

# Combining databases and knowledge bases for assisted browsing

**Robin Burke**

**Kristian J. Hammond**

Artificial Intelligence Laboratory

The University of Chicago

1100 East 58th Street Chicago, IL 60637

## Information finding and assisted browsing

Finding items of interest in a large multi-dimensional information space is a problem of increasing importance given the ever-increasing amount of information that is accessible on-line. Standard approaches such as keyword retrieval demand more specificity than the average user can supply. The user must know what he or she wants well enough to create a well-defined query in a query language. The alternative to querying is browsing. Browsing allows users who cannot specify exactly what they seek to rummage around in an information space to find it. Browsing in a large space presents additional problems, however. It is difficult to structure a browsing space so that users can move about in useful and efficient ways without getting lost and without having artificial restrictions on their means of access.

Both querying and browsing assume that there is a single location where information is located, but in many cases, multiple sources of information may be relevant and these sources may need to be gathered and combined to meet a particular need. Our approach to problems of information finding in large multi-dimensional information spaces is to employ *assisted browsing*. The user is provided a standard browsing interface to move about in an information space, but his or her progress in this space is monitored and relevant assistance is provided.

The aim of assisted browsing is to allow access

to information along a multitude of dimensions and from a multitude of sources without the user needing to be aware of these complexities. Since browsing is the central metaphor, we avoid forcing users to create a specific queries. At the same time, the intelligent assistance available in the system has the ability to draw in other sources of knowledge. Knowledge-based retrieval agents are aware of all of the dimensions of the information and present suggestions that lead the user's search in reasonable directions.

We have drawn our inspiration in this work from case-based reasoning theories of cognition [6,8,9]. Assisted browsing uses the cycle of "retrieve and adapt" that is fundamental to the case-based reasoning model. We also employ knowledge-based metrics of relevance and similarity that are at the core of many case retrieval systems [2,4].

## Find-me Systems

### The problem

We have implemented our assisted browsing approach in a series of systems called FIND-ME systems. The class of problems addressed by these systems is best explained through an example:

You want to rent a video. In particular, you'd like something like *Back to the Future*, which you've seen and liked. How do you go about finding something?

Do you want to see the sequel, *Back to the Future II*? Do you want to see another Michael J. Fox movie? Do you want to see *Crocodile Dundee*, another movie about a person dropped into an unfamiliar setting? *Time After Time*, another time travel film? *Who Framed Roger Rabbit?*, another movie by the same director?

The goal of the FIND-ME project is to develop systems that deal with this sort of search problem. We see this approach as applicable to domains in which there is a large, fixed set of choices and in which the domain is sufficiently complex that users would probably be unable to articulate their retrieval criteria. In these kinds of areas, person-to-person interaction takes the form of trading examples, because people can easily identify what they want when they see it. Many complex selection problems have these characteristics, for example, personnel selection. It is difficult to specify completely what kind of person is right for a job, but it is easier to look at a person's resume and come up with a response such as "Give me someone like this, but with more leadership experience."

The browsing interface is obviously a crucial component of these systems, but in this paper we focus on intelligent retrieval. FIND-ME systems have active *clerks* that constantly monitor the user's choices and retrieve alternative choices to those that the user is making. Each clerk implements a particular *retrieval strategy* that recalls plausible suggestions. The clerks also allow the user to critique the retrieved examples, and the feedback is used to retrieve further examples. The clerks' intervention decreases the chances that the user will become stuck in a particular corner of the information space, due to inadequate knowledge of the space.

The ability to critique the system's suggestions allows the user to push the retrieval system in a particular direction without having to explicitly articulate sets of features. This approach shares some characteristics of the use of relevance feedback in information retrieval [10] but with the

important difference that users see an explicit explanation of why each example was retrieved and can critique particular components of that explanation. In most relevance feedback approaches, the user selects some retrieved documents as being more relevant than others, but does not have any detailed feedback about the features used in the retrieval process.

## Video Navigator

We have used our experience in building other FIND-ME systems [7] in the construction of a system for browsing the set of movie videos available for rental in a typical video store. This system, VIDEO NAVIGATOR, draws from a database of 7500 movies based on a popular video reference work [11]. VIDEO NAVIGATOR serves as an intermediary between the user and this database.

The system uses a map of the video store that users can click on to explore the contents of database in a fairly standard, hierarchical, manner using the names of actors, directors, etc. The hierarchical type of search used in the map interface (and in many other search systems) can be an effective way to find a particular item when one or more of the features are known. However, more often than not, users will have a vague notion of what movie they want but not a something that could made into concrete query. By using the browsing interface, users can look around much as they would browsing in a video store.

When a user selects a movie to examine, the clerks spring into action. In VIDEO NAVIGATOR, there are four clerks: one recalls movies based on their genre, one recalls movies based on their actors, another on directors, and still another arrives a suggestions by comparing the user against the profiles of other users<sup>1</sup>. Whenever the user picks a movie to inspect, each clerk retrieves and suggests another related movie. It is as if the user has a few knowledgeable movie buffs following her around the store, suggesting

---

<sup>1</sup>For reasons of space, we have omitted discussion of the user profiling agent in this paper.

movies based on their particular area of expertise. The remainder of this paper discusses the operation of these information retrieval agents.

## Implementing retrieval agents

Consider the following representative interaction the VIDEO NAVIGATOR. The user selects the actress Glenn Close and looks at the information for the movie *Dangerous Liaisons*. The **Genre Agent** looks for other movies that share the same genre sub-categories as this movie. In this case, it looks for movies with the categories: “romantic drama,” “historical drama,” and “movies based on plays.” A database query joining these features retrieves about a hundred movies. These are ranked by the degree of overlap, and those with only one common feature are discarded. This leaves two dozen or so candidates. Only one movie, *Valmont*, shares all of three features, which makes it the best suggestion, appropriately so, since it was based on the same book as *Dangerous Liaisons*.

Suppose, however, that the user has already seen *Valmont*. The genre agent goes back to its list of possible candidates, which overlap on two genre features. In order to select among them, the genre agent must make additional discriminations, taking into consideration additional discriminating features that are relevant for genre.

The next place the genre agent turns is to the director feature, since directorial style is a significant component of genre. It first tries to see if any of these movies were directed by Stephen Frears, the director of *Dangerous Liaisons*. This step fails – the chances are small that in such a group there will be more than one movie by the same director. The genre agent has a fall-back position, however. It turns to a separate knowledge source – this time a semantic network of directors and their influences<sup>2</sup>. It looks at each movie under consideration and asks “How similar are this director’s influences to those of the

original movie?” This question is answered by using a marker passing operation in the semantic network. The distance in the network between two directors is a rough guide to the similarity of their influences. This information is factored into the evaluation of each of the candidate suggestions.

In this case, two movies, *Yanks* and *Enemies, A Love Story*, remain tied in first place. They are both romantic, historical dramas and their directors share some influences with Stephen Frears. The genre agent now invokes a third stage of discrimination, turning finally to the actors involved in the movies. First, it checks to see if any of the same actors appear in these movies and the target. When this operation fails, another knowledge source is employed, this one a knowledge base of actor types. This knowledge base categorizes actors into types such as “action hero” or “tragic heroine.” One important category is the multi-talented actor, who plays a wide variety of roles. The presence of such an actor in a movie is a clue that the film may be one in which versatile dramatic performance is important. This category is a point of similarity between *Dangerous Liaisons*, which stars John Malkovitch, and *Enemies, A Love Story*, featuring Angelica Huston, so this movie becomes the suggestion.

At this point, the genre agent has refined its criterion to romantic, historical dramas with Frears-like direction and versatile acting. This is not a query that a user could easily articulate, yet the agent can derive it by starting from the example and reasoning about what elements of that movie might contribute to the perception of genre. If the agent’s notion of genre does not match with what the user had in mind, the user can critique the agent’s retrieval by examining the factors that went into it. The user might, for example, decide that the historical aspect of *Dangerous Liaisons* was not that important. The genre agent would then drop “historical drama” from the features it considers and recompute its suggestion.

This example shows the application of two ba-

---

<sup>2</sup>The knowledge base of directors and their influences is built from information contained in [1].

sic operations: finding examples that are similar to the target under a certain metric, and discriminating between examples based on additional criteria. The genre agent uses genre categories to retrieve an initial set of possibilities and then applies knowledge of directorial style and actor types to discriminate among them. The other retrieval agents operate similarly. For example, the director agent uses the director knowledge base to find movies by related directors and then discriminates between them using a quality metric and genre categories.

## Architecture

The retrieval agents in VIDEO NAVIGATOR perform their similarity assessment by drawing together of several different kinds of information into the making of a recommendation. The database holds many details about each movie, but has no structure that helps make sense of those details. To reason about the features of movies, the agents turn to other knowledge sources: such as the knowledge base describing directors and their influences, and the knowledge base of actors and actor categories.

The architecture of the information retrieval agents in *Video Navigator* is shown in Figure 1. The three advisory agents interact with the video database through an intermediary: the database agent. The purpose of this design is to maximize modularity with respect to the database. The advisory agents issue KQML requests [5] to the database agent, which translates them into calls to the database. For example, the message in Figure 2 is sent by the director agent to retrieve the movies directed by Woody Allen. The database agent translates this into a simple table lookup and returns a list of matching propositions. For more complex queries, it can perform simple syntactic query optimization.

The retrieval strategies followed by each agent are implemented as plan trees containing conditionals. The steps in the plan are retrieval or discrimination operations. If an operation fails, for instance, if information is unavailable, there

are fall-back methods available. For example, if none of the actors in a target film have starred in other movies, the actor agent cannot perform its initial retrieval step. It will generalize its search by looking in the actor category knowledge base for other similar actors and then try to retrieve their movies.

## Conclusion

FIND-ME systems fill an important gap in information gathering research. They use existing archives and data-bases as resources to be mined on demand as part of the user's exploration of a complex domain. While there is growing recognition of the importance of the user in knowledge discovery and information gathering systems [3], many of these systems are built to be batch processors that learn new concepts or construct new knowledge bases independently of a user. In FIND-ME systems, users are an integral part of the information gathering process, even when they cannot specify exactly what information they seek.

Robustness in the face of user uncertainty is another important aspect of VIDEO NAVIGATOR and other FIND-ME systems. Most people's understanding of real world domains such as cars and movies is vague and ill-defined. This makes constructing good queries a difficult skill to master. We believe therefore that an information gathering system should always provide the option of examining a "reasonable next piece," of information, given where the user is now. The clerks in VIDEO NAVIGATOR, provide several reasonable next steps using a range of retrieval strategies.

## References

- [1] Andrew, G. *The Film Handbook*. Boston: G.K. Hall & Co. 1989.
- [2] Bareiss, R., & J. King. Similarity Assessment in Case-Based Reasoning. In *Proceedings of the*

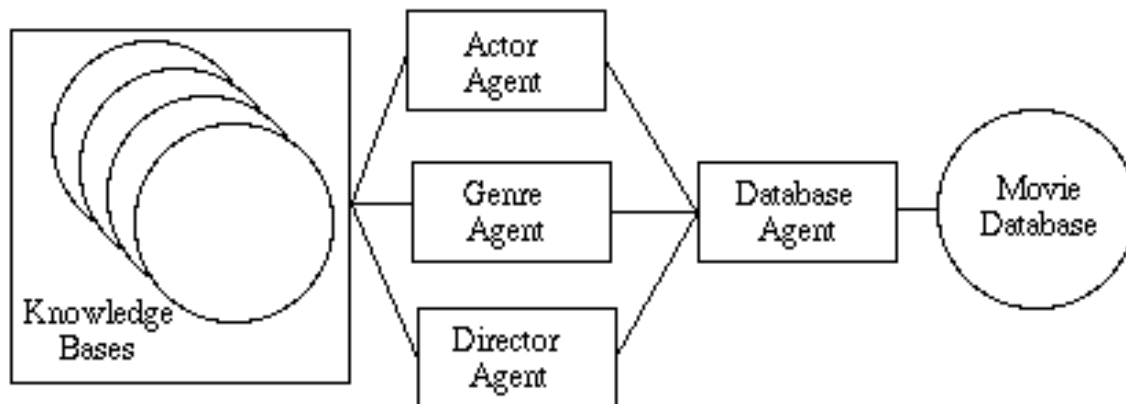


Figure 1: Video Navigator Architecture

```
(ask-all :sender director-agent
         :receiver db-agent
         :reply-with message-64
         :language parmenides
         :ontology movies
         :content
         (directed 'Woody Allen'
                   ?movie))
```

Figure 2: Example Message to the Database Agent

*Case-Based Reasoning Workshop I*. Defense Advanced Research Projects Agency, 1989.

[3] Brachman, R., & T. Anand. The Process of Knowledge Discovery in Databases: A First Sketch. In *AAAI Workshop on Knowledge Discovery in Databases*, (pp. 1-11). Seattle: AAAI, 1994.

[4] Burke, R. Retrieval strategies for tutorial stories. In *Case-based Reasoning: Papers from the 1993 Workshop* (pp. 118-124). AAAI, 1993.

[5] Finin, T., J. Weber, G. Wiederhold, M. Gene-sereth, R. Fritzson, J. McGuire, R. Pelavin, S. Shapiro, & C. Beck. Draft Specification of the KQML Agent-Communication Language. <http://www.cs.umbc.edu/kqml/kqmlspec.ps>, 1993.

[6] Hammond, K. *Case-based Planning: Viewing Planning as a Memory Task*. Academic Press. Perspectives in AI Series, Boston, MA. 1989.

[7] Hammond, K, R. Burke, & K. Schmitt. A Case-Based Approach to Knowledge Navigation. In *AAAI Workshop on Indexing and Reuse in Multimedia Systems*, (pp. 46-57). Seattle: AAAI, 1994.

[8] Kolodner, J. *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann, 1993.

[9] Riesbeck, C., & R. C. Schank. *Inside Case-Based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum, 1989.

[10] Salton, G., & M. McGill. Introduction to modern information retrieval. New York: McGraw-Hill, 1983.

[11] Wiener, T. *The Book of Video Lists*. Kansas City: Andrews & McMeel.