

# Don't Link Me In: Set Based Hypermedia for Taxonomic Reasoning

**H. Van Dyke Parunak**

Industrial Technology Institute  
PO Box 1485  
Ann Arbor, MI 48106  
(313) 769-4049, Fax -4064  
van@iti.org

## **ABSTRACT**

Hypermedia is often described as nodes of information with links between them, suggesting the conceptual model of a graph. A broader definition is a system of nodes of information through which people can move nonlinearly. Such a definition, while including graph-based hypermedia, also allows alternative implementations. This paper illustrates the need for alternative models by exhibiting a particular reasoning task for which navigating among nodes by way of explicit links is less effective than an alternative model of intersecting sets of nodes. The task is taxonomic reasoning, a particular kind of reasoning task that deals with the comparison and classification of highly similar nodes, in which an analyst viewing one node thinks not in terms of linking it to another node, but of including it in or excluding it from a set of related nodes.

This paper discusses this kind of reasoning and describes HyperSet, a set-based hypermedia system designed to support it. It compares HyperSet with other tools that support taxonomic reasoning, discusses the formal and implementational relationships between graph-based and set-based hypermedia, and defines the features that are required in a hybrid system that can concurrently support both set and graph manipulations.

## **KEYWORDS**

User models, Taxonomic reasoning, Interfaces, System architectures.

## **1. INTRODUCTION**

Hypermedia technology permits humans to move nonlinearly among related pieces of information, depending on their needs and interests. Traditionally, hypermedia is implemented according to a conceptual model based in graph theory. That is, the user thinks of the information as stored at the nodes of a graph, and moves from one node to the next over edges of the graph. This conceptual model is appropriate for knowledge tasks in which one node explains, amplifies, or otherwise elucidates another: a line in an outline leads to a paragraph or chapter; a word leads to its definition; a citation leads to the document cited. In general [GARG88] [CAGO88] [LANG90] (there are exceptions [HASW90]), links are binary and often directed, and the nodes that they connect differ in kind from one another.

For another kind of knowledge task a different conceptual model is more appropriate, a model based on set theory. This model facilitates manipulation of collections of similar

nodes that are assigned to one or more sets. Users move from one node to another in the same set, and from one set to another by way of nodes in the intersection of those sets. They do not think of nodes as linked directly to one another, but in terms of the sets to which they belong. Implementation of such a model in a conventional graph-based hypermedia shell is at best difficult, since the corresponding graph needs nondirectional links of arity greater than two. Yet the basic flavor of manipulation, moving nonlinearly and opportunistically through a collection of information objects, is functionally characteristic of hypermedia, and encourages us to enlarge the conventional view of hypermedia to accommodate it.

This approach should be contrasted with two other uses of sets in hypermedia.

1. In contrasting graph-based and set-based hypermedia, our focus is on the conceptual model that governs the user's interaction with the system, not the mathematical model underlying the implementation. Graph theory itself can be founded in set theory, and several formalizations of conventional graph-based hypermedia in fact use a set formalism [GARG88] [HASW90]. Still, conventional systems all envision the user as moving from one node of a graph to another, not from one set to another by way of a shared node, or from one node to another by way of a common set.
2. Even at the conceptual level, sets can be employed in several ways. Aggregation techniques require representing sets of nodes as a higher-level node that can be handled as a single entity, and expanded when needed into its components [HALA88]. The elements within an aggregation set may be undistinguished from one another in type (as in a common outline processor), or may fill specific slots in the aggregation (as in an idiomatic structure such as a Toulmin structure [TOUL69] or IBIS schema [CONK87]). This use of sets to model aggregations supports a different cognitive requirement from that discussed in this paper, and results in a different set of requirements for the interface.

Section 2 characterizes information tasks that are particularly well suited to a set-theoretic approach to hypermedia. Section 3 reviews existing tools against the tasks described in Section 2 and describes a prototype system that is tailored to these tools. Section 4 compares the graph and set formalisms, shows how they map into one another, and discusses the requirements for implementations to support both modalities. The possibility of such a hybrid implementation is extremely important, since knowledge workers are best served by systems that support the widest possible range of reasoning tasks, rather than excluding them from an important subset of human cognitive capabilities.

## 2. APPROPRIATE DOMAINS

The thesis of this paper is that adding a set-based interface to hypermedia is desirable because it provides a better fit to a specific kind of reasoning task than does traditional graph-based hypermedia. We begin by describing several examples of this kind of reasoning, and then abstracting its distinctive features.

A biologist returning from a field trip has collected three thousand different specimens of fungus, with extensive notes on each one describing the physical dimensions of the entire colony, the elevation at which it was found, the species of plant or kind of rock or soil on which it was growing, the kinds of insects she found crawling through it, the ambient temperature, exposure to sunlight, and relative humidity of its site, and other such details. She spreads her specimens and her notes on a large table and then groups and regroups them, noticing new features and exploring their correlation with those already observed, to try to bring similar specimens together and provide a basis for classifying them.

A linguist studying a newly discovered language is puzzling over a particular clause structure. He has collected thousands of words of text, and isolated within that corpus hundreds of examples of the clause in question, observing information such as its internal structure, the contexts in which it occurs, and the subject matter under discussion at each location. He records each example on a separate notecard and spreads them out on the floor of a large room that he can have to himself for several days. There he sorts them repeatedly into different piles, exploring how well they can be classified on the basis of features such as shared function words, subject matter, kind of literature, and classification of the higher-level construct in which each is embedded.

A research organization promoting the appropriate use of advanced technologies in manufacturing maintains a consulting service in which it visits small manufacturers and performs audits of their use of technology to help them identify opportunities for improving their productivity. Over the years it has collected hundreds of case reports, each one describing a single firm, its products, its economic structure, its use of technology, problems that it is facing, opportunities that were identified for it, and the result of taking those opportunities. Now the research organization wishes to search for trends in this body of information, to see if there are some general lessons that might be of use to its constituency. It needs a way to compare, contrast, group, and regroup the case studies.

The biologist, the linguist, and the research organization are performing variations of the task of *taxonomic reasoning*. This task has the following characteristics:

- The information objects being manipulated are highly similar to one another, so much so that the question of how to categorize or organize them is not immediately obvious. They are all (in the examples given) annotated specimens of fungi; or clauses; or case studies of interventions. In contrast, the information nodes in a graph-based hypermedia network typically are of several different types or levels of specificity (for example, a paper containing technical terms and the definitions of those terms; or the text of a government regulation and another document in which that regulation is cited).
- Taxonomic reasoning develops descriptions of each item along a set of dimensions, but these dimensions are not fully defined when one begins the reasoning process. For example, the biologist may notice only after some time that the presence or absence of a branching structure is a major distinguishing characteristic of the fungi under study.
- The basic activities that need to be supported to facilitate taxonomic reasoning are essentially set operations, such as sorting objects into sets based on their characteristics; looking together at the members of a single set (for example, to search for correlations among different characteristics); examining the different sets of which a single item is a member (to see whether there are relations among them); and generating new sets from old ones (for example, "Show me all the branching fungi that grow on this species of tree"). Presented with a given specimen, the analyst does not think of moving to another single specimen, but of the relation between this specimen and a set of others. To help compare different sets, it is useful to have available as well a simple statistical measure of similarity based on number of shared members.

### **3. A REVIEW OF TOOLS FOR TAXONOMIC REASONING**

Is a new model really necessary? Taxonomic reasoning deals with collections of information items: perhaps conventional hypermedia models could meet its needs. Alternatively, the result of taxonomic reasoning is a set of characteristics for each item (for example, color, size, shape), and a scheme for distinguishing groups of items by the values of their characteristics. Why won't a conventional database system fit the bill? In

this section, we review several hypermedia and non-hypermedia tools against the needs of taxonomic reasoning, ending with a description of a set-based hypermedia system.

### 3.1. Database

Before the advent of computers, large index cards with holes punched around the edges were the tool of choice for taxonomic reasoning. Characteristics were coded by notching the cards, and the cards could be sorted by inserting a wire through the holes and retrieving the ones that dropped off. Relational databases have largely replaced these keysort systems, and offer the kinds of set operations and the potential for computing correlations that taxonomic reasoning requires. In fact, some database systems were designed specifically for taxonomic retrieval [ESTA69]. Mathematically, it's easy to model set operations with a relational database, and database technology can serve effectively as the storage layer of a hypermedia system, whether graph or set based, as illustrated by several of the architectures in [MBB90]. However, as a user interface, conventional databases have two disadvantages.

1. The mode of interaction with a database is typically by form or query language, in which the user describes the desired items by constraints on the values of various fields. While this approach is appropriate for some applications, more commonly a user conducting taxonomic reasoning first sorts nodes together on the basis of vaguely perceived "similarity" and then examines the various fields to understand that similarity. When nodes contain photographs, a user may well sort together items that "look alike," and then use the resulting collection to generalize the nature of the similar characteristic. Thus taxonomic reasoning requires the ability to designate and not just to describe; to point to an item and associate it with other items in a tangible way, rather than through a query interface.
2. Databases presume that the fields can be defined in a separate analysis process, before data is loaded and manipulated, and adding new fields to an old file typically requires defining a new file structure and copying the old file into it. Taxonomic reasoning is much more efficient if fields can easily be defined dynamically and incrementally as reasoning progresses.

### 3.2. Graph-Based Hypermedia

Given a hypermedia environment supporting binary links, one can construct a first approximation to a set-based model by creating a node to represent each set and then linking each set node to its respective member nodes. In such a structure it is difficult to support set operations. For instance, after defining a set representing a given characteristic, one wants to review the nodes in the complement of the set to be sure that none of them in fact has the characteristic. It's easy in conventional hypermedia to view the nodes that *are* linked to a given set node, but hard to call up all those that *aren't*. There is also no way within a standard node-and-link environment to define operations (such as set union or intersection, or statistical measures of similarity) between pairs of sets. The fundamental problem is that the granularity of a conventional system is too fine, and most traditional systems do not provide the appropriate macro capabilities to construct operations on sets of links.

[HALA88] envisions, and IGD [FEIN88] implements, aggregate objects that permit a group of nodes to be manipulated as a single node. These innovations are motivated by the worthy desire of reducing complexity by facilitating hierarchical manipulations of collections of nodes. For example, one may wish to manipulate several section nodes as a single chapter, or several chapters as a single book. These aggregates have not been designed to support the distinctive cognitive requirements of taxonomic reasoning, and they do not lend themselves easily (i.e., without programmer intervention) to set operations (e.g., "Show me the entities in complex A that are also in complex B but not

in complex C"). Some implementations may be easily extended to support such a user view, and one objective of this paper is to motivate such extensions.

### 3.3. Relation to Hypercube Topologies

In certain applications, it is useful to restrict the topology of a hyperdocument to an N-dimensional hypercube [PARU89]. Such a structure is well suited to capturing relational information in a link-based structure, since each dimension of the hypercube can be associated with a separate characteristic of the specimens being studied, and a specimen is then located at the intersection in the hypercube corresponding to its particular values on each dimension. Perhaps the hypercube model obviates the need for a set-based interface.

The hypercube topology was invented to facilitate the comparison of two or more topics. For instance, in a two-dimensional hypercube, each topic occupies a column, and points of comparison occupy successive rows. Such a tool is a useful way to tease out some of the characteristics that distinguish specimens from one another, but is ill-suited for the process of sorting them into classes.

- If each dimension of a hypercube represents a different characteristic along which specimens can be compared, the hypercube will tend to have very high dimensionality, but relatively few values (often only two) along each dimension. Because display devices are two-dimensional, one can view only two dimensions (or characteristics) at a time. By contrast, a taxonomic analyst usually wants to see all of the characteristics of each specimen concurrently.
- The hypercube model does not permit multiple nodes to occupy the same intersection in the hypercube, but there may be several specimens that occupy the same intersection, particularly when not all of the relevant discriminating characteristics have been discovered yet.

### 3.4. The HyperSets Implementation

HyperSet currently exists as a working prototype that runs on IBM PC's. Its fundamental entities are *sets*, each containing zero or more *artifacts* (which are analogous to conventional hypermedia "nodes"). This section outlines the operations that HyperSet supports and describes an example application in the domain of literary analysis.

#### 3.4.1. The Capabilities of HyperSet

A user begins a typical HyperSet session by displaying a list in a scrollable window. This list contains either all of the artifacts in the collection or all of the subsets that have been defined. The user then selects an artifact (or a set) and proceeds to browse.

The basic browsing operation is not moving from node to node as in graph-based hypermedia, but moving from an artifact to one of the sets of which it is a member, and then to some other member of that set. The dynamic is not unlike traversing a link with multiple possible destinations, in which the user confronts the link as a distinct cognitive entity that requires a decision.

When at an artifact, the user can add it to or remove it from any set. When at a set, the user can remove any artifact currently in that set or add any artifact in the universe but not yet in the set. (Special restrictions apply to the universal set and the null set, which always contain all and none of the accessible artifacts, respectively.) Though these operations have the same effect as adding sources and destinations to multiple links in a graph-based system, the cognitive image is quite different. In HyperSets, I am not thinking of candidate destination nodes that I wish to reach from the current node, but of a set to which I wish it to belong.

HyperSet supports a full repertoire of set operations that generate new sets, including union, intersection, complement with respect to another set, and symmetric difference. It also supports editors to enter new artifacts and define new subsets.

The system can compute a simple correlation measure among sets. For any two sets  $A$  and  $B$ , this measure is

$$\left( \frac{|A \cap B|}{|A| |B|} \right)^{.5}$$

To motivate this measure, notice that the proportion of set  $A$ 's artifacts that are also in  $B$  is  $|A \cap B| / |A|$ , and the proportion of  $B$ 's artifacts that are common to  $A$  is  $|A \cap B| / |B|$ ; our measure is just the geometric mean of these two proportions. The measure is extended in the obvious way to more than two sets. This measure has the value 1 when the sets are identical, 0 when they are disjoint, and intermediate values when they intersect one another. Furthermore, for sets of given sizes, it is always greater when the sets are nested than when each has artifacts not included in the others.

One motivating application for HyperSet is the study of linguistic elements, which have a natural measure of location in the text from which they are drawn. Because the distribution of similar items is often of interest, HyperSet includes the ability to generate a density plot showing how the artifacts in a set are distributed [PARU81].

### 3.4.2 An Example Application

Among other activities, the Industrial Technology Institute provides an extension service for small manufacturers roughly analogous to the help offered to small farmers by an agricultural extension service. In the course of these interventions we generate a prose case report for each client.

Small manufacturers have been called the "foundation" of American manufacturing, generating about half of the value added and providing over half of the manufacturing jobs in this country [LURI89]. Our case reports contain a wealth of information on the challenges and opportunities facing this strategic sector of the economy, but mining them for this information has proven difficult. Cases do not follow a rigid format, so the information available differs from firm to firm, and is often anecdotal rather than the result of a formal survey. Yet, as one reads them, tantalizing hypotheses suggest themselves, about such characteristics as the relative frequency of different management structures, the correlation between employee skill levels and productivity, or the implications of the number of OEM's to which a firm is a supplier.

A simple preprocessor indexes each case study as a separate artifact into a HyperSet file. Then an analyst reads through each artifact, defining a set for each characteristic that seems important and entering each case into the appropriate sets. The flexibility of HyperSet permits the user to define sets "on the fly," while interacting with the data. Periodically, the analyst views all of the artifacts in a given set concurrently to see that they really are similar. This approach permits set definitions to emerge from perceptions of similarity that are at first vague and difficult to articulate, rather than forcing the analyst to develop rigid set definitions in advance. The user can easily add new sets to the collection as new discriminating features are observed. As the sets of artifacts develop, the analyst uses HyperSet's correlation measure to explore correlations among different characteristics.

HyperSet is only the first step in understanding a collection of artifacts such as our case studies. Once we learn how best to characterize these artifacts, more conventional database techniques offer increased efficiency in retrieval and confirmatory data analysis.

But arriving at that taxonomy in the first place is a challenging task, and one with which set-based hypermedia can give considerable aid.

#### 4. COMPARING THE TWO MODELS

To what extent are graph-based and set-based hypermedia simply different views of the same underlying constructs? To answer this question, this section develops and compares simple formalisms of graph-based and set-based hypermedia and the basic operations for browsing, authoring, and overview that each supports. It then extends these insights into requirements for a layered implementation that supports both user models. The formalisms presented here are not intended to be complete, but simply highlight the distinctive features that are relevant for comparing and contrasting the two models.

##### 4.1. A Simple Graph-Based Model

###### 4.1.1. Formalism

A graph-based hyperbase is an ordered triplet  $H_g = \langle N, A, L \rangle$  of nodes, anchors, and links, where:

- $N = \{N_1, \dots, N_m\}$  is a set of information nodes.
- $A = \{A_{i1}, \dots, A_{in}\}$  is a set of anchors, or subregions within nodes, such that each node  $N_i$  has one or more anchors  $\{A_{i1}, \dots, A_{in}\}$ .
- $L = \langle A_{ij}, A_{kl} \rangle$  is a set of links, or ordered pairs of anchors. In card-based systems, the second anchor is coextensive with an entire card, but the first must be a proper subset, at least for some links, if the network is to be more complex than a linear chain.

###### 4.1.2. Operations

The basic browsing operator is a function  $b : A \rightarrow A$  that maps a link's first anchor to its second. In card-based systems, the second anchor is that subregion of a node that includes the entire node, while in scrolling systems it may be a proper subregion. A deliberate side-effect of the mapping is displaying to the user the anchor (usually with some of its context) from the function's range. The basic authoring operation is defining links by identifying two anchors. That is, links are defined explicitly by walking manually through the traversal process. Common summary devices for set-based hypermedia include maps that show the network of links, and path macros (such as tours or back-up stacks) that identify subnetworks of reduced topology within the overall graph.

##### 4.2. A Simple Set-Based Model

###### 4.2.1. Formalism

A set-based hyperbase is an ordered pair  $H_s = \langle N, S \rangle$  of nodes and sets, where:

- $N = \{N_1, \dots, N_m\}$  is a set of information nodes, just as in the graph-based model.
- $S \subseteq 2^N$  is a set of subsets of  $N$ .

### 4.2.2. Operations

Set-based hypermedia uses two browsing functions, one  $f : N \rightarrow S$  that maps a node to one of the sets of which it is a member, and one  $g : S \rightarrow N$  that maps a set to one of its members. These are typically applied in alternation. As in graph-based hypermedia, a side effect of applying a browsing function is displaying to the user the selected element in the function's range. The basic authoring operations are turning on or off a node's membership in any set (starting either with a node or with a set), and generating new sets from existing ones by means of set operations (such as union, intersection, complement, and symmetric difference). Summary devices include displaying all nodes in a set; displaying all the sets to which a node belongs; correlation measures among sets and (for collections with an order parameter) density distribution plots. In fact, though HyperSet does not presently support it, the ready access it gives to collections of items makes it a natural interface to a complete statistical package.

### 4.3. Overlap between Graphs and Sets

Both approaches use the same concept of a node. But each requires some additional machinery not required in the other.

To build a graph-based system from a set-based one, we need to add the concept of an anchor, so that multiple links can originate at different points in one node.

To build a set-based system from a graph-based one, we need to allow links of arbitrary arity, so that traversal takes one not to another node, but to a list of accessible nodes (the "set"), from which the next node can be chosen. We also need to provide operators to perform set operations on pairs of these extended links.

### 4.4. Hybrid Implementations

Set and graph user models satisfy different reasoning tasks, but are not inconsistent with one another. It is desirable to implement both modalities in the same system, with sets manipulating collections of similar nodes and links joining these nodes to dissimilar ones to support such material as definitions of terms and antecedent literature. (HyperSet is only a prototype to explore the set-based interface and so, does not support conventional links.) The possibility of such hybrid systems, and of the development of still other user models besides sets and graphs, is a goal that should be kept in mind in designing layered hypermedia implementations [FUST90] [HASW90] [PARU90a] [PARU90b]. What is commonly described as the "link layer" in such implementations really needs to be something more general, an "association layer," that provides primitives from which both links and sets can be constructed.

Simple adjustments to the basic models of each kind of hypermedia permit such hybrid applications. In particular, a hybrid hyperbase that can support both set and graph-based operations is a triplet  $H_h = \langle N, A, S \rangle$  of nodes, anchors, and sets, where:

- $N = \{N_1 \dots N_m\}$  is a set of information nodes,
- $A = \{A_{11} \dots A_{mn}\}$  is a set of anchors, or subregions within nodes, such that each node  $N_i$  has one or more anchors,  $\{A_{i1} \dots A_{in}\}$ . (Set-based applications restrict anchors to be entire nodes.)
- $S \subseteq 2^A$  is a set of subsets of  $A$ . (Graph-based applications restrict these sets to be binary and ordered.)

Existing abstract models of hypertext differ in their support of these requirements. For example, [LANG90] requires links to be binary and directed, while [GARG88] and HAM [CAGO88] model them as binary and undirected; none of these can easily support a set-based system. The Dexter Reference Model [HASW90], in contrast, is sufficiently rich to support HyperSet.

## CONCLUSION

Reasoning tasks come in different flavors, many of which can be made easier by accessing information nonlinearly. So long as hypermedia is defined procedurally, as nodes connected by binary links, its value to some kinds of reasoning tasks will be severely restricted. One such task is taxonomic reasoning. Experience with the HyperSet prototype shows that an expanded model of hypermedia, one that supports a set based interface as well as a graph based one, is well suited to taxonomic reasoning. Under the most straightforward definition of each approach, neither set-based hypermedia nor graph-based hypermedia can be derived from one another, but a more general representation permits both to be derived as special cases, and thus allows the design of general-purpose hypermedia engines that support systems embodying both user models.

## REFERENCES

- [CAGO88] B. Campbell and J.M. Goodman, "HAM: A General Purpose Hypertext Abstract Machine." *CACM* 31:7 (July), 856-861.
- [CONK87] J. Conklin and M.L. Begeman, "gIBIS: A Hypertext Tool for Team Design Deliberation," *Proceedings of Hypertext '87*, 247-252.
- [ESTA69] G.F. Estabrook and R.C. Brill, "The Theory of the TAXIR Accessioner," *Mathematical Biosciences* 5, 327-340.
- [FEIN88] S. Feiner, "Seeing the forest for the trees: Hierarchical display of hypertext structure." *Proceedings of the ACM Conference on Office Information Systems* (Palo Alto, CA, March), 205-222.
- [FUST90] R. Furuta and P.D. Stotts, "The Trellis Hypertext Reference Model." In [MBB90], 83-93.
- [GARG88] P.K. Garg, "Abstraction Mechanisms in Hypertext." *CACM* 31:7, 862-870.
- [HALA88] F. Halasz, "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems." *CACM* 31:7, 836-855.
- [HASW90] F. Halasz and M. Schwartz, "The Dexter Hypertext Reference Model." In [MBB90], 95-133.
- [LANG90] D.B. Lange, "A Formal Model of Hypertext." In [MBB90], 145-166.
- [LURI89] D. Luria, "The Case for the Base: Thinking About the Michigan Manufacturing Foundation." *Modern Michigan* 2:1, 4-8.
- [MBB90] J. Moline, D. Benigni, and J. Baronas, editors, *Proceedings of the Hypertext Standardization Workshop*, NIST Special Publication SP500-178.
- [PARU81] H.V.D. Parunak, "Prolegomena to Pictorial Concordances," *Computers and the Humanities* 15, 15-36.

- [PARU89] H.V.D. Parunak, "Hypermedia Topologies and User Navigation," *Proceedings of Hypertext '89*, 43-50.
- [PARU90a] H.V.D. Parunak, "Reference and Data Model Group (RDMG): Work Plan Status." In [MBB90], 9-13.
- [PARU90b] H.V.D. Parunak, "Toward a Reference Model for Hypermedia." In [MBB90], 197-211.
- [TOUL69] S.E. Toulmin, *The Uses of Argument*. Cambridge University Press.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1991 ACM 0-89791-461-9/91/0012/0242...\$1.50