Fashionable trends and feasible strategies in information management

Karen Sparck Jones
Computer Laboratory, University of Cambridge
New Museums Site, Pembroke Street, Cambridge CB2 3QG, UK

The final form of this paper, based in an invited talk at *RIAO 88*
appeared in *Information Processing and Management*, 24, 1988, 703-711.

Abstract

This paper analyses current trends in information management, considers the problems they involve, and suggests some strategies for tackling these problems. The current goal is integrated, personalised information systems, to be reached via artificial intelligence. The argument is that the extent to which this goal can be achieved is limited because these systems are intrinsically heterogeneous, are for access to information, and deal in linguistically-expressed information; so the best strategy for building the systems that can be attained is via linguistically-oriented knowledge and inference. Evaluating these systems also presents problems because each use is unique, but evaluation is much needed and large-sample strategies for performance study can be devised.

In this paper I shall look at current trends in information management in an attempt to identify the real problems to be tackled and to suggest appropriate strategies for solving them.

1 Trends

There are currently three clear trends in information management.

1. The first is the development of 'multi-faceted' information systems. These involve very different types of information object, and of object use. The user manipulates a range of document types, for example tables, records, messages, letters, reports, or papers, carrying out a range of operations on these, for instance producing, modifying, annotating, describing, routeing, seeking, or scanning them. To support him in these activities he relies on second-order object types like indexes, directories, menus, and dictionaries.

In these systems the user moves freely from one type of object or activity to another; for example when writing a paper, the user may interrupt this to search a bibliographic file, and in turn interrupt his search to look at a thesaurus for other search words, or to send someone else an inquiry message. These systems are also characterised by having many users of common information resources, both in reading mode, as in consulting a common database, and in writing mode, as in modifying a shared schedule. They are further characterised by providing information resources for the individual user which depend on a larger information system, as with mail files, or bibliographic support for SDI services.

2. The second trend is the use of sophisticated terminal interfaces with different forms of display facility, multiple windows, etc, which allow linked and parallel operations in a

very flexible and convenient way. So, for example, in working on a paper, the user can simultaneously construct a diagram, amend a checklist, refer to displayed quotations, and so forth.

This trend is clearly linked with the first since sophisticated interactive facilities are needed to support the efficient use of a mixture of objects in a mixture of ways for a mixture of purposes.

3. The third trend, though in this case more one of aim than actuality, is the application of artificial intelligence. This is not just the adoption of AI techniques, e.g. in the use of frames for representation, or the introduction of AI processes, as in interpreting natural language sentences for indexing. Substantively applying AI means treating the system's information management as an activity requiring a knowledge base constituting a model of the world and inference procedures for exploiting this knowledge. For example a medical community information system would have on the one hand characterisations of diseases, diagnoses, and therapies, along with general biological information, and on the other characterisations of the people - patients and doctors - involved, the facilities available for treatment and so on.

The idea is that the effective interpretation of e.g. patient records, or accounts of research experiments, depends on a body of knowledge about medicine; and that if these resources are to be related as they need to be, for instance to support a clinician, this must depend on having a common body of knowledge whose use essentially involves inference processes. Thus, for example, the system may reason from the description 'patient has Snodgrass disease' extracted from a record to the description of a research paper referring to drug tests for breathing problems, via a knowledge of the properties of Snodgrass disease, in order to offer the doctor papers suggestive of a treatment for a patient with Snodgrass disease. This reasoning on knowledge is relevant not only to one-off background operations, for example indexing requests or documents for what they are about, but also to continuing foreground activities, for instance adapting to the user's continually changing needs as the system presents him with answers to successive questions during a consultation session.

This trend is related to two previous ones because it is motivated by the belief that terminal technology is inadequate in itself for complex information management, but that relying on the user as in 'outlining', or searching a common bibliographic database, places too great a burden on him.

2 Underlying assumptions

There are plenty of problems with the heady vision of intelligent information systems supporting the individual user, for example those of providing suitable inference capabilities for common sense reasoning. But I believe there are less obvious ones that are much more critical because of the tacit assumptions on which the vision rests. These assumptions are:

1) that we can have single integrated information systems, for example a system supporting alike the use of patient admission records by hospital administrators and the doctors' use of the research literature;

2) that we can have embedded personal information subsystems, for example have the doctor's own patient notes with annotations relating to his research interests, organised un-

der his research in progress headings and with links to his bibliography labelled 'evidence for X' or 'evidence against X', as a proper subpart of a general records library system;

3) that to achieve each of these reasoning on deep knowledge is required, for example that because a human has lungs a patient record is assigned to a chest disease clinic or data on patient findings are linked to the doctor's bibliographic information on unusual pulmonary diseases; and

4) that this reasoning has to be done within a unified framework, that is that the common fact that people have lungs is needed for both the administrator's and the doctor's purposes to be achieved.

3 Problems

The critical problems in all of this are of two kinds. There are problems

(a) of what we are trying to do, and
(b) of knowing whether we are succeeding in doing it.

3.1 (a) Problems with aims

There are several interacting problems here, all to do with heterogeneity as a fundamental and not just superficial property of these systens, and as one directly bearing on the user. The idea of an integrated information system is that we have information objects, i.e. objects that contain, embody, or convey information, and that these are manipulated in a way that depends on this to a greater or lesser degree, but always to some degree. For example I may say that I am writing a paper about user modelling, or alternatively that I am writing a paper for a language generation workshop; but even the label 'paper for generation workshop' is saying something about the paper that is meaningful to me in relation to its information content, as when I invoke it via the label to alter it, i.e. modify its content. But this simple account exploiting the common word "information" to suggest homogeneity is not really adequate. What we really have is a pervasive heterogeneity.

1) Thus we are dealing with objects of very different kinds, not just physically as with pictures and texts, but logically: pictures do not really have messages in the way texts do, and even with texts, for example a book, a news story, an acknowledgement signal, a purchase order, or a database entry, we have very different sorts of message. It is far from clear we can think of characterising all of these in the same sort of way to access the information they contain. This is even less clear when we consider second-order objects that we also need to access as first-order objects, for instance a table of contents or a technical thesaurus.

The presumption underlying the idea of an integrated system is that the user can access information as something independent of what it is or what it is for. But can one say that a news story and a thesaurus entry, for example, are about something, or contain information, in the same sort of way?

2) We have large differences of grain, both at the level of individual objects, as is evident when we compare, for example, a dictionary entry with a scientific article; and at the level of

the sets or groups of objects we also have to treat as units in information processing activities: for instance my today's mail, my meetings diary, my bibliography, the AI community bulletin board, the company sales database, the OCLC catalogue, the CA file. It is very unobvious that comparable or connectible levels of description are appropriate where the grain size is so different.

The presumption about personalisation within a larger information environment is that the embedded and host systems can be connected through a common descriptive apparatus. But labels suited to a few personal letters, for instance, are not necessarily right for an organisation, for either meaning or scaling up reasons: in the larger context labels that mean something to the individual may be meaningless or confusing, or not be sufficiently discriminating.

3) we are involved in different uses of information imposing different functional perspectives on it, not only as in, e.g., wanting a paper for quite distinct reasons, but as in e.g. writing a paper, editing a paper, indexing a paper, refereeing a paper, or seeking a paper.

The presumption is that because these are all information management activities, they require the same sort of information characterisation as a support. But a scenario for writing an intended paper is not necessarily much like a description for retrieving it: thus a scenario may consist of a list of headings in proposed text order, along with some notes to emphasise X, check Y, and link with Z, but an instruction 'check date' is not a plausible retrieval tag.

4) We are meeting quite different sorts of relevance need, for example extracting a personal record from a staff file to show a salary versus offering an abstract on a topic.

The presumption is that if searching is for something called information then we also have something called appositeness. But a database tuple matching a query, for instance, is not meeting a user need is the way a retrieved abstract text is.

This heterogeneity of things and their uses suggests there may be a real challenge in going beyond a merely aggregate system to a truly integrated one so, for example, instead of editing a document invoked as 'KSJ/DOC.3' and then, to get an illustration, pending this and invoking a trade statistics database with 'exports AND 1984 AND ...', I can just go: edit text 'import- export relations', find graph 'import-export relations'. Again, when we take the multiplicity of system users into account, there may be a similar challenge in achieving real personalisation so that, for example, instead of editing my document invoked with 'import-export relations' and, after scanning Joe's directory, getting Joe's document labelled 'new ideas on the balance of payments decline', I can just go: edit KSJ 'import-export relations', get JS 'import-export relations'.

Essentially in all of this we are up against two fundamental properties of any information system worthy of the name.

1) Its intrinsic indeterminacy: the same object can be described in different ways, i.e. the same piece of information can be differently described so that, for example, a document and a request, or two documents, do not match (as, inversely, the same description can be interpreted in different ways).

2) Its built-in private/public conflict: the way I think is not necessarily the way others think, but I want to be able to exploit information resources generated by and accessible to other people, i.e. a privately valid description may not be a publicly valid one.

4 Strategies

These are the problems for any attempt to build integrated, personalised information systems. Turning now to possible strategies for building them, can we, even if we allow for the reasons given for some limits to effectiveness, look to artificial intelligence, i.e. to reasoning on deep knowledge, to provide these systems? Do we need reasoning on knowledge of the world as on the one hand embodied in or referenced by documents, and as on the other expressed or presupposed by the user's inputs, to supply him with appropriate information?

In my view the answer is no; that is no if AI in the system has the same function as the human intermediary in many information-seeking environments has. This is because that function is not to supply information, but to supply *access* to information. There is no reason (even if it was feasible) for the system to replicate all the primary information contained in its objects that is the user's concern. This would still leave the user to find his way around. But as soon as the system provides effective means of access these are necessarily only limited leads into the information, and the user has to consider and evaluate this himself. The intermediary may function as a primary source for particular kinds of information, as in a question- answering system, but this is different: he is no longer an intermediary for that information, though for the user that information may be only a means to a further end.

The system's role is also limited from the other side, in that the intermediary is only acting on behalf of the user, and cannot be the user. That is the intermediary may try to simulate the user to find good access paths to information, but he can necessarily only do so very partially; the system does not, and cannot, have all the experience and thoughts of the user. In any case, even if the system could fully simulate the user, it would still have to rely on other components with intermediary functions. (Insofar as a user of a system is an information object for other users, as in a mail system, he is accessible to the system only by having a representation in a form the system recognises, either of an explicit sort like a mail identifier, or of the implicit sort provided by his actions. What other users independently know about a user is immaterial from the system's point of view.)

I am not arguing that reasoning on deep world knowledge has no role in an information system. We can in principle exploit AI- driven functional processors for specific tasks, like managing a group diary, or building a user model to support bibliographic access.

What I am arguing is that because the system has to deal with (first or second order) objects in which information is expressed in language and, further, in natural, and in public, language, processing has to be strongly tied to language. So it will in a material sense be processing using shallow rather than deep world knowledge. The system is necessarily concerned with *linguistic* information objects, i.e. with the case where information contained in, is information expressed by, these objects. Providing access to the information content of these objects implies taking account of the way it is expressed and hence, insofar as access to the objects can be facilitated by reasoning on knowledge bases, it will be on bases representing the way people talk about the world rather than the way it really is. Indeed the way people talk about it is the way it is.

This holds for processing bearing on any one type of information object or use. It applies even more to relating different types of object or use, and to supporting transitions between them. The heterogeneity of objects and uses is reflected in the existence of 'sublanguages' in different parts of the system, but these will be linkable for the same reasons and in the same way that natural languages are, through common elements in word and sentence meanings. This applies both to the variants of the same natural language used by different people or communities and to different natural languages.

Integration is thus achieved through linguistic linking, connecting one word or expression with another related one, and so is personalisation. In the first case the link is between pairs of information items or classes within the system, in the second the connection is between pairs where one side belongs to the system and the other to the individual user.

To promote effective information systems we have therefore to focus on methods of exploiting linguistic relationships, which may be complex as well as simple, as indicators of underlying referential relationships. This is where we have to look for the connectivity we are seeking.

5 Illustrations

I shall use two examples to illustrate the kind of thing I believe is required.

Example 1

The first of these is very simple, and is intended only to show how linguistic relationships can serve as a means of integrating and personalising a heterogeneous system. Menunet (Brooks and Sparck Jones 1985) was designed to deal with the limited version of the full case that is represented by a comprehensive set of conventional office utilities accessed and 1used via menus, where the user wants to be able to move conveniently from one menu to another without the tedious vertical tracking normally required with a large menu tree. The user should be able to jump across the tree, and specifically should be able to do this without having to know the name of the menu option he requires: in a large system there is no reason to suppose the user will remember the option name. At the same time, something more sophisticated than the usual help facility would be an advantage.

For instance, imagine the user has been in the word-processing utility and wants to send the document on which he has been working as a mail item. He would like to be able to move directly to the relevant point in the mail system, but cannot remember the relevant mail option name for the function 'send'.

The idea of Menunet is that relations between individual menu names for particular activities, i.e. between these linguistic objects, are established by synonymic or other cross references that can be dynamically exploited for ad hoc index menus which provide access to the menu options concerned.

Thus Figure 1 (a) shows some excerpts from menus connected with different utilities which each include options for some sort of sending activity, called "store" in the File menu, "send" in the Diary menu, and so forth. The different options for each menu have an associated list of keywords acting as alternative labels for, and hence access paths to, the specific concepts involved. For instance, "store", "send" and "archive" for the option "send" in the File menu. The keywords may have weights indicating the extent to which they are reasonable labels for the specific menu concept, here simply marked by W.

Then the user who wants to send something, but is not quite clear about where to go to do this, quotes "send" to the system, which constructs the index menu for "send" shown in Figure 1 (b). This gives all the menu options for which "send" is an access route because it appears in their keyword lists, showing they instantiate some form of sending, though the actual option name may be quite different. The options in the index menu are themselves ordered by the weights for "send" in the different origin menus. The user can then jump into the correct location represented by the specific option he has identified as appropriate. (The origin menu options can be amplified with descriptive glosses to help this.)

The basic strategy can of course be extended so composite index menus can be constructed for several items, as in the specification "send AND document"; the menus accessed need not only be utility command menus: they can be search menus higher in the overall system tree. The weights can moreover be personalised, either statically through initial settings to suit the individual's linguistic perceptions, or more interestingly by dynamic adaptation to reflect his patterns of menu movement through origin and index menus over time. This could in principle also lead to the introduction of new items in the keyword sets, taken from used options.

This example does not show the application of artificial intelligence in any sense: but it does illustrate the the basic idea of using linguistic connectivity to counteract heterogeneity.

Example 2

Our work on the notion of integrated inquiry systems provides a more sophisticated illustration and, moreover, one which does show how artificicial intelligence may be used in multifacetted information systems.

The idea here is that when the user seeks information, expressing this need as a natural language question, his need may be met in different ways from different types of information base, for example by information from a conventional database, from a bibliographic base, even from a knowledge base if such exists, and that insofar as these different types of base can respond to appropriately formulated versions of the input question, they should do so. The common input question will therefore be processed to derive a database query (Boguraev and Sparck Jones 1984), a document request (Sparck Jones and Tait 1984), and so forth, for searching the different types of base available. The first stage of processing is a common one, which uses a general-purpose natural language analyser to determine the meaning of the input as a natural language question and to represent this in a full and explicit, but task-independent form. Then for the database case, this is transformed into a formal data language query with appropriate terms and structure, as illustrated in Figure 2 (a) for a hypothetical building database. For document retrieval the initial question representation is processed to extract complex conceptual units defined by substructure types, i.e. as word senses linked by semantic case relations, from which alternative textual expressions of the conceptual unit are generated to search the document files: a request coordinates sets of these search variants, as shown in Figure 2 (b) for a notional search on a document file containing building manuals and similar literature.

Linguistic linkage is used in the database query derivation to relate natural language input terms to the appropriate predicate and argument terms for the domain and these in turn to their particular forms for the database implementation. Linguistic linkage in the document case relates syntactically different expressions of a search concept and, more importantly, lexical alternatives like synonyms and near-synonyms.

These linguistic processes, even though they include semantic inference in the database case to make explicit the relations between the elements of compound nouns, do not constitute any use of artificial intelligence. AI enters with the need to provide a knowledge base and inference capabilities to interpret 'difficult' questions for database access. With complex domains in particular, natural language questions may be couched in a form which is so remote and so elliptical that it may not be possible to obtain a legitimate formal query by mapping input terms and structures to domain ones. The appropriate form for an input query may only be obtained through inference on a characterisation of the domain (Boguraev, Copestake and Sparck Jones 1986).

1 For example, suppose that in the building database costs are explicitly attributes of building types. The question given in Figure 2 (c) cannot be directly mapped onto a suitable formal query. It is necessary to exploit a model of the domain which makes explicit the relationships presupposed by the question. Thus in the notional model of the domain constructed with the network formalism of Boguraev, Copestake and Sparck Jones 1986 shown in Figure 2 (c), activity types like building have costs, and are applied to building types that in turn have locations of particular types. An inference procedure can follow these links to derive a form of the question in which the missing item is inserted, so the question can be mapped onto a proper database query. We could in principle also promote further integration by using the knowledge base in this way in document request formulation, to extend variation with inferentially-derived expressions, as well as using it in the database case with which we are currently concerned.

In this work we are deliberately using a lexically-oriented semantic knowledge base with limited inference, because we believe this is the correct strategy for this information processing task. So to construct the network we can, but also must, build on the world descriptions represented by conventional dictionary definitions. Thus while the use of a structured knowledge base and inference on it means that we are involved with artificial intelligence, the knowledge base is shallow since it uses only two very general relations to link word senses and leaves much information about the world implicit in the linguistic elements of the network and the relations between them. It does not attempt to provide a reductionist account of the fundamental properties of the real world in a style common to much AI work. At the same time, the non-reductionist style means that the network naturally allows both for much of the richness of the reference world that is represented by the richness of the lexicon it encodes, and for the lexical variety that is important for information access.

5.1 (b) Problems with tests

It is also essential to recognise the problems that arise, in seeking to provide integrated, personalised information systems, with performance evaluation. Conventional information systems are not evaluated as thoroughly as they ought to and can be, though it is recognised that even these are difficult to evaluate. But it is much more difficult to evaluate interactive systems than conventional ones.

This is not only because interactive sessions are intrinsically much more complex, so it is difficult to disentangle the contribution some indexing or search method makes to performance from the effects of other system properties. There is the much more fundamental problem of non-repeatability: learning effects in the user make sessions unique. It is therefore very hard to get satisfactory experimental designs for performance evaluation.

Suppose, for example, that we want to compare an associative network of extracted keywords with a hierarchical thesaurus term display as devices which are available online as aids to the user in formulating and reformulating his search specification. We in principle want the individual user to try both devices for the same need. But whichever he uses first will affect his operations with the other, for instance because he has already seen relevant documents retrieved with the first device and knows how he got them. So along with ordinary experimental problems like those of relevance assessment and of coping with undiscriminating requests, plus those of controlling for session complexity, we have a need to get a genuine comparative evaluation over different types of method or device, and over variants of a given type.

There are three possible experimental strategies here:

(1) having real users and parallel non- or pseudo- users for the same starting need. But this implies non-authenticity: the non- users cannot properly simulate their corresponding real users, just because they are not real users.

(2) taking real user needs as starting points but having all the actual search sessions done by others. This means all the sessions are comparable with one another, but they are equally non-authentic.

(3) having real users applying sometimes one method for their need, sometimes another, but only one session for each need. This implies that there can be no session comparability for the same need, but all the sessions are genuine. This is, I believe, the only acceptable strategy, provided there is a large sample of needs for each method. It is essential to have a large sample for each method or variant studied, indeed of course the same size of sample, as also well-founded samples in all the required respects including, for instance, lack of bias from sample to need or method. But the intrinsic properties of on-line searching mean that even the best-designed and most carefully conducted experiments are blunt instruments since there are many "what if" questions which refer to individual interactive steps and so imply an unattainably fine comparative grain.

It is nevertheless essential to do much more evaluation - investigative or experimental - than is currently the case. This is not only because performance evaluation is always required, but also because of the extent to which, with interactive systems, feeling good at the terminal masks real system performance. The outcome of Blair and Maron's 1985 study is as much a caution about user perceptions as about the merits of the system studied: the users were happy though the objective results were poor. Proper experimental design for evaluating interactive searching, or any other forms of system use in multi-facetted information systems, needs much more work than it has so far had. But if we think that doing information research is a science we need to find out how to evaluate performance, so that we can show that X works better than Y; thinking that this is not necessary because all that matters is making the user feel good is not science: it is marketing.

References

Blair, D.C. and Maron, M.E. "An evaluation of retrieval effectiveness for a full-text document-retrieval system", *Communications of the ACM* 28, 1985, 289-299.

Boguraev, B.K., Copestake, A.A. and Sparck Jones, K. "Inference in natural language front ends for databases" in *Knowledge and data (DS 2): Proceedings of IFIP WG 2.6 working conference (1986)* (ed Meersman and Sernadas), Amsterdam: North-Holland (in press).

Boguraev, B.K. and Sparck Jones, K. "A natural language front end to databases with evaluative feedback" in *New applications of databases (ed Gardarin and Gelenbe), London: Academic Press, 1984.*

RIAO 88 Conference on user-oriented, content-based text and image handling*, Proceedings* *from Centre de Hautes Etudes Internationales d'Informatique Documentaire, Paris, 1988.*

Tait, J.I. and Sparck Jones, K. "Automatic search term variant generation", Journal of Documentation *40, 1984, 50-66.*

EXAMPLE 1 : MENU INDEXING

MENU	OPTION	KEYWORDS
File	store	store W send W archive W
	...	
Mail	send	send W transmit W post W
	read	read W scan W inspect W
	forward	forward W send W
	...	
Diary	send	send W enter W
	...	
WP	print	
	text	print W text W send W
	...	

Figure 1 (a)

INDEX	MENU		
"send"	Mail	send	(max W)
	File	store	.
	WP	print	.
	Diary	send	(min W)

Figure 1 (b)

EXAMPLE 2 : INTEGRATED INQUIRY

QUESTION :

maintenance costs for recreation
centres in rural communities

DATABASE QUERY : (BUILDING DATA)

```
 print cost-per-sq-ft
  where cost-type = maintenance
    and building-type = recreation
    and location-type = country
```

Figure 2 (a)

DOCUMENT BASE REQUEST : (MANUALS etc)

```
 ((maintenance cost) OR
  (costs of maintenance) OR
  (cost in maintaining) OR
  ( ... ))  AND
 ((recreation centre) OR
  (facilities for sport) OR
  ( ... ) ) AND
 ((the rural community) OR
  (communities in the country) OR
  ( ... ))
```

Figure 2 (b)

```
QUESTION :

maintenance costs in rural
 communities


INFERENCE ON KNOWLEDGE BASE

 eg VIA LINKS IN A NET STRUCTURE

                           * nodes above open arrows omitted

     /|\
      |
cost----->activity
----
          /|\ /|\
           |   |   /|\        /|\          /|\
    construct  |    |          |            |
         maintain<-----object----->locate<---
                        /|\                  |
                         |                   |
                     building                |
                     --------
                                             |
                                             |
                    -------------------------
                   |
                   |
                   |        /|\          /|\
                   |         |            |
                    --place----->reside<-----agent
                       /|\                    /|\
                        |                      |
                     country               community
                     -------

=> maintenance costs of buildings
   in rural communities

===> print cost-per-sq-ft
      where cost-type = maintenance
        and building-type = ANY
        and location-type = country
```

Figure 2 (c)