MeSOD: the Metric Spatial Object Data model for a multimedia application: Hyperbook

Masahiro Tabuchi and Yoichi Muraoka

School of Science and Engineering
Waseda University
3-4-1 Okubo, Shinjuku, Tokyo 169, Japan
(JUNET/BITNET): tabuchi@muraok.waseda.junet, mura001@jpnwas00

ABSTRACT

In this paper, a data model for multimedia applications, called MeSOD (Metric Spatial Object Data model), is proposed. This model maps entities in the real world onto metric spaces. The MeSOD is motivated by an electronic book system, called the Hyperbook system, that enables users to interact with the system using multimedia information.

The Hyperbook system is first introduced and the problems in implementing the multimedia communication is indicated. Then, the concept and the advantage of the MeSOD are discussed, and the informal definition of the MeSOD is summarized. The design of the MeSOD's applications developer's environment, MADE, is discussed, before the conclusion is presented.

1. INTRODUCTION

The recent progress in hardware and software technologies have provided us with more powerful and substantial personal-use computers, such as personal computers and workstations, than the minicomputers of several years ago.

Recent development and research efforts [1,2,3,5,6] in the application area have focused on how to make better use of these progressive technologies. For example, the SNAP system [1] based on the IFO model [2], one of the semantic data models, is an interactive schema manager that permits users to design and browse through schema visually and to query interactively. Also, the Intermedia [5,6], a tool designed to support both teaching and research in a university environment, contains multiple applications and mechanisms to link the contents of documents.

Advanced application systems, including multimedia applications, engineering applications, and expert database systems, require user-friendliness, availability and intelligence. Because these systems contain a database management system (DBMS), the facilities of DBMS, such as retrieving, storing, and sharing the data, must satisfy these requirements.

The Hyperbook system, introduced in this paper, is such an advanced application system. It is a self-contained, multimedia database system designed to be used as an electronic book. The system will enables users to "*read*" and "*write*" their own electronic book by retrieving and browsing in their own way, and storing the data in the proper form. It is now being developed at Waseda University. The goal of the Hyperbook project is to provide users with the capability of obtaining information as easily as they could from an ordinary book. The Hyperbook system allows users to query using multimedia information, such as image, voice, and text data. For instance, a Hyperbook system on birds would enable users to ask for the name of a bird by imitating its call and sketching its shape, as well as by describing other properties of the bird. Such multimedia form queries impress on users friendliness, availability, and intelligence.

User's queries in the form of multimedia information, such as the call and the shape of a bird, never exactly match the data in the multimedia database. Hence, a data model must be able to handle the *relationship degree*, that is the degree of relationship and includes the degree of resemblance or similarity, between the query and the data in the multimedia database.

Existing data models cannot handle this *relationship degree* between entities in the real world. While existing data models fundamentally assume users to retrieve with a keyword-form query, keywords are inadequate to represent the query for multimedia data. Hence, existing data models obviously lack the ability to handle multimedia data.

In the information retrieval (IR) field, the vector model of the keyword is well-known for representing similarity degrees between documents. Although this model enables users to retrieve documents with similarity between a query and a document vector, it is considerably difficult for users to select an adequate keyword to represent their query.

This paper introduces the Hyperbook system and proposes the Metric Spatial Object Data model (MeSOD) which is a clue adopted to implement the Hyperbook system. The MeSOD represents entities in the real world as points in a metric space and represents a *relationship degree* between these entities as a distance measured by a distance function defined on a metric space. The MeSOD enables users to define multiple metric spaces and lets them select a suitable metric space (and hence a suitable distance function) for their purpose.

By introducing a metric space, the Hyperbook system, based on the MeSOD, provides users with a user-friendly retrieval environment. For example, a metric space displayed on a CRT enables users to understand the relationship degrees visually. Users can visually retrieve and browse through necessary information, and judge

CH2686-4/89/0000/0396$01.00

whether a relation between data is close or distant in perspective.

This paper is organized as follows. The brief concept of the Hyperbook system is presented in section 2, and an informal definition of the MeSOD is discussed in section 3. In section 4, the design of the MeSOD's applications developer's environment, MADE, is discussed. Finally, section 5 is a conclusion.

2. THE HYPERBOOK PROJECT

The following discusses the concept of the Hyperbook system and gives an example of its use.

2.1 The Concept of the Hyperbook System

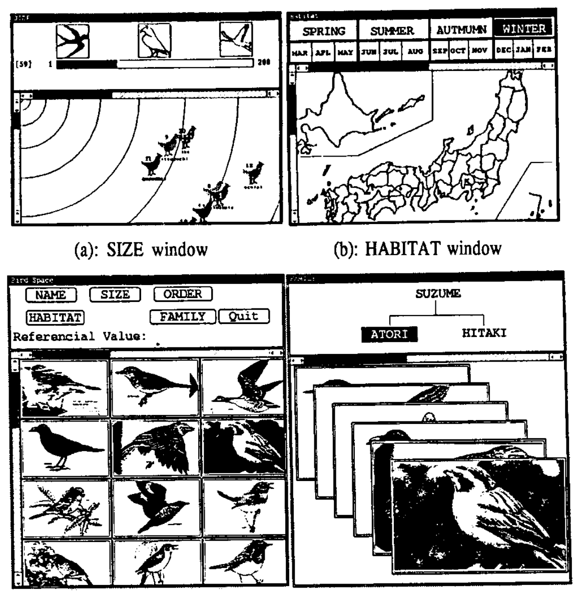
An electronic book system is superior to an ordinary book as it has many capabilities of storing, processing, and extracting information. However, with paper media, it is easier to retrieve and browse through necessary information than with electronic media, because existing electronic media tend to hide information. In a conventional electronic system, the stored information must be retrieved or browsed through by specifying queries by keyword expressions, just like fumbling in the dark. As keyword expressions are inconvenient and difficult for users, it is difficult to make good use of the potential power of the electronic media. Therefore, to be widely accepted by users, the Hyperbook system must provide them with a user-friendly environment in which they can query, retrieve, and browse easily. The fundamental strategy, adopted in the design of the Hyperbook system, is to provide users with a user-friendly and powerful retrieval environment.

The Hyperbook system is a multimedia database system. This system enables users to interact with the system using multimedia data, such as voice, image, and text data. For example, the system enables a user to ask for the name of a bird by entering an imitation of the bird's call or by entering a sketch of the bird's shape into the system. These multimedia form queries are user-friendly, but difficult for computers to handle. The user-friendliness is the capability of these queries to release users from the burden of expressing the queries by keywords. One of the difficulties for computers is that these queries never exactly match the data in the database. Consequently, a multimedia database management system based on the MeSOD, which functions as the nucleus of the Hyperbook system, requires the facilities to handle multimedia queries to enable multimedia communication.

2.2 An Example of Retrieval in Hyperbook

The following example explains an interactive process of retrieving birds in a Hyperbook system on birds.

Suppose that a user wants to know the name of a small bird seen near Mt. Fuji in the winter and that he/she cannot describe the details of the bird's features, but that he/she may be able to identify the bird if he/she sees it again. A SIZE window in Figure 1(a) lets the user specify the size of a bird with a slider. Note that the user need not know either the exact size of the bird or the internal structure of the SIZE data in the database, such that the SIZE data consists of the wing size and the body size. Though the SIZE window seems one-dimensional to users, it implies a two-dimensional metric space, whose coordinates correspond to the body size and the wing size of birds. This metric space contains the distance function which measures distances between birds in the space in terms of the size. By specifying an approximate size of the bird, the ordered set of the birds with the approximately same size is extracted. Therefore, a user needs neither worry about the burden of selecting an adequate keyword expression nor know the internal organization of the database.



(a): SIZE window (b): HABITAT window (c) : BIRD window (d): FAMILY window

Figure 1: A Retrieval Example in a Hyperbook on Birds.

If the user knows more detailed information about the bird, then he/she can narrow the selection further as follows. Figure 1(b) shows that the user indicates Mt. Fuji in the HABITAT window to specify that the bird inhabits an area near Mt. Fuji in the winter. In this case, the HABITAT window involves two metric spaces: the first one is a two-dimensional geographical space, and the second one is a one-dimensional time space. By taking a cartesian product of these two spaces together with the one above, the user can get a new metric space on which birds are displayed. While browsing through the area in this space specified above, the user may find a bird looking like the objective bird.

After finding the family name of the bird, the user may want to know the other birds that belong to the same family (Figure 1(c)). By specifying the family in the FAMILY window, the user gets the birds that belong to the family (Figure 1(d)). Thus, though at first a user may have vague knowledge about the objective bird, the user can interactively retrieve more knowledge by browsing through new information on birds in each visual metric space.

This interactive retrieval process is more useful in the multimedia environment than in the single medium environment, because the relationship degrees, represented as distances in the MeSOD, can be applied to match the user's queries with the multimedia data in the database. By directly entering the user's imitation of the birdcall or his/her sketch of the bird, for example, the user can get an ordered set of the birds with similar calls and shapes. Thus, the factor of the distance, the most important concept in the MeSOD, is useful because it enables users to represent their queries directly, and to retrieve information with vague queries.

3. MeSOD

In this section, the basic idea of MeSOD is informally introduced. (The formal description is given in [4].)

To begin with, a brief overview of the MeSOD is presented. The MeSOD maps entities in the real world onto a metric space, which corresponds to a topological space in mathematics. The MeSOD represents a relationship degree between entities in the read world, as a metric (distance) measured by distance functions defined on the metric space. The MeSOD enables users to define multiple metric spaces (and hence multiple distance functions) and to select a suitable distance function for their purpose. The user's selection of the distances reflects the movement of the user's viewpoint on purpose (subsection 3.1). The MeSOD is informally described in the subsection 3.2. The advantage of the information space organized by the metric phase is to enables users to represent vague queries with the relationship degrees (subsection 3.3).

3.1 Metric

The distance, the most important factor, is introduced into the MeSOD to measure the relationship degrees between entities in the real world. The distance between entities is short when they are closely related, and the distance is long when they are remotely related.

A conceptual space, where the relationship degree between entities can be measured in terms of some attribute value, is regarded as an information space. However, such relationship degrees between entities with an attribute value cannot be handled in conventional data models. As an example, let us take an entity called *name* whose domain is a string. Resemblance of names is conceivable from various viewpoints. For example, the pronunciation distance and the visual symbol distance are easily conceived as the name distance. The pronunciation distance corresponds to the non-similarity degree of each set of a vowel and a consonant in strings, and the visual symbol distance corresponds to visual non similarity degree in strings, that is spelling. These distances should be selected appropriately for the purpose and in the situation of retrieval of names in the database. For instance, the pronunciation distance should be employed by a user who wants to search with a name whose pronunciation he/she is not sure of, and the visual symbol distance should be employed by a user who has no confidence in his/her memory of the spelling.

Thus, the distance between entities corresponds to the relationship degree between them and may be changed by properly selecting the metric space. In other words, each entity always has a relationship degree with the others as a distance value. Also, choosing one of these metric spaces appropriately reflects the movement of the user's viewpoint of retrieving.

Additionally, by displaying a metric space properly, an application based on the MeSOD provides users with the retrieval environment where users comprehend easily how pieces of information are related to each other. The visual representation of a metric space enables users to understand the overall relationship degree between data in the database at a glance. Therefore, users can confirm and learn the relationship degrees between data visually. Moreover, users can define a personal metric space for personal use from those provided in a database system. For example, a user can collect frequently used data from a database, assemble a subset of the database, and define a metric space for the subset.

3.2 Informal Description of the MeSOD

The following informally describes the MeSOD.

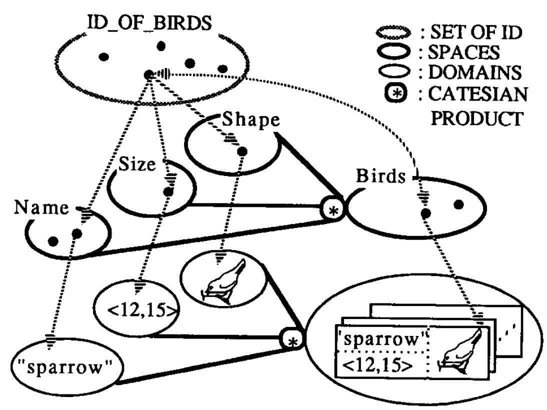


Figure 2: The Concept of a Space Object

Figure 2 illustrates the concept of a space object in the MeSOD. Let us suppose that a database of birds is being dealt with. 'Birds' corresponds to a set of object identifiers 'ID_OF_BIRDS', and each object has common attributes, such as Name, Size, and Shape. A metric space can be defined, for each attribute, such as the Name space, the Size space, and the Shape space. The value of each attribute is defined as a point in each space. The BIRDS space, in which a bird is defined, is composed of three **property spaces**: that is, Name, Size, and Shape. Thus by combining several spaces, more abstract space can be composed, that is called the **cartesian product space**. The distance functions on the BIRDS cartesian product space are defined as cartesian product distances composed of the distances of each property space, and they correspond to the relationship degrees between birds. Each of the property spaces has a "*part-of*" relationship with a cartesian product space.

Also, in case a user wants to add a special characteristic to a subset of birds, new spaces can be defined for it. These new spaces are called **specialized spaces**, and an original bird space is called a **generalized space**. For example, if a user wants to divide the BIRD space by the family of birds, then a specialized space which corresponds to a subset of birds in the same family can be generated. The user can add new attributes to each of the specialized spaces. Although these specialized spaces depend on the method used to divide an original space, a space to which all specialized spaces belong is automatically generated to manage them for every method of division. This space, called a **quotient space,** treats specialized spaces as points, and has the distance functions which measure the relationship degrees between specialized spaces. In the above example, a new quotient space includes all the specialized spaces whose birds belong to the same family. This quotient space maintains an "*is-a*" relationship between each of the specialized spaces and a generalized space.

3.3 Advantage of Retrieving by Distance

The advantage of retrieving by distance is to enable users to represent vague queries, including multimedia queries. This advantage is discussed using an example of a Hyperbook system on birds below.

This system stores not only the bird's properties: such as name, habitat, size, and call, but also the relationship degree between the values of the characteristics. It enables users to represent vague queries such as what kind of a bird seen in a forest or heard singing in a mountain valley is, and what kind of migratory birds are in Japan now. These queries are too vague to permit the selection of an adequate keyword to represent the contents of the queries. To respond to retrieval queries in a database, the relationship degrees between properties of birds are required.

3.3.1 Examples of Retrieval in the MeSOD: In the following, the suitability of the distance for vague queries, including multimedia form queries, is illustrated and indicated by retrieval examples based on distance. Two examples of retrieval are contrasted with those of the conventional data model: the relational data model.

Example 1: `What is a bird as small as a 'sparrow' ,that is seen in or around Tokyo in the winter ?`

This query example is very vague, because the size and the habitat of the bird is inexactly specified. This example includes the relationship degrees in size, place, and season.

<MeSOD>

By throwing '*winter*' and '*Tokyo*' into the HABITAT space, which is the property of birds space, and the 'size of *sparrow*' into the SIZE space, each neighborhood is extracted in each space. The intersection of these neighborhoods contains small birds such as the '*Japanese Great-Tit*' and the '*Ussurian Yellow-breasted Bunting*' as nominees. After that, by actually seeing a photograph of each nominated bird and hearing the birdcall, an objective bird can be retrieved.

Because, the BIRDS space is a cartesian product space, a user can query it by the relationship degrees between the relations composed of the SIZE space and the HABITAT space. In this case, the neighborhood of the query relation <sizeof("*sparrow*"), <"*Tokyo*","*winter*">> is extracted in the cartesian product metric space: SIZE * HABITAT.

```
/* get filed information */
SQL> fields BIRDS
   Name Area Season BodySize WingSize Family Order
/* identify attributes of the sparrow */
SQL>select BodySize, WingSize, Area, Season
   from BIRDS
   where NAME = 'sparrow'
Result: BodySize|WingSize|Area     |Season
             12|      15|Anywhere|Any season
                ........
SQL> select * from BIRDS
where BodySize between 10 and 14
and   WingSize between 13 and 17
and   Area is in <Tokyo, Kanagawa, Chiba, Saitama>
and   Season is in <winter, spring>
                ........
```

Figure 3: A Query Example with SQL

<Relational model>

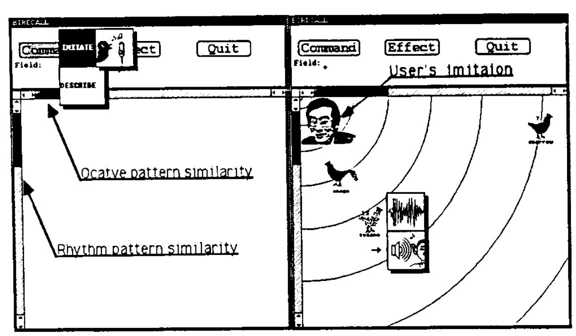
The procedures in Figure 3 are very burdensome and not user-friendly. Also, note that the bottom of the SQL statements cannot represent the specification of the extent of the query relation.

Example 2: `What is a small bird that makes a clicking sound, and has a short beak, and a round body ?`

This example is adequate for a multimedia form query. A user inputs an imitation of the birdcall and a sketch of the bird's shape into the system.

<MeSOD>

In the MeSOD, this query can be expressed by distance functions. The relationship degree between



(a) Specify input method. (b) Display neighborhood and verify.

Figure 4: BIRDCALL Space on a Window.

birds in terms of the call is measured by the voice distance function defined in the BIRDCALL space in Figure 4. Figure 4(a) shows a situation where a user selects to input his/her query by an imitation. After he/she inputs the birdcall by voice, the system responds by displaying a list of birds with birdcalls similar to the user's imitation as shown in Figure 4(b). In a similar way, the relationship degree between birds in terms of the shape is measured by the shape distance function defined in the SHAPE space. After entering his/her imitation and sketch into the BIRDCALL space and the SHAPE space, the user can retrieve and browse through each space or cartesian product space composed of these two spaces.

<Relational model>

In this query example, the searching based on keywords is almost impossible because of two difficulties. First, the imitation of the birdcall and the sketch of the bird's shape cannot be entered. Second, even if it were possible to enter the imitation of the birdcall and the sketch of the bird's shape, neither the imitation nor the sketch would match the bird's voice data or shape data in the database.

The MeSOD enables the user to represent queries freely, and is able to respond to such vague queries by introducing the concept of distance.

4. DESIGN OF THE MeSOD'S APPLICATIONS DEVELOPER'S ENVIRONMENT

In this section, the design of the MeSOD's applications developer's environment, MADE, is discussed. MADE includes a DBMS, based on the MeSOD, and several facilities, which supports the development of a window-based interface and the handling of special devices, to implement the Hyperbook system. The functions of the Hyperbook system include the capabilities of easy querying, retrieving, and browsing through the necessary information comfortably, and storing meaningful information.

MADE has been designed to make the development of the Hyperbook system easy. The design of MADE is based on an object-oriented approach. MADE and the concept of its objects are explained below.

4.1 MADE

MADE must support the following facilities for developers:

(1) to provide common user-interfaces with the MeSOD's application: that is, a Hyperbook system.

(2) to employ comprehensive tools easily and effectively.

(3) to manage the data in the database completely.

To meet these requirements, MADE consists of three layers to provide an extremely flexible architecture: that is, the user-interface layer, the application interface layer, and the MeSOD layer (Figure 5).

The user-interface layer contains visual objects, which know how to portray objects in the MeSOD layer (MeSOD's objects) and to receive messages (commands, queries and requests) from users through direct manipulation such as clicking and dragging a mouse, and typing on a keyboard. A visual object knows how to lay out oneself on a CRT according to spatial information acquired from the MeSOD layer. Because visual objects on a CRT enable users to manipulate and send commands to a visual object directly, they can visually represent queries and verify their results. Thus, a visual object plays an important role in the representation of the metric spaces and the interaction with a system.

The application interface layer is the interface for developers who write a Hyperbook program based on the MeSOD by directly making better use of the MADE capabilities, and it supports the mapping of a MeSOD's object to visual objects. An object in the application interface layer is a mapping object that connects a MeSOD's object and a visual object and it knows how to communicate with the two objects. An object in the application interface layer extracts the logical information of a MeSOD's object, processes it, and transmits the processed information to a visual object. Because the objects in the application interface layer know how to call functions in the other layers, in most cases developers can rapidly build an application by employing the facilities in only this layer.

The MeSOD layer has the responsibility for object management and persistence.

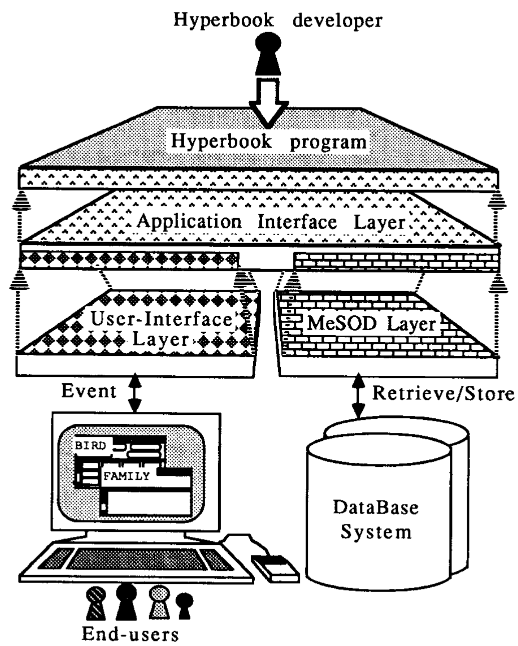


Figure 5 : Constitution of MADE and a Hyperbook Program

4.2 What is the MeSOD's Object

The MeSOD layer functions as a DBMS based on the MeSOD. The layer is implemented according to an object-oriented approach. The object-oriented concept in the MeSOD layer is explained below.

A MeSOD's object consists essentially of a private memory with a public interface, similar to SmallTalk. The private memory is a structured list of state variables, such as object name, element-of relationships, and space-type variables. The public interface contains primitive methods to instantiate new objects, to manipulate space or point objects spatially, and to extract the neighborhood of a query.

Two kinds of objects exist in the MeSOD layer. The first one includes space objects and point objects representing concepts of spaces and points respectively. These objects are frequently used to represent entities in the real world in the MeSOD's applications. The second one includes space-type objects which instantiate space objects and point objects, and are used in schema definitions in the MeSOD's applications. These objects are prototypes of metric spaces that contain common distance functions, such as distance functions of numbers, strings, and more complex-structured data types. For example, the space-type objects include the Number space-type, the String space-type, and the Bitmap Image space-type. These space-types are employed to instantiate the Age space object, the NAME space object, and the PICTURE space object.

MADE is written in Objective-C and C languages, where about 20 new classes are programmed in about 7,000 lines and C functions are programmed in about 1000 lines. MADE includes SunView, which is used to implement the visual objects in the user-interface layer. and the UNIFY database, which is used to manage the data dictionary of a database based on the MeSOD.

Currently, a subset of the Hyperbook system (Tiny-Hyperbook system) is built using MADE. The Tiny-Hyperbook system presents multimedia information, although the system enables users to query by only single media information (number and string).

5. CONCLUSION

This paper introduced the Hyperbook system and proposed the MeSOD, which is a clue adopted to implement the Hyperbook system. The Hyperbook system is a new electronic book with the capability of handling multimedia information. The Hyperbook system is intended to support several facilities: interaction with the system by multimedia information, customization of the retrieval environment, and management of multimedia information.

The stored information is retrieved by a query language in applications based on existing data models, and a user must know how information is organized in the database and how the language is used. A user is generally reluctant to study such details; therefore, a user must retrieve necessary information as if he/she were fumbling for information in the dark.

The MeSOD represents a world as metric spaces, and the Hyperbook system based on the MeSOD permits users to light up the direction (query) indicated in the subspace with several powerful spotlights (neighborhood). Furthermore, a user can freely combine and divide metric spaces to browse through information. While browsing through these metric spaces, the user can indicate a direction that is more explicit than the first indicated direction. The user of a system based on the MeSOD can retrieve and browse in his/her own way.

REFERENCES

[1]D. Bryce and R. Hull, "SNAP:a graphics-based schema manager", Int. Conf. Data Eng. 1986-2, IEEE Comput. pp. 151-164

[2]S. Abiteboul and R. HULL,"IFO:A Formal Semantic Database Model", ACM TODS, Vol. 12, No.4, Dec. 1987, pp. 525-565.

[3]G.Wilson and C. Herot, "SEMANTICS VS. GRAPHICS --TO SHOW OR NOT TO SHOW", Proc. of VLDB'80, pp. 183-197.

[4]M. Tabuchi and Y. Muraoka, "Proposal of the data model (MeSOD) based on metric space", Bulletin of the Centre for Informatics, Waseda University, Vol 7, SPRING, 1988, pp. 20-29 (in Japanese).

[5]N. Yankelovich, et al, "Intermedia: The Concept and the Construction of a Seamless Information Environment", *Computer*, Jan. 1988, pp. 81-96.

[6]N.Meyrowitz,"Intermedia: The Architecture and Construction of an Object-Oriented Hypermedia System and Applications Framework", Proc. OOPSLA 86, Sept. 1986, pp. 186-201.