Using Metadata for the Intelligent Browsing of Structured Media Objects

William I. Grosky, Farshad Fotouhi, Ishwar K. Sethi
Multimedia Information Systems Group
Computer Science Department
Wayne State University
Detroit, Michigan 48202
and
Bogdan Capatina
Ford Motor Company
Dearborn, Michigan 48120

Abstract

Interacting with a multimedia information system is different from interacting with a standard text-based information system. In this paper, we present the design of a system called *Content-Based Hypermedia (CBH)*, which allows a user to utilize metadata to intelligently browse through a collection of media objects. We describe the approach we use to model data in order to make it browsable, explore our approach to browsing, which we call *metadata mediated browsing*, indicate how metadata is used in the concept of *similarity*, present the architecture of our system, and discuss indexing techniques for similarity browsing using content-based metadata and approaches to clustering which generate higher-level metadata to help the user browse more effectively.

1 Introduction

Interacting with a multimedia information system is quite different from interacting with a standard text-based information system. In any such system, the real-world objects which comprise its domain are directly represented through their properties and indirectly represented through their relationships to other real-world objects. In text-based systems, however, all properties and relationships are presented in a textual format to the user. In standard relational systems, each real-world object has a unique textual identifier and has properties whose values can be textually presented. Even in object-oriented systems, which allow the representation of more complex properties, such as those which are set or sequence valued, or those whose values are other objects, information is presented in a textual format.

In a multimedia information system, however, there exist representations of objects which are not textually based. These representations consist of portions of images (static visual representations of objects), videos (dynamic visual representations of objects), and audios (aural representations of objects). When these representations are included in the domain of an information system, they can be used in two distinct fashions: as real-world objects themselves, hereinafter called *media objects*, having properties and participating in relationships, one can treat them as one treats other first-class objects and seek to gain information about them; or, as user-recognizable surrogates for the real-world objects which comprise their content, one can use them in the process of seeking information about the corresponding non-media objects which they represent[1].

Concentrating on the second case above, we work under the assumption that any information concerning a media object which can be used to infer information regarding its content (i.e., the corresponding non-media objects which it represents) is an example of *content-based metadata*. Media objects are rich in information concerning the non-media objects which they represent [1], the most important of this information being the identity of the given non-media objects. This information may be gleaned in three ways: manually, where a user specifically inserts into the system that, say, a particular region of an image is a visual representation of the person *Bill*; automatically, where the system itself uses various feature matching techniques to derive a similarity between a media object m_1, which represents an unknown non-media object e_1, and another media object m_2, which represents a

[1] For the sake of simplicity, we ignore such media objects as a digitized photograph of a photograph.

known non-media object e_2, the result being that e_1 and e_2 are actually identical; and semi-automatically, where the system works in conjunction with user initiated actions to identify the contents of a media object.

Knowing the identities of the various non-media objects which are represented by a media object is quite powerful. By seeing or hearing a media object, the user of a multimedia information system can gain information, through his or her own knowledge, concerning the represented non-media objects which may not be explicitly modeled by the system. Even if all such information is explicitly represented in the system and is capable of being queried on and textually answered, simply viewing or hearing the appropriate media object can invoke an emotional reaction not possible via a simple textual interface.

For this reason, a system which would allow a user to intelligently browse through a collection of media objects would serve a very useful purpose. Such a system, called *Content-Based Hypermedia (CBH)*, is currently under development by our group[2]. In the remainder of this short presentation, we describe the approach we use to model data in order to make it browsable, explore our approach to browsing, which we call *metadata mediated browsing*, indicate how metadata is used in the important concept of *similarity*, present the overall architecture of our system, and discuss various research topics connected to our approach, concentrating on indexing techniques for similarity browsing using content-based metadata and approaches to clustering which generate higher-level metadata to help the user browse more effectively.

2 The CBH Data Model

The use of CBH can best be understood through the definition of a *CBH-schema*, which is nothing more than an object-oriented schema over non-media objects which has undergone a transformation which will shortly be explained. An object-oriented schema consists of various hierarchical structures which we can classify into three domains. These are the *class hierarchy*, the *nested object hierarchy*, and the *complex object hierarchy*. The *class hierarchy* is a hierarchy of classes in which an edge between a pair of classes represents an *is-a* (specialization/generalization) relationship; that is, the subclass is a specialization of the superclass and the superclass is a generalization of the subclass. The *nested object hierarchy* is a hierarchy of classes in which an edge between a pair of classes represents either an *is-part-of* (aggregation) or an association relationship. Finally, the *complex object hierarchy* is the union of the nested object and class hierarchies. Thus, an edge between a pair of classes represents an *is-a*, *is-part-of*, or an association relationship. The class hierarchy and the nested object hierarchy are viewed as special cases of the complex object hierarchy.

To transform our original object-oriented schema into a CBH-schema, we first add a class of images and a class of videos[3]. Each image is actually a complex object, comprising various regions having semantic content. Similarly, each such region, itself, may be decomposed into various subregions, each having some semantic content. This decomposition follows the complex object structure of the non-media objects represented by the given regions. That is, if non-media object o_2 is a part of non-media object o_1, and o_1 has a representation r_1 appearing in some image (as a particular region) or video (as a particular sequence of regions), then, cases exist where r_1 would have a component r_2 which is a representation of object o_2[4]. For example, a window is part of a building. Thus, the region of an image corresponding to a building may have various subregions, each of which correspond to a window. We call these image regions having semantic content *semcons* (iconic data with semantics).

To the resulting schema, we now add a class of semcons. We note that this class *is-part-of* the class of images. Attributes of this class of semcons are based on various extracted features such as shape, texture, and color, which are used for determining when one semcon is similar to another, and thus represents the same non-media object. We note that semcons as well as their attributes are considered as metadata.

To each non-media class, we then add a set-valued attribute *appearing-in*, which leads from each instantiation of that class to the set of images and videos where its corresponding semcon appears[5]. We also add an attribute *represents* to the class of semcons which leads from each semcon to the non-media object which that semcon represents. We note that a non-media object can be an instantiation of one or more classes. If $\mathcal{S}$ is the original object-oriented schema, the resultant schema, $\mathcal{S}_{CBH}$, is now defined to be the CBH-schema corresponding to $\mathcal{S}$.

[2] The media objects in our system are currently restricted to be images and videos.

[3] Technically speaking, a video may be considered as a sequence of objects from the class of images in conjunction with other information.

[4] This would not be the case where r_2 is occluded in the particular media object or is just not visible due to the placement of the sensor with respect to the three-dimensional non-media object o_1.

[5] Technically speaking, *appearing-in* also carries information concerning where in the image and video the given semcon is located.

Informally speaking, it is now possible to view an image or video, specify a particular semcon within this media object, and find out information concerning the non-media object corresponding to this particular image region. For example, viewing an image of Professor Smith, it is now possible to navigate to a set of images and videos containing representations of the students of Professor Smith. We explain how this is accomplished from the user's viewpoint in the next section.

3 Browsing Data and Metadata in CBH

In our system, browsing data or metadata is done in a uniform fashion. Browsing in a populated database under schema $\mathcal{S}_{CBH}$ is quite different than it would appear from the previous definitions, however. We recall that the user is browsing only through media objects. To implement this look and feel of the system, we give the implication to the user that each non-media object in a given class of $\mathcal{S}_{CBH}$ is replaced by the set of media-objects in which the given non-media object appears. More formally, if class C in schema S and class C_{CBH} in schema $\mathcal{S}_{CBH}$ are two corresponding classes, as far as the user is concerned, the set of instantiations of class C_{CBH} consists of the union of the sets *appearing-in(i)*, for i an instantiation of class C in schema S. The reality of the implementation is, of course, quite different.

Whenever viewing a particular media object, the user can choose a particular semcon, r, for further examination. One of the actions the user can carry out is to see the value of any attribute, a, defined over a non-media object with respect to one of the, perhaps, many classes of which it can be an instantiation, and which the given semcon represents. This is accomplished in the CBH-schema $\mathcal{S}_{CBH}$ by calculating *represents(r).a*, after selecting the desired class. If the value of this attribute is a simple data type (e.g., integer, real, or string), this value is textually presented to the user. If, however, this attribute's value is another (non-media) object, the user is allowed to browse through a set of media objects, each of which contains a representation of this latter non-media object. This approach easily generalizes to set-valued attributes. In a similar fashion, the user can follow an association (relationship). For example, if semcon r is chosen by the user and the non-media object *represents(r)* participates in a binary relationship with a collection, S, of other non-media objects, then the user is allowed to browse through the set of media objects, consisting of each media object which contains a representation of a non-media object from the collection S. See Figure 1 for a browsing path from an image of a person *Bill* to an image of Bill's office and how it is mediated by particular relationships among corresponding non-media objects.

When a particular semcon is chosen, the user views a scrolling menu of choices, which includes each attribute and relationship in which the non-media object represented by the particular semcon participates. Through the use of filtering commands, the user will be able to navigate through paths composed of many relationships and attributes and restrict the collection of media objects at the final destination. For example, choosing a particular semcon which is an image of person *Joe*, a filtering command of the form **self.co-worker.residence, where self.co-worker.residence.city = 'Livonia'**, will take the user to a collection of media objects which represent the residences of co-workers of *Joe* who live in Livonia.

The most likely browsing scenario is the use of metadata as an aid to browsing the data itself. A ubiquitous example of this is to navigate along *similarity* paths. Such a path would proceed from a given semcon to the set of image objects containing semcons similar to the given semcon[6]. An illustration of such navigation would be to proceed from the image of a particular person to a set of images of persons having similarly shaped eyes. Supporting such a browsing path critically depends on techniques for robust metadata generation in the form of image features, as various sorts of feature matching approaches will be utilized. These browsing paths are much more complicated to support than those mentioned in the previous paragraph and utilize iconic indexes in their implementation [2, 3, 4, 5]. We discuss these indexes in more detail in Section 6.

Metadata, itself, may also be browsed through for various reasons. An illustration of this is when metadata is an intermediate point on a browsing path which otherwise contains some data items. One such situation is when the chosen semcon is the entire media object. Then the allowable attributes and associations over which browsing and filtering are allowed are the ones defined for the class of images and videos in schema $\mathcal{S}_{CBH}$. An example browsing path would be to proceed from a given image to the set of all images photographed by people living in the same city as the person who shot the given image[7]. Another example of browsing through metadata occurs when the user, for whatever purpose, views the output of an image

[6] The hope is that the non-media objects which two similar semcons represent are identical.

[7] The identity of the person who shot a given image is, of course, metadata.

processing routine on a given image or video. This may be done in an environment where CBH is being used as part of a testbed for image interpretation researchers.

4 The CBH Hypermedia Web - Metadata Mediated Browsing

A user would view CBH as a linked collection of media objects having certain recognizable portions which correspond to non-media objects modeled by the information system. By choosing a certain portion of an image or video, the user would be able to find out various property values of the corresponding non-media object, as well as navigate to other media objects which represent non-media objects in various relationships with the given non-media object. These media object portions are the semcons, a type of metadata which mediates a natural, user-centered navigation style enabling the user to discover properties and relationships among the modeled non-media entities. We call this navigation style *metadata mediated browsing*.

In our system, the user's starting point in this web is akin to a Mosaic home page, where we use clustering techniques [6, 7] to construct various higher-level groupings (higher-level metadata) of media objects which the user might be interested in viewing. We discuss various clustering techniques in Section 7.

A question arises as to how the user actually chooses the appropriate media portion (semcon). One semcon may contain another and clicking a mouse over the contained semcon may also indicate that the user wants information concerning the containing semcon. For example, clicking a mouse over *Joe's eyebrow* may indicate that one wants information on *Joe*, *Joe's head*, *Joe's face*, or *Joe's eyebrow*. Each of these may be non-media objects modeled in the system. For example, instead of asking for representations of the residences of co-workers of Joe who live in Livonia, as in the previous section, one may want to browse all representations of persons with similarly shaped eyebrows.

Thus, there must be a way for the user to indicate which level of resolution is wanted. In our approach, media objects are capable of being packed and unpacked into their component semcons at any arbitrary level of resolution. We will be experimenting with various approaches of doing this, one of which will be highlighting the boundary of the appropriate semcon so the user will know which level of resolution the system will choose for a particular mouse click. Also, since a non-media object can be an instantiation of many classes, the user will be able to choose the class in which to view a particular semcon. This corresponds to the notion of *role* in relational systems.

As one can see from the previous discussion, CBH is hypermedia-based. What is new is that CBH will allow navigation to be initiated from metadata in the form of media components and is formally based on an underlying object-oriented database schema. A related and quite interesting hypertext system, which does not concentrate on media objects, is that of [8], which is based on a generalization of the entity-relationship data model. Current hypermedia systems [9, 10, 11] lean more to presentation issues rather than browsing media objects by content and generally treat each media object as a single entity, so that the anchor point of a link must be the entire media object. These systems also do not discuss such issues as index design to make browsing and filtering more efficient. There are, however, quite powerful formal models of hypertext [12] which, though not delving into all the details necessary to implement our approach, are compatible with it. In particular, the concept of *resolver* and *accessor* functions, as presented in the Dexter model [12] seem suited for implementing *similarity* paths.

5 System Architecture

Figure 2 illustrates the logical architecture of the CBH database module, the most important component of our overall system. Conceptually, this module consists of a standard alphanumeric database, a semcon database, a feature database, and a media object database. Physically, of course, all this information can be stored in a single database. The standard alphanumeric database holds information concerning the non-media objects which are being represented in our system, while the uninterpreted media objects reside in the media object database. Metadata reside in the semcon database and the feature database. We note that features which reside in the feature database are properties of semcons.

Upon inserting a media object into our system, the appropriate semcons must be identified. While our present goal is to completely automate this process, our version of CBH utilizes a semi-automated approach. The user must roughly outline the various semcons utilizing a mouse. This rough outline will then be refined utilizing various image processing tools. Depending on the nature of the semcon, various features will then be automatically extracted. The user then has a choice: either to explicitly indicate to the system the corresponding non-media object which that semcon represents, or to invoke a user-mediated similarity match which finds the set of non-media objects which this semcon could possibly represent. In processing this information, the system must efficiently store the location and shape of the given semcon. For images, we are using a linear quadtree-like file structure to indicate each semcon's spatial extent.

6 Indexing for Similarity Navigation Using Content-Based Metadata

We have previously examined two classes of approaches for retrieving image regions based on similarity. One class of approaches deals with the design and manipulation of indexes for shape-based similarity retrieval [2, 3, 4, 5]. The other set of techniques is concerned with the representation of image spatial knowledge in order to retrieve image regions based on the similarity of spatial relationships among the various objects appearing in the given images [13]. Since, for images, a semcon is region-based and may have a complicated spatial structure, both of these approaches may be used to efficiently find matching semcons.

More specifically, we have proposed a data-driven, shape-based similarity retrieval approach utilizing local feature-based iconic index structures. Given any structural feature-based shape representation technique and a quantitative method to measure the similarity (or difference) between any two features, a feature index tree can be created. Such a tree can be in main memory [4] or in secondary memory [2]. Given a feature of the input image, the best matching feature in the feature index tree can then be efficiently found. Since each feature indicates which semcons it is contained in, and, in turn, each semcon indicates the media object it is contained in, we can utilize this tree to efficiently generate a set of media objects to navigate to for a particular *similarity* path.

In these implementations, each feature description is considered to be atomic. When a particular feature is found in an image, various semcons are hypothesized to be present in the image. We have shown mathematically and have experimentally verified, however, that under some very general assumptions, an index based on hierarchical features is more computationally efficient than one based on non-hierarchical features [3]. Since we have seen that semcons may contain other semcons, a natural hierarchical structure to feature construction exists. We are now applying our preceding work to this new environment.

The type of semcon we have been discussing up to now has been static. That is, a semcon is considered to be a region of a single image. There is also the notion of a *dynamic* semcon, that is a sequence of corresponding regions in a video. We believe that by using this notion, certain dynamic behaviors can be characterized, such as walking and dancing. Constructing indices for these types of semcons presents a very interesting research problem.

We are also exploiting our previous work in spatial indexing [13] in the design of our system. This work presented a spatial access method which utilized a data structure called an SB^+*-tree*, which is based on a B^+-tree. The motivation behind this research is that such a structure will allow commercial databases an access method for spatial objects without a major change, since most commercial databases already support a B^+-tree as an access method for text data. The SB^+-tree is a hybrid of existing spatial access methods. For each axis of the space, a set of index points is generated, an index point being created whenever a new minimum bounding rectangle begins or ends. These index points are then used to create the corresponding SB^+-tree. The number of SB^+-trees generated is dependent upon the number of dimensions of the approximation of the spatial object. We have developed an algorithm for performing spatial join between two spatial relations using the SB^+-tree. Through simulation, it is shown that the use of an SB^+-tree in performing such a join is much better in performance than that of an R-tree.

7 Generating Higher-Level Metadata through Clustering

Our system can be viewed as a network of data-containing components (nodes) connected by links. By the term *node*, we mean every instantiation of all classes which comprise the CBH-schema $\mathcal{S}_{CBH}$, and by the term *link*, we mean every relationship that can exist between the different classes which comprise the CBH-schema $\mathcal{S}_{CBH}$. The user gains knowledge through browsing the network. Knowledge is stored not only in the individual nodes but in the relationships between linked nodes (another form of metadata). Each user makes their own path through the network based on their own interests, allowing for less cumbersome structuring of the information. The users have more direct contact with the information, and are more involved with structuring the information in their own way, compared to a traditional database system.

It is unreasonable to expect each user to adapt to the underlying structure of the network. Rather, the network should adapt itself to each user. As hypermedia networks grow larger and larger, the ability to make them more personal becomes ever more important. It is easy for users to become lost among the nodes of a hypermedia network, especially when the relationships between the nodes are not natural to the user. Allowing each user to have their own view of the information reduces this information overhead.

We have previously implemented methods of autonomously monitoring a user's progress through a hypertext document and collecting information about their travels [6]. This information has been used to help the user navigate through the hypertext, as well as to generate clusters (using genetic algorithms and neural networks) and personal views of the document [7].

Clustering allows for higher-level concepts (metadata concerning ways of categorizing a group of nodes rather than a single node), allows the breaking of a single larger hypermedia network into appropriate modules, allows for views over the network, and allows the user to make changes at the cluster level without affecting the hypermedia network itself. Current clustering techniques in hypermedia systems are either structurally driven, where the connectivity of the nodes delineates the clusters, or concept driven, where the keywords (either in the content of the nodes themselves or in the meta-information about each node) delineate the clusters. Structurally driven clustering does not take into account how the user navigates through the nodes and links. Concept driven clustering has difficulty if the hypermedia network contains foreign languages or non-textual items. Both techniques rely heavily on the network's authors. We feel, however, that the user should really be involved in the clustering process. Each user sees different relationships between data and a hypermedia network should allow each user to mold the hypermedia information to their personal needs. See Figure 3.

In [7], we have used a genetic algorithm to delineate the clusters. Finding an optimal clustering of the nodes in a hypermedia network is equivalent to checking every partition of the set of nodes. As this is an exponential problem, it is too costly to find the optimal clustering for a network with more than a few nodes. Using an adaptive algorithm, we have found a very good clustering within a reasonable amount of time.

Metadata in the form of node clusters are generated by representing the user's tour through the hypermedia by a list of items, each item being a regular-expression-like string of the form $CNID : (PNID((CAID.weight - NNID@)^+),)^+$, where CNID is the identifier of the current node (media object), PNID is the identifier of the previous node (media object), CAID is the identifier of the current anchor (semcon), NNID is the identifier of the next node (media object), and weight is an integer indicating the number of times a particular path segment was traversed.

For example, consider the graphical representation of a hypermedia web as shown in Figure 4, where arabic letters represent node identifiers and Greek letters represent anchor (semcon) identifiers. Each time the user browses through a particular hypermedia web, he or she creates a path (a sequence of nodes visited and anchors activated). The set of paths that a user has taken which is shown in Figure 5 may be graphically represented as shown in Figure 6. WEB represents the hypermedia system CBH, itself, where each path begins and ends. The list data structure which captures this information is shown in Figure 7. These lists are then used by the genetic algorithm discussed above. In this algorithm, the payoff function is composed of two separate partial payoff functions: weight similarity and neighborliness. Weight similarity promotes clusters containing nodes with similar weights, while neighborliness promotes clusters containing neighboring nodes. We have also experimented with a neural network approach to this problem.

8 Conclusion

We have presented the outline of a system for content-based browsing of media objects and have shown how our previous work in indexing techniques for image and spatial databases, as well as in clustering techniques for hypermedia networks, lend themselves to the implementation of such a system. Metadata is central to the approach we have taken as it mediates the user-centered navigation style which this system supports. More specifically, the semcon class of metadata mediates any type of navigation, while the feature class of metadata mediates the navigation along *similarity* paths. We call this type of navigation *metadata mediated browsing.*

Image semcons are represented by using a linear quadtree-like data structure to delineate each semcon's spatial extent. This spatial extent is currently being specified by the user during insertion of the media object containing the given semcon. The identity of the non-media object represented by a given semcon is also specified at this time, either directly by the user or by a semi-automated procedure using feature matching. How to represent semcons which have a temporal modality is currently under study.

Features are extracted from each semcon, depending on its type. These features are indexed as discussed above, in order to efficiently find all semcons containing a given feature.

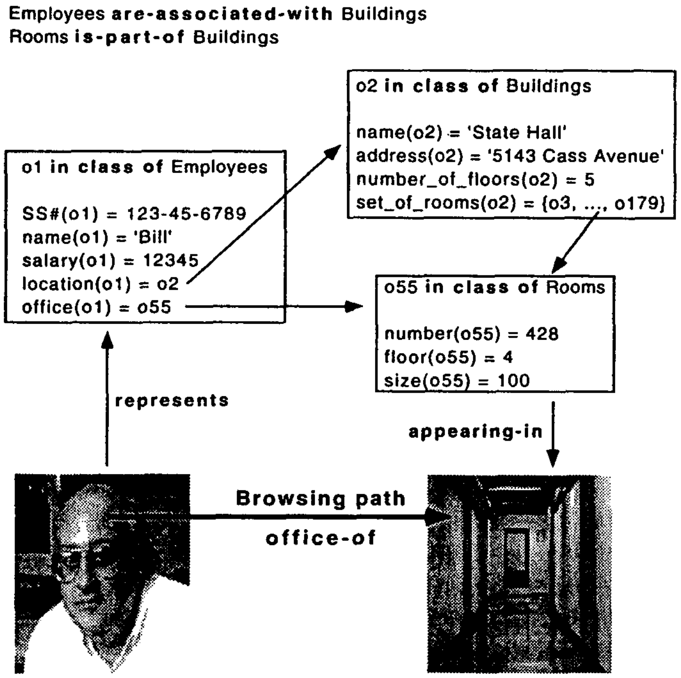

Figure 1: An Example Browsing Path Between Media Objects

Our short term goal is to have a working prototype for images and then to extend this prototype to include and videos and video indexes.

References

[1] W.I. Grosky. Multimedia information systems. *IEEE Multimedia*, 1(1):12–24, Spring 1994.

[2] W.I. Grosky, P. Neo, and R. Mehrotra. A Pictorial Index Mechanism for Model-Based Matching. In *Data and Knowledge Engineering*, pages 309–327, 1992.

[3] W.I. Grosky and Z. Jiang. Hierarchical Approach to Feature Indexing. *Image and Vision Computing*, 12(5):275–283, June 1994.

[4] W.I. Grosky and R. Mehrotra. Index-Based Object Recognition in Pictorial Data Management. *Computer Vision, Graphics, and Image Processing*, 52(3):416–436, 1990.

[5] I.K. Sethi and N. Ramesh. Local Association Based Recognition of Two-Dimensional Objects. *Machine Vision and Applications*, 5:265–276, 1992.

[6] A. Johnson and F. Fotouhi. Automatic Touring in a Hypertext System. In *Proceedings of*

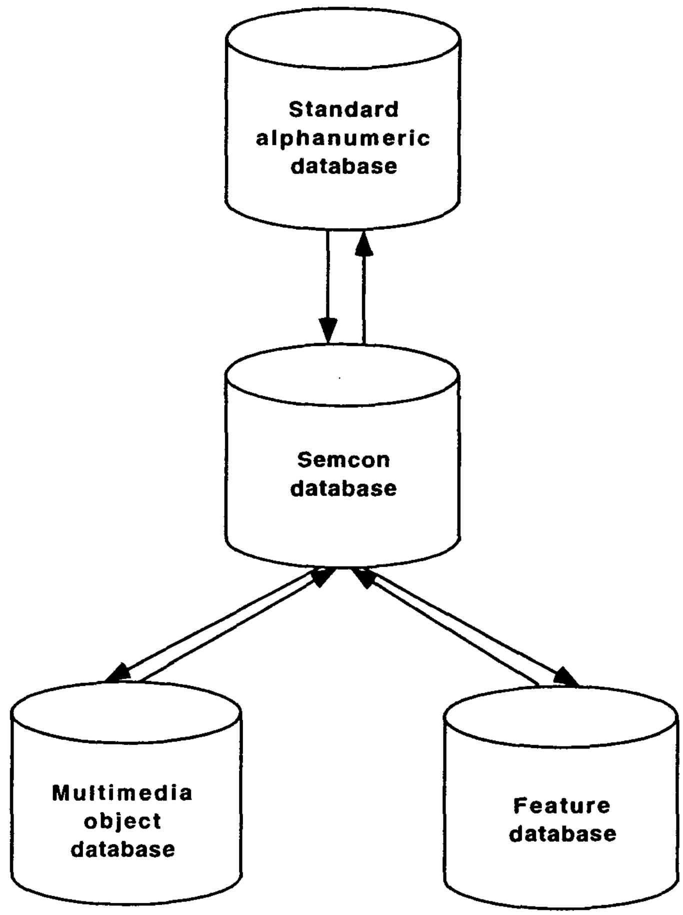

Figure 2: CBH Database Module Architecture

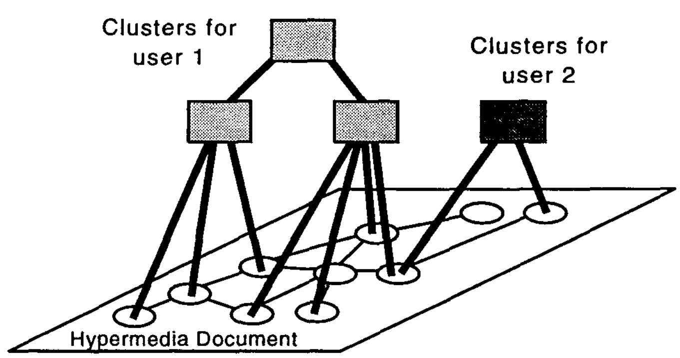

Figure 3: Conceptual Hypermedia Clustering

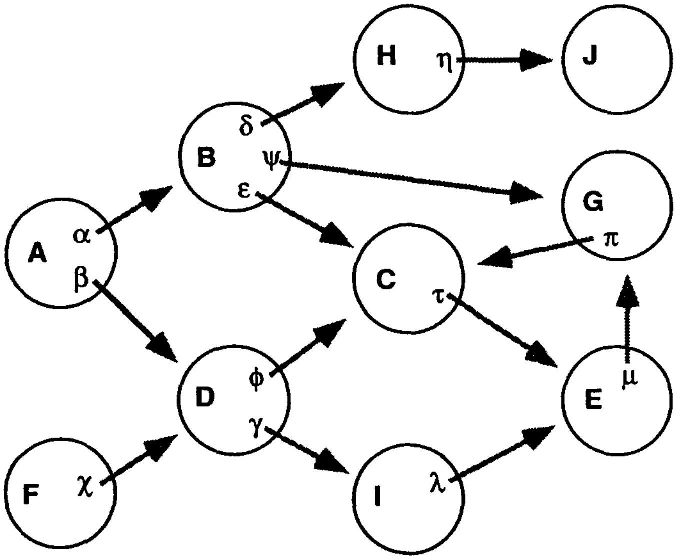

Figure 4: Graphical Representation of a Hypermedia Web

WEB → Aβ → Dγ → Iλ → E → WEB
WEB → Aβ → Dφ → Cτ → E → WEB
WEB → Aα → Bδ → Hη → J → WEB
WEB → Fχ → Dγ → I → WEB

Figure 5: The Paths a User Takes through the Hypermedia Web

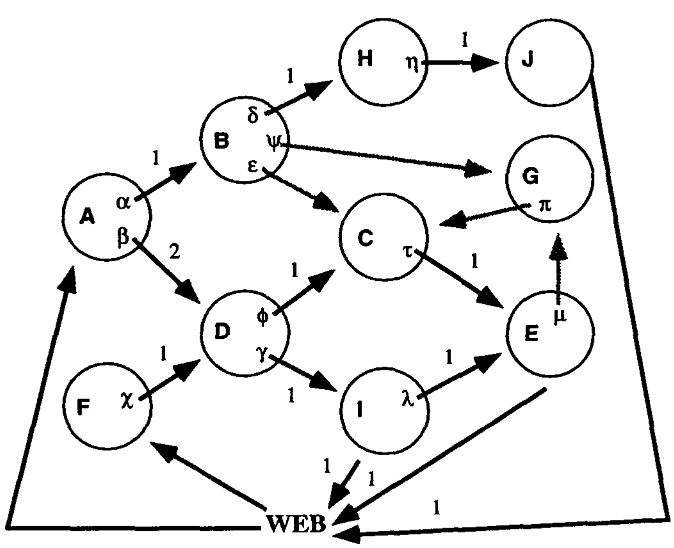

Figure 6: Graphical Representation of the Paths

A: **WEB(α.1–B@β.2–D@),**
B: **A(δ.1–H@),**
C: **D(τ.1–E@),**
D: **A(γ.1–I@φ.1–C@), F(γ.1–I@),**
E: **C(WEB.1–WEB#), I(WEB.1–WEB@),**
F: **WEB(χ.1–D@),**
H: **B(η.1–J@),**
I: **D(λ.1–E@WEB.1–WEB@),**
J: **H(WEB.1–WEB@)**

Figure 7: List Representation of the Paths

the IEEE International Phoenix Conference on Computers and Communications, pages 524–530, Tempe, Arizona, 1993.

[7] A. Johnson and F. Fotouhi. Adaptive Clustering of Hypermedia Documents. *Journal of Database Management*, To Appear.

[8] F. Garzotto, P. Paolini, and D. Schwabe. HDM - A Model-Based Approach to Hypertext Application Design. *ACM Transactions on Information Systems*, 11(1):1-26, 1993.

[9] L. Hardman, D.C.A. Bulterman, and G. van Rossum. The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. *Communications of the ACM*, 37(2), February 1994.

[10] K. Hirata, Y. Hara, N. Shibata, and F. Hirabayashi. Media-based Navigation for Hypermedia Systems. In *Proceedings of the Fifth ACM Conference on Hypertext*, pages 159–173, Seattle, Washington, 1993.

[11] D. Lucarella, S. Parisotto, and A. Zanzi. MORE: Multimedia Object Retrieval Environment. In *Proceedings of the Fifth ACM Conference on Hypertext*, pages 39–50, Seattle, Washington, 1993.

[12] F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30–39, February 1994.

[13] F. Fotouhi, A. Ibrahim, and S. Hasan. SB^{+}-Tree: An Efficient Index Structure for Joining Spatial Relations. Submitted for publication.