Dagobert Soergel
College of Library and Information Services
University of Maryland
Office: (301) 405-2037 Home: (703) 823-2840
Fax: (301) 314-9145 E-mail: ds52@umail.umd.edu or dsoergel@umail.umd.edu

Information structure management

A unified framework for indexing and searching

in database, expert, information-retrieval, and hypermedia systems

Abbreviated version

Abstract

Database, expert, information-retrieval, and hypermedia systems share an essential unity and a unified framework can improve their design. The systems differ along a variety of dimensions but only as a matter of degree. A new definition of a system structure generalizes search and inference operators that are applicable to any type of searching. The basic structure consists of objects (entities) and relationships (links) to make statements about objects. To enhance searching and inference, the structure includes neighborhoods (sets of objects with their relationships), especially offspring neighborhoods and ancestor neighborhoods, and connections (named chains of links). Navigation and query-based searching are but different metaphors for the same basic search and inference processes, such as spreading activation (with Boolean and weighted searching as special cases), hierarchical inheritance, and structure matching. Examples from various contexts illustrate the approach.

Full version: Fidel et al., ed. Challenges in indexing electronic text and images. Medford, NJ: Learned Information; 1994 (ASIS Monograph Series). Chapter 7, p. 111 - 156.

(Based on a paper presented at the
54th ASIS Annual Meeting, Washington, DC October 27 - 31, 1991)

Table of contents

1 Introduction and overview

This paper presents a unified design that integrates the functions of database systems, expert systems, Information Storage And Retrieval (ISAR) systems, and hypermedia systems, a design that offers generalized search and inference operators applicable to any type of searching - by subject, by citation, by prerequisite, or by any other criterion, following any kind of link - and thus make searching both simpler and more powerful. We call such a system an **information structure management system**, emphasizing the key element of structure in common to all types of systems encompassed in the unified approach proposed.

Distinctions often made, particularly the distinction between traditional (bibliographic) retrieval systems and typical hypermedia systems, are not absolute but a matter of degree in a multidimensional design space (Section 2). A general structure and data model for unified information structure management systems (Section 3) provides the basis for discussing search (or navigation) (Section 4) and system construction or indexing (Section 5). Section 6 revisits the unity theme by analyzing the multiple functions of objects (entities, nodes) and relationships (links) in an information structure management system.

Throughout the paper we will introduce examples from one context and then generalize the principle to other contexts. For example, consider Boolean searching which is common in ISAR systems. It might be quite useful in hypermedia systems to extend navigation, which leads from a starting document to a citing document or a supporting document, to a Boolean search, where the user could start from two documents and look for documents that cite **both** (known as cocitation) or that support **both**. The examples have been chosen with a view to demonstrating the richness and power of the proposed structure.

2 The multidimensional design space for information structure management systems

Starting from the differences between a prototypical (bibliographic) retrieval system and a prototypical hypermedia system, this section analyzes the multiple facets or dimensions that characterize systems or, more precisely, individual searches; these dimensions can vary independently from each other, spanning a multidimensional design space.

A few introductory examples are useful as a backdrop for the general discussion. When I start preparing a lecture on an new topic, I might first do a literature search, starting with a suitable subject descriptor that leads to a list of documents and pick a document to read first, or I might remember a document that would give a good start. The starting document might lead to a document cited, or it might teach me that I must find out about another topic first and do another search; if there are no such leads, I will simply proceed to another document from the retrieved list.

The reader of a book might follow the king's advice: "Begin at the beginning and go on till you come to the end: then stop", reading one chapter after the other, one paragraph after the other. Or she might deviate from the author's sequence, consulting, for example, the table of contents or the diagram showing the logical dependencies between chapters found in the front of some books. Or she might deviate form the linear path by looking at some of the footnotes or by following a cross-reference.

Some people may read the newspaper from beginning to end, but most of us pick certain sections, scan the headlines, read the first paragraph of a story and perhaps go on in that story, or jump on to the next headline. In an electronic newspaper the reader can enter an interest profile, a set of terms as a starting point from whence a search leads to a list of stories or articles in order of importance or grouped by subject.

In each of these examples, the search for information is a journey that evolves in a series of steps. Each step starts from a point that is known and moves to a new point that is yet unknown. The user must find the next stop on the journey, the item of information to be consulted next; an ISAR/hypermedia system must support the user in this retrieval task. The term *user* is used here in the broadest sense, including *searcher* and *reader*.

An item of information can be any text object - from a book to a sentence - or any other media object, such as a picture, a map, or a segment of sound or a moving picture sequence that can be replayed. The term *media object* is a broad term that covers all of these; specifically, many media objects consist only of text (*text objects*) or contain a large proportion of text. The general framework presented here deals more broadly with any type of *object*, *entity*, or *item*. An object is also called a *node* when the emphasis is on the place or role of the item in a network of links. In the literature a media object is also called *frame* (in the sense of all the information fitting on one screen, not to be confused with frames used for knowledge representation), *notecard*, or simply *card*.

Reading a whole document, such as a book or journal article, is a special case of search: The question on what chapter (or section, or paragraph) to read next is seldom posed explicitly; by default the reader goes on to the next chapter (or section, or paragraph). But there exists a retrieval problem nevertheless, and a book solves this through its arrangement, its table of contents or an outline in flow chart form as mentioned above.

A book that is available as a hypermedia base makes the user (reader) aware of choices and suggests detours or radical deviation from the default sequence. It does so by supporting retrieval through a network of links among paragraphs (or sections, or chapters) that allow the reader to select (or retrieve) another section starting from a section just read. This kind of retrieval is an integral part of reading. On the other hand, when the user is faced with the question what entire document to read next in his quest for information, he turns to a bibliographic ISAR system. There he cannot start from a paragraph he just looked at; most systems require a query formulation as a starting point; a query formulation contains subject descriptors or author names, etc. There are systems that establish links between whole documents, such as a database of full-text journal articles in which the user can jump from a reference in the bibliography of article 1 directly to the referenced article 2.

The basic retrieval operation is always the same: Starting from a known object - a media object or a descriptor - the user follows links provided by the system to find one or more other objects. On an abstract level, using a known media object as a starting point to find descriptors for further searching is the same thing as using a known descriptor as a starting point to find media objects for perusal. Even Boolean searching works both ways: The user may start from two descriptors and look for all media objects that can be reached from **both** descriptors, or she may start from two media objects and look for descriptors that can be reached from **both** of these objects. (Descriptors in common to two relevant media objects are likely to be good descriptors for further search.) Moreover, as will become clear below, the same principles apply for searching in a database system that deals with any type of objects, such as food products, and that uses relationships or links to store data about these objects.

There are differences between searches and, by implications, between systems. Analyzing these differences is important as a basis for system design (Table 1). Table 2 shows the dimensions that span a space in which any **search** can be located.

From a user's point of view the best system is one that supports any type of search equally well, no matter where it falls in the multidimensional design space defined by Table 2. As a practical matter, most systems support some types of searches and some search features better than others; in that sense, the dimensions discussed can be considered as dimensions for the classification of **systems**.

Table 1. **Parameters or dimensions for analyzing searches and systems**

"Traditional" information retrieval systems and hypermedia systems are designed to support certain kinds of searches.

The differences can be analyzed along a number of dimensions shown below.

These differences are a matter of degree.

An integrated information structure management system should support all types of searches, adapting to user needs and preferences.

Each of the dimensions shown in Table 2 merits a few comments.

Type of starting object(s). A typical hypermedia search starts from a paragraph, a picture, or an audiovisual object and follows links to other such objects. A typical bibliographic search starts from a subject or author and follows links from these to whole documents.

Method of finding good starting objects and Role of the starting object in the entire search. In the typical hypermedia search, the user looks at objects that actually provide some of the information needed and uses these same objects as starting points for the next search step. In a bibliographic ISAR system, the starting object - a subject descriptor or author name - is not of interest in and of itself but only as a means to find target objects. The user must deliberately search her own memory or a thesaurus to find useful subject descriptors to start the search. In the typical hypermedia search, the user never worries about finding starting objects because the items of information she reads or looks at will also lead to further interesting items. In the typical bibliographic ISAR system, finding the right starting objects becomes a major concern.

Method of specifying the starting object(s) and link types: either (1) clicking on elements displayed on the screen (often a default value) and (2) keying in their identifiers. (1) is available in modern bibliographic systems: With a document record displayed, the user can click on an author name or subject descriptor in the record and thereby initiate a new search. This is a matter not of the basic system nature but of the user-system interface.

Type of target objects found. The typical hypermedia search leads to the media objects themselves, a paragraph, a picture, or the full text of a document. The typical bibliographic search leads to references to documents; the user must find the actual text. But some systems first retrieve references and then, in a second step, provide access to full text. (There may or may not be links, such as citation links, between documents.)

The following two dimensions are expressions of the degree to which a search is *interactive*.

Role of the results of a single search step in the entire search. What does the user do with the search results? She may consider the results of one search step merely as a piece of a mosaic being built through many steps - the berry-picking approach to assembling information typical of (but by no means exclusive to) hypermedia searches (The term berry-picking was actually coined by Bates 1989 in a paper suggesting a highly interactive and incremental approach for **bibliographic** retrieval). Or the user might use the object(s) found as stepping stones for further searching, without or in addition to using them as sources of information in and of themselves, also fairly typical of hypermedia searches. Or the user might look at the results of a search step as the final answer set giving all the needed information, the system having done all the work assembling the answer without assistance from the user through interaction.

The number of search steps in an entire search (information seeking episode). The prototypical hypermedia search has many search steps, the typical bibliographic search just one.

Table 2. **Dimensions for analyzing searches and systems** (simplified)

Dimension	**Typical hypermedia search**	**Typical bibliographic search without interaction**
Type of starting object(s)	A paragraph, a picture, an audiovisual object	A search key: A subject descriptor, person, organization, etc.
Method of finding good starting objects	Natural encounter during a search	Deliberate selection
Role of starting object in the entire search	Mostly objects of value in their own right	Mostly objects used only for searching
Method of specifying starting object(s) and link types	Object currently examined as default starting object Selecting link type from display on the screen	Entering elements from the keyboard
Type of target objects found	Full text, picture, etc.	References to documents
Role of the results of a single search step in the entire search	Piece of a mosaic being built Stepping stone for further searching	Final answer set of documents to be read
The number of search steps in an entire search	Many	Few
Number of objects looked for or found in one search step	Few (1 - 5)	Many (10 -20 and up)
Number of objects examined upon retrieval before moving on to the next search step in an entire search	Few	Many
Granularity: Size of objects looked for or found	Individual paragraphs, images, or sound objects	Whole documents
Completeness and complexity of search specification for each search step	Partial, often implicit Simple	Complete, often carefully worked out Complex
Ability of the user to augment the data base	Often built in as an essential part of the system	Usually no-existent

Number of objects looked for or found in one search step. In the typical hypermedia search, the user is looking for just the next object to look at. The search following links from the object being examined generally leads to one or a few other objects. In the typical bibliographic search, the user is looking for a list of references, perhaps 10 or 20, to be used as the final answer set; a search may retrieve many more.

Number of objects examined upon retrieval before moving on to the next search step. In a hypermedia search the user often examines only the first of several retrieved objects and immediately uses that object as the starting point in a next search step. The user of a bibliographic search typically selects and examines a number of relevant documents from the retrieved set and initiates additional search steps only if the information received is not sufficient or points out another information need.

Granularity: Size of objects looked for or found. Very specific access to individual paragraphs, images, or sound packages (such as a passage within a piece of music) vs. searches for whole documents (Books, book chapters, journal articles, etc.).

Completeness and complexity of search specification required in each search step. The definition of a search step developed so far - starting from a single starting object and selecting all target objects reachable by a given type of link - is oversimplified. While this simplicity will do for some searches, others use a much more complex search specification, such as a Boolean query formulation or a query formulation using a relationship with multiple arguments. Devising such a query formulation requires effort. The typical bibliographic ISAR system works best with a complete and carefully worked out search specification. In the typical hypermedia search the user is not even aware that she uses search specifications as she selects a link type for a search starting from the object being examined. A search may consist of one very complex step or of many simple steps.

Ability of the user to augment the data base. Some hypermedia systems allow the user to add new objects and new links, either public or private; searching and the user's own work of reading, note-taking, commenting, and writing are not separate activities but all integrated in the interaction with a hypermedia system. Allowing user feedback to update and improve the database would also be a great benefit in bibliographic and fact retrieval but is much less common there.

Introductory example

The example of a digital library design given on the next three pages illustrates the approach discussed in sections 3 and 4. Having this example in mind should make it easier to grasp the principles as they are laid out in the text.

The example consists of the following parts:

> The conceptual schema for the envisioned digital library in a graphical display.
>
> A search for a single concept in a screen mockup. The single concept is expressed as an OR combination of descriptors from the Alcohol and Other Drug Thesaurus, making use of the descriptor relationships from the thesaurus
>
> A search for two concepts ANDed.

3 The structure of a unified information structure management system

A **information base** (defined to include database, knowledge base, bibliographic database, and hypermedia base) is made up of **objects** (also called **entities** or **nodes**), **relationships** (also called links), **neighborhoods** (also called **regions**, **clusters**, or **virtual composites**), **queries**, and **paths and scripts** (See Table 3 for definitions). The data base literature uses the terms entity and relationship, the hypermedia literature uses object and link.

Table 3. **Elements of information structure**

Objects (entities, nodes)

Relationships (links)

Neighborhoods

A neighborhood is any group of objects, particularly a group identified through relationships or links with one or more other objects or neighborhoods, together with the relationships or links that exist among the members of the group. A search results in a neighborhood .

Queries

A **query** specifies a search and thus a neighborhood; that specification is dynamic: the neighborhood contains the objects meeting the search criteria at the time the query is invoked.

Paths and scripts

A path is a special type of object or node that defines a neighborhood of objects and specifies a sequence of these objects. (A path leads the user along a given sequence of paragraphs and figures in a hypermedia base, thus emulating a traditional document.) A script is a special type of object consisting of instructions that orchestrate the display of other objects to a user.

Neighborhoods, queries, paths and scripts, and relationships can be treated as objects.

3.1 Objects (entities, nodes)

A standard hypermedia system covers media objects - paragraphs, pictures, sound objects - without differentiation of object types, and many systems do not differentiate between link types either. A typical database covers objects such as persons, organizations, food products, or technical products and uses distinct relationship types to express data about them. A generalized database/hypermedia base structure can cover a wide variety of object types and represent almost nay type of information (see Table 4 for examples).

Hypermedia systems with untyped objects and links do nor represent meaning except at a very coarse level. However, the future belongs to systems that deal with meaning and can thus provide more intelligent support. For example, Carlson 1990 presents a system that deals with strategies, objectives, issues and their causes and allows for sophisticated retrieval and processing through its rich semantics of object types and link types. A hypermedia system in the area of mathematics might include assertions (mathematical theorems) as object so that they can be linked to other objects, such as persons or proofs.

Table 4. **Object types. Examples**

Media object (text, graphics, sound) of any size	Person or organization
Path	Concept (subject, topic)
Database, data set	Computer program
Assertion	Organism
Problem	Food product
Strategy	Building
Objective	Work of art
Issue	Technical product, device (anything from a screw to an engine to an entire airplane)
Situation, circumstance (which may be the cause for an issue)	
	Person

3.2 Relationships (links)

An information base consists of stored objects (or references to objects that exist outside) and statements about these objects. The statements consist of a relationship with one or more arguments, with each argument slot filled by a specific object value, for example *Smith 1991 supports Miller 1990* (a statement about two media objects), or *Brownie has-ingredient Chocolate* (a statement that makes an assertion about the world directly). A more complete form of the second statement, using a relationship with 6 arguments, is *Brownie has-ingredient [Chocolate, rank 3, percent of weight 20%, percent of dry 25%, [purposes: improving taste, chewy consistency]]*.

A relationship that has two arguments is called **binary**; it establishes a **link** between its two arguments. Hypermedia systems use binary relationships called links. Database and expert systems use many binary relationships but also relationships with three or more arguments. We use the specific term link is used when the emphasis is on binary relationships, particularly in the context of using these relationships for navigation in a hypermedia base, and the more general term relationship whenever the context requires it.

Table 5 gives examples of relationship types; these are discussed in the following to illustrate the wide range of information that can be handled in a unified information structure.

Some relationships can be applied to many types of objects. For example, *produced-by* can link any type of object, such as a media object, computer program, or food product to a person, organization, machine, or even an event. *Contradicted-by* could link a mathematical theorem to a counterexample or the campaign statement of a politician to a set of facts. *Continued-by* is a navigational link that applies primarily to media objects but could also be used to indicate the sequence of elements in a classification or the sequence of courses in a meal; *continued-by* has three arguments so that it can express the fact that an object can participate in many paths and the continuation object depends on the path. Finally, the relationship *has-narrower-term* illustrates that thesaurus relationships are no different from any other relationships in the system, such as *deals-with*, and can be used the same way in searching.

Other relationships are specific to an object type. The following examples are specific to media objects and data sets: *Has-summary* leads from a long to a short form (a reader could also pursue such a link the other way if she needs more detail than the summary provides). *Has-prerequisite* serves a user who does not understand a media object and needs to acquire additional knowledge before reading it; *has-prerequisite* is also useful for constructing a didactically sound path. *Has-same-content-as* can be used by the system to select from several media object that say the same thing the one that best fits the user's language ability and cognitive style, to warn a user who is about to read something that merely repeats what she just read, or, conversely, suggest a different presentation if the user has trouble understanding.

Table 5. **Relationship types (link types)** (types defined based on meaning)

General relationship types	**Relationship types applying primarily to media objects**
produced-by	
has-target-audience	*has-prerequisite*
supported-by	*has-summary*
contradicted-by	*has-same-content-as*
praised-by	*is-simplified-from*
criticized-by	*is-later-version-of*
object deals-with topic/ *topic discussed-in object*	*is-written-in*
described-in	*describes* (the reciprocal of *described-in*)
includes	*illustrates*
has-special-case	**Relationships on food products**
has-narrower-term	Food product *has-ingredient* [Food product, rank, total %, solids %, [purpose list]]
[object, path] *continued-by* object	Food product *underwent-process* [Process, equipment, temperature, duration, place/stage, sequence no., [purpose list]]
Relationships on issues, objectives, strategies	
media object *helpful-for* problem/ problem *help-in* media object	Food product *has-constituent* [ChemSubst, rank, total %, solids %]
circumstance *causes* issue	
objective *addresses* issue	**Relationships for user model**
objective *addresses* cause	person *has-interest* [topic, intensity]
strategy *aims-at* objective	person *has-knowledge-of* [topic, depth]
strategy *assigned-to* organization	person *reads-language* [language, fluency]
	media object *readable-by* [object1, object2]
	media object *processable-by* object

Has-ingredients is used to model actual data about a food product. This relationship is complex to accommodate the complexity of the data. *Brownie has-ingredient Chocolate* does not tell the whole story; one also wants to know that chocolate has rank three in order of predominance, amounts to 20% based on total weight, 25% based on solid weight, and has the purposes *improving taste* and *chewy consistency*. If you thinks this relationship is too complex, consider the ingredients of a pudding: If it is made with fluid milk, milk is listed as the first ingredient on a total weight basis, but if it is made with dry milk, milk is listed as the third or fourth ingredient, even though the composition is essentially the same, as could be seen from ingredient information on a dry weight basis. Statements formed with the relationship *has-ingredient* serve many functions: they inform the user about the ingredients of a food, they let the user find all foods containing chocolate chips or chocolate in any form, they let the user find all ingredients that have been used to achieve *chewy consistency*, etc.

An information system should also include **user models**, that is data about users (persons, organizations, machines), their characteristics and needs. User model data are often kept separately, but the system envisioned here includes them in the information base just like any other data, represented just the same way through relationships. So are data about the relationship between users and other objects. For example, *readable-by* refers to the physical readability of a media object by object1 (a person or device, such as a computer) with the aid of object2 (another device, such as a microform reader or a disk drive of a certain kind). *Processable-by* refers to ability of the agent object to do something with the data read, e.g., a file being processable by a computer program or a text being processable by a person (I can physically "read" a printed Swahili text, but I cannot process it).

The groupings in Table 5 should not be construed too narrowly. For example, *has-prerequisite* could be used to link computer programs A and B where running A requires B. *Written-in* applies to both documents and computer programs. *Has-ingredient* applies also to drugs.

Object types and relationship types together specify what kind of data can be expressed in a hypermedia base/data base; they define the **conceptual schema** of the information base.

Note. On the surface, a statement about the ingredient of a food product may look quite different from a link between two text objects, such as Document A *criticizes* Document B. But they can be expressed in exactly the same format, and there are many advantages in doing so. An information structure management system combines the selective and inferential power of database and expert systems with the interface and navigational power of hypermedia systems. One integrated conceptual schema is more parsimonious than several conceptual schemata, since many relationship types apply across content areas. More importantly, an integrated schema allows for useful relationships that would be much more cumbersome in separate systems. For example, a user could find all food products having 20% or more chocolate, select one of them, follow a link to a text describing it or a picture depicting it; or he could define the neighborhood of all media objects that describe the selected product or any of its ingredients. Or a user could apply a program that computes the nutrient content data of a prepared food product based on ingredient data and nutrient data for the ingredients and then pick a nutrient on the list and retrieve all media objects that discuss the importance of this nutrient.

Table 5. **Relationship types (link types)** (types defined based on meaning) (repeated)

General relationship types

produced-by

has-target-audience

supported-by

contradicted-by

praised-by

criticized-by

object deals-with topic/
topic discussed-in object

described-in

includes

has-special-case

has-narrower-term

[object, path] *continued-by* object

Relationships on issues, objectives, strategies

media object *helpful-for* problem/
problem *help-in* media object

circumstance *causes* issue

objective *addresses* issue

objective *addresses* cause

strategy *aims-at* objective

strategy *assigned-to* organization

Relationship types applying primarily to media objects

has-prerequisite

has-summary

has-same-content-as

is-simplified-from

is-later-version-of

is-written-in

describes (the reciprocal of *described-in*)

illustrates

Relationships on food products

Food product *has-ingredient*
[Food product, rank, total %, solids %,
[purpose list]]

Food product *underwent-process*
[Process, equipment, temperature,
duration, place/stage, sequence no.,
[purpose list]]

Food product *has-constituent*
[ChemSubst, rank, total %, solids %]

Relationships for user model

person *has-interest* [topic, intensity]

person *has-knowledge-of* [topic, depth]

person *reads-language* [language, fluency]

media object *readable-by* [object1, object2]

media object *processable-by* object

3.3 Neighborhoods and queries

A neighborhood (also called region or cluster) is any group of objects together with the relationships that exist among the members of the group (Table 6). More generally, a neighborhood can in turn contain neighborhoods. A neighborhood is not merely a set since it also includes the relationships between the members. The term was chosen in keeping with the spatial and navigation metaphor often associated with hypermedia systems.

Usually the members of a neighborhood are selected based on their relationship with one or more other objects. In the simplest case, a neighborhood consists of all objects that can be reached from a starting object through links of a given type - the neighbors of the starting object. For example, the neighborhood consisting of all documents criticizing a given document, which is assembled by starting from the given documents and following links of the type *criticized-by*.

More generally, there is a direct association between neighborhood and search: A query leads to a neighborhood, and to each neighborhood corresponds a query. Including queries as objects allows for any kind of statements one wishes to make about a query and thus about the neighborhood defined by the query.

Table 6. **Neighborhoods**

A neighborhood is any group of objects or neighborhoods, particularly a group identified through links with one or more other objects, together with the relationships that exist among the members of the group.

Examples

All objects *dealing with* a given topic

All objects *produced-by* a given person P, e.g. all music pieces composed by Gershwin

Search: person P *produces* object

All documents *criticizing* a given document D, together with the relationships among them (such as one of these documents citing another)

Search: document D *criticized-by* document

All pictures *illustrating* a given paragraph

All persons *producing* a given document

All food products *having-ingredient* a given food product

All food products that occur in a *has-ingredients* statement in the ingredient role associated with the purpose chewy consistency

All objects selected by the user during the course of a search for later examination and processing

3.3.1 Neighborhoods based on hierarchical relationships

Neighborhoods based on hierarchical relationships between objects are very important for searching (Table 7). This section gives a brief definition, a fuller discussion of their role in searching is given in Section 4.7. Following a hierarchical relationship downward yields an *offspring neighborhood*. The first three examples are from a bibliographic context and need no further explanation. An offspring neighborhood of media objects lets the searcher specify a scope for a Boolean search condition; he can require two subject descriptors to co-occur in the same paragraph (a very strict condition) or in an entire book (the neighborhood of all paragraphs included in the book, a much looser condition). An offspring neighborhood of concepts implements inclusive (hierarchically expanded) searching, searching for a descriptor and all its narrower descriptors (easily done in MEDLINE and other systems using commands such as *explode* or *cascade* or, in DIALOG, an ! following the descriptor). An offspring neighborhood of assertions supports a thorough search for evidence disproving the assertion: In a search for counterexamples that disprove a general mathematical theorem, look for counterexamples for any of the special cases as well. These examples illustrate the power of the general concept of an offspring neighborhood; one formalism handles three seemingly different situations.

The converse of an offspring neighborhood is an **ancestor neighborhood**. The ancestor neighborhood for a paragraph consists of all the media objects in which the paragraph is included: the article containing the paragraph, the journal issue containing the article, the journal volume to which the issue belongs, and finally the journal itself.

Ancestor neighborhoods provide an approach to hierarchial inheritance and are thus important in searching. Consider a medical journal article dealing with hepatitis. A paragraph in that article might say "Five patients were treated with ... . Three patients responded to treatment in 7 days ... ". A user searching for paragraphs on the treatment of hepatitis would not find the paragraph in question unless the system considers the hepatitis link from the document as a whole. For another example assume that the user is interested in an **explanation of AIDS for the educated lay person**. The system must select articles not only based on their subject matter but also based on the intended audience. The intended audience can often be seen from the journal, such as Scientific American, in which an article appears. All articles in Scientific American should inherit the attribute *has-target-audience* Educated lay person.

An object that heads an offspring neighborhood plays an important role as **organizing node**. Links associated with an organizing node are either non-inheriting or inheriting. **Non-inheriting** links pertain to the organizing node as such or to the totality of the nodes under it; for example, Scientific American *deals-with* All of science is a non-inheriting link since it applies only to the journal (the totality of articles) as a whole, not to individual articles. **Inheriting links** are introduced as a space-saving device. An inheriting link applies to every object in an offspring neighborhood formed along a given relationship; for example Scientific American *has-target-audience* Educated lay person is an inheriting link since it applies to each and every article *included-in* Scientific American. It is more efficient to assign the *has-target-audience* once to Scientific American an inhering link than to assign it to every single article.

Table 7. **Neighborhoods based on hierarchical relationships**

Offspring neighborhoods

A book and its chapters (one level down)

A book and its chapters, subchapters, sections, paragraphs (all the way down)

A journal and its volumes
A journal volume and its issues
A journal issue and its articles.

A concept and its narrower concepts.
Example: {cognitive processes (the top concept), apperception, cognitive mapping, serial ordering, associative processes, mental concentration, ideation, thinking}

An assertion and its special cases

Ancestor neighborhood

A paragraph in a journal article, the article itself, the issue, the volume, the journal

Paragraph linked to treatment
Journal article linked to hepatitis and to a person as author
Journal linked to written for the general public

3.4 Neighborhood links

Neighborhoods give rise to neighborhood links, which are essential in searching. Neighborhood links are defined in terms of links between atomic objects as follows (Table 9): A *from-neighborhood link* is any link from any object in that neighborhood. As an example, consider a search for all text objects criticizing a given document; such text objects can be reached from the criticized document following *criticized-by* links. The search should find text objects criticizing the document as a whole **or any part of it**; thus it should start from the offspring neighborhood arising from the top document node and follow the relationship *criticized-by* from any object in that offspring neighborhood. A *to-neighborhood link* is any link that ends up in an object in that neighborhood. There can be a neighborhood on one or on both ends of a link. A good example of a link from a neighborhood to a neighborhood is a journal-to-journal citation link that exists whenever any article from journal A cites any article from journal B.

Table 9. **Neighborhood links**

From-neighborhood link

A link from any object that is in neighborhood N

To-neighborhood link

A link to any object that is in neighborhood N

This finishes the discussion of the basic elements of an information structure. The remaining sections of Chapter 3 add elements that make for a richer structure to express still more types of information or knowledge and do more powerful searches.

3.5 Connections

A connection is a chain of several links. A link takes the searcher only one step, but often he wants to go several steps at a time; then he takes a connection. For example, the chain

[Greek vase *has-instance* Object-1, Object-1 *depicted-in* Slide-567]

leads from the descriptor Greek vase to a slide depicting one. The **connection type** is

[Concept *has-instance* Object, Object *depicted* in Slide].

A connection type is defined as a chain of links, each link belonging to a specified link type. We include "chains" consisting of a single link, called **direct connection**; a connection consisting of two or more links an **indirect connection**. For examples see Table 10.

An indirect connection can be identified by a name, and that name can then be used just like a simple link, except that the system - transparent to the user - follows several steps.

Often an indirect connection has a meaning equivalent to a direct connection. For example,

[Person P *affiliated-with* Organization O,
Organization O *has-phone* Phone number N]

is for many searches equivalent to

Person P *has-phone* Phone number N

The system should tell the user who employs a direct connection about equivalent indirect connections, which offer added possibilities for reaching relevant objects. To do so, the system must know such equivalencies and other relationships between connection types.

In some important cases, the length of a connection chain is not known beforehand, for example, if the searcher wants to chain hierarchical *includes* relationships all the way down. For this situation, we introduce the (*) notation illustrated in the following example:

[Subject-1 *includes* Subject-2 (*)]

Starting from Subject-1, this connection leads to other subjects included in Subject-1, an arbitrary number of levels down. For example,

cognition and memory *includes* **cognition**
  cognition *includes* **cognitive processes**
    cognitive processes *includes* **apperception**
    . . .
    cognitive processes *includes* **thinking**

Table 10. **Connections and relationships between them**

Sample connection type

[Concept *has-instance* Object, Object *depicted-in* Slide]

Direct and indirect connections

A **direct connection** connects two objects through a single link

Examples

Connection 1: [Media object A *commented-by* Media object B]

Connection 2: [Media object A *criticized-by* Media object B]

Connection 3: [Person P *has-phone* Phone number N]

An **indirect connection** connects two objects through two or more links via intermediate objects

Examples

Connection 4: [Media object A *proposes* Theory T,
Theory T *commented-by* Media object B]

Connection 5: [Media object A *proposes* Theory T,
Theory T *criticized-by* Media object B]

Connection 6: [Person P *affiliated-with* Organization O,
Organization O *has-phone* Phone number N]

Relationships between connections (knowledge used for intelligent search support)

Connection 2 ***isa*** Connection 1

Connection 5 ***isa*** Connection 4

Connection 4 ***equivalent-to*** Connection 1

Connection 5 ***equivalent-to*** Connection 2

Connection 6 ***equivalent-to*** Connection 3

3.6 Statements about statements or statements as objects

As discussed in Section 3.2, an information base consists of statements created by relating one or more objects using a given relationship type. Statements can themselves be objects that can participate in relationships. For example, if there is a statement *data set A supports assertion B*, and document C disputes that statement, the relationship of C is neither to the data set A alone nor to the assertion B alone but to the statement linking them. Similarly, if there is a statement *document A criticizes document B* and C takes issue with that criticism, the proper linkage is not from C just to A or just to B but to the critique link between the two.

The truth of statements in an information base is seldom absolute. Thus it should be possible to make statements about statements indicating a **surety or probability value**. More precisely, the surety of a statement may depend on whom you ask; to account for this, we can introduce the relationship

statement *is-believed-by* [object, strength]

where the object could be a person or the system itself (for belief-strength indications representing the general consensus of the system builders).

3.7 Knowledge about the structure of knowledge: Relationships between object types and relationships between relationship types. Object types and relationship types as objects

When a searcher specifies *text document* as the object type to be found, the system should also find *journal articles*, *government reports*, etc. Likewise, when a user after reading a paragraph wants to find paragraphs commenting on it and thus specifies the relationship type *commented-by* to lead to other paragraphs, the system should also use the more specific relationship types *supported-by* and *criticized-by*. In order to do this, the system must know the hierarchy of object types and the hierarchy of relationship types.

The simplest way to represent this hierarchy is to treat object types and relationship types as objects and define relationships between them. On a formal level, relationships between object types are no different from relationships between any other object, and the system should treat them that way both internally and in the user interface. The operation to see the object types that fall under *document* should be no different from the operation to see all objects included in a given book or all concepts narrower than *system of government*.

3.7.1 **Relationships between object types**

Table 11 shows some relationships between object types (entity types).

Table 11. **Relationships between object types**

Journal article *isa* Document	Visual object *isa* Audiovisual object
Government report *isa* Document	Sound object *isa* Audiovisual object
Text object *isa* Media object	Audiovisual object *isa* Media object
Map *isa* Visual object	City *is-part-of* address

3.7.2 **Relationships between relationship types**

Table 12 shows some relationships between relationships types (link types). The *commented-by* example already illustrated the use of a relationship type along with the relationship types under it (in the example *criticized-by* and *praised-by*). This example works also the other way around: A user who starts from a paragraph and wants to find all paragraphs criticizing it, should follow not only *criticized-by* but also *commented-by*, using an ancestor neighborhood of relationship types.

Table 12. **Relationships between relationship types**

Media object *criticized-by* Media object	***isa***	Media object *commented-by* Media object
Media object *praised-by* Media object	***isa***	Media object *commented-by* Media object
Topic *discussed-in* Object	***inverse-of***	Object *deals-with* Topic

Universal relation or link as the top of the isa hierarchy

In indexing: relationship type not specified

In searching: any relationship type acceptable

4 Searching

Sophisticated and powerful searching is the purpose of the whole elaborate structure. Searching can at times be a very complex task; however, much of this complexity can be hidden from the user who is willing to accept choices by the system, and the system can help the user designing his own searches by guiding him through the process and displaying menus of options where applicable. Complexity behind the scenes adds power and makes the user's life easier.

4.1 Definition of search

A total search consists of one or more **search steps** (Table 13). For example, a user begins with a problem she needs to solve; the *help-in* link leads to media objects that can help in solving the problem. In a second step she starts from one of these media objects and uses the *has-prerequisite* link to find media objects she needs to acquire required background knowledge . She selects one of these, finds it too difficult, and uses the *has-simplified-form* link to find a media object that is easier to read. To give another example, a user starts with a term and, using links among terms and concepts, finds the preferred term as well as broader and narrower concepts (a concept neighborhood). In the next step she starts from this concept neighborhood and, using the *discussed-in* link, finds media objects.

Each search step starts from something that is known, often a **starting object or neighborhood** and leads to one or more **target objects or neighborhoods** that, one hopes, contribute to the goal. Each object or neighborhood found along the way may contain some (or all) of the information needed and/or may serve as a stepping stone to further information, as the starting point in the next search step.

Table 13. **Search: Definition**

Total search = series of search steps, each leading

> from something known (for example, a known object or neighborhood)
>
> to something unknown but expected to be helpful (one or more helpful target objects or neighborhoods)

An object or neighborhood encountered may be helpful because it

> contains some of the information needed and/or
>
> serves as a stepping stone to other objects or neighborhoods.

Searches differ in the method used to get from the known to the targets. In the search examples given above, the method is navigation based on links (binary relationships). The user specifies his query by selecting an object on the screen as the starting object and selecting a type of link to follow. The same result could be achieved by formulating an explicit query. More complex cases, particularly searches using higher-order relationships, require an explicit query. The distinction between search as navigation and search as query-based retrieval is more a matter of perspective than of the basic nature of the search, They are two different metaphors for the same process. The search procedures corresponding to these metaphors are different, but this is a matter of degree rather than absolute difference (see Section 2). The nature of the starting object (for example, media object vs. subject descriptor) may also play a role in determining the perspective. As the discussion and the examples in the next section make very clear, the principle is always the same: The user starts from an object or neighborhood and, following a given link type, finds other objects or neighborhoods.

The typical hypermedia search uses the navigation metaphor. The starting object is the media object (a paragraph of text, a picture, etc.) currently on the screen, and the user expects to find just one, or at most just a few, media objects to look at next. The media object found then becomes the starting point for the next search step, and so forth. A system might facilitate this process by showing on the screen a "map" of the links between objects, an outline being a special case.

The typical bibliographic search uses the query metaphor. The starting object is a subject descriptor or an author name, and a link type is often specified. The user won't be surprised finding 30 or 50 or even 200 documents. A query can also be more complex, combining several starting objects and requiring that a target be reachable from all. Query formulations are also required for searching data represented through relationships with many arguments, for building elaborate inferences into a search, or for deriving new data through processing. Such query formulations can get quite complex.

The result of a search is a set of objects or neighborhoods. Thus the result of a search is a neighborhood, and the query can be seen as the top node of that neighborhood.

The following sections discuss navigation searches moving from the simple - single-criterion searches starting from a single object - to the complex - combination searches with neighborhoods as targets and searching with hierarchical inheritance.

4.2 Specification of a search based on relationships

This section discusses in detail how to specify a search based on using relationships from one or more starting objects to identify target objects. For ease of explanation the discussion uses the navigation metaphor, but the concepts apply more broadly to any search based on relationships.

In each search step the user must tell the system what he needs through a **search specification (query formulation)** consisting of the four elements shown in Table 14.

Table 14. **Elements of a search specification (query formulation)**

The general type of target objects or neighborhoods

One or more search criteria, each consisting of

> A starting object or neighborhood
>
> A connection condition: the permissible connections from the starting object or neighborhood to the target objects or neighborhoods.

The format in which the target objects or neighborhoods found should be displayed.

The **target object or neighborhood specification** expresses what the user wants to find - concepts, persons, media objects, offspring neighborhoods of media objects, ancestor neighborhoods of media objects, assertions, etc. Specifying neighborhoods as targets is significant in a Boolean AND search as explained below.

As a **starting object or neighborhood** for the first search step the user may employ either an object she knows (a problem, a media object, a term, a person) or she may select an object from an initial menu. Or the user may enter a **starting object type**, such as **problem**, and in return be shown a menu of possible values to select from.

How to get from the starting object to the target objects wanted is specified in a **connection condition** which consists of **one or more permissible connection types**.

Often the user's purpose is achieved best by allowing any of a number of related connection types; in the examples that follow, several connection types to be used in parallel are shown in {}. As an example, assume a user is looking at a given media object A and wants to find media objects in which A is criticized. The user should use the following connection types in parallel:

{Connection 1 [Media object A *commented-by* Media object B],
Connection 2 [Media object A *criticized-by* Media object B],
Connection 4 [Media object A *proposes* Theory T,
Theory T *commented-by* Media object B],
Connection 5 [Media object A *proposes* Theory T,
Theory T *criticized-by* Media object B]}.

Searches differ in formal complexity: A search can use a single search criterion or a combination of search criteria. It can start from a single object or from a neighborhood. The targets specified can be single objects or they can themselves be neighborhoods. The following discussion expands on search criteria and their combination.

4.3 Single-criterion search starting from a single object.

This is the simplest kind of search. In a single-criterion search, a single connection going into a target object is sufficient to select that object (Table 15).

Table 15. **Single-criterion search starting from a single object**

Examples

Starting object: A subject descriptor

Target objects: Media objects

Connection condition: *discussed-in*

Starting object: A media object

Target objects: Descriptors

Connection condition: *deals-with*

(The descriptors found might be useful as starting points in further search steps)

Starting object: A building

Target objects: Media objects

Connection condition: {*depicted-in*, *discussed-in*}

Starting object: A book (only the top node for the book as a whole, excluding the nodes for parts of the book)

Target objects: All objects of any type

Connection condition: *Universal link* (any link type)

Starting object: A book (top node only)

Target objects: All objects of any type

Connection condition: Any one-link or two-link connection

4.3.1 **Single-criterion search starting from a single object with neighborhoods as targets**

In the examples in Table 15 atomic objects are specified as targets for selection. Sometimes on may want to retrieve whole neighborhoods, for example whole documents rather than single paragraphs (Table 16). Put differently, the whole document should be shown if any of its subordinate objects (sections, paragraphs) are found. In that case, the search targets can be specified as document offspring neighborhoods. The system finds any media object that fulfills the search criterion and then identifies the whole document to which it belongs. Thus specifying neighborhoods as targets of a single-criterion search does not affect retrieval per se but what information is displayed once an object is found. This is in contrast to combination searches where, as we shall see, it makes a big difference for retrieval whether atomic objects or neighborhoods are specified as targets.

Table 16. **Single-criterion search starting from a single object with neighborhoods as targets**

Example 1

A search for all whole documents that deal with a given descriptor, such as **Drug treatment**. The descriptor could be assigned to the document as a whole or to any section or paragraph included in it.

Starting object: Subject descriptor **Drug treatment**

Targets: Offspring neighborhoods from whole document nodes

Connection condition: *discussed-in*

A *discussed-in* link from the descriptor to any one element (section, paragraph) of a target neighborhood is sufficient.

Display: Show the whole document node, under it any sections that either has the descriptor assigned to it or includes a paragraph that has the descriptor assigned to it. Highlight the objects to which the descriptor is assigned.

Example 2

A search for all food products to which a person (or person class) is allergic. This search is complex because *allergic-to* statements can refer to food products, such as milk, or to a chemical substance, such as lactose, and because the offending allergen may be in any ingredient of the food. In other words, this search must find all food products that directly or indirectly contain something the person is allergic to.

Starting object: A person

Targets: Neighborhoods consisting of a food product, all food products reachable from it through *has-ingredient* links or chains of *has-ingredient* links, and all chemical substances reachable from any of these food products through *has-constituent* links

Connection condition: *allergic-to*

An *allergic-to* link from the person into any element of a target neighborhood is sufficient.

Display: Show the whole food and the offending ingredients or constituents.

4.4 **Single-criterion search starting from a neighborhood**.

A single-criterion search starting from a neighborhood can start from any object in the neighborhood (Table 17); it is an implied OR search which leads to many more objects than a search starting from a single object in the neighborhood.

The first example is an inclusive subject search: The system first builds the offspring neighborhood consisting of a descriptor and all its narrower descriptors and then starts from any descriptor in that neighborhood (rather than be limited to the one descriptor that heads that part of the hierarchy) to find media objects via the connection type *discussed-in*. This is an implied ORing of the starting descriptors. In searching MEDLINE this can be achieved by prefixing a descriptor with EXPLODE.

The second example illustrates the usefulness of this method even better. An assertion is contradicted when any of its special cases is contradicted. So to see all objects that contradict an assertion, start from the assertion itself and find all objects linked via *contradicted-by*, but then also start from each of the special cases and follow the same link, that is, start from the offspring neighborhood.

The third example shows how to find all the media objects linked to a given book: Start from the book node itself but also from all the nodes dependent on the book node via a chain of *includes* links, that is, an offspring neighborhood.

The fourth example defines a geographical area as a neighborhood of places and then looks for all businesses located there.

This type of search is very powerful since the user can define any starting neighborhood she wants to, using a search of arbitrary complexity. In effect, the user has the system do a preliminary search to assemble the starting neighborhood for the main search. Some systems provide shortcuts for defining certain kinds of neighborhoods. For example, when searching MEDLINE one can use EXPLODE (or, in Dialog, !) to define the offspring neighborhood of a subject descriptor (the set consisting of the subject descriptor and all descriptors under it in the hierarchy).

Table 17. **Single-criterion search starting from a neighborhood**

Any of the objects in the starting neighborhood can serve as a starting point for the search, vastly increasing retrieval (implied Boolean OR).

Examples

Starting neighborhood:	A subject descriptor and all its narrower descriptors (MEDLINE EXPLODE) Example: {cognitive processes (the top concept), apperception, cognitive mapping, serial ordering, associative processes, mental concentration, ideation, thinking}
Target objects:	Media objects
Connection condition:	*discussed-in*
Starting neighborhood:	An assertion and all its special cases
Target objects:	All objects of any type
Connection condition:	*contradicted-by*
Starting neighborhood:	A book and all its chapters, sections, and paragraphs.
Target objects:	All objects of any type
Connection condition:	The universal link
Starting neighborhood:	A city and all locations in a 100 mile radius
Target objects:	All businesses
Connection criterion:	*is-location-of*

4.5 Combination search (Boolean AND or weighted search)

Single-criterion searches are simple but often overly general. More specific selection requires using two or more search criteria simultaneously - a **combination search** (Table 18): For a target object to be selected in a combination search it must be reachable by two or more connections, normally coming from different starting objects. A combination search with direct connections each using a single link type is a straightforward Boolean AND. A combination search with connections of arbitrary length with arbitrary link types is called **spreading activation** in the context of semantic networks. The formalism described here lets the user specify anything in between.

Table 18 gives some simple and familiar examples. The first example is a plain Boolean AND search where the searcher specifies two descriptors and wants documents that deal with both.

The second example shows the reverse: Starting from two objects known to be relevant, she can find all descriptors that are used in indexing both objects. Those should be good candidate descriptors to find more relevant objects, much better than the descriptors used in indexing just a single relevant object. With the general search operator suggested here this second search follows exactly the same format as the first.

Example:

Documents A and B are known to be relevant

Document A *deals-with* alcohol, apperception, impairment, teenagers

Document B *deals-with* apperception, spatial, sex differences

Clearly **apperception** is a much more plausible descriptor for finding more relevant documents than **alcohol** or **sex differences**.

In an extension of this method, the searcher specifies the set of all known relevant objects as the starting neighborhood and the system ranks descriptors by the number of documents they index.

Table 18. **Combination search (Boolean AND) with single objects as targets**

Find all objects that satisfy two or more search criteria simultaneously.

Examples:

Starting from two descriptors, follow the link type *discussed-in*. The objects reached from **both** descriptors are the objects that deal with both.

Starting from two objects, follow the link type *deals-with* . The descriptors reached from **both** objects have been used in indexing both, as in the example already given in the text:

Document A *deals-with* alcohol, apperception, impairment, teenagers

Document B *deals-with* apperception, spatial, sex differences

Descriptor found: Apperception

Starting from two documents, find all documents *citing* both (co-citation).

4.6 Combination searches with neighborhoods as targets

With combination searches it makes a big difference whether the targets are restricted to atomic objects or whether neighborhoods are admitted and, if so, what type of neighborhood (Table 19). Searching for neighborhoods uses to-neighborhood connections. A to-neighborhood connection exists whenever there is a connection to any element of the neighborhood, greatly increasing the possibilities for simultaneous satisfaction of two search criteria. By proper definition of target objects or neighborhoods, the searcher can require, for example, that two terms co-occur in the same paragraph, in the same book chapter, or in an entire book, or in an entire journal issue, or in the neighborhood formed around a document and all documents it cites, or in all documents that originated from a research project. The examples show that this concept has very broad application.

Specifying neighborhoods as the targets of a search requires definition of the kind of neighborhood desired, such as offspring neighborhoods starting at whole document nodes or citation neighborhoods starting at any media object following citation links one step (or two steps, or n steps). Fortunately, defining target neighborhoods is just the same as search specification: Instead of specifying an individual starting object, specify a starting object type and the connection type(s) to be used to reach other elements of the neighborhood.

Present hypertext systems do not allow the specification of neighborhoods as search targets. Thus it is not possible to conduct a whole-document level Boolean search even if Boolean searching is implemented. Many hypertext systems allow Boolean searching for an "index search" but not for a search using the typical hypertext links. In bibliographic ISAR systems the search level supported depends on the type of descriptor used. Subject descriptors assigned through explicit indexing are assigned to documents as a whole. With text words, one can specify as the search target a whole document or a paragraph or a sentence.

The concept of a neighborhood type as search target is so fundamental and has such broad applications that two more examples are in order. The first example (Table 20) generalizes the idea of finding descriptors in common to two relevant objects and using them as descriptor candidates for further searching. The search now targets **descriptor neighborhoods** that can be reached from two relevant objects. In this example, the system identifies broad descriptors in common to two objects even though each broad descriptor is represented by a different one of its narrower descriptors in each of the two documents. Such descriptors can be extremely useful for further search.

Table 19. **Combination search (Boolean AND) with neighborhoods as targets**

A neighborhood satisfies a search criterion if any of its objects satisfies it.

Examples:

Starting from two descriptors,
find all neighborhoods of a given type indexed by both.

> Examples for neighborhood types that make sense here:
>
> Offspring neighborhoods starting from the top node for a whole document, such as book, journal article, or report, along the link type *includes*.
>
> Ancestor neighborhoods of any media object along the link type *includes*.
>
> The neighborhood consisting of a document and all the document it *cites*.

Table 20. **Combination search (Boolean AND) with neighborhoods as targets 2.**

Starting from two objects, find offspring descriptor neighborhoods that can be reached from both through *deals-with*

Example:

> Documents B and C are known to be relevant
>
> Document B *deals-with* apperception, spatial, sex differences
>
> Document C *deals-with* thinking, verbal, bilinguals

Even though there are no descriptors in common, it is clear to the reader that both deal with cognitive process. The system can infer this quite simply: from both documents one can reach the descriptor neighborhood

> {cognitive processes (the top descriptor), apperception, cognitive mapping, serial ordering, associative processes, mental concentration, ideation, thinking}

Any of the descriptors in that neighborhood are good candidate starting points for finding more relevant documents.

4.7 Hierarchical inheritance through ancestor neighborhoods as search targets

The neighborhood consisting of a media object and all its ancestor objects is of particular interest since it supports the important process of hierarchical inheritance (Table 21). The following example illustrates the point:

Assume **Scientific American** published an article **The spread of AIDS** which includes a section on **drug treatment**. Table 21 contains the relevant statements in a hypermedia base. The section should clearly be found in a search for material on the **Drug treatment** of **AIDS** that is written for the **Educated lay person**. A Boolean AND search, combining the three required elements, with ancestor neighborhoods as targets will accomplish this. The ancestor neighborhood of the section consists of three elements, namely

{D359 (the section), D355 (the article), D243 (Scientific American)}

Any link into any of the elements is a neighborhood link into the ancestor neighborhood, thus the ancestor neighborhood meets all three search criteria and is retrieved:

From **Drug treatment** the system reaches the section D359,

from **AIDS** the article D355,

form **Educated lay person** the journal D243

We can describe this type of search differently: A Boolean AND search is satisfied if a proper connection exists either to the lowest level object itself or to any of its ancestors through an inheriting link. This is hierarchical inheritance: For purposes of this search, the lowest level object inherits selected connections into any of its ancestors.

Table 21. **Hierarchical inheritance through specifying ancestor neighborhoods as the search targets.**

D359 (the section **Drug treatment**) *included-in* D355 (the article **The spread of AIDS**)

D355 (the article) *included-in* D243 (Scientific American)

D243 (Scientific American) *has-target-audience* Educated lay person
or Educated lay person *is-target-audience-of* D243

D355 (the article) *deals-with* AIDS or AIDS *discussed-in* D355

D359 (the section) *deals-with* Drug treatment
or Drug treatment *discussed-in* D359

Query: Find ancestor neighborhoods that meet the following criteria:

Reachable from **Educated lay person** through *is-target-audience-of* AND

Reachable from **AIDS** through *discussed-in* AND

Reachable from **Drug treatment** through *discussed-in*

Ancestor neighborhood:

{D359 (the section), D355 (the article), D243 (Scientific American)}

This neighborhood meets all three search criteria; it can be reached

from **Educated lay person** through *is-target-audience* leads to D243

from **AIDS** through *discussed-in* leads to D355

from **Drug treatment** through *discussed-in* leads to D359

5 Indexing

Indexing in the general sense is the creation of any type of explicit relationship or link. Such relationships can be created by the creator of an object, by specially appointed editors/indexers, and by users. The following examples illustrate the range.

The author of a document creates *includes* and corresponding *continued-by* links from the top node for the entire document to each section, and from each section to paragraphs and figures, whether these links are presented through arrangement in a printed document or through explicit links in a hypermedia base. By her choice of words, the author also creates a link between each word in the text and the text section in which the word occurs. Specially appointed editors and "indexers" (in the common, much more narrow usage) create further links, for example, links between subject descriptors and a document as a whole (the most common form of "subject indexing") as well as links between subject descriptors and individual document sections or even individual paragraphs. An indexer may introduce (create) subject descriptors and create links between them. In that example, the indexer is perhaps better called thesaurus builder. Many might not even call establishing links between subject descriptors "indexing", but it is establishing links and requires intellectual decisions. Having stated the fundamental sameness one should add that there are also differences: establishing links between descriptors does involve a different link type, requires a different type of thinking, and has more far-reaching, system-wide consequences.

5.1 Indexing with hierarchical inheritance

When dealing with an object, such as a document, that has multiple objects under it, the indexer must decide at which hierarchical level a relationship should be made. She should use a principle known from semantic networks and frame systems: If a relationship applies to all or almost all subordinate objects, establish the relationship at the superordinate object as an inheriting relationship. If the relationship applies only to a specific object, establish the relationship only for the specific object. This principle ensures parsimonious indexing with no ill effect for searching, provided hierarchical inheritance is applied, for example by specifying ancestor neighborhoods as targets as described in Section 4.2.5.

Table 22 gives examples. The first example deals with a hierarchy of media objects: a journal, an article, a section of the article. A statement that holds for all objects (e.g., all sections) under a superordinate object (e.g., an article) should be made at the superordinate level.

The second example deals with a hierarchy of descriptors; it is a restatement of a well-known indexing rule: Rather than assigning several specific descriptors that are all children of the same broader descriptor assign the one parent descriptor that includes them all.

The third example also deals with a hierarchy of descriptors, this time in a thesaurus-building context. Rather than giving the same RT (Related Term) relationship for several specific descriptors that are all children of the same broader descriptor give the RT relationship for the one parent descriptor that includes them all.

Table 22. Indexing with hierarchical inheritance

Example 1: Media object hierarchy

Indexing the section **Drug treatment** (D359) in the article **The spread of AIDS** (D355) in **Scientific American** (D243). Previous indexing resulted in the following statements:

D359 (the section) *included-in* D355 (the article)

D355 (the article) *included-in* D243 (Scientific American)

D243 (Scientific American) *has-target-audience* Educated lay person

D355 (the article) *deals-with* AIDS

The following statements are true for the section (D359): *deals-with* Drug treatment, *deals-with* AIDS, *has-target-audience* Educated lay person. However, only the first is unique to the section, the other two are already included for superordinate objects. Thus there is no need to repeat them; only one statement needs to be added:

D359 (the section) *deals-with* Drug treatment

Example 2: Considering the descriptor hierarchy in descriptor assignment

If Document D *deals-with* apperception

Document D *deals-with* cognitive mapping

Document D *deals-with* . . . (serial ordering, associative processes, mental concentration, ideation, thinking)

are all true, the indexer should assign the broader descriptor:

Document D *deals-with* cognitive processes

Conversely, if cognitive processes *discussed-in* Document D is true,
this relationship inherits down to apperception, cognitive mapping etc. Thus,

thinking *discussed-in* Document D is true through inheritance.

A complete search for thinking should start from the ancestor neighborhood.

{thinking, cognitive processes, cognition and memory (a still broader descriptor)}

Example 3: Considering the descriptor hierarchy in establishing thesaurus relationships

Each of the descriptors under cognitive processes is related to intelligence. But instead of eight relationships apperception *has-Related-Term* intelligence, cognitive mapping *has-Related-Term* intelligence, etc., we should establish just one inheriting relationship

cognitive processes *has-Related-Term* intelligence

6 Review: The functions of objects and relationships

This section returns to the major theme of this paper, the essential unity of database systems, expert systems, ISAR systems, and hypermedia systems through a discussion of the various functions that objects (entities) and relationships (links) play in a unified system.

6.1 The functions of objects

Objects have five major functions (Table 23):

First, objects participate in relationships that model actual data. One can make a statement about chocolate chip cookies only if *chocolate chip cookie* is an object in the information base.

Second, objects, specifically media objects, represent data for assimilation by users. A paragraph, a figure, a tone document are meant to transmit information or, more generally, to enter the users cognitive or affective sphere.

Third, an object may serve as a starting point or a query element to access other objects. Some objects are introduced primarily for the access function; for example concepts to be used as subject descriptors are introduced not a objects of interest in their own right but to serve as access points for retrieval of media objects, software objects, organizations, persons, food products, or whatever. However, concepts may also be useful in themselves: Their hierarchical relationship may convey information and their definitions may be of interest. The thesaural relationships given for a subject descriptor (Broader Term, Narrower Term, Related Term) may also contribute to the definition and thus help a user whose final information need is clarification of the meaning of a term. This illustrates a general point: An object can, and often does, serve several functions simultaneously.

Fourth, objects provide a focus for the organization of the information base; this function is very important for hypermedia bases. The head of an offspring neighborhood, such as the top node for a book leading to all the chapters, or a path object serve this function.

Fifth, objects serve as focal points for relationships that pertain to all elements of a neighborhood (inheriting relationships). For example, the relationship Scientific American *has-target-audience* Educated lay person is introduced with the idea that it holds for all the articles *included-in* Scientific American (hierarchical inheritance).

Table 23. **Functions of objects (entities, nodes)**

An object can serve one or more of these functions

Participate in relationships that model actual data

Represent data for assimilation by users (media objects)

Serve as an access point that leads to other objects

> Example: Subject descriptor
> But: Entry for subject descriptor also useful in itself, as in a dictionary.

Provide a focus for the organization of the database

Serve as focal points for relationships that pertain to all elements of a neighborhood (inheriting relationships)

6.2 Functions of relationships (links)

Relationships have two functions (Table 24). As with objects, one function of relationships is to model actual data. The *has-ingredient* relationship for food products is an example. Another example is data set *supports* assertion. That is a factual statement made possible by including the relationship type *supports* in the conceptual schema.

The second function of relationship or links is to point to other objects. Most relationships established to model actual data can also be used as pointers: On the one hand, I may want to know the ingredients of chocolate chip cookies; I use the relationship *has-ingredient* for its substantive information value. On the other hand, I may want to use this relationship to find all foods containing chocolate chips, in which case I use it as a pointer for retrieval purposes. Links between media objects are established primarily for their pointer value. Media object 1 *includes* media object 2 does make a statement about these two objects, but the main purpose of the link is to point the reader of media object 1 to media object 2.

The observation that objects (entities) and relationships (links) serve multiple functions, some more familiar in the database world and some more familiar in the hypermedia world, underscores the advantage of the unified view presented in this paper. It shows that existing systems set up primarily to store factual data can be navigated (provided proper software support) and existing systems set up primarily for navigation can be used to look up facts.

Table 24. **Functions of relationships (links)**

Modeling actual data

Examples:

Food product 1 *has-ingredient* [food product 2, ...]

Data set *supports* assertion

Pointing to other objects

Most links

Examples for links established primarily as pointers:

Media object *includes* media object

Media object *continues* media object

Concept *discussed-in* media object

References

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

Carlson, David A. 1990.3; Ram, Sudah.
HyperIntelligence: The next frontier.
Comm. ACM. 1990.3; 33(3): 311-321.

Croft, Bruce W.; Turtle, Howard 1989
A retrieval model for incorporating hypertext links.
Hypertext '89 Proceedings. New York: ACM; 1989.11: 213-224.

Evans, Peter 1991
Personal communication.
Many of the ideas in Section 2 were stimulated by conversations with Peter Evans, at the time a doctoral student at the University of Maryland.

Frisse, Mark E. 1988.7
Searching for information in a hypertext medical handbook.
Comm. ACM. 1988.7; 31(7): 880-886.

Frisse, Mark E. 1989.11; Cousins, Steve, B.
Information retrieval from hypertext: Update on the Dynamic Medical Handbook project.
Hypertext '89 Proceedings. New York: ACM; 1989.11: 199-212.

Frisse, Mark E.; Cousins, Steve B.
Models for Hypertext.
J. of the American Society for Information Science. 1992.3; 43(2): 183-191.

Halasz, Frank G. 1988.7
Reflections on notecards: Seven issues for the next generation of hypermedia systems.
Comm. ACM. 1988.7; 31(7): 836-852.
A seminal paper which provided an important takeoff point for the ideas developed here.

The Information Navigator
by IME (Information Management and Engineering)
990 Washington Str., Dedham, MA 02026-6790
(617) 320-0303 FAX (617) 320-0793

Thompson, R. H.; Croft, W. B. 1989
Support for browsing an intelligent text retrieval system.
International J. for Man-Machine Studies. 1989; 30: 639-668.

Tompa, Frank Wm. 1989.1
A data model for flexible hypertext database systems.
ACM Transactions on Information Systems. 1989.1; 7(1): 85-100.