes on, and SAM suggests a journal run that, accepted by the user, leads to two journal names, confirmed by the user. The subsequent search finds 42 documents, and the advice of undertaking a zoom on those documents is displayed. The user accepts, confirms three terms, and finds 82 documents, that he examines. Finally, the coach suggests a zoom on the documents classified as topical or useful.

This sample session contains enhanceable situations only. Examples of critical situations are presented in the following section.

## Evaluation

We performed two experiments for evaluating the effectiveness of strategic help. More in detail we aimed at:

1. Evaluating the importance and usefulness of the strategic help given by the system, by measuring effectiveness of each suggestion (in terms of usefulness and comprehensibility), performance of the system, and user's satisfaction.
2. Analyzing the strategies followed by the users, by means of information gathered on user's behavior.

### *A First Validation Experiment*

In the FIRE 1995 experiment the *less successful users* were analyzed in detail for understanding the reasons of their low performances (Brajnik et al., 1996). The data (logs and video recording) of the search activity of those users have been used for the first experiment, in which we simulated their activities with the new version of FIRE. The aim was to discover if the system acts in an effective way in the critical situations experienced by 1995 less successful users.

The simulation was straightforward: we simply started the search as those users did and followed the suggestions of the system. The results are positive: the system's hints and advices, if followed by the users, would have allowed them to prevent or resolve *all* the critical situations of the 1995 experiment.

For instance, a user looking for documents about expert systems to diagnose failures in cooling systems for fission reactors, input as the first query: "fission reactors *and* cooling systems," that retrieved no documents. The user than *and*-ed a new term, and the query became: "fission reactors *and* cooling systems *and* failure diagnosis." In such a situation (critical, as previously defined), FIRE explains that the new query would retrieve again no documents. FIRE's hint is to try expanding existing facets, and especially the second one, because the term *cooling system* has a very low posting count. To expand the query, FIRE's advice is the application of an automatic thesaurus browsing; by accepting this advice, the user is given a list of terms, among which the promising "fission reactor cooling and heat recovery," "fission reactor safety," and "fission reactor operation." If these three terms are selected and added to the proper facets, we obtain two useful documents.

Another user input the following query: "(reactor *or* nuclear) *and* (cooling system *or* machining) *and* (failure diagnosis *or* failure *or* diagnosis)," and retrieved no documents. The user than deleted the term *failure diagnosis* from the third facet. The user resorted to this query after a frustrating sequence of four searches retrieving no documents, and this is probably the reason for such a curious query structure: he was just applying the *trial-and-error* strategy (and losing patience). Again, the situation is classified as critical. Besides suggestions similar to those seen in the previous example, FIRE suggests the advices journal run and area scanning. Journal run leads to a list of 10 journals, among which the interesting *Transactions of the American Nuclear Society* and the proceedings of the conference *Water Chemistry of Nuclear Reactor Systems.* If the user selects them, the system proposes a list of 84 documents with several useful ones (three in the first 10 positions). Area scanning leads to a list of classification codes containing the promising "Fission reactor protection systems, safety, and accidents." If selected, the system retrieves four useful documents in the first 10 positions.

This first experiment was useful to understand if the system is effective in the situations previously identified, but this does not necessarily mean that it can have a positive impact on searches performed by real users. We needed to evaluate FIRE observing real users using it.

### *Towards a More Systematic Experiment*

#### *Experimental design*

According to Robertson and Hancock Beaulieu (1992) the second experiment is classified as a *diagnostic* and *laboratory* experiment, in which both *quantitative* and *qualitative* measurement methods have been used. According to Harper and Hendry (1997), it is a *light-weight* experiment. The rationale for these choices are that it was not appropri-

(a) Did you easily understand the messages explaining the suggestions?
(b) Did the activities proposed by the system achieve the results that you expected?
(c) Did the suggestions appear in the right context?
(d) Were the suggestions useful for your search?
(e) Was the ranking of the suggestions adequate?
(f) Did you clearly understand the reasons that triggered each suggestion?
(g) Have you had any problem in interacting with the strategic help?
(h) Would you have preferred to get help on your explicit request only?
(i) Do you think that the strategic help feature modified the results of the search?
(j) Do you think that the strategic help feature modified your search strategy, or would you have done the same things even without such a feature?
(k) Did you have any problem in classifying the documents?
(l) Are you satisfied with your search?

FIG. 10. Some of the questions asked in the interview.

ate to test FIRE on a large number of users at this stage of the prototype's development, and, at the same time, it was important to gather information about both system and user's behavior in different situations.

Only six participants took part in the experiments: four of them were graduate students in computer science and the other two were researchers at the Department of Mathematics and Computer Science of the University of Udine. The participants used FIRE performing their searches within the INSPEC20K database. Each participant used FIRE with a different information problem. Each graduate student was given an induced information problem, designed by the experimenters (two of these four information problems were used in the FIRE 1995 experiment). These information problems are called *artificial* because they do not arise from a real user's information need. The two researchers were asked to formulate an information problem concerning with their research interests. These information problems are called *realistic.* As recommended in Borlund and Ingwersen (1997), the artificial information problems were designed with the aim of providing the participants with a *simulated work task situation,* and are mixed with real information problems.

Data gathered by the experiment belong to the following categories:

1. *User's satisfaction.* Satisfaction was measured by means of a structured interview at the end of each search. During the interview each user was asked to express judgments about (see Fig. 10): comprehensibility and contextuality of suggestions, correctness of the suggestion's order (i.e., if the more promising suggestions appear at the beginning of the list), quality of the interaction with the system (ease or difficulty to use), influence, and usefulness of the strategic help.
2. *User's behavior.* Information about user's behavior was gather by means of the logged sessions, audiorecording, and interview.
3. *Performance.* Performance was calculated by means of the classical measures precision, recall, and E-measure (defined here as the average between precision and recall), on the basis of the documents classified as useful by the users at the end of their sessions. These measures were possible for the artificial information problems only, because the useful documents existing in the database were known a priori. Given the low number of users, we did not do any statistical analysis of these data.

Each experimental session was divided into three phases:

1. *Training.* Each participant was trained to use FIRE on a trial information problem. During the training the user was encouraged to ask every information needed to better understand FIRE functionalities. This phase lasted about half an hour.
2. *Testing.* The participant was informed about the way the experiment would have been conducted. Then he was given the information problem to be solved. During the search, one experimenter was continuously available to give technical support (e.g., to explain user interface commands and options) but not to give strategic or terminological support. Another experimenter monitored the session on another computer. Each session was logged and user's activity was audio recorded. This phase lasted about 1 hour.
3. *Interview.* At the end of their search session, users were interviewed to gather their judgments and remarks. The first part of the interview was focused on the topics of interest for the experimenters, while the second part was open, to give the users the chance to freely express their feelings about the system.

*Results and discussion*

In this section we discuss both qualitative and quantitive results. Given the limited number of participants and their common background in computer science, no claim of statistical significance can be made, and the conclusions that we can draw are of course limited. Therefore, our analysis is focussed mainly on qualitative results.

The comprehensibility of suggestions has been judged positively by all users. Nevertheless, in some situations, a misunderstanding of a good suggestion caused its rejection by a user and eventually led to a stalling situation (see earlier). Remarkably, this happened with a user who was quite expert in the use of IR systems. The reason is probably that the user interpreted in a wrong way some suggestions, because of his past experience in the use of different IR systems. Novices are probably more ready to accept suggestions, even when they do not understand exactly their meaning (sometimes they just want to see what happens), because they are not biased by their past experience.

Correctness of suggestions ranking and contextuality of the suggestions have been judged in different ways by the users. Some users said that some suggestions seemed too general, and therefore, it was difficult to judge both contextuality and correctness of the order. This is probably due to a lack of better explanations of those suggestions by the system, which should clarify the link between a suggestion

TABLE 1. Activities performed by users.

| | | Tactics | | | | | | Stratagems | | | Total | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Parallel | | | | | | | | | |
| | | Fix | MA | AB | MTB | MCB | Rel feed | Area scan | Journ run | Auth search | | |
| A1 | A | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 4 | |
| | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| A2 | A | 1 | 1 | 2 | 0 | 0 | 2 | 1 | 0 | 2 | 9 | |
| | S | 0 | 0 | 1 | 1 | 0 | 1 | 3 | 0 | 0 | 6 | 15 |
| A3 | A | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 1 | 1 | 6 | |
| | S | 0 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 | 4 | 10 |
| A4 | A | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 3 | 8 | |
| | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |
| R1 | A | 1 | 0 | 2 | 0 | 0 | 1 | 1 | 3 | 2 | 10 | |
| | S | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 12 |
| R2 | A | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 4 | |
| | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 5 |
| User | | 3 | 2 | 5 | 2 | 1 | 5 | 5 | 4 | 6 | | |
| Exec | | 3 | 3 | 10 | 2 | 1 | 8 | 9 | 6 | 12 | | 54 |

that can be valid in various situations and the specific situation in which it is provided.

The participants judged positively the quality of interaction with the system. In particular, they appreciated the wide variety of search activities proposed, their proposal without explicit help requests and without interrupting users activity, and the control of the interaction kept by them.

For what concerns influence and usefulness of the strategic help, almost all the users said that the system was helpful because it speeded up the searching process. At the same time, they also said that it did not change their search strategy, and that they would have applied the same search activities even without a strategic help. This is quite interesting because, comparing (on the basis of the logs) the behavior of these participants with the behavior of participants to the FIRE 1995 experiment, it is manifest that the strategies followed are completely different: in 1995 experiment, nobody ever executed a stratagem, while all the users kept on applying tactics for query reformulation. We see three explanations for this apparent contradiction:

1. The six participants in this experiment are more expert in database searches than the participants in FIRE 1995 experiment, and thus they exploit a wider range of search activities.
2. The users were not aware of the influence of the strategic help, which actually modified their search strategies in a smooth way. Beyond not perceiving strategic problems, users do not perceive strategic help. This hypothesis is also supported by a learning effect observed: some users, after having accepted a suggestion and experienced its usefulness, autonomously applied the same search activity in subsequent situations.
3. Even if stratagems were also possible in the 1995 version of FIRE, their application was not as easy as in the new version, so they were not applied by the FIRE 1995 experiment participants.

From logs analysis we observed that the users actually intensively exploited the strategic help by accepting many suggestions, and in general, applied different search activities. Table 1 shows the activities performed by the six users (in the first column, A1–A4 had an artificial information problem, R1 and R2 a realistic one) after an accepted suggestion (A in the second column) and activities autonomously selected by users (S) from the menu Search Activities. MA stands for morphological analysis, AB for automatic browsing, MTB for manual thesaurus browsing, and MCB for manual browsing of the classification schema (see earlier). User is the number of users who have applied a particular search activity, while Exec is the total number of executions of each search activity.

The total number of tactics executions (27) equals the total number of stratagems executions, and this equilibrium is also valid for each single user. The low number of executions of tactics like manual thesauri and classification schema browsing, compared with the high number of executions of tactics like automatic thesauri browsing and relevance feedback, confirms that users strongly preferred applying less demanding activities. Moreover, manual browsing was executed after automatic browsing only, which probably means that it was applied only when users felt that the terms proposed automatically by the system were not sufficient to accurately formulate their query.

The number of search activities autonomously executed by the users (13) is much lower than the number of activities suggested by the system and accepted (41). On the one side, this is due to the time needed by the users to get acquainted with a new system: they cannot exploit immediately all its functionalities. On the other side, this can also mean that users found that the activities suggested were sufficient to continue the search, and did not feel the urgency to apply other activities.

TABLE 2. Number of terms added to the query obtained by system suggested tactics (S) and user's selected tactics (U).

| | | Parallel | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | MA | AB | MTB | MCB | Rel feed | Total | |
| A1 | S | — | — | — | — | 1 | 1 | |
| | U | — | — | — | — | — | — | 1 |
| A2 | S | 2 | 2 | 0 | — | 1 | 5 | |
| | U | — | 0 | 1 | — | 4 | 5 | 10 |
| A3 | S | — | 1 | — | — | 1 | 2 | |
| | U | — | 1 | — | 0 | 1 | 2 | 4 |
| A4 | S | 1 | 0 | — | — | — | 1 | |
| | U | — | — | — | — | — | — | 1 |
| R1 | S | — | 0 | — | — | 0 | 0 | |
| | U | — | — | 1 | — | — | 1 | 1 |
| R2 | S | — | 2 | — | — | 1 | 3 | |
| | U | — | — | — | — | — | — | 3 |
| Total | | 3 | 6 | 2 | 0 | 9 | | 20 |

The — stands for a never applied tactic, the 0 for an applied tactic giving no terms.

An analysis of the temporal distribution of search activities indicates that users applied tactics for query reformulation especially at the beginning of their sessions, and actually almost half of them were applied at the beginning of the session, before the first search. After the first search, the paths followed vary a lot across different users, but they are perfectly consistent with the berrypicking and behavioral models. When users find useful documents, often they switch to chaining activities, like searching for other documents written by the same authors; in other cases they browse the database by means of area scanning or journal run, or they just keep on reformulating the query. Anyway, the useful documents are gathered during all the searching process and not only by means of a final query.

Very seldom some users applied the trial-and-error strategy (Chen & Dhar, 1991), and, in general, users seldom applied activities in complete autonomy (i.e., activities not suggested by the system and not belonging to those listed in FIRE Search Activities menu). Three out of six users, after the initial query formulation, continued their session by means of accepted suggestions and FIRE Search Activities menu only.

Table 2 shows that the most effective (the one providing more interesting terms) tactic was the relevance feedback, followed by the automatic thesauri browsing, whereas manual browsing of the classification schema was applied just once and returned no useful terms.

The total number of searches made by means of user's queries (i.e., on explicit user request) is 31, and is almost equal to the total number of searches made by means of stratagems (i.e., made by the system), that is, 27. Sequences of two or more searches that produced no hits are quite rare. We have observed four of these situations only, and in three cases the sequence consisted of just two "no hits" searches. The remaining one consisted of four searches, but in this case the user had already retrieved all the useful documents contained in the database, and he was trying some weird queries. This confirms that the new version of FIRE helped in overcoming the stalling situations experienced by participants to 1995 experiment. Useful documents have been retrieved in equal number by means of user's query (17) and stratagems (19). The most effective stratagems have been area scanning and author search.

Comparing performances obtained by users in this and in 1995 FIRE experiment, we can see slight differences only. For the two information problems used also in the previous experiment, the E-measure was 35.2 and 14.3, while in this experiment it is 36 and 18, respectively (considering the useful documents and giving equal weight to recall and precision). If we consider not only the useful documents retrieved, but also the topical ones, the precision increases considerably, reaching even 100% in one case. This probably means that it was difficult to distinguish between usefulness and topicality of documents. Users confirmed during the interview that they had problems in judging the documents.

Calculating measures of precision and recall the way we did has the drawback that what we actually measured is a mixture of two different skills: system's skill in retrieving topical or useful documents, and user's skill in judging the documents. Moreover, if users are not experts in the respective domains, the only situation that can be simulated by means of the experiment is a user who starts a search on an unfamiliar topic. Ellis (1989a, 1989b) says that, in this context, users apply starting activities; in particular, they try to find some interesting references that can be used both to better understand the matter and to apply chaining activities for finding other documents. This means that, during starting activities, users are not necessarily interested in retrieving many documents, because they are aware of their poor ability of judging them. It is important to take into account these facts when using induced information problems in an evaluation experiment.

## Conclusions and Future Work

A model of user interfaces for IR providing strategic support in a collaborative coaching framework has been proposed, implemented, and evaluated. The evaluation demonstrates the effectiveness of the approach.

We are planning to improve the prototype in various ways. We will extend the knowledge bases of the system to include more tactics and stratagems and new kinds of knowledge not considered so far (knowledge about the user and knowledge about the user–system interaction modalities). We are also taking into account other knowledge engineering tools beyond the classical rule-based approach used in the current version of FIRE, like Bayesian networks (Horvitz, Breese, Heckerman, Hovel, & Rommelse, 1998). We are also improving the Terminological Aid Module and reimplementing the user interface of the system as a Java applet accessible on the Web.

Finally, we will implement a strategic help module for a probabilistic IR system. This might lead to reassess the well-known probabilistic versus boolean IR dichotomy: it is maintained that probabilistic systems perform better than boolean ones (but see Hersh et al., 2001); however, it might well be the case that a strategic help module might be more effective for the latter kind of systems, perhaps leading to better performances than probabilistic ones.

## Acknowledgments

Many thanks to Marcia Bates and to an anonymous referee for many detailed and useful comments.

## References

Baeza Yates, R., & Ribeiro Neto, B. (1999). Modern Information retrieval. Harlow, England: Addison-Wesley.

Bailey, B.P., Konstan, J. A., & Carlis, J.V. (2000). Adjusting windows: Balancing information awareness with intrusion. In P. Kortum & E. Kunzinger (Ed.), Proceedings of the 6th conference on human factors and the Web: Doing business on the Web. Austin TX.

Bates, M.J. (1989). The design of browsing and berrypicking techniques for the online search interface. Online Review, 13(5), 407–424.

Bates, M.J. (1990). Where should the person stop and the information search interface start? Information Processing & Management, 26(5), 575–591.

Belkin, N. (1980). Anomalous states of knowledge as a basis for information retrieval. The Canadian Journal of Information Science, 5, 133–143.

Belkin, N. (1996). Intelligent information retrieval: Whose intelligence? In U. Konstanz (Ed.), ISI '96: Proceedings of the fifth international symposium for information science (pp. 25–31). Konstanz.

Belkin, N., Cool, C., Kelly, D., Lin, S.-J., Park, S.Y., Perex-Carballo, J., & Sikora, C. (2001). Iterative exploration, design and evaluation of support for query reformulation in interactive information retrieval. Information Processing & Management, 37(3), 403–434.

Belkin, N., Marchetti, P. & Cool, C. (1993). Braque: Design of an interface to support user interaction in information retrieval. Information Processing & Management, 29(3), 325–344.

Belkin, N.J., Cool, C., Stein, A., & Thiel, U. (1995). Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. Expert Systems with Applications, 9(3), 379–395.

Borlund, P., & Ingwersen, P. (1997). The development of a method for the evaluation of interactive information retrieval systems. Journal of Documentation, 53(3), 225–250.

Brajnik, G., Mizzaro, S., & Tasso, C. (1996). Evaluating user interfaces to information retrieval systems: A case study on user support. In SIGIR96, 19th international conference on research on development in information retrieval (pp. 128–136). Zurich, Switzerland.

Chen, H., & Dhar, V. (1991). Cognitive process as a basis for intelligent retrieval system design. Information Processing & Management, 27(5), 405–432.

Cummings, G., & Self, J. (1989). Collaborative intelligent educational systems. In D. Bierman, J. Breuker, & J. Sandberg (Eds.), Proceedings of the 4th international conference on artificial intelligence and education, volume 3 of Frontiers in Artificial Intelligence and Applications (pp. 73–80). Amsterdam, the Netherlands. IOS.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1998). Human–computer interaction. (2nd ed.). New York: Prentice-Hall.

Efthimiadis, E.N. (1996). Query expansion. In M.E. Williams (Ed.), Annual Review of Information Science and Technology (ARIST), 31, 121–187.

Ellis, D. (1989a). A behavioural approach to information retrieval system design. Journal of Documentation, 45(3), 171–212.

Ellis, D. (1989b). A behavioral model for information retrieval system design. Journal of Information Science, 15, 237–247.

Ellis, D., Cox, D., & Hall, K. (1993). A comparison of the information seeking patterns of researchers in the physical and social sciences. Journal of Documentation, 49(4), 356–369.

Ellis, D., & Haugan, M. (1997). Modelling the information seeking patterns of engineers and research scientists in an industrial environment. Journal of Documentation, 53(4), 384–403.

Fenichel, C.H. (1981). Online searching measures that discriminate among Users with different types of experience. Journal of the American Society for Information Science, 32(1), 23–32.

Greenberg, J. (2001a). Automatic query expansion via lexical-semantic relationships. Journal of the American Society for Information Science, 52(5), 402–415.

Greenberg, J. (2001b). Optimal query expansion (QE) processing methods with semantically encoded structured thesauri terminology. Journal of the American Society for Information Science, 52(6), 487–498.

Guida, G., & Tasso, C. (1994). Design and development of knowledge-based systems—From lyfe cycle to methodology. Chichester, UK: John Wiley & Sons.

Harper, D.J., & Hendry, D. G. (1997). Evaluation light. In Proceedings of the second MIRA workshop (pp. 53–56). Available online at http://www.dcs.gla.ac.uk/mira/workshops/padua_procs/. Also, Technical Report TR-1997-2, Department of Computing Science, University of Glasgow, Glasgow.

Hersh, W., Turpin, J., Price, S., Kraemer, D., Olson, D., Chan, B., & Sacherek, L. (2001). Challenging conventional assumptions of automated information retrieval with real users: Boolean searching and batch retrieval evaluations. Information Processing & Management, 37(3), 383–402.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998). The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In Proceedings of the 14th conference on uncertainty in artificial intelligence (pp. 256–265). Madison, WI: Morgan Kaufmann.

Hsieh-Yee, I. (1993). Effects of search experience and subject knowledge on the search tactics of novice and experienced searchers. Journal of the American Society for Information Science, 44(3), 161–174.

Kay, J. (Ed.) (1999). UM99—User modeling: Proceedings of the seventh international conference. Wien: Springer-Verlag.

Korfhage, R.R. (1997). Information storage and retrieval. New York: John Wiley & Sons.

Lancaster, F.W. (1979). Information retrieval systems: Characteristics, testing and evaluation (2nd ed.). New York: John Wiley & Sons.

Larson, R.R. (1991). The decline of subject searching: Long-term trends and patterns of index use in an online catalog. Journal of the American Society for Information Science, 42(3), 197–215.

Mandala, R., Tokunaga, T., & Tanaka, H. (2000). Query expansion using heterogeneous thesauri. Information Processing & Management, 36(3), 361–378.

Manglano, V., Beaulieu, M., & Robertson, S. (1998). Evaluation of interfaces for IRS: Modelling end-user searching behavior. In Colloquium on information retrieval (IRSG)—Draft (pp. 137–146).

Marchionini, G. (1989). Information-seeking strategies of novices using a fulltext electronic encyclopedia. Journal of the American Society for Information Science, 40(1), 54–66.

Marchionini, G. (1995). Information seeking in electronic environments. Cambridge, UK: Cambridge University Press.

Meadow, C.T. & Cochrane, P.A. (1981). Basics of online searching. New York: John Wiley & Sons.

Porter, M.F. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137.

Robertson, S.E., & Hancock Beaulieu, M.M. (1992). On the evaluation of IR systems. Information Processing and Management, 28(4), 457–466.

Sullivan, M., Borgman, C.L., & Wippern, D. (1990). End-users, mediated searches and front-end assistance programs on dialog: A comparison of learning, performance and satisfaction. Journal of the American Society for Information Science, 41(1), 27–42.

Sutcliffe, A.G., Ennis, M., & Watkinson, S.J. (2000). Empirical studies of end-user information searching. Journal of the American Society for Information Science, 51(13), 1211–1231.

Twidale, M.B., Nichols, D.M., Smith, G., & Trevor, J. (1995). Supporting collaborative learning during information searching. In CSCL'95 Proceedings of computer support for collaborative learning (pp. 367–374). Bloomington, IN.

Vollaro, A.J., & Hawkins, D.T. (1986). End-user searching in a large library network: A case study of patent attorneys. Online, 10(4), 67–72.

Wenger, E. (1987). Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication and knowledge. San Mateo, CA: Morgan Kaufmann.

Wilson, T. (1999). Models in information behaviour research. Journal of Documentation, 55(3), 249–270.

Xie, H. (2000). Shifts of interactive intentions and information-seeking strategies in interactive information retrieval. Journal of the American Society for Information Science, 51(9), 841–857.