A hypermap data model for integrating multimedia and geographical information systems*

Jonás Arturo Montilva C.

University of Los Andes
Faculty of Engineering
Computer Science Department
Data & Knowledge Engineering Research Group
Mérida, Venezuela
e-mail: jonas@ing.ula.ve

Abstract

Adding multimedia data types to a digital map increases tremendously its semantic capability and creates a new instrument for cartography and spatial analysis in Geographical Information Systems (GIS). The notion of hypermap has emerged as a response to the need for integrating GIS and Multimedia. This paper describes a *hypermap data model*, an object-oriented data model that enhances the ability of a GIS to manipulate maps by adding multimedia data types (e.g., unformatted text, still images, audio, animation, and video).

The model has three key benefits: (1) it provides the concepts, constructs, operations and rules needed to define and manipulate hypermaps; (2) it can be used as a framework for the implementation of multimedia GIS ; and (3) it serves as a valuable research tool for investigating the integration of GIS and Multimedia Systems at a conceptual level.

1 Introduction

To enhance the functionality of geographical information systems (GIS) with multimedia capabilities is becoming an important research direction. A traditional GIS has the ability to create, access, analyze, and manipulate spatial data, about a given region of the earth, using 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 tesselations. Adding multimedia data types, such as unformatted text, still images, audio, animation and video, to a GIS increases the semantic power of its maps, by allowing the user to manage new types of information that were not possible, not even conceivable, with traditional cartography.

The notion of hypermap has recently emerged from the idea of extending the functionality of a GIS with the potential of Hypermedia Systems to manipulate multimedia data types. A

* *Proceedings of PANEL95 - XXI Latin American Conference on Informatics*. (Canela, Brazil, Julio 29 - Agosto 4). Instituto de Informática. Universidad Federal do Rio Grande do Sul. pp. 1139-1150.

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 data models have been proposed yet. This paper introduces a hypermap data model. This model can be used for designing and implementing hypermap database management systems, as well as for modeling hypermap applications in GIS.

The *hypermap data model* was designed based on the conceptual integration of two existing data models: a spatial object-oriented data model proposed by S. A. Roberts, et al., [RGH91] and a hypermedia reference model [MON92]. A software integration method was applied for integrating the concepts, constructs, operation and rules of both models. This method is described in detail elsewhere [MON92, MON93]. The details of how the hypermap data model was designed using the mentioned method is given in [MON95].

The paper is organized as follows. Section 2 defines the concept of hypermap based on the notions of hyperdocument and digital maps. The method used for designing the model is described very briefly in section 3. The concepts, constructs, operations and rules of the hypermap data model are given in Section 4. Finally, the implementation of the model and the concluding remarks are discussed in section 5.

2 The concept of hypermap

The notion of hypermap arises from the integration of hyperdocuments and digital maps. The concept of hyperdocument comes from the Hypermedia-Multimedia technologies; whereas digital maps are the object of study and manipulation in Geographical Information Systems. The definition of each of these two concepts is given first before introducing the notion of hypermap.

2.1 Hyperdocuments

A hyperdocument is a non-sequential 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. A hyperdocument is created, used, and maintained by a multimedia system. A multimedia system is a software tool that integrates into a single framework the functionality of a database management system, a graphical user interface system, and an information retrieval system [CON87].

As a specialized database management system, a multimedia system allows its users to create, access, and maintain multimedia data into a hypermedia database.

As a graphical user interface system, a multimedia system creates a highly interactive environment based on the intensive use of windows, sounds and pointing-and-clicking actions based on a mouse. The visual items of the information nodes (i.e., the textual, graphical, image, animated or video data) of a hyperdocument are visualized using 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 one information node to another, by activating the buttons displayed on a window.

Finally, from the information retrieval perspective, a multimedia system is a powerful medium for the associative and indexed access to the multimedia information stored in the hyperdocument nodes. Information retrieval in a multimedia system is mainly associative, that is, any kind of association (e.g., similarity, conceptual, structural, spatial, etc.) between the information nodes of a hyperdocument is done through links. Access to an information node, or to one of its items of information, is provided by the linking capability of the system.

2.2 Digital maps

GIS are concerned with the collection, organization, storage, analysis and display of data that represent geographical entities. Geographical entities are characterized by: (a) having a location in a 2D/3D space referred to the earth' surface, as well as other spatial properties (e.g., direction, form and area); and (b) maintaining spatial relationships between them (e.g., neighborhood, distance and spatial inclusion).

The unit of manipulation and display in a GIS is a digital map. 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. The spatial properties --geometrical and topological -- are represented in a digital map using *points, lines* and *areas*, which are defined by 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.

2.3 A definition of hypermap

On the basis of the given definitions of hyperdocument and digital map, we can now define a hypermap 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. Objects on a map are associated with multimedia information through links. Both the name and the location of an object on a map are used as buttons that when are activated cause the retrieval and display of its associated multimedia information, as shown in Figure 1.

A hypermap is a database that has associated a graphical user interface. The hypermap database stores the information nodes as composite objects whose components are either maps or multimedia objects. The links that associate the information nodes in the database are also objects. The graphical user interface uses windows to display the active information node and buttons to display the links associated with the active node.

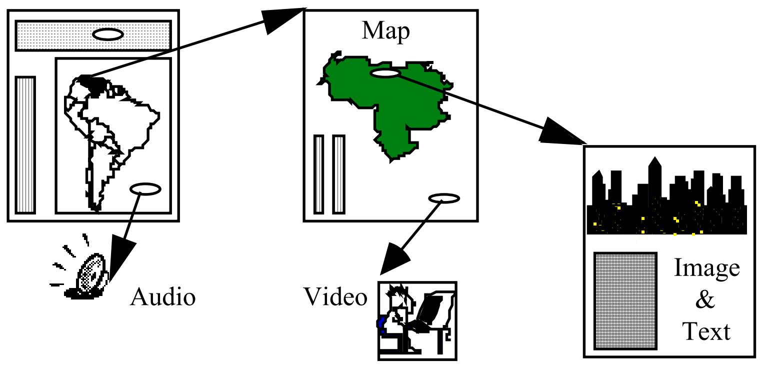

Figure 1. A hypermap example

3 The design of the hypermap data model

The hypermap data model was designed using an integration methodology. This methodology creates a specific-purpose data model based on the integration of two or more existing data models. A set of database requirements that justifies the creation of a new data model must be given as input to the method.

The integration method is composed by four phases: pre-integration, conceptual analysis, conceptual comparison and conceptual integration, as shown in Figure 2. The pre-integration phase is concerned with the selection of: (1) the data models to be integrated, called here *input models*; (2) the integration strategy (extension, transference and combination) to be used; and (3) the order of integration to be applied if more than two models are to be integrated. The conceptual analysis phase entails the identification and description of the features of each model being integrated. Its purpose is to gain an understanding of the properties of each model. For each participating model, a *core* -- a subset made of selected constructs, operations and rules of the model -- is defined based on the integration requirements. Each model core is then represented, or formally defined, using an appropriate modeling notation. The result of the representation process of each model core is referred here to as *core schema*. The conceptual comparison phase involves the identification of similarities and differences between the features of the model cores. The resulting list of similarities and differences is used in the next phase to help identify the points of integration between the cores. Finally, in the conceptual integration phase, the model cores are merged using the core schemata and according to the selected strategy. The integrated model is validated against the given set of integration requirements and refined by iterating the process until an appropriate solution is found.

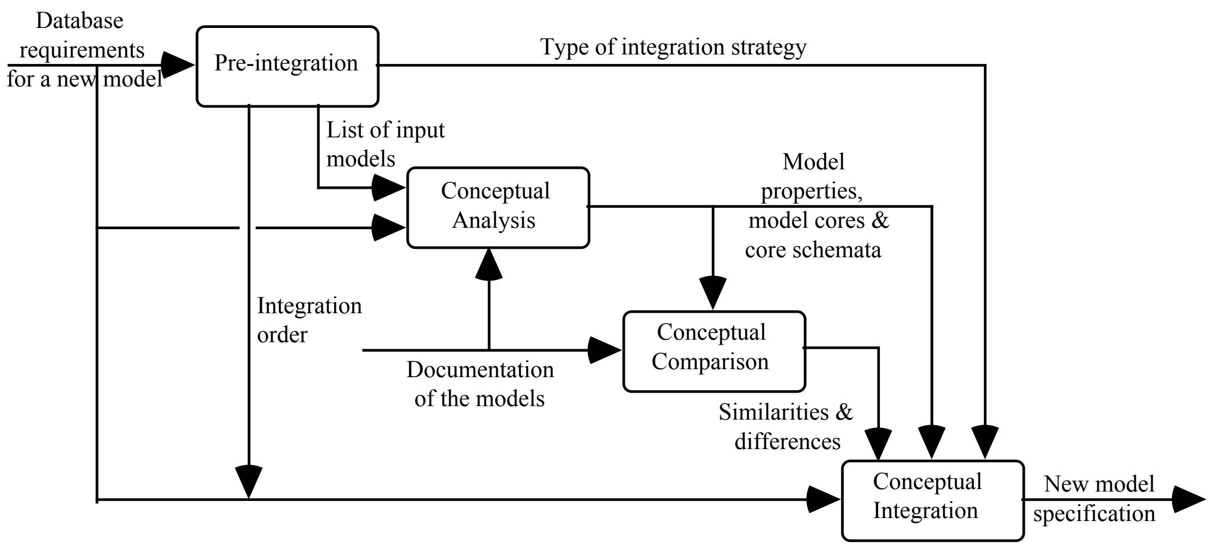

Figure 2. The Integration Methodology: Phases and information flow

For the design of the hypermap data model, two models were selected as input 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 representation of the same object, which is essential for modeling spatial databases. The HMR model, on the other hand, is a reference model. It is based mainly on the Lange´s hypermedia formal model [LAN90] and is extended with some features drawn from the Dexter Hypertext Reference Model [HAL94]. The data models of KMS [AKS88], Notecards [HAL88], and HyperCard also influenced the design of the HMR model. A description of each of these two models and the details of the application of the integration methodology to the hypermap model design is given in [MON95].

4 The hypermap data model

The hypermap data model is described here based on the three properties that characterize a data model: (1) the paradigm(s) used for its design and its associated view of the world (i.e., the way of looking at the world used by the designer and users of the model); (2) the concepts that it supports; and (3) its three components: constructs, operations, and modeling rules.

4.1 The paradigm and view of the world

The hypermap data model was designed based on the object-oriented paradigm. In this paradigm, the real world is modeled in terms of five constructs: *object, classes, attributes, methods* and *messages* [KIM90, GOL89]. The mapping between these constructs and the elements of the real world they represent are shown in Table 1. An ontological view of the world, proposed by M. Bunge [BUN77], is used here to define the meaning of the object-oriented constructs. According to this view, the world is composed of things. A thing may be an *entity* (concrete object) or a *concept* (*abstract object).* All things possess *properties*. Things that possess the same set of properties form a *kind*. The set of all values associated with the properties of a thing at a given time is called the *state of the thing*. A thing changes its state with time. A change of state, called *state transformation*, is caused by an *event*.

Element of the View	Object-oriented Construct
Entity or Concept	Object or Instance
Kind	Class
Property	Attribute
State of a thing	Values of the object's attributes
Action	Method
Event	Message

Table 1. The meaning of the object-oriented constructs

The hypermap data model is an object-oriented data model that, besides the mentioned object-oriented constructs, provides a set of specialized constructs (see Figure 3). A hypermap is an object-oriented database that represents information about an application domain in terms of objects, called *nodes of information*. A node describes the properties of an entity or concept of the application domain using items of information. An item is an object that contains multimedia information in the form of text, graphics, images, sounds, videos or maps. The relationships between the entities or concepts of the application domain are captured by links. A link is an object that relates two different nodes in an organizational, associative, spatial or active way.

4.2 The concepts supported by the model

As an object-oriented database, a hypermap must comply with the concepts that characterize the object-oriented (OO) paradigm. The hypermap data model supports different OO concepts such as object identity, abstract data types with encapsulation, classification, specialization, multiple inheritance, etc. (See Table 2).

Object identity is supported through surrogates, i.e., a unique system-defined identifier, which is assigned to each object during its creation. Each object has associated a unique class. A class defines the intension and extension of its instances in an integrated manner. That is, in addition to define the structure and behaviour of its instances (class intension), a class also groups all its instances as a collection (class extension).

Message passing is supported. Objects respond to messages. A message indicates the operation (method) that should be performed by the receiver object. A database operation is performed by sending a message directly to the corresponding class.

The structure of the database is modeled by a set of classes. Classes are associated by two types of relationships: superclass/subclasses and composition. The first type creates a hierarchy of classes in which each subclass inherits the attributes and methods defined by its superclass(es). In the composition relationship, a composed class is defined by means of the aggregation of other classes, called components. A composite object is an aggregate of references to other objects, called components.

Table 2 shows the concepts supported by the hypermap model. They are classified by the domain from which they were taken. The details of how these concepts are supported by the hypermap data model are given in [MON93]. A definition of each of these concepts can be found in the related literature (see, for example, [ZDO90, KIM90, LAU92, PAR89, MOL90]).

Object-oriented concepts	**GIS concepts**	**Multimedia concepts**
Surrogate-based object identity	Vector-based spatial data representation	Hypertext-Hypermedia
ADT & encapsulation: private and subclass visible	Multiple representation of geometrical properties	Organizational, associative spatial and active linking
Classification: intension integrated with extension	Topological relationships	Navigation by activation of buttons
Class specialization with multiple inheritance	Map generalization through different scales	Composition of units by organizational linking
Composition by attribute aggregation		
Persistence by reachability, orthogonal to instances		

Table 2. Concepts supported by the hypermap data model

4.3 The constructs of the model

Besides the general-purpose object oriented constructs -- object, class, attribute, method and message --, the hypermap data model has a set of specialized constructs for supporting the modeling and manipulation of hypermap databases. These constructs are implemented as classes.

A meta-model (a database schema of the model), expressed in terms of the Object Modeling Technique [RUM91], is presented in Figure 3. The structure and behaviour of selected classes of the model are shown in Figures 4-7.

According to the meta-model, a hypermap is a composite object made of spatial information units. It forms a network of multimedia information nodes related by instances of the class Link. Each information node is a composite object. It is composed by a set of instances of two classes: Item or Button, which are specialized into more specific classes. 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 multimedia information.

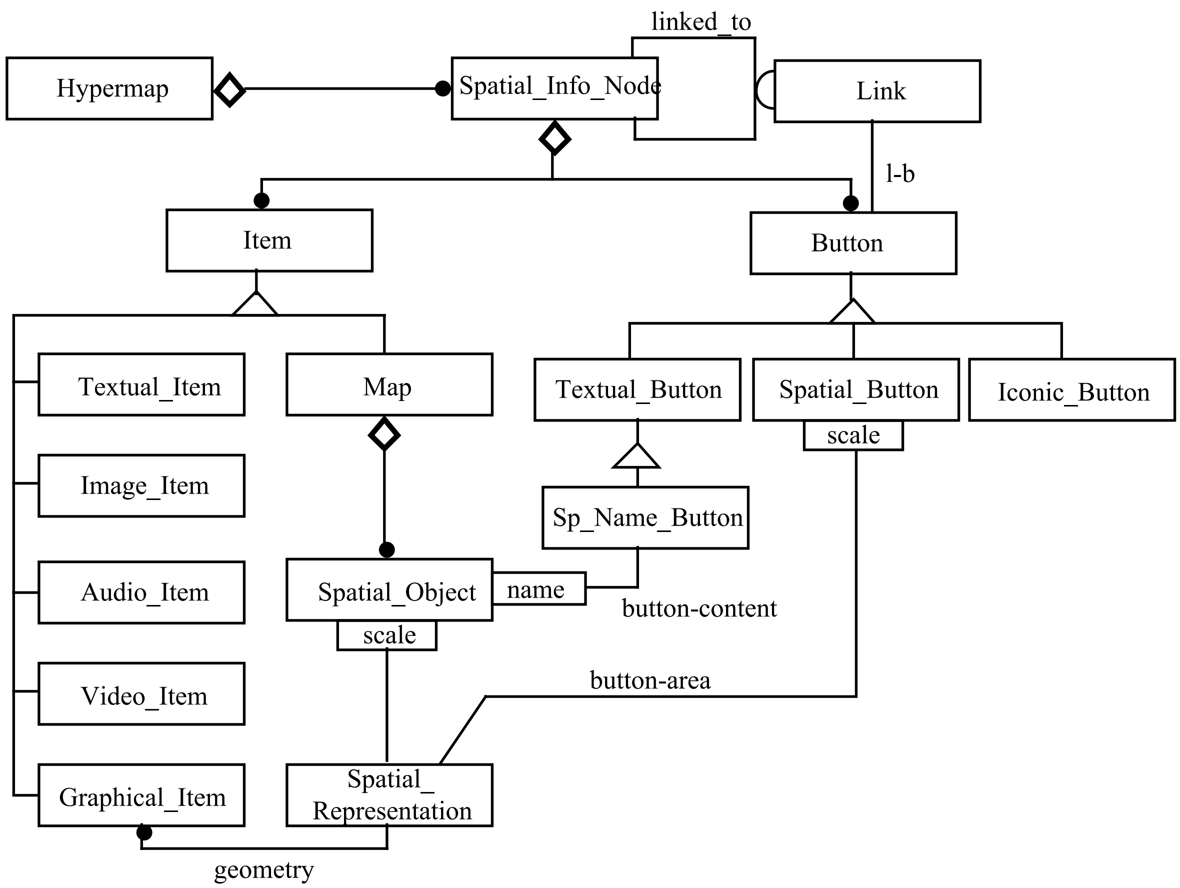

Figure 3. The schema of the hypermap data model

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 a name and 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.

Since the model supports only a vector representation of spatial data, each geometry of a spatial object is 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 emerges 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.

4.4 Operations of the model

The operations of the hypermap model are defined by its classes. Database operations, such as instance creation, object persistence, instance selection, and attribute projection, are provided by the root of the class hierarchy of the model and inherited by all its subclasses. This root, called OBJECT, is an abstract class that provides the behaviour needed to manage: (1) the persistence of the instances created by the user; and (2) the retrieval of instances from the database. The structure and behaviour of the root class are shown in Figure 4. The instance creation method *new* is the constructor of persistent objects. The methods *getAttribute* and *setAttribute* allows the user to retrieve or update, respectively, the value(s) of a given attribute of the instance that receives the message. The method *save* stores the state of the receiver into the database.

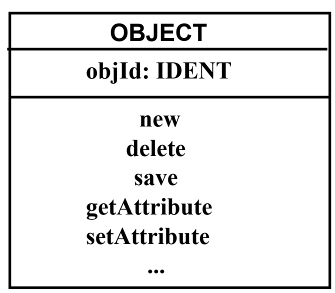

Figure 4. The class OBJECT

The attributes and methods of the classes Hypermap, Spatial_Info_Node and Link are shown in Figure 5. A hypermap database should be opened before being used and closed after its use. The method *displayNode* presents in a given window the current spatial information node. The *editNode* method changes the display mode of the node to the edit mode, which allows the user to update the content of the node (e.g., add, relocate, and delete its items and buttons).

The items and buttons that conform a spatial information node are organized into an area or information space. Each item or button has a position relative to the origin of this space. The activation of a button on the screen (*activateButton*) causes the button to activate the associated link, whose final effect is to display the destination node or perform the action indicated by the content of the destination anchor.

Figure 6 illustrates part of the attributes and methods of the classes Item, Button and Spatial_Button. *Item* is an abstract class that generalizes the common structure and behaviour of the different multimedia and map items. The bounding box of an item is the area required by its content when the item is displayed. Methods for display, edition, compression, decompression are redefined for each subclass of Item.

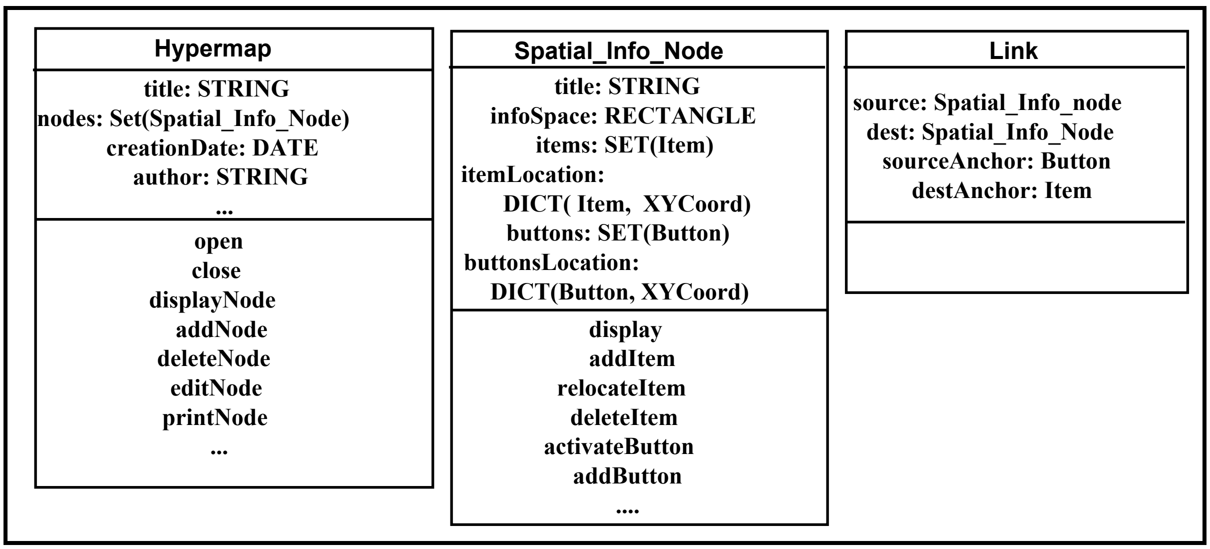

Figure 5. The classes Hypermap, Spatial_Info_Node and Link

Button is also an abstract class. The operations for navigating through the hypermap are provided by this class. 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 sounds, animation, and videos.

Spatial objects on a map may have associated two types of buttons: Spatial_Button and Sp_Name_Button. They are classes of buttons for linking a map to another map or activating a multimedia item, such as a sound, an animation or a video related to the map. A Sp_Name_Button is associated with the name of the object and behaves as a textual button. A Spatial_Button takes the shape of the spatial representation (geometry) of the object for a given scale. Clicking on any location of a spatial object on a map causes the activation of its spatial button.

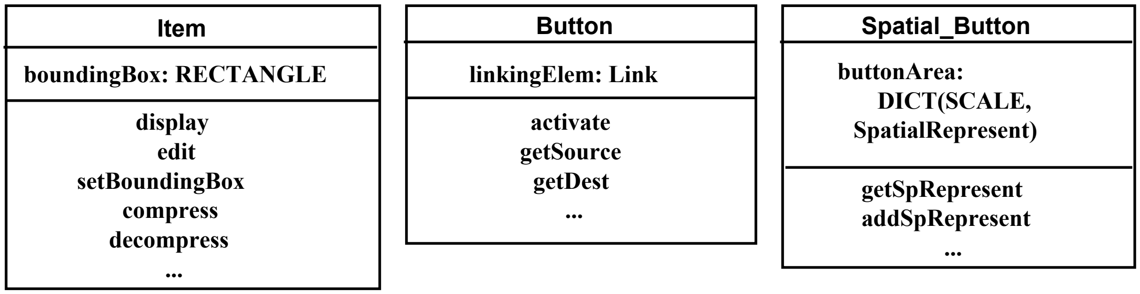

Figure 6. The classes Item, Button and Spatial Button

The classes for creating and manipulating maps in a hypermap are presented in Figure 7. The Map class, a subclass of Item, preserves all the spatial structure and behaviour of the map type defined by the OO-GIS data model [RGH91]. This feature is the major distinction between a hypermap and those hypermedia systems that display maps as images, but don't provide the mechanism for manipulating the objects on the maps. In contrast, a hypermap allows the user to manipulate interactively the spatial objects associated with each of its maps. The spatial analysis operations that characterize a GIS are provided by the Spatial_Object and Spatial_Representation classes. The user is therefore allowed to perform, on any map of a hypermap, operations such as intersection, overlay, buffering, calculation of distances, areas, and perimeters, graphical operations (e.g., rotation, translation) and scale transformation.

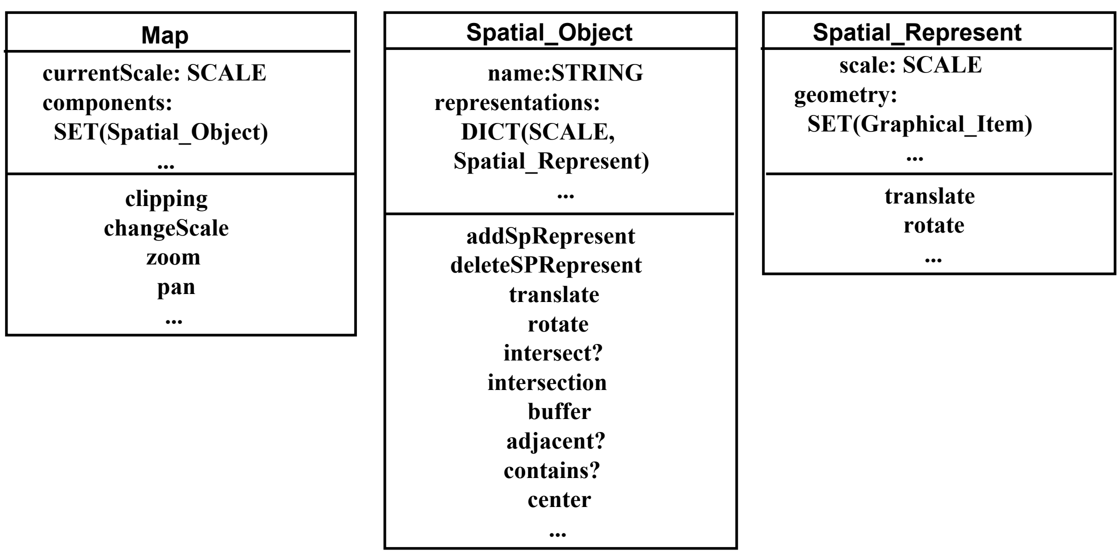

Figure 7. Classes for the creation and manipulation of maps

7 Conclusions

We have introduced, in this paper, a novel specific-purpose data model for the creation, manipulation, navigation, and access of hypermap databases. This model is the result of integrating spatial object-oriented databases and hypermedia databases at a conceptual level. A software integration method was used for designing the model. This method helped to deal with the complexity inherent to the integration of concepts, constructs, operations and rules of the two models that were used as input to the design process.

The notion of hypermap in GIS is very recent and the related literature is limited. Our definition of hypermaps is inspired on that given by Laurini, Milleret-Raffort and Thompson [LAU90, LAU92]. The main differences are that our definition: (1) uses an object-oriented approach; (2) incorporates sound, animation, and video; (3) 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 (4) is the result of applying a software integration method to extend the notion of hyper-document with digital maps.

The hypermap data 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 in C++ has been developed for implementing the classes of model in a XWindow platform. PEX and Xlib has been used for implementing the windowing and spatial analysis operations.

The model has three key benefits: (1) it provides the concepts, constructs, operations and rules needed to create, access, manipulate and maintain hypermaps, i.e., multimedia geographical databases; (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.

References

[AKS88] Akscyn, R.M., McCracken, D.L., and Yoder, E.A. KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations. *Comm. ACM*, Vol.31, No. 7, July, 1988, 820-835.

[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.

[BUN77] Bunge, M. *Treatise on Basic Philosophy, Vol. 3: Ontology I: The Furniture of the World*. Reidel, Boston, 1977.

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

[CON87] Conklin, J. Hypertext: An Introduction and Survey. *Computer*, Vol.20, No.9, September, 1987, pp.17-41.

[GOL89] Goldberg, A. and Robson, *A. Smalltalk-80, The Language*. Addison-Wesley, 1989.

[HAL88] Halasz, F.G. Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Comm. ACM*, Vol.31, No.7, July, 1988, 836-852.

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

[KIM90] Kim, W. *Introduction to Object-Oriented Databases*. The MIT Press, Massachussets, 1990.

[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.

[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.

[MOL90] Moline, J., Benigni, D., and Baronas, J. (Eds.) *Proceedings of the Hypertext Standarization Workshop*, National Institute of Standards and Technology, U.S Department of Commerce, January 16-18, 1990.

[MON92] Montilva, J.A. and Roberts, S.A. *A Methodologyological Framework for the Integration of Data, Information, and Knowledge Management*.. Research Report No. 92-20, University of Leeds, School of Computer Studies, Leeds, UK, September, 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.* Technical report. University of Los Andes. Faculty of Engineering. Data & Knowledge Engineering Research Group. Mérida, Venezuela, 1995.

[PAR89] Parsaye, K., Chignell, M. Khoshafian, S., and Wong, H. *Intelligent Databases: Object-Oriented, Deductive Hypermedia Technologies*, John Wiley and Sons, 1989.

[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.

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

[ZDO90] Zdonik, S.B. and Maier, D. (eds.) *Readings in Object-Oriented Database Systems*, Morgan Kaufmann, 1990.