Formalizing a Hypermap Object Model for Multimedia-Geographical Information Systems[*]

Jonás Arturo Montilva C.

Computer Science Department , University of Los Andes
Faculty of Engineering, Data & Knowledge Engineering Research Group
Mérida, Mérida, 5101, Venezuela

ABSTRACT

The notion of hypermap emerged as a response to the need for adding to geographical information systems (GIS) the linking and semantics capabilities of hypermedia. A hypermap is a new and powerful instrument for thematic cartography and spatial analysis in GIS. It extends the ability of a GIS to represent, store, manipulate, and display digital maps with the associative linking and interactive capabilities of hypermedia and the semantics power of dynamic multimedia data (i.e., audio, video and animation). A hypermap increases the semantic expressiveness of a GIS, by allowing its users to navigate and manipulate multimedia spatial information, which were not possible, not even conceivable, with conventional GIS.

Although the notion of hypermaps has been around for at least five years, a formal model that describes precisely this notion has not been published in the literature. A formal definition of hypermaps is needed in order to standardize the concepts and facilitate its implementation.

This paper presents a *hypermap object model*, a description of the collection of classes needed to create hypermap applications in GIS. This model is formalized using graph theory and the Z++ notation, an object-oriented extension to the widely-used Z specification language. The model can be used as a conceptual framework for the design of multimedia GIS; and as an object-oriented design for building a class library in an object-oriented programming language. The model has been fully implemented as a library of classes in C++ and it is actually been used for the development of hypermap applications.

Keywords:

hypermaps, multimedia maps, multimedia & geographical information systems, spatial object-oriented models

1. INTRODUCTION

Geographical information systems (GIS) are a kind of information systems that have the ability to create, access, analyze, and manipulate spatial data, about a given region of the earth, using digital maps as the unit of manipulation. Maps are created and manipulated by a GIS using graphical data types, such as points, lines, areas and cells called tessellation. Extending this set of graphical data types with multimedia data types (i.e., unformatted text, still images, audio, animation and video) increases enormously the semantic power of maps. This integration allows the user to manage new types of information that were not possible, not even conceivable, with the cartographic facilities of a conventional GIS.

The notion of hypermap emerged from the idea of extending the functionality of a GIS with the potential of hypermedia to manipulate multimedia data types. A hypermap is a hyperdocument extended with geographical references, in which the location of an object on a map is used to retrieve its associated multimedia data through mouse clicking. The notion of hypermap is relatively new. Although several definitions of this notion are found in the literature [LAU90, WAL90, LAU92, BOU92, MON93], no attempt to formalize its concepts has been published yet. The literature on hypermaps is characterized by the absence of formalism. Wallin [WAL90], for instance, discusses only the main ideas behind the notion of hypermap. Laurini, Milleret-Raffort and Thompson [LAU90, LAU92] give a more detailed, but informal, definition of hypermaps and discuss some aspects of the implementation. The use of object-orientation for the design and implementation of hypermaps is not considered by any of the previous authors. A first attempt to formalize the ideas behind hypermaps is presented in [MON93]. The model described in this paper emerged from this previous work.

We believe that formal definitions of hypermaps are needed in order to standardize the concepts and facilitate its implementation. In this paper, we make a contribution to the process of formalizing the notion of hypermaps. We propose first a formal definition of hypermaps based on

[*] Proceedings of *II International Conference on Information Systems Analysis and Synthesis - ISAS'96.* Orlando, USA, Julio, 1996. p.64-70.

graph theory. This definition is then used to design and formalize a *hypermap object model*; that is, a collection of re-usable object-oriented classes that can be used for the development of hypermaps. The model is described using OMT, an object-oriented modeling technique [RUM91]. The structure and behaviour of each class of the model is formalized using the Z++ technique [LAK91], a formal specification language based on the so-called Z Notation [SPI92].

The model can be used for designing and implementing hypermap tools for building multimedia GIS applications. It can also be used as a re-usable object-oriented design for developing multimedia GIS.

The paper is organized as follows. Section 2 presents a formal definition of hypermap. The hypermap object model is described in section 3. The formal specification of the classes of the model are given in Section 4. The implementation of the model is briefly mentioned in Section 5. Finally, the concluding remarks are discussed in section 6.

2. A FORMAL DEFINITION OF HYPERMAPS

The notion of hypermap is the result of integrating hyperdocuments and digital maps. A hyperdocument is a non-sequential multimedia document whose structure is a network or hierarchy of nodes (called cards, units, notecards, frames, etc.) composed by multimedia information (i.e., text, graphics, images, audio, animation and video) and interconnected by links. A hyperdocument can be organized, written and read in any order determined only by links between the information nodes.

On the other hand, digital maps are the object of study and manipulation in GIS. A map is a model of the spatial distribution of significant entities of a portion of the earth' surface [BUR86]. Properties of the entities existing on the earth' surface are classified into spatial and non-spatial properties. In the vector approach to GIS, spatial properties --geometrical and topological -- are represented in a digital map using graphical data types, such as *points, lines* and *polygons.* These data types are the basic modeling constructs used to represent in a GIS the geometry of spatial entities and their location in space with reference to a given coordinate system. An important component of a map is the legend, which associates the non-spatial attributes (non-spatial represented properties) to the spatial objects (spatially represented entities) displayed on the map.

A hypermap can now be informally defined as a hyperdocument about a geographical region, in which the spatial properties of its entities are represented by one or more maps linked to multimedia information in the form of text, graphics, images, audio and video. Multimedia data can be used to represent the non-spatial properties of the objects contained in a map. Objects on a map are connected with multimedia information through links. Both the name and the location of an object on a map can be used as buttons that when are activated (pressed) cause the retrieval and display of its associated multimedia information, as shown in Figure 1.

More formally, a hypermap can be defined as follows:

Definition 1: A *hypermap* is a directed graph *G = (N, L)*, in which *N* is a finite set $\{n_i\}$ of information nodes and *L* is a finite set $\{l_K\}$ of links. Each link l_k □ *L* connects two nodes n_i and n_j; i□j, n_i, n_j □ *N*.. Each node n_i □ *N*. contains a finite set $\{b_m\}$ of buttons and a finite set $\{i_j\}$of items of the following types: map, text, graphics, image, audio or video.

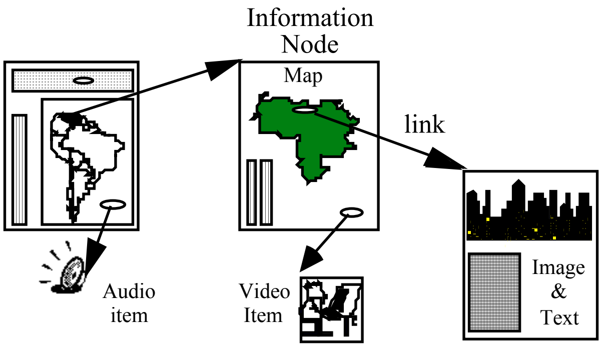

Figure 1. A hypermap example

The structure of an information node is shown in Figure 2. It should be noted that a map is considered here as another item of information. An information node may consist of a single map, a set of information items including one or more maps or a set of multimedia items only.

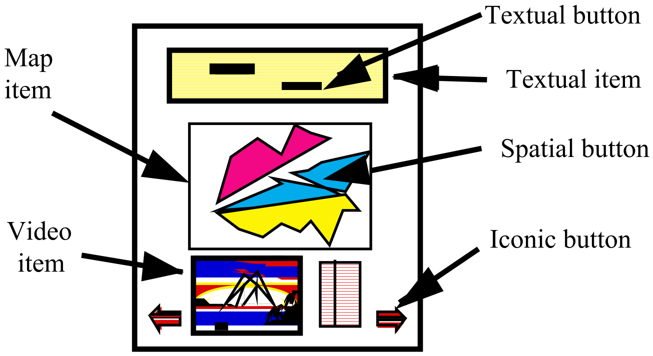

Figure 2. Structure of an information node

Note that, in the given definition, we concentrate only on the static or structural properties of a hypermap. The dynamics of a hypermap, that is, the behavioral properties that allows a user to navigate through the hypermap´s nodes, are omitted from the definition. The reason of this is that hypermap dynamics is associated with the graphical user interface that allows the creation and manipulation of hypermaps.

We, therefore, see a hypermap as an object-oriented database that has associated a separated graphical user interface. The hypermap database stores the information nodes as composite objects whose components are either maps or multimedia objects. The links that connect the information nodes in the database are also objects that have associated displayable objects, called buttons. Buttons are shown and activated in the graphical user interface. They allow the user to navigate through the information space of the hypermap.

The graphical user interface of a hypermap creates a highly interactive environment based on the intensive use of windows, sounds and pointing-and-clicking actions based on a mouse. The visual data of the information nodes (i.e., text, graphics, images, maps, animated or video clips) of a hypermap are displayed on windows. Audio data, on the other hand, is reproduced using the audio output facilities (e.g., headphones) provided by the hardware. The user is allowed to move from the active node (the currently displayed node) to another visual or audio node, by activating the buttons displayed on the window.

3. THE HYPERMAP OBJECT MODEL

The design of the hypermap object model is the result of integrating two models: the object-oriented spatial data model of S.A. Roberts, et al. [RGH91], called here *OO-GIS model*, for brevity; and a hypermedia reference model (*HMR model*) described in [MON93]. The OO-GIS model was chosen because of its expressive power for representing spatial objects, and its semantics for modeling and manipulating multiple geometrical representations of the same object, which is essential for modeling spatial databases. The HMR model, on the other hand, is a reference model based mainly on the Lange´s hypermedia formal model [LAN90] and is extended with some features drawn from the Dexter Hypertext Reference Model [HAL94]. A description of each of these two models and the details of the application of the integration methodology to the hypermap model design are given in [MON95].

An object model is a term used in Object-Orientation to refer to the definition of a set of classes associated with a given domain or object of study. It captures the structure and behavior of the objects (i.e., concepts or entities) that comprise the domain. The hypermap object model is, therefore, a representation of the components of a hypermap. It describes the classes needed to create hypermap applications. This set of classes are shown in Fig. 3 using the OMT object modeling notation [RUM91].

According to this model, a hypermap is a composite object made of spatial information nodes. It forms a network of information nodes (also called, units, cards, or pages) related by instances of the class *Link*. Each information node is also a composite object. It is composed by a set of instances of two classes: *Item* or *Button.* The class *Item* is specialized into the following subclasses: *Textual_Item*, *Graphical_Item*, *Image_Item*, *Audio_Item*, *Video_Item* and *Map*. These subclasses provide the operations needed to create, manipulate and display spatial and multimedia information.

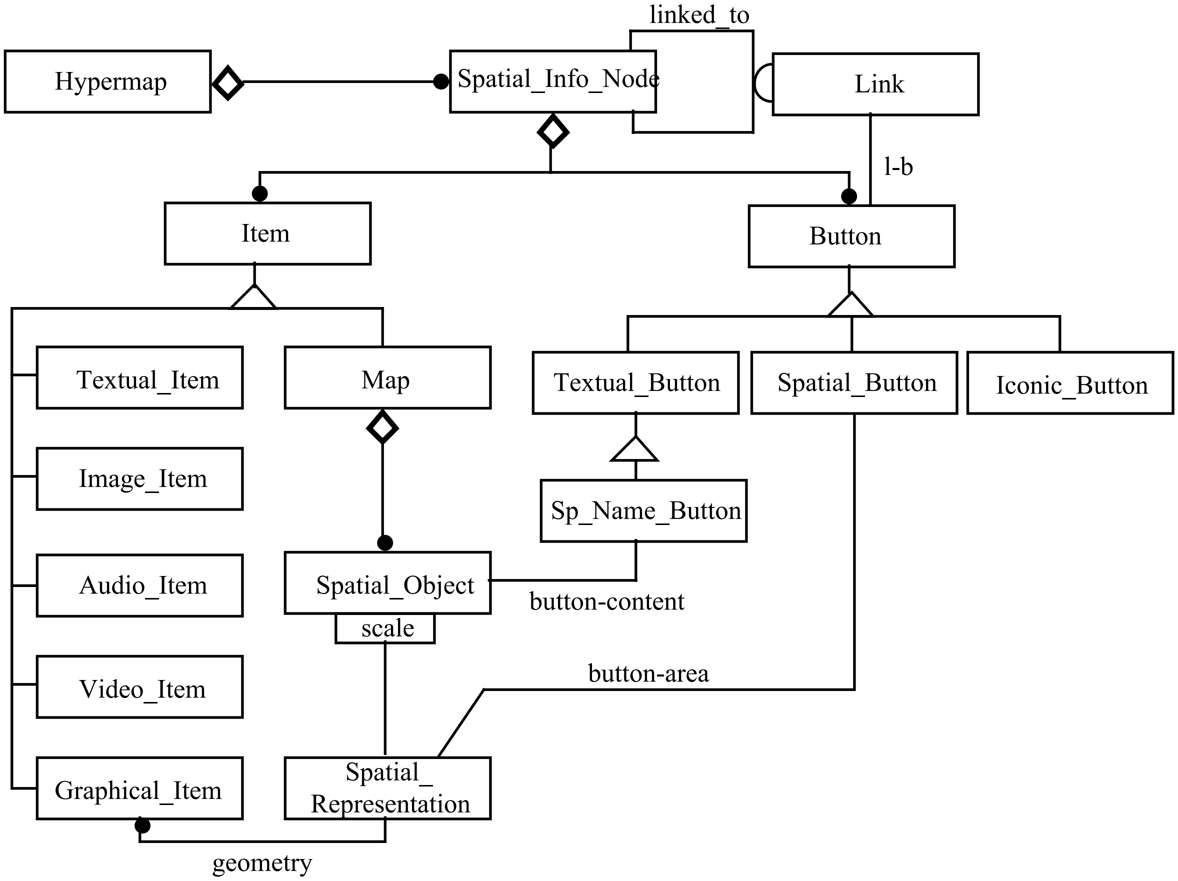

Figure 3. The hypermap object model

It should be noted that a map is defined as a special kind of item. As defined by the class *Map*, a map is composed by a set of spatial objects. Each spatial object has associated one or more spatial representations. Each spatial representation is an object that defines the geometry of the object for a given scale. This feature allows the user to create and manipulate maps of the same region at different scales.

The model supports only a vector representation of spatial data. The location and shape of a spatial object are, therefore, defined by a set of graphical objects, i.e., points, lines and polygons. However, a raster representation can easily be incorporated by specializing the *Image_Item* class.

The model includes the textual and iconic buttons used in most hypermedia systems. Two new types of buttons emerged from the integration of the OO-GIS and HMR data models. The first type of button, called *Sp_Name_Button*, uses the spatial object's name to link the object with another information node. The second one, called Spatial_Button, is related to the spatial representation (geometry) of a spatial object. The shape of a spatial button is the geometry of the object, that is, the button is activated by clicking on any location of the object's geometry.

4. A FORMAL DEFINITION OF THE HYPERMAP CLASSES

The object model shown in Fig. 3 enumerates the classes needed to build hypermaps. It also shows the generalization, composition and associative relationships between classes. To fully describe the model, the details of each class must also be provided. Instead of using the OMT technique for this purpose, we preferred to apply the Z++ Notation [LAK91]. The reason for this is that the Z++ notation produces a formal and more detailed definition of classes than OMT. As a formal specification language, Z++ provides a collection of mathematical constructs for specifying the attributes and methods that comprise a class.

The formal specifications of the main classes of the hypermap object model are given in this section. For space reasons, some details of the specifications are omitted.

The *Hypermap* class is defined in an equivalent way to a undirected graph class. An instance of this class is comprised by a set of instances of the *SpInfoNode* class and a set of instances of the *Link* class. Each instance of the latter connects two instances of the former. Most of the operations defined by this class are equivalent to those commonly supported by a graph [MON96].

The structure of a *Hypermap* instance is made of a title, a set of spatial information nodes, a set of links between information nodes. One of the nodes is made the root of the hypermap. It is the unit to be displayed when a hypermap is opened for manipulation or navigation. An extensive set of operations for editing, visualizing and manipulating hypermap instance are included in the Hypermap class specification, as shown next:

```
CLASS Hypermap
    OWNS
        title: STRING ;
        nodes:     P SpInfoNode  ;
        linksP Link     ;
        root: SpInfoNode    ;
        ...
    INVARIANTS
        ∀ l: links □ l.source □ nodes □
                    l.destination □ nodes ;
        root □ nodes;
        ...
    OPERATIONS
        INIT:              ---> ;
        AddTitle STRING  ---> ;
        AddNode:       SpInfoNode ---> ;
        AddLink: Link ---> ;
        DeleteNode:    SpInfoNode ---> ;
        DeleteLink:    Link ----> ;
        DisplayNode: WINDOW  SpInfoNode---> ;
        EditNode:      SpInfoNode --->
                            SpInfoNode  ;
        ...
END CLASS
```

Each hypermap is comprised by a set of spatial information nodes. Each of these nodes contains spatial or multimedia information. Each *SpInfoNod*e instance has a title, a set of items of information (i.e., the multimedia items or components of the node), and a set of buttons associated with the links between information nodes. Operations are provided for manipulating, editing and visualizing spatial information nodes, as shown in the following schema.

```
CLASS SpInfoNode
    OWNS
        title:       STRING ;
        items:           P Item ;
        buttons:         P Button  ;
        infoSpace:       RECTANGLE ;
        ....
    OPERATIONS
        INIT:            ---> ;
        addTitle: STRING ---> ;
        addItem:  Item --->   ;
        addButton:       Button ---> ;
        delItem:  Item --->   ;
        delButton:       Button --> ;
        Display:  WINDOW   --->  ;
        EditItem  Item ---> Item ;
        EditButton:      Button ---> Button ;
        ...
END CLASS
```

Each instance of *SpInfoNode* is composed by a set of instances of the *Item* class (or any of its subclasses) and a set of instances of the *Button* class (or any of its subclasses). *Items* and *Button* instances are organized into the information space (*infoSpace*) of the spatial information node, as a set of non-overlapping rectangular areas.

The *Link* and *Button* classes are formally defined as follows:

```
CLASS Link
    OWNS
        source:    SpInfoNode ;
        destination:     SpInfoNode ;
        button:    Button ;
    ...
    INVARIANTS
        source □ destination ;
        ...
    OPERATIONS
        linkedTo:  ---> SpInfoNode ;
        ..
END CLASS
```

```
CLASS Button
    OWNS
        assocLink:       Link ;
        frame:           RECTANGLE ;
        location:  POINT
        ...
    OPERATIONS
        INIT:      ---> ;
        Display:  ---> ;
        Edit:---> ;
        inNode:   ----> SpInfoNode ;
        toNode:  ----> SpInfoNode
        activate:  ---> ;
        ...
END CLASS
```

Each instance of *Link* connects two different instances of *SpInfoNode* in one direction: from the source to the destination node. Each links has associated a button, which allows the user to navigate, in the graphical user interface, from one node to another. Each instance of *Button* occupies a rectangular area, called frame. This area is located a given position in the information space of the associated node. Navigation through the hyperspace is performed by the methods *toNode* and *activate*. The former returns the destination node of the corresponding link of a button; whereas, the latter signals the hypermap to display the destination node.

Four subclasses of buttons are provided by the model. The Textual_Button class allows the user to associate a button to a substring of a textual item. The Iconic_Button class defines buttons whose shapes are icons. They are useful for navigating, performing actions or playing audio tracks, animation, and video clips. *Spatial_Button* and *Sp_Name_Button* are used for navigating from a map to another map or multimedia information node.

The formal specification of the Item class is illustrated next together with the *VideoItem* class specification.

```
CLASS Item
    OWNS
        infoSpace:      RECTANGLE ;
        inNode:         SpInfoNode ;
        location:  POINT ;
        ...
    OPERATIONS
        INIT:      ---> ;
        Show:      ---> ;
        Edit: ---> ; ...
END CLASS
CLASS Video Item EXTENDS Item
    OWNS
        item: VIDEOCLIP
    OPERATIONS
        Play:  --->  ;
        Reverse:  ---> ;
        FastForward:  ---> ;
        ...
END CLASS
```

The model supports a rich set of multimedia data types. Text, graphics, images, audio tracks, animation and video clips can be used as data types in the information nodes of a hypermap. The *Item* class provides common structure and behaviour to all these types of multimedia items. Each item occupies a rectangular space (*infoSpace*) that is used to organize the information space of the corresponding spatial information node.

Maps are also considered as items of information. A *Map* instance is made of a set of *SpObject*. instances; each of them has associated a set of *SpRepresentation* instances, one for each scale used by the hypermap. A spatial representation describes the geometry of the spatial object at the corresponding scale. The geometry of a spatial object is made of a set of graphical items (i.e., points, lines or polygons).

Parts of the formal specifications of the *Map*, *SpatialObject* and *SpRepresentation classes* are given next. More details of these specifications are given in [MON96].

```
CLASS Map EXTENDS Item
    OWNS
        title:      STRING ;
        spObjects:      P SpObject ;
        scale:          SCALE ;
        legend:         P LEGENDUNIT;
        ...
    OPERATIONS
        Display:  WINDOW ---> ;
        ChangeScale:  SCALE ---> ;
        AddSpObject: SpObject ---> ;
        ...
END CLASS

CLASS SpatialObject
    OWNS
        name:     STRING ;
        reps:     SCALE ∅  SpRepresentation;
        button:   SpNameButton ;
    OPERATIONS
        AddSpRep: SCALE  SpRepresentation --->;
        DelSpRep:  SCALE  --->  ;
        Display:     WINDOW  SCALE  --->  ;
        ...
END CLASS
CLASS SpRepresentation
    OWNS
        geometry:       P GraphicalItem  ;
        scale:          SCALE ;
        button:         SpatialButton;
        ...
    OPERATIONS
        AddGeogObj: GeogObject --->   ;
        DelGeogObj:  GeogObject --->   ;
        Display:  WINDOW    --->   ;
        ...
END CLASS
```

Each *SpatialObject* instance may have associated two types of buttons. The first type, SpNameButton, is associated with the object's name and behaves as a textual button; whereas the other type, *SpatialButton*, is related to each spatial representation of the object. A *Spatial_Button* takes the shape of the spatial representation (geometry) of the object for a given scale. Clicking on any location of the spatial object's geometry on a map causes the activation of its spatial button. This is an important difference between our hypermap model and other approaches (e.g., [LAU92], [ERW92]), in which the clicking area of the spatial object is defined as a rectangular button.

In general, the major distinction between our definition of hypermap and spatial hyperdocuments (i.e., hypermedia applications that display maps as images) is that our approach supports all the functionality provided by a GIS. The user is, therefore, allowed to manipulate the map items of a hypermap. By applying the operations defined by the *Spatial_Object* and *Spatial_Representation* classes, the user may perform, on any map of a given hypermap, operations such as intersection, overlay, buffering, calculation of distances, areas, and perimeters, graphical operations (e.g., rotation, translation) and scale transformation.

Spatial hyperdocuments do not provide this functionality. They are limited to the display of the map. Despite navigation and linking to/from maps are supported, maps don't provide the mechanisms for manipulating their objects. In contrast, a hypermap allows the user to manipulate interactively the spatial objects associated with each of its maps.

5. IMPLEMENTING THE MODEL

The hypermap object model has been fully implemented as a library of classes using the C++ object-oriented language in a UNIX platform. PEX and Xlib were used for implementing the windowing and spatial analysis operations. Xlib provides the support needed to display the spatial information nodes on windows. The PEX graphical library was used to provide the graphical support needed by the *SpRepresentation* and *GraphicalItem* classes. The details of the implementation are given in [CHA95], [OCH95].

6. CONCLUSIONS

The main contributions of this paper are the formalization of the notion of hypermap as a directed graph and the introduction of a hypermap object model that can be implemented in any object-oriented language.

The notion of hypermap in GIS is relatively recent and the related literature is limited. Our definition is inspired on that given by Laurini, Milleret-Raffort and Thompson [LAU90, LAU92]. The main differences are that our definition: (1) is formally established; (2) uses an object-oriented approach; (3) incorporates sound, animation, and video; (4) uses a more comprehensive approach to the internal organization of nodes, which are defined as composite objects comprised by one or more multimedia items or maps, instead of only one item as defined by the Laurini's approach; and (5) is the result of applying a software integration method [MON93] to extend the notion of hyperdocuments with digital maps.

The hypermap object model is being used for the design and implementation of a hypermap database system, an object-oriented software tool for creating, manipulating and maintaining hypermaps. A library of classes that implements the model has been fully developed in C++ under a XWindow platform. A hypermap application that describes different geographical and economics themes of the State of Mérida in Venezuela has been implemented [CHA95, OCH95].

The model has three key benefits: (1) it provides the concepts and classes needed to create, access, manipulate and maintain hypermap applications; (2) it can be used as a framework for the implementation of multimedia-geographical information systems; and (3) it serves as a valuable research tool for investigating the integration of GIS and Multimedia at a conceptual level.

7. REFERENCES

[BOU92] Boursier, P. and Mainguenaud, M. Spatial Query Languages: Extended SQL vs. Visual Lanuguages vs. Hypermaps. 5th Int. Symp. on Spatial Data Handling, Vol. 1, 1992.

[BUR86] Burrough, P.A. *Principles of Geographical Information Systems for Land Resources Assessment*. Oxford University Press, 1986.

[CHA95] Chavez, J.L. *Conversión y extensión de una librería de clases de objetos espaciales bajo X Window* (in Spanish). Undergraduate Thesis. Universidad de Los Andes. Facultad de Ingeniería. Departamento de Computaciónm Mérida, Venezuela, 1995.

[ERW92] Erwin, S.M. A Hypermedia GIS - The Massachusetts Turnpike Study. *Computer Environment and Urban Systems*, Vol.16, No.4, 1992, p.375-383.

[HAL94] Halasz, F. and Schwartz, M. The Dexter Hypertext Reference Model. *Comm. of the ACM*. Vol. 37, No. 2, February, 1994.

[LAN90] Lange, D.B. A Formal Model of Hypertext. In Moline, J., Benigni, D., and Baronas, J. (Eds.) *Proceedings of the Hypertext Standarization Workshop*, National Institute of Standards and Technology, 1990.

[LAK91] Lano, K. C. Z++ an object-orientated extension to Z. In Nicholls, J.E. (Ed.) *Z User Workshop*, Oxford 1990, Workshops in Computing, Springer-Verlag, 1991. p. 151-172.

[LAU90] Laurini, R. and Milleret-Raffort, F. Principles of Geomatic Hypermaps. *Proc. 4th Int. Symp.* Spatial Data Handling. Zurich, 1990.

[LAU92] Laurini, R. and Thompson, D. *Fundamentals of Spatial Information Systems*. Academic Press, London, 1992.

[MON93] Montilva, J.A. *An integration methodology applied to the design of a data/knowledge model for multimedia and spatial applications*. Ph.D.Thesis. University of Leeds, School of Computer Studies, Leeds, UK, January, 1993.

[MON95] Montilva, J. A. A methodology for the design of specific-purpose data models applied to multimedia and geographical information systems. *Proc. of the 5th Int. Symp. on Systems Research, Informatics and Cybernetics - ISAS´95,* Baden-Baden, Alemania, Agosto, 1995.

[MON96] Montilva, J.A. *SpGraph: A Spatial Object Model for Suppotring Graph-Based Applications.* Research report. University of Los Andes, Computer Science Department. Data & Knowledge Engineering Research Group. 1992 (Submitted to the International Conference on Information Systems Analysis and Synthesis - ISAS´96).

[OCH95] Ochoa, J.G. *Desarrollo y aplicación de una librería de clases para hipermapas* (In Spanish). Undergraduate Thesis. Universidad de Los Andes. Facultad de Ingeniería. Departamento de Computación, Mérida, Venezuela, 1995.

[RGH91] Roberts, S.A., Gahegan, M.N., Hogg, J., and Hoyle, B. Application of object-oriented databases to geographic information systems. *Information and Software Technology*, Vol.33, No.1, 1991, pp.38-46.

[RUM91] Rumbaugh, J. et al. Object-Oriented Modeling and Design. Prentice-Hall, NY. 1991.

[SPI92] Spivey, J.M. *The Z Notation: A Reference Manual*. 2nd. Ed. Prentice Hall International. 1992.

[WAL90] Wallin, E. The Map as Hypertext. Proc. of the 1st European Geographical Information Systems Conf. Amsterdam. April. 1990. pp. 125-134.