USING IODYNE:

ILLUSTRATIONS AND EXAMPLES

Eric H. Johnson

OVERVIEW

IODyne (pronounced "iodine") is an Internet client program that allows one to retrieve information from servers by dynamically combining information objects. Information objects are abstract representations of bibliographic data, typically titles (or title keywords), author names, subject and classification identifiers, and full-text search terms.

IODyne is being developed as part of a library research project, and so the emphasis on bibliographic data. The retrieval paradigm described here can be extended to other kinds of data as well.

IODyne embodies several key ideas:

- simultaneous connection to, and querying of, multiple servers, regardless of the query language or protocol required by each;
- abstraction of commonly understood kinds of searches (title, author name, subject, etc.) from the particular representation of the searched data on the server. Regardless of the particular bibliographic server you are connected to, you always do the same kinds of searches in the same way;
- construction of searches as persistent, arbitrarily complex, storable, and recallable objects, rather than the end result of a series of steps that cannot be recalled;
- caching of search results in the client, as well as set combination (e.g., set union, intersection, and difference) of search results in the client (various kinds of client-server load-balancing optimizations are possible, doing some kinds of set combinations in the client and others in the server);

- normalized internal representation of abstract queries to facilitate set combination and other set-theoretic operations such as query subset recognition;
- a modeless, visible, multiple-windowed user interface in which, rather than learning commands, you learn how information objects behave and how to manipulate information objects. You do information retrieval in IODyne by moving information objects among different information spaces and combining information objects in various ways in these spaces;
- visual, navigable, and simultaneously viewable information spaces—such as subject thesauri and classification schema—to facilitate search term suggestion.

Underlying all these ideas is the desire for the best use of each computing resource involved in getting information from the source to the user. The client-server paradigm is a sound one, but the prevalent use of World Wide Web browsing software and server-side CGI binaries does little to use the power of client machines to their best advantage, particularly in the realm of information retrieval. We are now in the situation where the desktop computers commonly used for Web browsing, though they rival the capacity and speed of the servers from which they retrieve information, are essentially being used as dumb terminals. Granted that much of this power goes toward supporting "nice" user environments and presenting multimedia content, there is, however, much more that can be done by client machines to enhance the *informational* aspects of client-server interaction and take much of the query processing load off the server.

As for the user interface, HTML forms provide only the barest level of interactivity for the user of an information retrieval system. Server-intensive processing must occur between each stage of retrieval feedback to the user, not only to process the query but to generate the HTML required to display the results as well as the particular controls required for the next stage of query refinement. This is another artifact of the outdated host-terminal paradigm, where the host must compose the terminal's screen, which impinges on client-server interaction in WWW-based information retrieval. Server administrators have enough to worry about without also having to write and maintain code for user screens.

The presentational power of desktop user environments, especially those which support drag-and-drop operations, allow for far higher levels of interactivity than is possible with HTML forms. For an information retrieval environment, a useful interface would allow you to literally "draw" queries on the screen, send any parts of these queries to any number of connected servers at any time, would give you different kinds of feedback in multiple windows, and so on. Though well within the capabilities of any currently available desktop PC, such an environment is all but impos-

sible to implement in HTML, though with "frames" available in most Web browsers you might be able to do it in a limited and unsatisfactory modal sort of way.

The HTTP protocol is another obstruction to good information retrieval interaction in that it imposes far too much overhead for the needs of query and relevance-feedback interchange. In such cases, connection-based client-server communication, with lower-overhead protocols tailored to the needs of the particular kind of data exchanged, are far more desirable.

None of this is to say that Web browsers have no place in an information retrieval environment. Web browsers are fine for displaying documents once they have been located and for moving between documents in the browsing sort of way allowed by hypertext. This was the intent when Mosaic was first developed. In the interests of maintaining an open standard for digital library documents, the IODyne client will support both HTML and SGML displayers through CCI as well as other kinds of viewers applicable to particular collections being accessed.

IODyne toolbar. All menus and tool controls (for Keywords, Keyword in Context, and Concept Space) are here, as well as buttons for initiating searches. You use the **Repositories** menu to control connections to repositories.

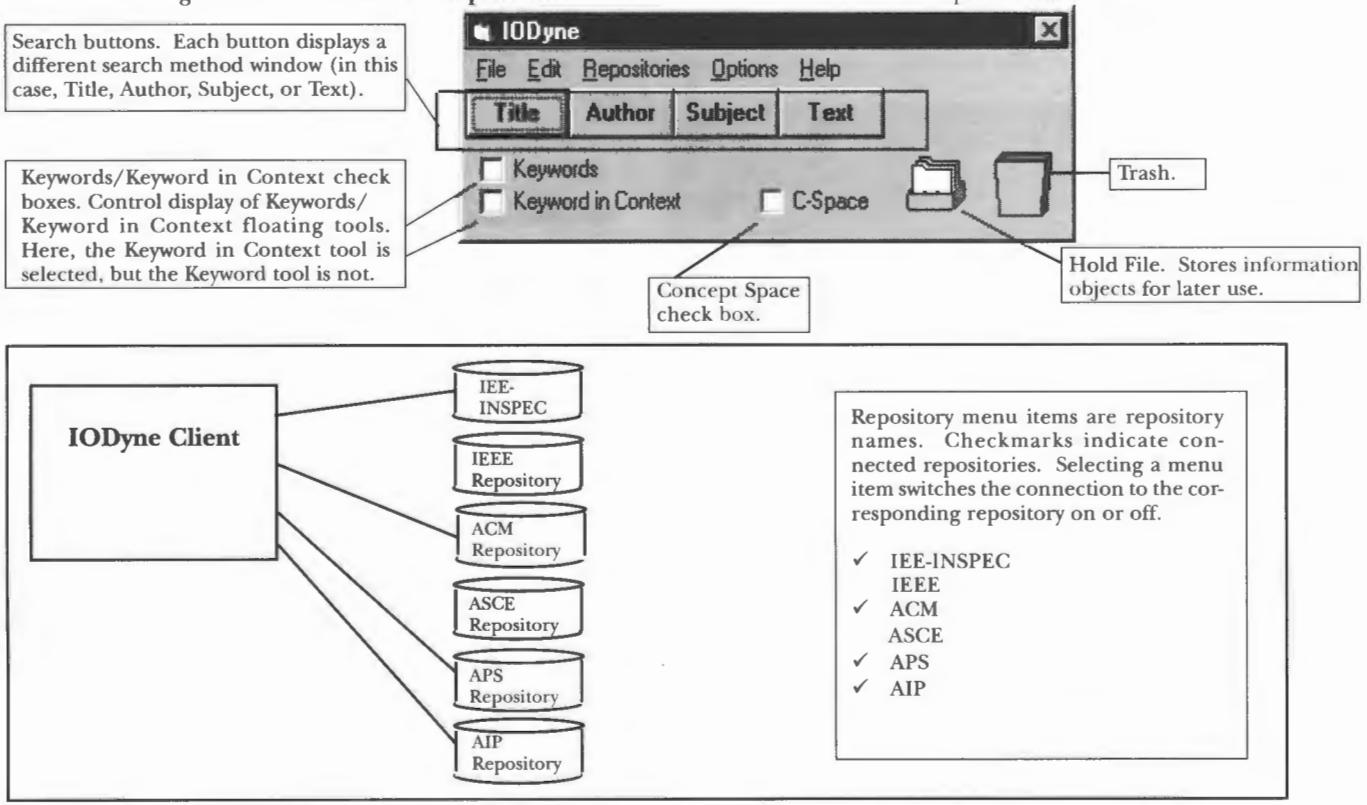
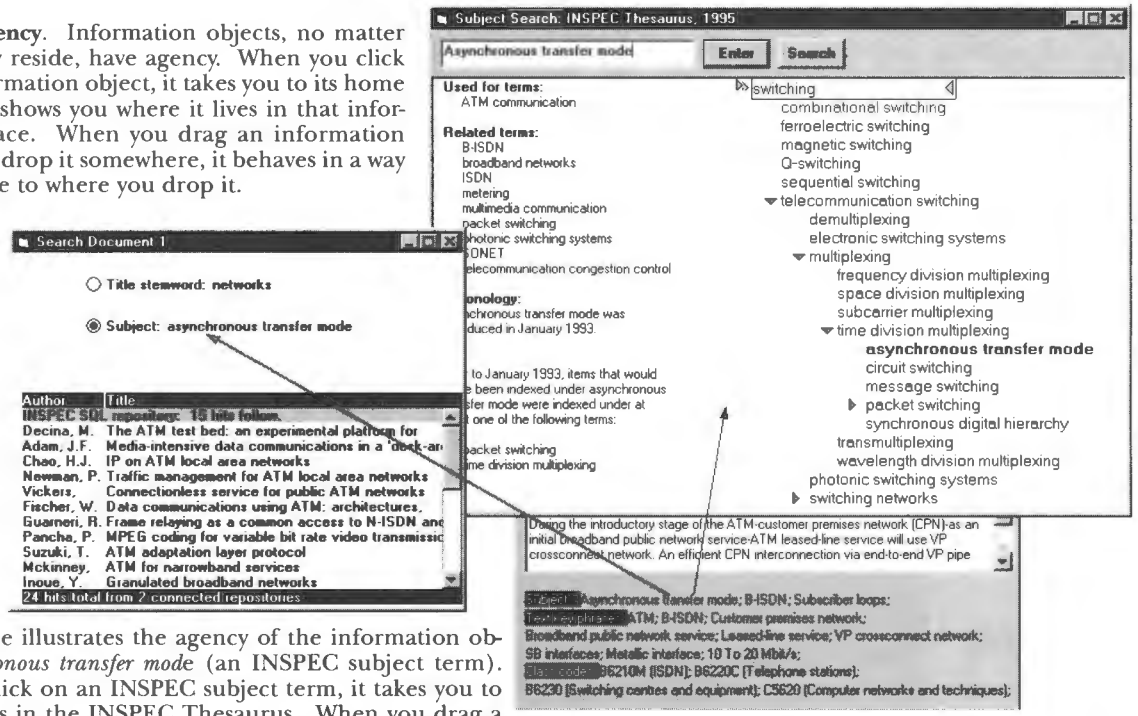


Figure 1. The IODyne Toolbar

Object Agency. Information objects, no matter where they reside, have agency. When you click on an information object, it takes you to its home space and shows you where it lives in that information space. When you drag an information object and drop it somewhere, it behaves in a way appropriate to where you drop it.



This example illustrates the agency of the information object *Asynchronous transfer mode* (an INSPEC subject term). When you click on an INSPEC subject term, it takes you to where it lives in the INSPEC Thesaurus. When you drag a subject term (action indicated by thick gray arrow) and drop it on a search document, the subject term makes the search document retrieve records indexed by that subject term from the connected repositories. Other kinds of information objects (e.g., author, title, text) work the same way.

Figure 2. Agency of the Information Object *Asynchronous Transfer Mode*

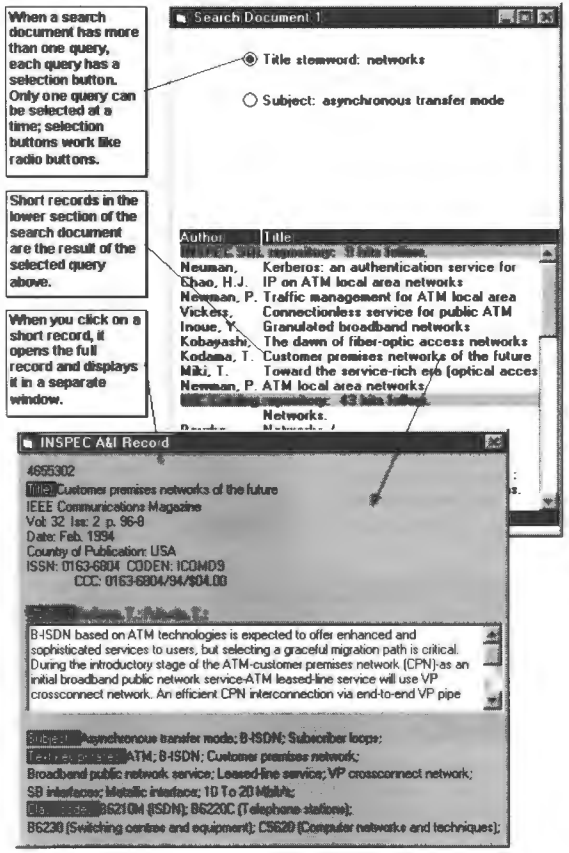
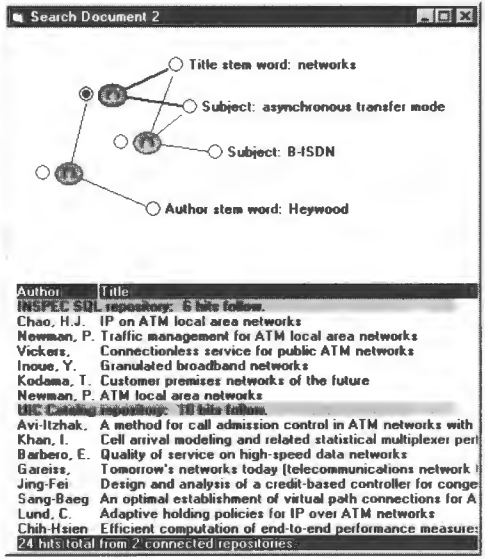
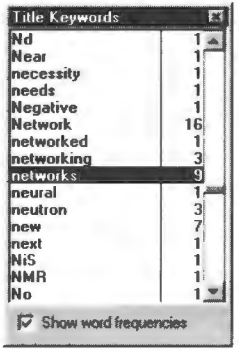


Figure 3. Search Documents

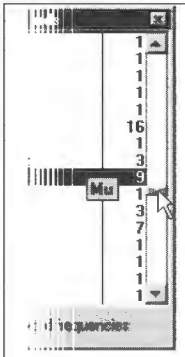
Search Documents. You construct queries on search documents, which you can save and open again just like documents in other applications. You can drag information objects from other windows onto search documents (as well as between search documents). Using drag and drop editing, you can build arbitrarily complex query trees combining different types of information objects. Combination nodes can have more than two children, and any node can have multiple parents. Like simple queries, combination nodes have selection buttons. Selecting a combination node highlights the node, all of its children (transitive as well as direct), and the branches connecting them. The short records displayed are for that selected combination of queries.



***Keywords* and *Keyword* in *Context* tools.** As you type a search term, these floating tools (which you can modelessly turn on and off at any point during search term entry) give you feedback as to how many items your search will retrieve.



In this example, the word **networks** has been typed into the Title Search dialog. The Keywords tool (left) shows that the word is in the database and 9 titles contain it. The Keyword in Context tool (below) shows the actual titles that contain the word. As you continue to type other words, the Keyword in Context tool culls the list to show you which items will be retrieved by all the words you have typed.



The Keywords scrollbar has a modified slider that shows you what part of the alphabet you will land in when you release the mouse button. A metaphor is tabs on the side of a dictionary. The keywords list can be very long (maximum of over 2 billion words); this feature can take you immediately to any part of the list no matter how long it is

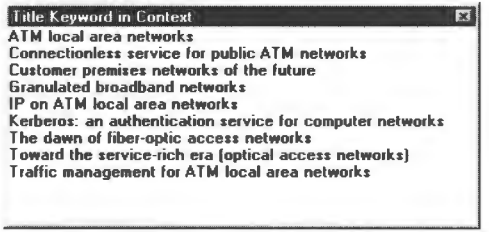
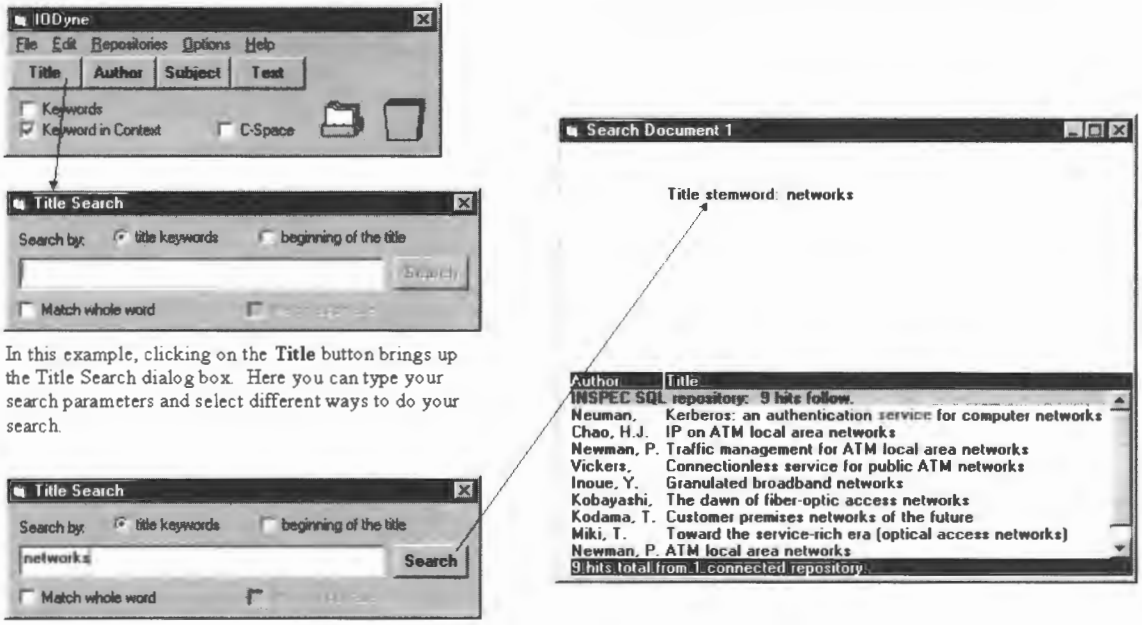


Figure 4. *Keywords* and *Keyword* in *Context* Tools

Creating search parameters. You can create search parameters in a number of ways. The simplest is to treat the search buttons on the IODyne toolbar as regular buttons: click on them and a search dialog appears for you to enter your search term and select various options.



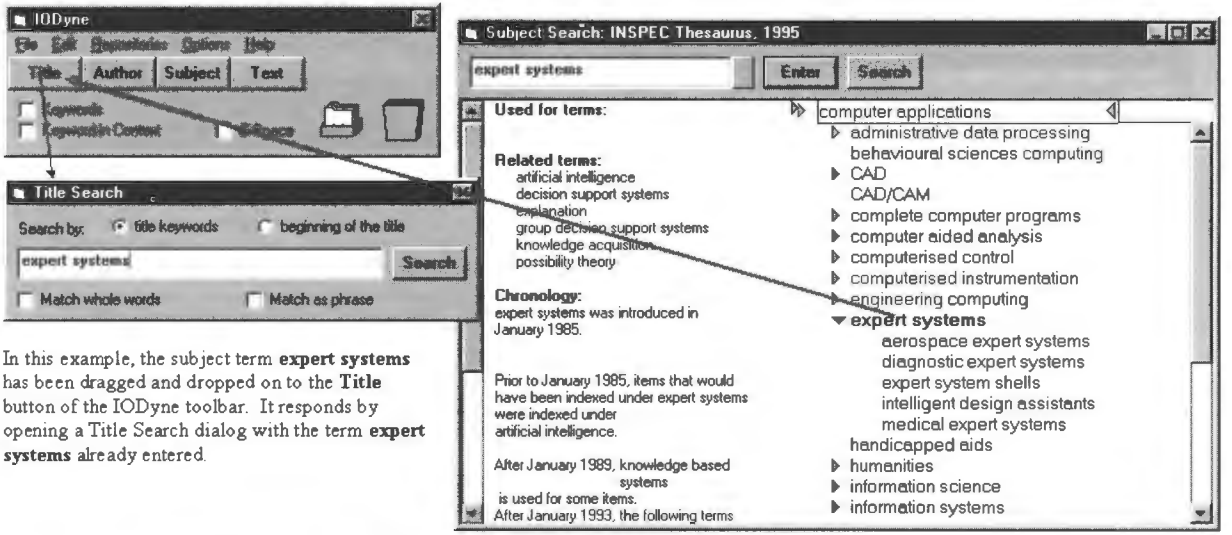
In this example, clicking on the **Title** button brings up the Title Search dialog box. Here you can type your search parameters and select different ways to do your search.

Here the word **networks** has been typed into the Title Search dialog. Clicking on the **Search** button puts the search parameter and attributes (what kind of search it is, how to match the search term to the database fields, etc.) on to the search document as shown at right.

Figure 5a. Creating Search Parameters

Although you can click on buttons in IODyne to open dialogs and do searches, IODyne supports other ways of using buttons.

You saw in the example above how clicking on the **Title** button opens the Title search dialog with an empty search parameter field. You can drop an information object from another window onto the **Title** button (or any of the buttons on the IODyne toolbar) to open the search dialog with the content of the dropped object already entered as the search parameter.



In this example, the subject term **expert systems** has been dragged and dropped on to the **Title** button of the IODyne toolbar. It responds by opening a Title Search dialog with the term **expert systems** already entered.

You can then edit the term to suit your search and change any of the options in the dialog, exactly as if you had typed it yourself. Here the s has been taken off the end of expert systems and the option to search for it as a phrase selected. Clicking on the Search button then puts the search on a search document and searches the connected repositories.

Figure 5b. Creating Search Parameters

Clicking on the **Search** button doesn't always put the search on the search document where you would like it. For this reason IODyne allows you to drag a Search button and drop it on the spot where you would like the search parameters displayed.

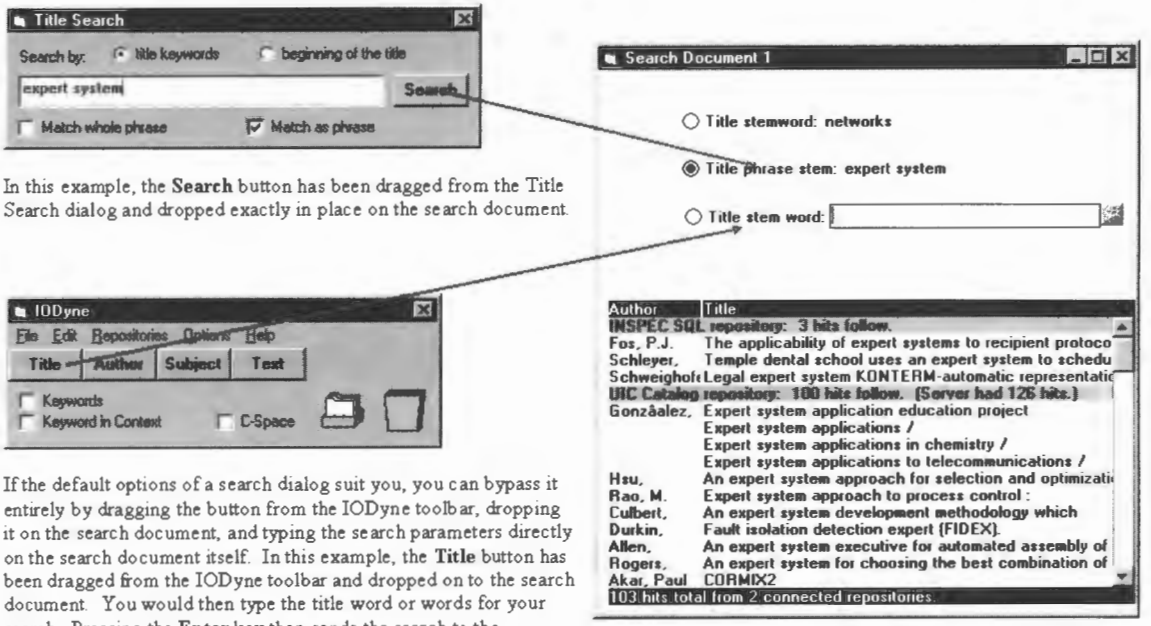
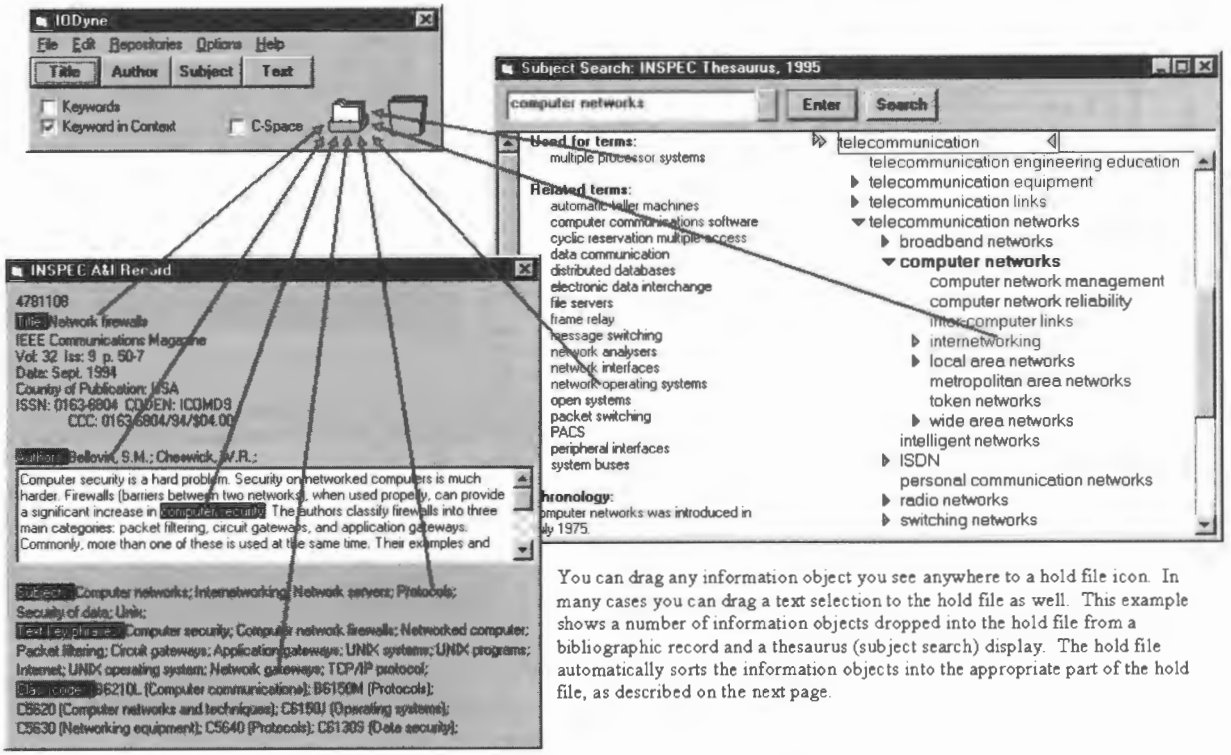


Figure 5c. Creating Search Parameters

In this example, the **Search** button has been dragged from the Title Search dialog and dropped exactly in place on the search document.

If the default options of a search dialog suit you, you can bypass it entirely by dragging the button from the IODyne toolbar, dropping it on the search document, and typing the search parameters directly on the search document itself. In this example, the **Title** button has been dragged from the IODyne toolbar and dropped on to the search document. You would then type the title word or words for your search. Pressing the **Enter** key then sends the search to the connected repositories.

Using the Hold File. The hold file employs the metaphor of the index card file to help you store keywords, subject terms, and even entire bibliographic records. You can make a hold file part of a search document so that when you save it the hold file and the data in it becomes part of the saved document.



You can drag any information object you see anywhere to a hold file icon. In many cases you can drag a text selection to the hold file as well. This example shows a number of information objects dropped into the hold file from a bibliographic record and a thesaurus (subject search) display. The hold file automatically sorts the information objects into the appropriate part of the hold file, as described on the next page.

Figure 6a. Using the Hold File

Opening the hold file and using its contents. Open the hold file by clicking on it.

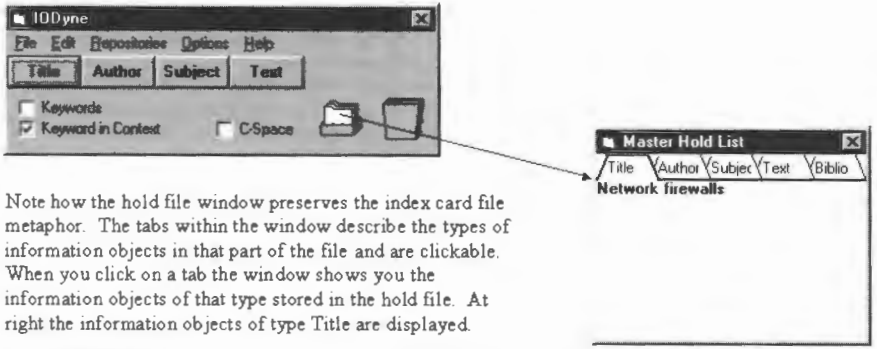
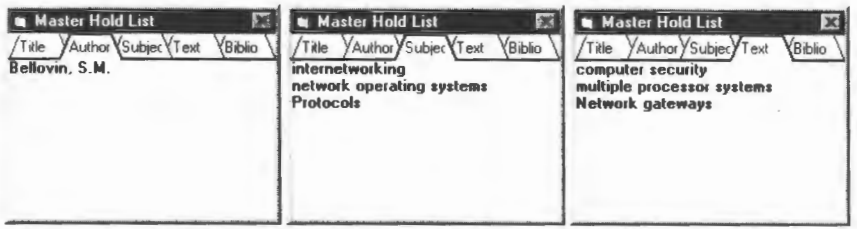


Figure 6b. Opening the Hold File and Using Its Contents

Note how the hold file window preserves the index card file metaphor. The tabs within the window describe the types of information objects in that part of the file and are clickable. When you click on a tab the window shows you the information objects of that type stored in the hold file. At right the information objects of type Title are displayed.

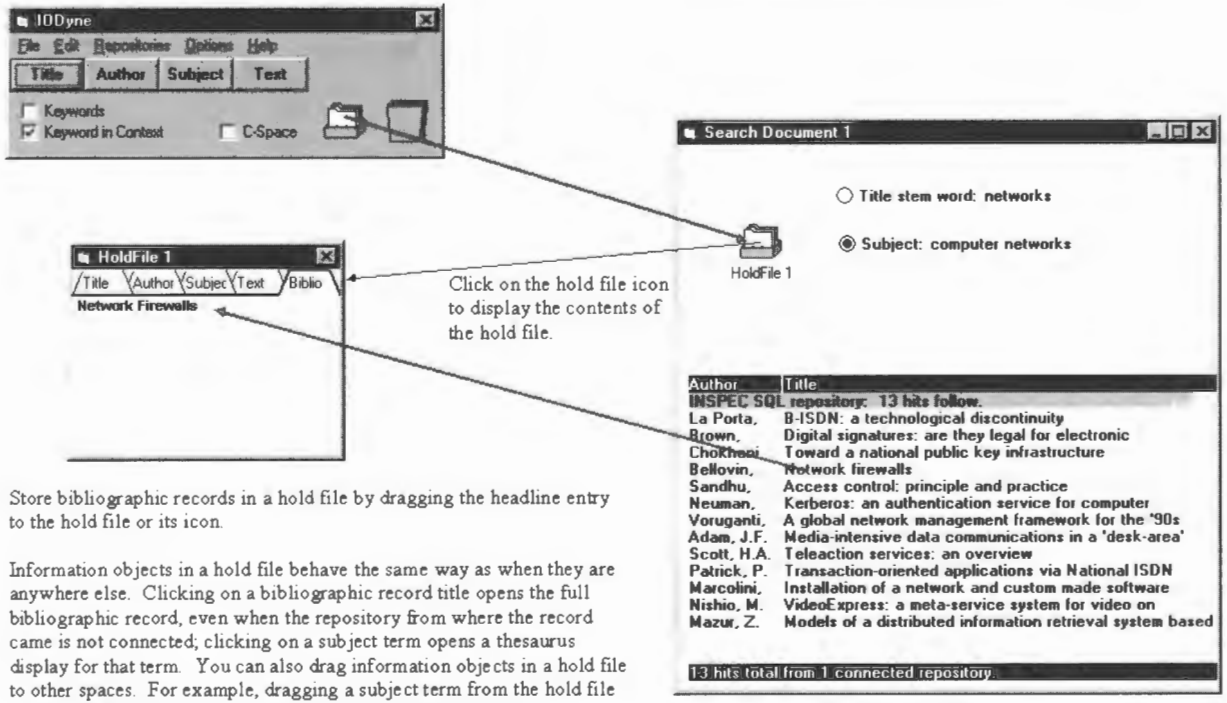
The title **Network firewalls** in the example at above right corresponds to the article title dropped into the hold file in the example on the previous page. Information objects of other types are under the other tabs. Seeing where the other information objects were dragged from, you should expect to see the following under the other tabs:



When you drag names appearing in the author field of a record into the hold file, it automatically files them under Author. Terms you drag from subject fields or thesaurus displays are put under **Subject**. However, terms listed under **used for** in a thesaurus display are considered as Text since they are not preferred subject terms; this is the case for the term **multiple processor systems**. Terms dragged from miscellaneous parts of records such as text key phrases and selections from abstracts are also considered **Text.** The actual type ascribed to a given field displayed on a record is configured by the manager of the repository.

The **Biblio** tab is for storing bibliographic records, described below.

Making local copies of hold files. You can make a local copy of the hold file dragging the hold file icon to a search document and dropping it there.



Store bibliographic records in a hold file by dragging the headline entry to the hold file or its icon.

Information objects in a hold file behave the same way as when they are anywhere else. Clicking on a bibliographic record title opens the full bibliographic record, even when the repository from where the record came is not connected; clicking on a subject term opens a thesaurus display for that term. You can also drag information objects in a hold file to other spaces. For example, dragging a subject term from the hold file to a search document performs a subject search on all connected repositories.

Figure 6c. Making Local Copies of Hold Files