ELSEVIER

Interacting with Computers 10 (1998) 321–351

Interacting with Computers

Towards a cognitive theory of information retrieval

Alistair Sutcliffe*, Mark Ennis

Centre for HCI Design, School of Informatics, City University, Northampton Square, London EC1V 0HB, UK

Abstract

A framework for constructing a cognitive model of users' information searching behaviour is described. The motivation for the framework is to create explanatory and predictive theories of information searching to improve the design of information retrieval (IR) systems. The framework proposes a taxonomy of components for process models of the information seeking task, information need types and knowledge sources necessary to support the task. The framework is developed into a preliminary version of a cognitive theory of information searching by the addition of strategies and correspondence rules which predict user behaviour in different task stages according to information need types, facilities provided by the IR system and knowledge held by the user. The theory is evaluated by using claims analysis based on empirical observations of users information retrieval and by a walkthrough of an IR session to investigate how well the theory can account for empirical evidence. Results show that the theory can indicate the expert strategies which should be followed in different task contexts but predictions of actual user behaviour are less accurate. The future possibilities for employing the theoretical model as a tutorial advisor for information retrieval and as an evaluation method for IR systems are reviewed. The role and potential of cognitive theories of user task–action in Information Retrieval and Human Computer Interaction are discussed. © 1997 Elsevier Science B.V.

Keywords: Information searching; Cognitive task models; Theory; Framework; Scenarios

1. Summary

This paper proposes a synthesis theory of information retrieval that builds on previous cognitive models of the IR process and incorporates experimental evidence of user information seeking behaviour. The paper sets out a modelling framework for investigating information retrieval and reports our progress to date in developing the pre-theoretical framework into a theoretical model.

* Corresponding author.

PII S0953-5438(98)00013-7

The major components of the framework are a process model of information searching, and knowledge representations necessary to support the process. The process model contains activities which describe information searching as the cognitive tasks and strategies that dictate physical or cognitive user actions. Strategies may be considered as plans or methods that encapsulate expertise, and some mistakes that model novices' sub-optimal behaviour.

The four main activities performed in the retrieval process are: problem identification, need articulation, query formulation and results evaluation. Problem identification formulates the user's goal or information need. Information needs are expressed as properties which describe the query in terms of complexity, intended target and specificity of expression. Initial search strategy selection is driven from the need type definition. Need articulation input is restricted to natural language expressions of a particular need which are parsed into knowledge structures containing high level concepts and semantic propositions. Query formulation transforms the conceptual need into the keywords and query syntax employed by the computerised retrieval system. Results evaluation enables the comparison of the retrieved results against the information need and the adaptation of the searches direction accordingly. Evaluation strategies are triggered by volume, relevance and precision parameters of the results set. These four activities form the iterative cycle of information seeking. Various routes through this cycle are possible depending on strategy selection rules and the outcome of interaction with the computer supported retrieval system.

Strategies represent the user's information searching skill. Seven strategy rule sets are used within the process model for problem identification; need articulation; query formulation; results evaluation; query reformulation; and process strategies.

Two levels of knowledge representations are incorporated within the model as knowledge sources and query instance models. Knowledge sources are stereotype models which describe the knowledge held by a user and influence strategy selection. Query instance models hold domain specific information and describe the propositional content of the query, including knowledge relevant to a query that a user may extract from long-term memory; the system thesaurus or retrieved documents.

Knowledge sources are categorised into domain, device, information resource and IR knowledge. Domain knowledge describes knowledge of facts, concepts, and their relationship in a specific domain. Device knowledge is composed of functions provided by the task support facilities (TSFs) and knowledge of the syntax and semantics of the system's retrieval language. TSFs are a description of the user interface features offered by a system. This enables the model to evaluate different systems through the encoding of the facilities they provide to support the user's task. Information resource knowledge describes the user's awareness of the searchable databases and their properties. IR knowledge describes the search strategies which may be employed as expert behaviour.

Query instance models are composed of three semantic primitives (concepts, terms and relationships) that are used to construct schema of an evolving information need. Concepts are high level aggregations of information that relate to some entity or goal. Terms express the attributes of the search target in a device query language. Relationships associate terms and concepts into knowledge structures. Queries are formed by extracting elements from these query instance models.

To integrate the components of knowledge representations and behaviour we use correspondence rules. These predict how the process model will operate according to inputs describing the user's knowledge, system facilities and an information need.

To test the theory we conducted experiments of users information seeking with the MEDLINE database. Observations taken from this experimental data are used to assess the theory's claims. Scenarios are used to run the process model to generate explanations for observed behaviour and search performance, e.g. poor search result due to premature termination. Scenario analysis shows that the theory does explain strategy selection during information searching but it only does so at a general level. It cannot account for hybrid strategies and only partially for sub-optimal performance.

The modelling framework we report provides a means of scoping the information searching problem. While there are many aspects we have not covered, in particular the problem of how the user's conceptual model of a query may develop, the framework does elaborate previous models of information searching, e.g. Kuhlthau's process model [10] and Marchionini's framework [42], with strategies and rules that provide a much richer account of behaviour. Although the current theory is preliminary and inadequate to deal with several aspects of observable user behaviour, it is at least a starting point. The initial scenario-based validation demonstrates that the theory has explanatory power, and can account for how expert strategies should be selected for a particular context. Predictions of individual user behaviour are beyond the scope of the current theory and may remain so. User searching strategies may be inconsistent for several reasons. First, individuals may adopt strategies from memory even if they were sub-optimal; moreover, different search styles may be necessary for particular types of IR system. Even though the prognosis for predicting individual behaviour is poor, as this would require acquiring more knowledge of the individual user, much can be achieved by modelling users as generic groups.

Possible applications of the theory are a tutoring or intelligent help system for information retrieval based on the process model and its rule set; a walkthrough evaluation method to test the utility of IR systems according to the task support provided and as a decision support system for configurable information retrieval systems. The cognitive theory could provide predictions of expert strategies for a particular search need thereby acting as an expert system advisor using human intermediary knowledge. Such approaches have been explored in intelligent information retrieval [7], although by employing domain specific knowledge as well as generic strategies. Our approach has also modelled novice users, so it can suggest how they may be helped by pointing out common mistakes, suggesting expert searching strategies and by indicating the system facilities to consult at each stage in the search process. This may be developed towards configuration support for different types of users.

2. Introduction

Several authors have proposed user models of information retrieval, for instance Belkin's Anomalous State of Knowledge (ASK) model [1]; however, no comprehensive account of user's information seeking behaviour has been reported. Ingwersen [2, 3] has proposed a framework for modelling information retrieval in his polyrepresentation theory, and noted the need for improved cognitive theories of information retrieval

(IR). While most would acknowledge that information searching is an iterative type of task, e.g. Markey and Atherton's pearl growing model [4] or Belkin's ASK model, and that uncertainty and unpredictability characterise information searching tasks [5], few detailed cognitive models of the search process have been proposed, in contrast to the many detailed models of machine based searching (e.g. [6, 7]).

Several user models of information searching have emphasised the changing nature of the user's information need; for instance, Bates' berrypicking model [8,9] describes information seeking behaviour in response to exploratory needs in which the information target evolves as a result of feedback from retrieved results. Similarly, Belkin's ASK model characterised how a user's perceived need changes during the search process as a consequence of feedback from retrieved results. Kuhlthau [10] proposed a six-stage model of the searching process composed of high level goals: initiation, problem definition, source selection, formulating queries, examining results, and extracting useful information. This provides a process model which could be elaborated to account for different ways that users may implement information seeking tasks. The scripts of information searching of Belkin et al. [11] indicate how high level goals may be achieved by searching behaviour linked to a particular search context that the user encounters. In conclusion, there have been a number of models published which contributed differing views on the IR process. However, user behaviour is inconsistent in the search strategies adopted even for the same search need and system [12, 13]. None of the models to date have been able to account for how user search behaviours should be organised for optimal performance; furthermore, the models have not been synthesised into a common theory which draws on the various contributions.

Although there are not large numbers of experimental studies on information retrieval there are several reports which do provide evidence that cognitive modellers can draw upon. How retrieved results are displayed to users can alter search strategies [14, 15] while the task complexity, reflected in the information need, also affects user's strategies [16]. The user's model of the system and domain knowledge impact on the search process [17, 18]; moreover, user motivation and the importance of the information need also impact on search duration and the effort a user will employ [18]. Rouse and Rouse [19] in a review of empirical studies, summarise a wide variety of variables that can effect searching behaviour, including payoff, costs of searching, resource available, amount of information sought, characteristics of the data and conflicts between documents. Cognitive models of information searching need to account for how these factors affect the search process and provide accounts that are based on experimental evidence.

From the human–computer interaction (HCI) perspective cognitive modelling of information searching is related to models advocated in the AMODEUS project which provides bridging models ranging from human information processing architectures [20] to cognitive models of user system interaction implemented as a computational model (PUMS [21]) and software engineering/interaction models with design rationale [22]. Computerised implementations of the human information processing architecture EPIC [23] and GOMS [24] have been used to predict expert performance and timings in small deterministic tasks. These investigations focused on small-scale scenarios of interaction to obtain precise predictions of behaviour, whereas information retrieval implies more broadly based models.

Hence, our approach is to model cognitive tasks at a high level of granularity and sacrifice some precision. Our approach to cognitive task modelling draws on models of human action [25], models of reasoning and levels of user knowledge proposed by Rasmussen [26] and elaboration of action models to describe goal directed tasks, realised in the walkthrough methods [27].

The main alternative to the modelling tradition is the task artefact cycle of Carroll et al. [28]. This is appealing in its focus on designed artefacts embedding HCI theories with claims analysis to extract knowledge and theoretical assertions about why a particular design should have good usability. Although the task artefact tradition does offer an entry point in which psychological theories, principles, etc. may be introduced via psychological rationales, it does not give a comprehensive account of the psychological process of interaction. In attempting to build a cognitive model of the information searching task and describing activity in the context of information retrieval systems, we propose a framework that can embed many claims about information searching tasks and interaction with information retrieval systems.

This paper takes the cognitive modelling theme forward by arguing for a synthesis theory of information retrieval that takes differing modelling viewpoints into account and is based, as far as possible, on experimental evidence. Developing such a model is a complex task, so this paper sets out the framework to assist this process and reports our progress to date in developing the pre-theoretical framework into a theoretical model. Since cognitive modelling needs to demonstrate its utility beyond the intellectual endeavour of theorising, the final part of the paper looks forward to the way in which cognitive models may be applied in improving the human computer interface of information retrieval systems.

3. A cognitive framework for modelling information searching

First we should explain that by information searching we are considering a range of behaviours from goal directed information searching to more exploratory information browsing. We also wish to account for information retrieval from relational databases as well as bibliographic databases. Many empirical studies and theoretical investigations have focused on text searching where needs tend to vary from exploratory, evolving searches to more specific searches. In relational databases empirical studies are less numerous, and searches tend to have a specific target in mind, but this may not always be true. The first distinction we wish to draw is between goal directedness of information seeking and the influence of the user's task. Three starting points for information seeking are considered:

- goal directed information searching when the user has a specific target in mind;
- serendipitous or exploratory information seeking when no specific goal is present besides the intention to explore an information repository;
- embedded information seeking when the user's goal is motivated by an 'external' task, e.g. a doctor wishes to find information to confirm a diagnosis. This is a specialisation of goal directed searching, although the degree of task embedding may vary. In

research–investigation tasks information seeking becomes an important task in its own right and exploratory searching is observed, whereas in operational tasks information searching is a smaller component within the external task and the user's goal will generally be more specific.

Most models of information searching (e.g. [3]) consider three agents: the user, the expert intermediary and the retrieval system. In contrast to this view the framework we propose uses just two agents, the user and the information search support system. The reason is that we wish to model expert user behaviour, which would incorporate an intermediary's skills, and create a framework that helps evaluate which aspects of the information searching task are best carried out by people or computer systems. The dual agent model allows different versions of human expertise to be posited, so different levels of computerised support can be specified. The major components of the framework are a process model of information searching, and knowledge representations necessary to support the processes.

3.1. Process modelling

Two components are proposed for process models: first, activities as high level components that describe the information searching task in general terms. Activities, in turn are composed of strategies that dictate user action, which may be either observable, physical actions or cognitive actions. Strategies may be considered as plans or methods to deal with specific aspects of the information searching problem.

3.1.1. Activities

Information searching falls into four major activities:

Problem identification: this involves identifying the initial goal or information need. This activity may form part of the external task or become general problem solving in unstructured tasks. If the problem or information need is complex, decomposition methods are used to split the problem into smaller components. Different problem components may be prioritised for the search. The end point is that a need is known, even if that need is general and implies simple exploration of an information repository.

Articulating needs: once a need has been identified it may be expressed as concepts or high level semantics. These are refined into lower level terms which are utilised in queries. Terms may be acquired from the user's long-term memory or from external sources, such as the system thesaurus or paper-based keyword indexes. Unwanted indexing terms and inappropriate keywords have to be filtered out. An intersection between the process and knowledge representation is apparent in the extent to which a user's goal is considered to be held in memory or explicitly articulated. The act of articulation in language invariably causes refinement of concepts and information needs.

Query formulation: the complexity of this activity depends on the sophistication of the IR system and the user's skill in generating queries. Hence complex queries can be formed if the user is skilled in Boolean query languages and such a language is available in the retrieval systems (e.g. SQL). Alternatively, query formulation may be trivial with a hypertext system when the user merely recognises search terms as hotspots and follows information links. Two sub-activities may be recognised; first identification of search terms and then transforming these into the query language supported by the search system.

Evaluating results: once information is retrieved the user has to decide whether to accept the retrieved results or to continue searching. Three sub-tasks are involved. First, the user has to scan the result set or examine the contents in detail. Various scanning and sampling strategies are possible. Then a decision has to be reached as to how useful the retrieved results are and whether they are sufficient to meet the need. Once this decision is reached the user has to elect to either accept the results or decide how the query should be changed. Evaluation activity will depend on the richness of the user's domain knowledge; the more you know about a subject area, the better one can determine the relevance of the search results. However, the complexity of this activity is also determined by characteristics of the machine. An IR system which allows for browsing results and relevance feedback will offer more possibilities than a simple display interface [14].

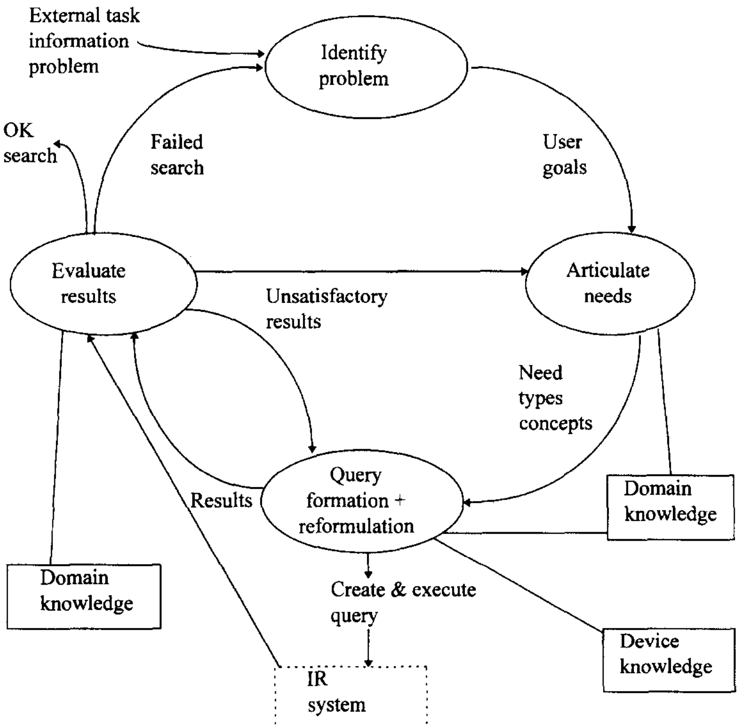

Fig. 1. Process model of information searching activities and knowledge sources.

The cycle of information seeking and its principal activities are summarised in Fig. 1. Problem identification forms the entry point. Searching then proceeds around the cycle with Evaluation leading into Query-reformulation. The knowledge sources which influence activity are the user's knowledge of the domain and the IR system. These are elaborated later. In addition, there is the user's knowledge of the current query which may be held in working memory during a search session. The model of the current query will contain query terms, concepts drawn from domain knowledge which are relevant to the query, and descriptions of the retrieved results. This model will evolve as the query progresses, and reflects the user's anomalous state of knowledge.

3.1.2. Strategies

Strategies represent information searching skill. Good strategies are a facet of expertise and can be observed when expert intermediaries search information sources. Other strategies may be sub-optimal or naive and are used to model an ordinary user's approach to information seeking. Ordinary users may, of course, learn expert strategies given sufficient opportunities. Some strategies are domain dependent and may be employed by domain experts, although these cannot be described in a generic model.

Strategies can be grouped approximately but not exclusively in each of the four principal activities, although this categorisation is influenced by the query cycle, as query reformulation introduces new strategies not necessarily in the initial stages of searching.

3.1.2.1. Problem identification. Most strategies for this activity are in the realm of general problem solving; for instance, divide and conquer describes horizontal partitioning of a complex problem into separate areas, whereas top down decomposition is a vertical approach that deals with complexity by splitting the problem into sub-components. Alternatively, means-ends analysis can discover goals in problems of causation or consequence. These strategies can be used to identify an information goal in complex problems; however, as problem identification is an activity shared with many tasks besides information searching, we do not elaborate routine problem solving strategies in the framework. An early strategic decision is to choose between detailed querying in which effort is invested in getting the query 'right first time', hence a complex query is formed for a search, and iterative querying in which queries are simple and the search progresses by assessing feedback over many query cycles. Other general strategies depend on the search goal. An exploratory goal implies a browsing style of search, whereas a more specific need suggests querying.

3.1.2.2. Articulating needs. When the information searching cycle starts, the principal strategy is one of refinement to find lexical terms which express the searcher's goal. High level descriptors of the goal (commonly referred to as concepts) are refined into terms. The user's ability to carry out this strategy effectively depends on their domain knowledge, so if this is poor, they have to acquire search terms from the environment (e.g. indexing terms, thesauri and controlled vocabularies). Another approach is to generate examples of possible search results or to cite analogues and similar cases, and then try to extract search terms from them.

3.1.2.3. Query formulation. The users' abilities in this activity are highly constrained by their knowledge of query languages and the search device, so it is only worthwhile to describe general strategies, such as either forming complex queries with Boolean logic or using simple queries and search iterations to find appropriate results. Describing the reasoning involved in query formation at the syntactic level depends on variations in query language syntax. The strategies we propose describe whether a user is likely to choose to use complex or simple query language according to their knowledge and information need. If the users' device knowledge is high then they are likely to adopt a complex search strategy for a complex information need, with the converse being true if they have low device knowledge.

3.1.2.4. Evaluating results. Users may either decide to browse a result set or to view the content of retrieved items. Different sampling strategies are possible, for instance serial search, scanning, or systematic sampling of the retrieved set. Strategy selection will depend on how the IR system presents the results and the quantity of retrieved items. When large numbers of items are retrieved, scanning and sampling will be necessary, whereas inspection of contents will be possible with a small number of items. Judgements of usefulness of the retrieved results depend on the user's domain knowledge, although other strategies may be used such as assessing the authority of the author. Depending on the usefulness of the results the user chooses to either accept the search or reformulate the query.

3.1.2.5. Query reformulation. These strategies are part of query formation activity in later cycle stages and are closely coupled to evaluation. There are two levels of strategies; first, decisions on which way the query needs to be changed; then the results of this decision invoke lower level tactics for reformulating queries. Broadening is a response to a small results set, and involves either reducing the number of search terms or introducing disjunctions (ORs). Other means of broadening a query include lexical treatment such as stemming search terms by removing suffixes. Narrowing is the converse strategy to reduce the size of a result's set by either adding more search terms or converting disjunctions into conjunctions (ANDs). Negations (NOTs) may also be used to exclude unwanted items.

Other means of implementing broadening and narrowing strategies is by the use of controlled vocabularies, or thesauri. Searches may be narrowed by choosing more specific terms in a hierarchical schema, also reducing synonyms and adding related terms can help to narrow a search. Searches may be broadened by the opposite action; adding synonyms and removing related terms, and substituting more general for specific terms. If value domains are being queried in relational databases, broadening may be achieved by relaxing constraints; for instance, extending a range on a date field or increasing the number of value instances. Increasing constraints narrows queries, with the inverse treatment of decreasing ranges or removing the value instances, e.g. Find all Customers with birth dates = 1950, 1962, 1955.

3.1.2.6. General strategies. These describe an approach to the search process motivated by the type of information need. Such strategies may be considered as plans formed by the user; however, planning may not be explicit in information retrieval which is frequently driven by situation action in the sense of Suchman [29]. In that perspective general strategies may be seen as constraints on the onset of searching behaviour. These choices may be a function of user's motivation, searching skill, or the importance of the need. They determine the approach to searching in the problem identification stage and influence when information searching may terminate. These strategies also evaluate the usefulness of the retrieved results against the importance of the information need and user motivation.

3.1.2.7. Consultation strategies. When the users' knowledge is inadequate, one option is to consult an external source of knowledge. This source may be a human expert intermediary or a computer-based information retrieval system. As the framework focuses on human computer interaction, these strategies will direct the user to consult the system to supplement their knowledge. For instance, when the user possesses insufficient domain knowledge, the system thesaurus may be consulted to acquire more terms and concepts.

3.2. Knowledge representation

This component complements the process model as knowledge held by the user will impact on search strategies. The knowledge representation framework is decomposed into two levels:

Knowledge sources: these are stereotype models which describe knowledge held by the user; for instance, domain knowledge may be high, medium or low. These models influence strategy selection.

Query instance models: these hold domain specific information at a lower level of detail to describe the knowledge relevant to a query that may be extracted by the user from their long-term memory, system facilities, such as thesauri, or from retrieved documents.

The latter models are less strongly related to process models; instead query instance models address a different, yet central problem in information retrieval—that of mapping terms and concepts held by the user to indexing and content descriptors of the target information. Ingwersen's polyrepresentation theory [3] falls into the query instance category although it does so at a high level of abstraction and covers some process issues as well.

3.2.1. Knowledge sources

We distinguish four knowledge sources held by the user.

Domain knowledge: this describes knowledge of facts, concepts, and terminology in a specific domain. High domain knowledge enables users to assess retrieved results more effectively, and provides a richer set of concepts and terms for query formation.

Device knowledge: this source has two sub-components; first, knowledge of the facilities afforded by a particular system, e.g. whether it has a thesaurus, keyword listing, reusable query library, etc. All of these task support facilities (TSFs) may be present in some ideal system; in reality, only a sub-set will be implemented, and hence accessible by the user. The current list of TSFs is given in the following table, organised in three task groupings:

Support type	Facilities
Browsing support	Thesaurus
	Concept maps—visualisation of thesaurus or classification schemes
	Metadata dictionary—description of available databases and their contents
	Database indexing structures
	Hypertexts
Query formation support	Boolean query languages
	Syntax directed editors
	Keyword queries (non-Boolean—implicit ANDs)
	Query-by-pointing (hypertext concept maps or hotspots on realistic domain images)
	Query-by-example (find similar/related items)
	Pre-formed queries
	Reusable queries
Evaluation support	View results summary—number of hits
	Results summary browser—title, abstract, etc.
	View results—contents
	View and mark/reuse results (relevance feedback)

When a particular IR system is being evaluated the TSFs supported by the application are listed in a system model. We do not include machine based search strategies such as statistical or inferential techniques as these properties of the machine are not within the scope of the framework. A more efficient search engine will support the user's task more effectively, but the need for the facilities to support the search task remains. Hence for a search system such as OKAPI [30,31], we would not describe the internal features of the search engine which uses a combination of different search techniques, but we would describe the user interface features as TSFs. In OKAPI's case this would involved adding TSFs to the above list such as an active thesaurus which prompts the user with additional search terms. The TSFs in a system model describe an IR support system in terms of its high level components or services. Process model rules are triggered to consult this model if the user does not possess the necessary domain or device knowledge and requires task support. If the desired TSF is found the strategy will succeed, otherwise failure is reported.

The second sub-component is knowledge of the query language implemented in the device, such as SQL, other Boolean query languages, or use of restricted natural language. Such knowledge is complex and may not be specifically linked to a particular device; for example, a user may transfer their expectations from one device to another leading to misconceptions, e.g. a user types in keywords with Boolean operators to a system which in fact uses a stop list to throw away 'and' 'or' and 'not'.

Information resources: these describe the user's awareness of the searchable databases and their properties. This source may be considered to be an extension of device knowledge. In an ideal world databases should be transparent to the user; however, that is rarely so. Knowledge of the database subject domain, costs, and response time are key features of expert intermediary knowledge which enable them to direct user searches to appropriate databases.

IR knowledge: this describes knowledge of searching strategies which have been covered earlier.

3.2.2. Query instance models

These are composed of three semantic primitives which are used to construct hierarchical or networked schema that describe an information need as it evolves during searching.

Concepts: are higher level aggregations of information that are related to some entity or goal. Concepts are expressed in linguistic terms, but providing a strict definition to distinguish them from keyword terms is difficult. A pragmatic justification for distinguishing concepts from terms is to describe higher order semantics which would not usually form query terms and lower level descriptors of a search target which are useful as keywords, although we recognise that concepts can be refined into search terms. In practice, concepts are entities, higher order categories in hierarchical schema and generally stated needs.

Terms: express attributes of the search target in language which may be used by a search mechanism to match against the indexing terms or contents of a target document. Terms in entity–relationship parlance will be attributes and their value domains, while they are keywords in bibliographic searches.

Relationships: these associate terms and concepts into knowledge structures using relationship types drawn from semantic databases and unified bibliographic and relational schema:

- specialisation-of: relates a more detailed, less abstract term to its parent
- generalisation-of: relates a more general, abstract term to its child
- aggregation (part-of): composes terms into a whole at the same level of abstraction in a class hierarchy
- instance relations: associates two terms according to a user defined functional relationship
- alternative-for: expresses a synonym for a term in a thesaurus

Query instance models are heterarchies which preserve levels of abstraction as the search is refined from higher level concepts to lower level terms. These models are formed during the search process by recruiting components from domain knowledge held in the user's memory, supplied by the machine in the form of thesauri and extracted from retrieved documents. Fig. 2 illustrates a query instance model for a medical domain.

Queries are formed by extracting terms from these models. This may be a simple process of taking terms as keywords and entering them into a search system; alternatively, the terms may have to be constructed into a complex Boolean structure according to the syntax of a query language. In the framework we propose operators for changing the query instance model, but no transformation procedures for constructing queries. The operators

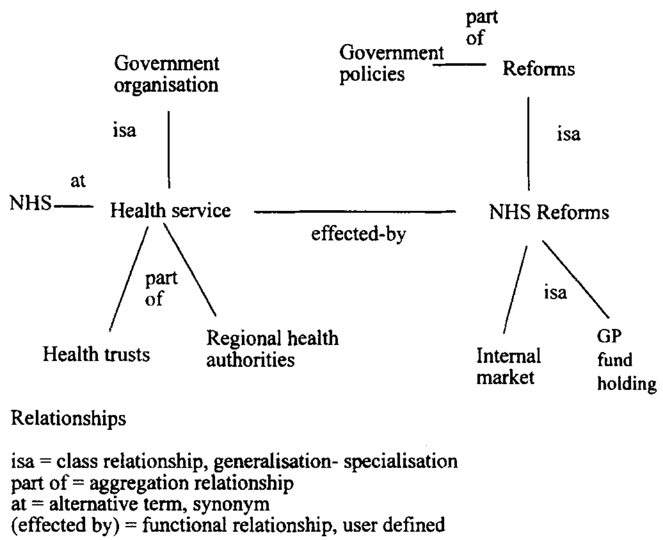

Fig. 2. Model of concepts and terms in a query instance model.

for building query instance models are:

Create model <initiates construction of a new model >
Delete model <deletes an existing model >
Add term <adds a new term as a leaf node to the tree >
Remove term <prunes the tree by removing a node >
Replace terms <substitutes a term at a location the tree >
Change relationship type <changes attribution of the relationship between two nodes >

Strategy rules in the process model invoke operations to construct the query instance model; for instance, Refine Query strategy adds more nodes to the tree to create a more elaborate model.

3.2.3. Information need types

As search strategies are determined by the type of need [14] reflected in the user's goal, there is considerable motivation to taxonomise need types. Although these are not strictly a knowledge source held by the user, or a model of the query, we include them in this section as they describe different states of knowledge which pertain at the onset of searching. Such states, and hence need types, may change as searching progresses when the user discovers new needs after evaluating retrieved results.

We express need types in contrasting pairs:

Known/unknown need: this describes the state of knowledge held by the user about the information to be acquired.

Variable/fixed: characterises how specific and important the user's need is and whether it will change during the search process or not. Exploratory goals are more likely to be variable as the search target changes during exploration.

Precise/general target: a precise target is when the user knows either the identity or a specific description of the search target; this is contrasted with a search when users possess only high level concepts. Note that precise knowledge of the target is a function of known-ness of the need, but it is possible to have an unknown yet specific need (e.g. when you have forgotten a specific reference but you know it is in the database somewhere).

Simple/complex targets: searches may involve locating a single item, or in complex targets several items may be required; furthermore, in some searches the search targets may have dependencies. These form links which have to be followed, so the first item has to be found which contains clues to the second, etc. In Hypertext, the designer has already constructed the path for the user.

Values/text: values or numeric queries are familiar in relational databases, and for date constraints in bibliographic searches. With multimedia databases this distinction is becoming blurred; however, value/text queries have different connotations in terms of strategies, e.g. use of ranges in value domains to narrow queries.

Finally, we consider a more complex set of needs, which we have not elaborated in our current framework:

Functional types: these describe the argumentation semantics of the user's goal, e.g. questions of comparison, cause and effect, history-sequence, evaluation, justification, time, quantity, identity, attributes, group-membership. Such taxonomies have been proposed in the explanation literature as question/answer categories [32,33]. It is interesting to note that these functional types might be described as patterns, e.g. a comparative need type implies two sub-queries describing each alternative that has to be compared. This line of inquiry is part of our future work.

4. Developing a theory of information searching

To integrate the components of knowledge representation and behaviour we use correspondence rules. These predict how the process models will operate according to a set of preconditions describing the user's knowledge and an information need. As developing a process model that accounts for activities and strategies requires a large number of rules, we have adopted a modular design to minimise undesirable interactions between rules and to facilitate validation.

4.1. Modelling approach

The addition of rules rapidly makes paper based modelling intractable, as it is difficult to check their effect in paper based 'box and arrow' representations of theories. Accordingly, we have implemented the theory in a computational cognitive architecture. This approach has been adopted in HCI; for example, SOAR has been used by Young and Blandford [21] to implement Programmable User Models (PUMS). However, SOAR is a complex system which is difficult to program and more recent PUMS models have dropped this architecture in favour of simpler hand-crafted implementations.

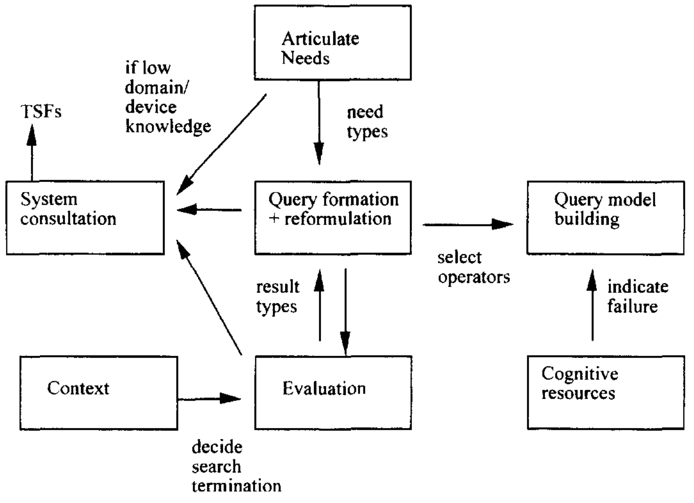

Fig. 3. Model of interaction between the rule sets.

We chose COGENT because it provided an easy to use, yet neutral environment for theory development. Unlike SOAR, COGENT does not provide a problem solving and learning mechanism but we did not requires these features. An improvement over SOAR is provision of configurable constraints on human information processing, such as working memory, selective attention, and spreading activation in long-term memory. These parameters allow plausible cognitive models to be constructed and evaluated. COGENT has a considerable track record of research usage and has demonstrated its effectiveness in the implementation of a wide range of theories [34–36].

4.2. Process model development

Rules governing behaviour within the process model were grouped into six sets. The rules have different preconditions according to the knowledge sources and other inputs they test; similarly, the action clause may either alter a state variable (e.g. update a user's memory), invoke an action, or predict failure if the preconditions are not satisfied. The first three rule sets are linked to activity components in the process model, the forth set consists of system facility consultation rules, which in the implemented model are interleaved with activity rules, while the last two global rule-sets influence process termination and execution of the other strategy selection rules. Fig. 3 illustrates the interaction between the rule sets. As space precludes listing all the rules, an example is given of a rule in each group, and further examples are listed in Appendix A.

Articulation rules trigger Query formation rules by their action clauses. Results from the query trigger evaluation rules, which in turn invoke Query reformulation rules. System consultation and Context rules are tested during the search cycle and indicate either the need for system facilities or search termination.

4.2.1. Articulating needs

This rule set takes an input need type and profile of the user's knowledge and selects the initial strategy. Strategy selection rules choose an optimal strategy given a need type and level of domain knowledge. Rules in this set have the general format:

IF <need type > AND <knowledge level = high/medium/low > THEN Strategy X
with an example:
IF <Need-type = Unknown > AND <Domain-knowledge = high > THEN Refine-concepts

This rule selects expansion of the user's mental model of the query by refining concepts towards search terms. Concepts may be acquired from either the user's long-term memory or facilities provided by the system, i.e. a thesaurus. Describing a need type as unknown means that the target is ambiguous or not clear, rather than completely unknown.

IF <Need-type = Known > AND <Domain-knowledge = low > THEN Reuse Query
IF <Need-type = General > AND <Domain-knowledge = high > THEN Refine concepts
IF <Need-type = General > AND <Domain-knowledge = low > THEN Extract terms, Construct Query (Trial-and-Error-search)

The first of these two rules will trigger a strategy to reuse a previous query from the user's memory, which will be present if the known need has been processed before, otherwise a system consultation rule will be invoked to search the system's reusable query library. The second rule invokes a search for further concepts to refine the general need into a more specific query; this may be achieved by consulting the system thesaurus for additional search concepts and terms. Should the system model not contain a thesaurus TSF, the third rule triggers a guesswork strategy utilising whatever inadequate search terms the user may extract from the problem statement.

4.2.2. Query formation and reformulation

These rules govern strategies to select complex or simple queries, according to the user's device knowledge. Further rules operate at the instance level of query building to make the query structure more, or less, elaborate.

IF <Need-type = simple > THEN Use Keyword Query
IF <Need-type = complex > and <Device-Knowledge = high > THEN Use Boolean query
IF <Need-type = complex > AND <Device-Knowledge = low > THEN Use multiple Keyword Query and Iterate.

Reformulation rules depend on a result type (e.g. Narrow/Broaden query) which is returned from the evaluation activity (see following section). These rules advise on query implementation and can be refined to trigger the appropriate rule according to the query type. For instance, if a query is of type Boolean then the AND/OR rule is triggered.

IF <Narrow-Query > AND <Query-type = text > THEN Add search terms

IF <Narrow-Query > AND <Query-type = text, Boolean > THEN Replace ORs with ANDs
IF <Narrow-Query > AND <Query-type = value > THEN Reduce value range or add values

4.2.3. Evaluation

Evaluation rules determine reformulation strategies by deciding whether to accept the results, or reformulate the query, and if so, what strategy to adopt. Other rules indicate the strategy for processing the results set according to the quantity and relevance of the retrieved results, i.e. the number which is useful to the user as a percentage of all the retrieved results.

IF <Results = Many > AND <Domain knowledge = High > THEN Scan results
IF <Results = Many > AND <Domain knowledge = Low > THEN Select Random sample
IF <Results = Many > AND <Relevance = Low > THEN Narrow-Query
IF <Results = Few > AND <Relevance = Low > THEN Broaden-Query

Assessment of relevance is a complex problem [37,38]. At present we adopt a simple model whereby the user judges the number of relevant items in the current retrieved set, so this is more closely related to precision, rather than the definition of relevance in terms of the useful results present in the database compared to the useful results actually retrieved. The relevance variable can be quantified with user defined values for the Low/high, etc. parameters.

4.2.4. System consultation rules

These are triggered when the user does not possess sufficient knowledge and needs to consult the system for terms, concepts or other facilities. These rules are embedded in the activity rule sets. If the system facility is present the rule succeeds, otherwise it fails and signals a possible problem in completing the task
IF Domain-knowledge = low AND Refine-concepts THEN Consult system thesaurus

4.2.5. Context rules

These are global rules which affect different parts of the search process and express the constraints of time, user motivation and importance of the need on the IR task. Context rules mainly apply to the evaluation activity where they determine whether the process should terminate; however, some also influence initial query formation.

IF Time = short AND Results = Few THEN Terminate
IF Motivation = low AND Results = Few THEN Terminate
IF Need-type = important AND Results = Few THEN Continue

As before, the variables for time, motivation and results can be quantified with user defined parameters.

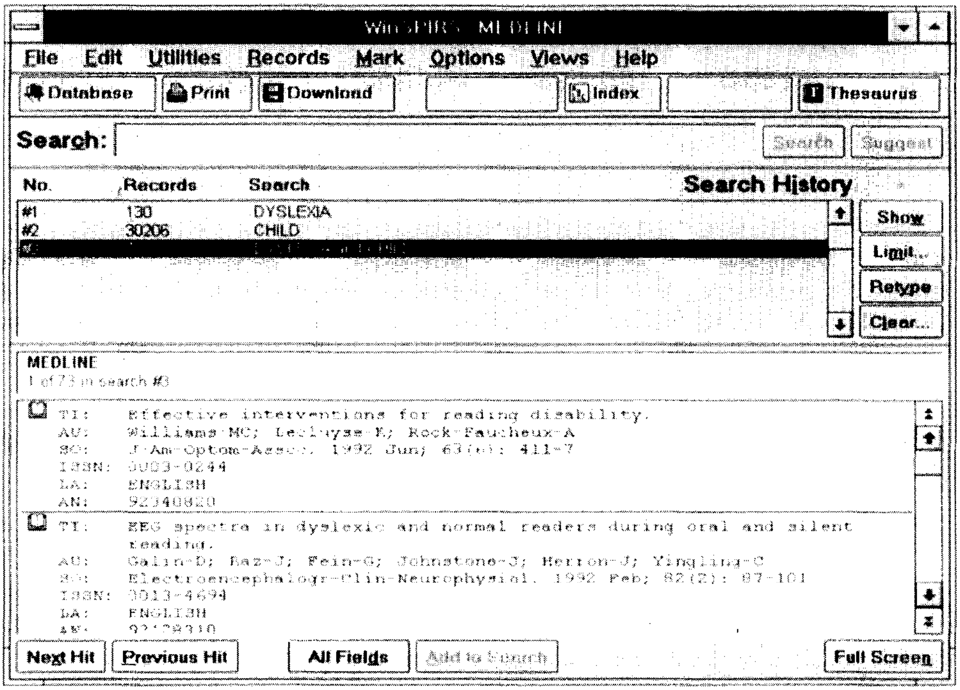

Fig. 4. MEDLINE top level interface.

4.2.6. Cognitive resource rules

These are global rules which are not currently used in the process model implementation. These rules are part of COGENT's built-in features and model limitations on working memory, spreading activation in long-term memory and selective attention. These rules will be employed in the query instance model to specify how errors in human information processing may affect the reliability of query building and modification.

This selection of rules is reported as work in progress to illustrate how the framework will be developed into a more predictive theory. Implementation of the rule set in COGENT is underway and further progress will be reported when the working system has been tested with scenarios drawn from observed user behaviour.

4.3. Evaluating the theory's claims

To test the theory we have conducted experiments of user information seeking with the MEDLINE[1] database. Users were given several information seeking tasks, varied according to our need-type taxonomy. The user interface showing a results summary being displayed during a query session is illustrated in Fig. 4. Think aloud protocols and performance data were recorded to investigate the strategies adopted by users with specific

[1]MEDLINE is the copyright of the US National Library of Medicine and SilverPlatter.

profiles and to see how observed data agree with the theory's predictions. This work is being reported in a following paper (Sutcliffe and Ennis, in preparation). Here we focus on one specific observation taken from these experimental data and use it to assess the theory's claims. If the theory can give plausible accounts for why the observed behaviour should occur then this, at least, validates the theory at the explanatory level.

4.3.1. Claims analysis

The observation is the premature termination problem. We found, as have other researchers, that users frequently terminate searches when they have found only a small number of the available items which could answer their need [38]. Users' judgement of relevance of retrieved data for their information need is complex [13]; however, users are often satisfied with a few retrieved results which are useful for their task, when there are many more potentially useful items in the database. To test the theory the rules are analysed by backward chaining from the observed situation to see what explanations they can provide.

First, there is a motivation explanation, expressed in the rule: IF Motivation = low AND Results = Few THEN Terminate. If the need-type is not seen to be important and the user motivation is not high then accepting a few retrieved results will be adequate. However, there may be other reasons, for instance, poor domain knowledge may hinder assessment of the results, so the user does not know that they are a poor match for a need: IF <Results = Few > AND <Domain knowledge = Low > THEN Accept sample. Note this rule is overridden if the need type is important. Finally, device knowledge may play a part if the search has completed some iterations and the user finds the system complex to use as they possess limited knowledge. This rule expresses the path of least cognitive resistance: IF <Results = Few > AND <Device knowledge = Low > THEN Accept sample. These claims can be investigated by examining protocol evidence of user's reasoning during different stages of the searching task. Furthermore, follow-up interviews can elicit the user's perception of what factors bear upon their decisions. From a preliminary analysis of our experimental evidence we have data to confirm the first two explanations; however, the third looks doubtful. Users appear not be daunted by the device, instead they use compensating strategies of more search iterations to make up for their lack of skill in searching.

4.3.2. Scenario analysis

Another method of validation is to describe an information searching scenario and account for the development of a search by tracing through the process model and strategy rules. As with the claims analysis the scenario is based on observed evidence from the experiments.

The information search task was 'Investigate the increases and decreases in efficiency within the NHS (national health service) since the introduction of formal clinical auditing.' This is a complex information need which was characterised as Known, as it was not a general exploratory search, Fixed as the goal implied specific information about the efficiency impact of clinical audits, and Precise as the areas of investigation were reasonably well described. In functional terms the query involved causation as the effect of introducing clinical auditing (the cause) on NHS efficiency (the result) has to be discovered.

The user was a final year medical student who had good domain knowledge for medicine as a whole, but this not for this area, hence, the user profile was set as Domain Knowledge = medium. Device knowledge was considered to be high as the student was familiar with the MEDLINE system and its Winspirs user interface.

The user proceeded to create the first query with the keywords 'NHS' and 'Clinical Audit' and 'increased or decreased efficiency'. The theory's predictions are that the user should extract terms directly from the query statements as domain knowledge does not provide additional terms necessary from the Refine strategy. The implicated rules are:

IF <Need-type = Known > AND <Domain-knowledge = medium > THEN Extract terms
IF <Need-type = Fixed > AND <Domain-knowledge = medium > THEN Extract terms
IF <Need-type = Precise > AND <Domain-knowledge = medium/low > THEN Extract terms

The rules indicate that the strategy Extract terms will be followed. This might be extracting terms from an external source, such as a written search need as observed in the experiment, or it might be extracting terms from the user's working memory if a search need is spoken. The theory does not as yet distinguish between such starting points. The search was submitted with Boolean operators so we may infer that a complex search strategy had been invoked, following the rule:

IF <Need-type = complex > and <Device-Knowledge = high > THEN Use Boolean query

This part of the scenario begs the question how complex or simple needs are designated. So far we have adopted a pragmatic definition that if a need statement is not a simple find of items belonging to a set (e.g. Identity, Attribute or Group membership function types) then it is considered to be complex. The definition of what is judged to be a complex query also involves pragmatic criteria. Queries containing more that two search terms or queries containing Booleans, or queries containing a constraint clause in relational databases queries (e.g. Find Customers where Age <21) are also rated as complex.

The query returned a moderate number of results (25). The user scanned the results summary (see Fig. 4 for the user interface) which displayed the title, authors and source journal for each document. Scanning was not exhaustive and after viewing the first 10–12 items, the user judged that the retrieved items did not contain the answer to the query. Part of the problem was the over-restrictive term 'NHS' which did not unify with document's indexed terms or contents such as the acronym expansion National Health Service. Suspecting this was the problem the user proceeded to revise the query by adding synonyms 'N.H.S.' and 'Health service' and 'United Kingdom', using general domain knowledge. The theory rules invoked are first to select the strategy to process the results:

IF <Results = Few > AND <Domain knowledge = Medium > THEN View results
IF <Results = Many > AND <Domain knowledge = Low > THEN Scan results

In fact, the users scanned the results summary rather than viewing the contents of the retrieved items. This illustrated the problem of predicting individual user behaviour and setting levels for parameters which may be individually different. When 25 results is set to 'Many' scanning is selected, but when 25 is set to 'Few' viewing is preferred. Although viewing the results might be more effective, users often tend to choose cognitively lazy strategies. For example, they may be able to rely on domain knowledge of authors and journals to judge relevance rather than inspecting abstracts. The following rules indicate the evaluation response:

IF <Results = Few > AND <Relevance = Low > THEN Broaden-Query

This selects a Broadening implementation strategy:

IF <Broaden-Query > AND <Query-type = text > THEN Remove search terms
IF <Broaden-Query > AND <Query-type = text > THEN Add synonyms for search terms
IF <Broaden-Query > AND <Query-type = text, Boolean > THEN Replace ANDs with ORs

Three rules could be applied; in fact, the user chose the second to add synonyms for NHS. Determining exactly which strategy should be applied is beyond the scope of the current theory as it requires more detailed knowledge of the query contents and domain knowledge. Another complexity is that there are two interpretations for addition of the term 'United Kingdom'. The user may have used two strategies in reformulating the query, broadening the NHS components to get more hits, while adding 'United Kingdom' to exclude references to health service reforms in other countries.

IF <Narrow-Query > AND <Query-type = text > THEN Add search terms
IF <Narrow-Query > AND <Query-type = text > THEN Add NOT to exclude unwanted items

Alternatively, adding 'United Kingdom' may be interpreted as adding part of the synonym for national health service. This relies on the user's inference that the National Health Service refers to the United Kingdom NHS. Dealing with selection of mixed strategies is a further challenge to theory development.

Continuing the scenario, the second search iteration produced results confirming the user's expectation. The result set ($N = 34$) was scanned as before; however, the retrieved items still had low relevance, so the broadening strategy continued. The user added a more general term 'reforms' for 'clinical audit', by recruiting domain knowledge that clinical audit was one of the many recent reforms carried out in the UK national health service. This change is modelled by the rule:

IF <Broaden-Query > AND <Query-type = text > THEN Substitute specific terms with more general search terms

However, this is not exactly what the user did. Instead, the original specific term was retained as well, consequently the search produced more results, many of which referred to reforms which did not involve clinical audit. This was a sub-optimal strategy and the user realised the mistake. This illustrates that while the theory does not model errors explicitly it can help analysing sub-optimal strategy selection. The results set ($N = 53$) was scanned as before but given the increased set and the low relevance of many of the items only the first few item summaries were inspected.

IF <Results = Many > AND <Relevance = low > THEN Scan results sample

This time a narrowing strategy was selected.

IF <Results = Many > AND <Relevance = Low > THEN Narrow-Query

The user consulted the system thesaurus

IF <Narrow-Query > AND <Domain-Knowledge = medium > THEN Acquire terms

The forth query added several terms which were related to 'efficiency increases/decreases' 'general standards', 'quality of care', 'Quality assurance', 'performance' and 'effectiveness'. These were included with OR relations. The user also added synonyms for decreased efficiency 'inefficiencies' but interestingly not for increased efficiencies.

IF <Narrow-query > AND <Query-type = text > THEN Add related search terms

The addition of these terms had the desired effect. Some of the retrieved items were relevant to the question, even though the size of overall results set did not change dramatically ($N = 24$), because disjunctions had been used in the query. Again a hybrid strategy had been employed, the user wished to narrow the query by adding terms but wanted to avoid being over-restrictive with conjunctions. The theory predicts the strategy selection in general terms; however, human experts employ subtle combinations of strategies within one query cycle.

The final results set was scanned and a sample of the results inspected by reading the article abstracts to verify that they were relevant to the need. At this point the user terminated the search having selected five relevant articles, even though some of the retrieved results were not inspected in detail and could have been useful. Furthermore, the relevant, retrieved results as a percentage of those potentially available in MEDLINE was small (2.5%). This demonstrates a satisfying strategy by the user who found some articles which could answer the search need, but the search was not exhaustive. The theory rules which suggest such behaviour are:

IF <Relevance = high > AND <Results = Many > THEN View results sample
IF <Search-cycle > 3 > THEN
IF <Motivation = medium > AND <Results = Relevant > THEN Terminate
IF <Need-type importance = low > AND <Results = Relevant > THEN Terminate

Although we rewarded our subjects for their participation, we suspect their motivation was not as high as it might be in an examination or if a professional life-saving judgement depended on the search outcome. Even so the subject had continued with a search which took 15 minutes, and had persisted until an adequate if not optimal answer had been found. The choice of three search cycles as a termination threshold for low motivation is an arbitrary figure which we can modify in light of further empirical evidence.

In conclusion, the theory does explain strategy selection during the process of information searching but it only does so at a general level. It cannot account for hybrid strategies and only partially for sub-optimal performance. We now turn to the final topic, how the modelling framework and the implemented theory might be used in user interface design to support information retrieval.

5. Utilising the theory in interface design

Three possible applications are proposed. First, a tutoring or intelligent help system

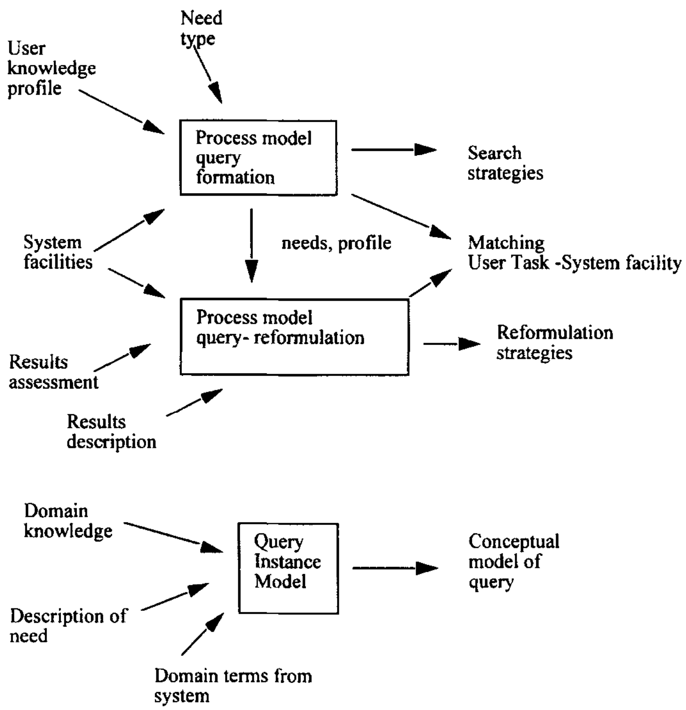

Fig. 5. Process and query-instance model inputs and outputs.

could be developed from the process model and its rule set. Second, a walkthrough evaluation method could be used to test the utility of IR systems according to task support factors. Finally, the process model may have potential as a decision support system for configurable information retrieval systems, or as an intelligent, adaptable user interface.

5.1. Intelligent help for information searching

The rules for query formation and reformulation can be developed as an intelligent help/ strategy advisor for novice searchers. We have developed the prototype for this application in the Intelligent Strategy Planner (ISP) within the INTUITIVE system [39]. The ISP advised users on query broadening and narrowing according to the number of retrieved results. Rules triggered advice depending on a user defined parameter for a too-many/too-few results boundary. The rules were also sensitive to media type so whereas many relational database records may be acceptable the same number of videos would not be. An alternative interface allowed the user to view the actual number of retrieved results and then inform the system of their judgement by too-many/too-few buttons. The system presented advice as text explanations of narrowing or broadening strategies and then highlighted the appropriate part of the query to alter. Hence, if broadening was required and the query contained ANDs these where highlighted with the suggestion to change them to ORs.

Preliminary evaluation of this facility indicated that users found the advice practical and effective [39]. The rules in the theory have evolved from the INTUITIVE rule set. Since they account for the level of the user's knowledge, advice can be tailored to a user profile.

5.2. Walkthrough evaluation method

The process model and strategy selection lends itself to development of an evaluation walkthrough method, albeit at a general level of assessing information retrieval system functionality. User profiles and need types can be used to step through the process model to investigate which task support facilities should be present to help users complete each stage of the information searching task. The inputs to the current model and its outputs are shown in Fig. 5. If the type of need, number of results, and a user assessment of their relevance can be fed into the model, then the system can advise useful strategies to follow and the necessary system support facilities for each stage. The output may be expressed as questions for the evaluator which explore the adequacy of each facility, e.g. is a thesaurus present, does it cover the user domain in sufficient depth, is it accessible, easy to use, etc.

When the need for a TSF is identified then the actual system can be inspected with standard usability methods (e.g. [40,41]) to ascertain whether the TSF can be found easily by the user, provides appropriate functionality and gives adequate feedback.

5.3. Configurable information retrieval systems

As information searching is becoming more complex with the spread of the world-wide web, computer support for information searching will need to become more adaptable to the user's need and context. Furthermore, as individual search styles are inconsistent even when the need and system are the same [13], customised systems will be necessary to fit individual needs. The implemented theory could be developed into a configuration advisor. The system could be provided with a user profile and set of search needs from which it could indicate the necessary TSFs that the system should provide. These could be recruited from a library of IR services which support different aspects of the user's task.

6. Discussion

The modelling framework we report provides a means of scoping the information searching problem. While there are many aspects we have not covered, in particular the problem of how the user's conceptual model of a query may develop, the framework does elaborate previous models of the information searching problem (e.g. [3]). The importance of the user's knowledge of the search system and domain knowledge and how these influence searching behaviour has been pointed out by Allen [14,15]. The difference between information searches which are exploratory versus specific has been recognised in several theories, although more emphasis has been placed on exploratory, evolving needs (see [1,2,9]). Ingwersen [3] goes further by describing need types in a two-dimensional taxonomy of known/unknown versus fixed/variable goals. We have extended this approach to account for different types of database search and pointed towards more elaborate taxonomies that account for the semantic contents of the user's goal. Unfortunately, categorising need types is not easy. Furthermore, the user's need type may change during the search process, so it remains to be seen how reliably correlations of user behaviour and search needs will be.

Progressing from a taxonomy to a cognitive IR theory breaks new ground, as far as we are aware. Our activities can be mapped onto Kuhlthau's process model [10] and Marchionini's framework [42]; however, elaborating the process model with strategies and rules provides a much richer account of behaviour. While this is an ambitious undertaking, and the current theory is preliminary and inadequate to deal with several aspects of observable user behaviour, it is at least a starting point. The preliminary validation reported in this paper demonstrates that the theory has explanatory power, although it can only account for how expert strategies should be selected for a particular context. Predictions of actual user behaviour are beyond the scope of the current theory and may remain so. User searching strategies may be inconsistent, as reported by Iivonen [13], for several reasons. First, searches may be based on memory. Individuals may adopt strategies which worked in the past even if they were sub-optimal; moreover, different search styles may be used as a consequence of experience with particular types of IR system [14]. Even though the prognosis for predicting individual behaviour is poor, as this would require acquiring more knowledge of the individual user, much can be achieved by modelling users as generic groups.

The cognitive theory we propose has the potential to provide predictions of expert strategies for a particular search need. In this mode it could act as an expert system advisor which encapsulates human expert intermediary knowledge. Such approaches have been explored in intelligent information retrieval [7], although by employing domain specific knowledge as well as generic strategies. Our approach has also modelled novice user needs for system support, so it can suggest how novice users may be helped in two ways. First by suggesting expert searching strategies and secondly by indicating the system facilities to consult at each stage in the search process. This may be developed towards configuration support for different types of user.

There are several outstanding problems with the model. It relies on interpretation of categories such as 'complex', 'known needs' by the model builder. Although heuristics are proposed to help the categorisation of needs, these inevitably rely on judgement in their interpretation. Similarly, the operation of the model depends on judgements of relevance during the evaluation phase. Here the model builder has to make a reasonable guess as to what the user would consider to be relevant in the retrieved results. Alternatively, judgements could be obtained from the user directly, but assessment of relevance is idiosyncratic in many cases and often reflects changes in the user's need [38]. In spite of this limitation, given that relevance judgements can be obtained, the model will suggest strategies for either accepting the results or reformulating queries.

From the HCI perspective the cognitive task model of information searching is a higher level theory than most of the models proposed in previous work [20,21]. The models we report describe tasks at a higher level of granularity than the detailed models in GOMS and model human processor theories [23,24]. We believe that high level models will be useful in design because they will address a wider range of phenomena than previous HCI theories, although the connection between theory and design will not be direct. For instance, the TSFs we propose will have to be translated into implications for design in a specific IR system. In some sense the strategies are similar to an expert system modelling any stable body of human expertise. So, even if the predictive power of the theory is limited, the process model and its strategies represent a repository of expert intermediary knowledge which can be refined with testing. The process model predicts behaviour at a high level of granularity, as do the process models of risk related decision process reported by Fox and Das [43]. The query instance model, which we have not elaborated as yet, is related to argumentation models in medical decision making which can predict instance level decisions within a domain [43]. These models have also been implemented in COGENT and rely on scenario-based testing for their validation and improvement.

In out future work we will improve the theory by further scenario-based testing. This is a slow process with paper-based models, but once the model implementation is complete, scenario testing is rapid. This will allow rapid debugging of the rule-sets, and validation against a wide range of scenarios drawn from experimental data. The process of formalising the theory within rule sets collates current knowledge from empirical studies and other theories. As such, the theory we report represents at a minimum a repository of knowledge about information searching tasks, even though it is inadequate in its current version.

Acknowledgements

Mark Ennis is supported by an EPSRC postgraduate studentship under grant number 95306422. The research was partially support by ESRC Cognitive Engineering project ‘Modelling information strategies and resources’.

Appendix A. Rule-sets

Appendix A.1. Procedures

Procedures are triggered by strategy selection rules and are subsequently used as preconditions which are tested by further rules that either determine user action or use of a system facility.

Construct query: build a query with terms and syntactic structure. Execute query by submitting it to a search system.
Acquire terms: find terms to construct a query or to refine higher level concepts.
Extract terms (from requirement): take terms directly from the problem statement either in a text or from a spoken need.
Refine-concepts: decompose or refine higher level concepts into terms.
Consult-system thesaurus/index: search for terms in a thesaurus or classification (indexing) structures.
Explore information sources: investigate the contents of available information sources, e.g. different databases or information providers.
Browse: search in an exploratory manner.

Scan results summary: scan titles of retrieved documents, may be carried out systematically or randomly.
Sample results: view contents of some of the retrieved results, may be carried out randomly or systematically.
View results: inspect content of retrieved items.

Appendix A.2. Articulation rules

These rules take a need type and user knowledge profile as input and select a strategy.

IF <Need-type = Known > AND <Domain-knowledge = high > THEN Reuse Query
IF <Need-type = Unknown > AND <Domain-knowledge = high > THEN Refine-concepts
IF <Need-type = Fixed > AND <Domain-knowledge = high > THEN Extract terms from requirement and Refine-concepts
IF <Need-type = Variable > AND <Domain-Knowledge = high > THEN Explore Information Sources
IF <Need-type = Precise > AND <Domain-knowledge = high > THEN Extract terms from requirement, Construct Query
IF <Need-type = General > AND <Domain-knowledge = high > THEN Refine concepts
IF <Need-type = Simple > AND <Domain-Knowledge = high > THEN Construct Query
IF <Need-type = Complex > AND <Domain-Knowledge = high > THEN Decompose Info Need
IF <Need-type = Known > AND <Domain-knowledge = low > THEN Reuse Query
IF <Need-type = Unknown > AND <Domain-knowledge = low > THEN Explore Information Source, Browse
IF <Need-type = Fixed > AND <Domain-knowledge = low > THEN Extract terms from requirement and Acquire terms
IF <Need-type = Variable > AND <Domain-knowledge = low > THEN Browse
IF <Need-type = Precise > AND <Domain-knowledge = low > THEM Extract terms, Construct query

Appendix A.3. Consultation rules

These rules trigger use of the system facilities when the user has low domain or device knowledge.

IF <Acquire-terms > AND <Domain-knowledge = low > THEN Consult-system thesaurus
IF <Refine-concepts > AND <Domain-knowledge = low > THEN Consult System Thesaurus

IF <Refine-concepts > AND <Domain-knowledge = low > THEN Consult System Index
IF <Construct-Query > AND <Device-knowledge = low > THEN Use Query-syntax Editor
IF < Construt-Query > AND <Device-knowledge = low > THEN Consult Query-construction help
IF <Reuse-Query > AND <Domain knowledge = low > THEN Search Query Library

Appendix A.4. Query formation and re-formulation

These rules govern first strategies to be selected complex or simple queries, according to the user's device knowledge. Further rules operate at the instance level of query building to make the query structure more, or less, elaborate. This contains knowledge of query languages necessary for constructing complex queries with Booleans, etc.:

IF <Need-type = Simple > THEN Use Keyword Query
IF <Need-type = Complex > and <Device-Knowledge = high > THEN Use Boolean Query
IF <Need-type = Complex > AND <Device-Knowledge = low > THEN Use Multiple Keyword Query and Iterate.

Reformulation rules depend on a result type which is returned from the evaluation activity (see following section). These rules advise on query implementation and can be refined to triggered the appropriate rule according to the query type. For instance, if a query is of type Boolean then the AND/OR rule is triggered.

IF <Narrow-Query > AND <Query-type = text > THEN Add search terms
IF <Narrow-Query > AND <Query-type = text > THEN Remove synonyms for search terms
IF <Narrow-Query > AND <Query-type = text > THEN Add NOT to exclude unwanted items
IF <Narrow-query > AND <Query-type = text > THEN Add related search terms
IF <Narrow-Query > AND <Query-type = text > THEN Replace ORs with ANDs
IF <Narrow-Query > AND <Query-type = value > THEN Reduce value range or add values
IF <Narrow-Query-Specificity > THEN Replace General terms with Specific
IF <Broaden-Query > AND <Query-type = text > THEN Remove search terms
IF <Broaden-Query > AND <Query-type = text > THEN Add synonyms for search terms
IF <Broaden-query > AND <Query-type = text > THEN Remove related search terms
IF <Broaden-Query > AND <Query-type = text > THEN Replace ANDs with Ors
IF <Broaden-Query-Specificity > THEN Replace Specific terms with General
IF <Change-Query > AND <Query-type = text > THEN Replace terms

Appendix A.5. Evaluation

The evaluation activity rules determine reformulation strategies by deciding whether to accept the results, or reformulate the query and, if so, what strategy to adopt. Other rules indicate the strategy the process is to follow to deal with the results set according to the quantity of retrieved results and precision of the retrieved results, i.e. the number which is useful to the user as a percentage of all the ones retrieved.

IF <Results = Many > AND <Domain knowledge = High > THEN Scan results
IF <Results = Many > AND <Relevance = low > THEN Scan results sample
IF <Results = Many > AND <Relevance = high > THEN View results sample
IF <Results = Many > AND <Domain knowledge = Low > THEN Select Random sample
IF <Results = Many > AND <Relevance = Low > THEN Narrow-Query
IF <Results = Many > AND <Granularity = Too high > THEN Narrow-Query-Specificity
IF <Results = Many > AND <Relevance = Low > THEN Change-Query
IF <Results = Few > AND <Relevance = Low > THEN Broaden-Query
IF <Results = Many > AND <Granularity = Too low > THEN Broaden-Query-Specificity
IF <Results = Null > THEN Broaden-Query
IF <Results = Null > OR <Results = Few > AND Query-already reformed THEN Change-Query
IF <Results = Few > AND <Device knowledge = Low > THEN Accept sample
IF <Results = Few > AND <Domain knowledge = Low > THEN Accept sample
IF <Results = Many > AND <Relevance = high > THEN Accept

Appendix A.6. Context rules

These are global rules which affect different parts of the retrieval process and express constraints of time, user motivation and importance of the need on the IR task. Context rules mainly apply to the evaluation activity where they determine whether the process should terminate; however, some also influence initial query formation.

IF Time = short AND Results = Few THEN Terminate
IF Motivation = low AND Results = Few THEN Terminate
IF Need-type importance = high AND Results = Few THEN Continue
IF Motivation = low AND Results = Relevant THEN Terminate
IF Need-type importance = low AND Results = Relevant THEN Terminate

References

[1] N.J. Belkin, Information concepts for information science, Journal of Documentation 34 (1987) 55–85.
[2] P. Ingwersen, The cognitive viewpoint in IR, Journal of Documentation 48 (1) (1993) 60–64.

[3] P. Ingwersen, Cognitive perspectives of information retrieval, interaction elements of a cognitive IR theory, Journal of Documentation 52 (1) (1996) 3–50.
[4] K. Markey, P. Atherton, ONTAP: online training and practice manual for ERIC database (ED106109), Syracuse University, ERIC Clearing House on Information Research, Syracuse, NY, 1978.
[5] C.L. Borgman, The users mental model of an information retrieval system: an experiment on a prototype on-line catalogue, International Journal of Man–Machine Studies 24 (1985) 47–64.
[6] C. Van Rijsbergen A new theoretical framework for Information retrieval, in: Proceedings of ACM SIG-IR Conference, Pisa, Italy, 1986, pp. 194–200.
[7] W.B. Croft, Approaches to intelligent information retrieval, Information Processing and Management 23 (4) (1987) 249–254.
[8] M.J. Bates, Information search tactics, Journal of the American Society for Information Science 30 (4) (1979) 205–214.
[9] M.J. Bates, The design of browsing and berrypicking techniques for the on-line interface, On-line Review 13 (5) (1989) 407–424.
[10] C. Kuhlthau, Longditudinal case studies of the information search process of users in libraries, Library and Information Science Research 10 (3) (1998) 257–304.
[11] N. Belkin, C, Cool, A. Stein, U. Thiel, Scripts for information seeking strategies, in: Case-Based Reasoning and Information Retrieval—Exploring the Opportunities for Technology Sharing. Papers from the AAAI Spring Symposium Series, Technical Report SS-93-07, 1993, pp. 8–17.
[12] D. Davidson, The effect of individual difference of cognitive style on judgments of document relevance, Journal of the American Society for Information Science 28 (5) (1977) 273–284.
[13] M. Iivonen, Searchers and searchers: differences between the most and least consistent searchers, in: Proceedings of the 18th ACM-SIGIR conference, Seattle, ACM Press, 1995.
[14] B.L. Allen, Cognitive research in information science: implications for design, Annual Review of Information Science and Technology 26 (1991) 3–37.
[15] B.L. Allen, Perceptual speed, learning and information retrieval performance, in: Proceedings of the 17th SIGIR Conference, Dublin, ACM Press, 1994, pp. 71–80.
[16] A. Large, J. Besheshti, A. Breuleux, A. Renaud, A comparison of information retrieval from print and CD-ROM versions of an encyclopedia by elementary school students, Information Processing and Management 30 (4) (1994) 499–513.
[17] D.A. Michel, What is used during cognitive processing in information retrieval and library searching? Eleven sources of search information, Journal of American Society for Information Science 45 (7) (1994) 498–514.
[18] T. Jacobson, D. Fusani, Computer, system and subject knowledge in novice searching of a full text multifile database, Library and Information Science Research 14 (1) (1992) 97–106.
[19] W. Rouse, S. Rouse, Human information seeking and design of information systems, Information Processing and Management 20 (1-2) (1984) 129–138.
[20] P. Barnard, Bridging between basic theories and the artefacts of Human Computer Interaction, in: J.M. Carroll (Ed.), Designing Interaction, Psychology at the Human Computer Interface, Cambridge University Press, 1991.
[21] R.M. Young, A.E. Blandford, The state of PUMs: a revised manifesto, AMODEUS Report UM/WP26, June 1994.
[22] V. Bellotti, Integrating theoreticians and practitioners' perspectives with design rationale, in: S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, T. White (Eds.), Proceedings of INTERCHI-93, ACM Press, 1993 pp. 101–106.
[23] D.E. Kieras, S.D. Wood, D.E. Meyer, Predictive engineering models using the EPIC architecture for a high performance task, in: Proceedings of CHI-95, Human Factors in Computing Systems, ACM Press, 1995.
[24] B.E. John, D.E. Kieras, Using GOMS for user interface design, which technique, ACM Transactions of Computer Human Interaction 3 (4) (1996) 287–320.
[25] D.A. Norman, Cognitive engineering, in: User Centred System Design: New Perspectives on Human-Computer Interaction, D.A. Norman, S.W. Draper (Eds.), Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.
[26] J. Rasmussen, Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human

performance model, IEEE Transactions on Systems, Man and Cybernetics, SMC-13(30) (1983) 257–266.
[27] C. Lewis, P. Polson, C. Wharton, J. Reiman, Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces, in: J.R. Chew, J. Whiteside (Eds.), Proceedings CHI-90, ACM Press, 1990, pp. 235–241.
[28] J.M. Carroll, W.A. Kellogg, M.B. Rosson, The task–artefact cycle, in: J.M. Carroll (Ed.), Designing Interaction, Psychology at the Human Computer Interface, Cambridge University Press, 1991.
[29] L.A. Suchman, Plans and Situated Action, the Problem of Human–Machine Communication, Cambridge University Press, 1987.
[30] M. Hancock-Beaulieu, Query expansion: advances in research in on-line catalogues, Journal of Information Science 18 (1992) 99–103.
[31] S.E. Robertson, M. Hancock-Beaulieu, On the evaluation of IR systems, Information Processing and Management 28 (4) (1994) 457–466.
[32] E. Sarantinos, P. Johnson, Explanation dialogues: a theory of how experts provide explanations to novices and partial experts, Research Report, Department of Computer Science, Queen Mary Westfield College, University of London, 1992.
[33] M.T. Maybury, Custom explanation: exploiting user models to plan multi-sentential text, in: Proceedings of the Second International Workshop on User Modelling, Hawaii, USA, 1990.
[34] J. Fox, R. Cooper, J. Farringdon, T. Shallice, Building computational models of cognition, in: B. Silverman (Ed.), Proceedings of the Workshop on Expert Judgement, Human Error and Intelligent Systems, 10th European Conference on Artificial Intelligence, Vienna, 1992.
[35] R. Cooper, T. Shallice, Soar and the case for unified theories of cognition, Cognition 55 (2) (1995) 115–149.
[36] R. Cooper, J. Fox, J. Farringdon, T. Shallice, A systematic methodology for cognitive modelling, Artificial Intelligence 85 (1996) 3–44.
[37] L. Su, The relevance of recall and precision in user evaluation, Journal of The American Society for Information Science 45 (1994) 207–217.
[38] C.L. Barry, User defined relevance criteria: an exploratory study , Journal of the American Society for Information Science 45 (3) (1994) 149–159.
[39] A.G. Sutcliffe, I. Bennett, A. Doubleday, M. Ryan, Designing query support for multiple databases, in: D. Gilmour (Ed.), Proceedings of INTERACT-95, Lillehammer, Norway, 1995.
[40] J. Neilsen, Usability Engineering, Academic Press, 1993.
[41] A.G. Sutcliffe, M.V. Springett, From user's problems to design errors: linking evaluation to improving design practice, in: A. Monk, D. Diaper, M.D. Harrison (Eds.), People and Computers VII, Cambridge University Press, 1992, pp. 117–135.
[42] G. Marchionini, Information Seeking in Electronic Environments, Cambridge University Press, 1995.
[43] J. Fox, S.K. Das, A unified framework for hypothetical and practical reasoning: lessons from medicine, in: D. Gabbay, T. Hunter (Eds.), Proceedings of the International Conference on Formal and Applied Practical Reasoning, Berlin, Springer, 1996.