A Logical Data Model for Integrated Geographical Databases

M. Gargano E.Nardelli

**Istituto di Analisi dei Sistemi ed Informatica, C.N.R.
V.le Manzoni 30, 00185
Rome, Italy (+39-6-770031)**

Abstract

In this paper we investigate an approach to the design and the development of geographical database systems based on the integration of different specialized databases. This approach is supported by an extension of the relational model which makes it possible to separate interaction issues from integration issues guaranteeing the independence between the logical level and the physical level. The introduced extended logical model guarantees to the user a uniform and expressive view of geographical entities, still allowing him to refer and access to them through both descriptive and geometrical/spatial characteristics. On the other hand, the logical model provides to the designer a formal methodology and a formal frame for the correct specification of the geographical information at the logical level. An experimental prototype of an architecture based on the proposed logical data model is under development in the framework of the research project "MULTIDATA", sponsored by the Italian National Research Council (CNR).

1.Introduction

The growing need of new applications in non traditional fields, such as for example office automation, interactive computer aided design, geographic data processing, health data processing etc., have stimulated the interest of many researchers to extend database capabilities to satisfy the requirements of such application fields. Such requirements and the problems arising from them are usually more complex than those characterizing traditional commercial applications. For example such domains often involve complex data that usually refer to real world entities to which geometrical properties are associated and among which topological relationships exist.

Geographical information systems are a significative example of the attempt of extending database capabilities to deal with complex data. Geographical information systems are in fact concerned with the representation and manipulation of both descriptive and spatial/geometric data. The former may for example refer to the textual and numeric information related to a geographical entity (e.g. the flow or the name of a river, the density of population of a given region) while the latter may for example specifiy the spatial/geometric characteristics (e.g. the shape and the location of the given region).

The available geographical information systems usually do not offer the number of advantages that database technology does. In this context, the main goal of database researchers is to put the foundations for the definition of a geographical information system that manages descriptive and spatial/geometric data in integrated way and that guarantees all the functionalities of the traditional DBMS's.

To pursue such an integration goal by exploiting the available technologies, a widely used approach to the realization of geographical databases rests on the coupling of traditional database management systems (DBMS) and specialized systems for geometrical data or image data processing. The rationale of such an approach relies both on the wide availability and reliability of the current DBMS's and the powerful capabilities of specialized systems for the spatial/geometrical data processing. In fact, since DBMS's have been dealing since early '70 with alphanumeric data and with typical requirements of commercial and business applications, they are now widely spread and able to process descriptive data in a very powerful and reliable manner. On the other side, geometric data and image data handling applications, with their strong requirements for manipulating huge quantities of spatial data and efficient algorithms for exploiting the geometrical and topological characteristics of such data, have lead during the last two decades to the development of specialized environments providing very powerful capabilities for efficiently processing geometrical and topological properties of spatial data.

The positive consequence of coupling different specialized technologies is that, with some little efforts, one is able to build a system managing geographical data, by simply setting up a communication channel between DBMS's (usually relational) and specialized systems for geometrical/spatial data. Furthermore, most of these systems guarentee on the average good performances, whenever they are designed for specific applications.

On the other hand the lack of a really integrated DBMS leads to several drawbacks, both under the design and the interaction point of view. Section 2 discuss this topic in detail and points out that such drawbacks are principally due to the fact that the integration directly relies on the physical implementation, without a reference to an integrated logical model. As a consequence, a way to overcome these drawbacks is to define logical design methodologies and a logical frame for the integrated modeling of geographical information.There is therefore need of a suitable logical model, but the nowadays available data models are not capable to accomplish this aim.

It is in fact well known that, though the relational model defined by Codd [Cod70] constitues a powerful tool for logical modeling of traditional applications thanks to its validated theoretical basis, so as it stands it does not support an effective representation and an efficient manipulation of geographical information [RFS88, O&M88, S&V89]. On the other side, the newest approaches to the modeling of geographical data, based on the object oriented paradigm [LRV88, M&K88], has been until now considered not enough strong for the logical modeling of data owing to the lack of a well-defined underlying theoretical frame [Ban88]. In fact, even though such approaches seem very promising, only recently formal definitions and axiomatizations have been proposed [A&K89, B&K89]. Thus, though a complete formalization could produce a well-defined model (as the relational one) for the object oriented paradigm, it is still a highly debated research topic.

We believe that the relational model capabilities, with respect to the framework of geographical applications, have still to be fully exploited. Thus, in this paper we define (i) a suitable extension of the relational model that provides expressive and powerful primitives for integrated modeling of geographical data, (ii) introduce an extended algebra providing operators which directly model the kinds of manipulation of geographical data performed by the end-user andthat furnishes a basis for the formal definition of high level data manipulation and definition languages. From the designer point of view, the model offers a frame and a formal methodology for the correct specification of the geographical information at the logical level. From the user point of view it guarantees an expressive and uniform view of the database content. Finally, the model supports the definition of high level data manipulation languages and of file structures that efficiently represent spatial/geometric information.

The rest of the paper is organized as follows. In section 2 specific characteristics of the architecture of geographical information systems realized coupling different subsystems are discussed, and the relevant motivations of the integrated approach proposed in this paper are pointed out. Section 3 defines a non traditional domain for modeling the geometrical component of the objects and introduces the extended relational model. The extended relational algebra and the relative operators are defined in section 4, where many examples are also presented. Section 5 contains the concluding remarks. Finally, the extended relational algebra is formally defined in the Appendix.

2.Architectures of geographical information systems.

In this section we discuss some specific characteristics of current geographical information systems based on the coupling of traditional DBMS's and spatial/geometric data handling systems. In the discussion their principal drawbacks are underlined and the relevant motivations of the integrated approach proposed in this paper are pointed out.

1) **Physical separation** A strong separation exists between alphanumeric database for the management of descriptive information and spatial/geometric data handling system. By separately treating the alphanumeric and spatial/geometrical information is possible to exploit validated and well sperimented DBMS for the management of alphanumerical data, and ad hoc file structures, as for example the quad trees [Sam84], for the representation of spatial/geometric data. A drawback is that the physical separation of the two databases leads to a fragmentated representation of geographical entities. In fact, as such systems are lacking of a logical model that represents descriptive and spatial/geometric information in integrated way, spatial/geometric information is generally codified in order to be represented, at the logical level, in the alphanumerical DBMS (usually relational) [C&F80], [Gro84]. The user has two different views: the descriptive one, that refers the object by its description, and the geometric one, that refers the object by its shape and location. Such views are correlated, but the user does not perceive the integrity of objects he manipulates. Under the database design point of view this is a drawback too, because geographical database designers and application programmers can not rely on high-level data modeling and manipulation capabilities.

2) **The language** The interaction between the user and the system is essentially accomplished by means of user friendly interfaces. The query language is generally developed to guarantee a set of basic funtionalities depending on the requirements of the specific application. As Fig. (a) points out, the lack of a logical model makes the query language partially resting on the physical level and this fact leads to two principal drawbacks. First, the interaction between the end-user and the system often implies the knowledge of implemention details that an end-user should not aware of. Second, the overall system maintenance often needs of a direct manipulation of the implemented code, for example by rewriting programs in low level languages. As far the integration matter is concerned, by means of carefully developed interfaces, the language allows the user to friendly refer the objects either by spatial or descriptive characteristics. In such way a sort of integrated management of geographical objects is simulated. But in reality, the integration is realized at the physical level by means of pointers directly connecting the two databases. These ad hoc solutions often lead to inefficiency because they are lacking of generality. In fact, access paths are generally designed on the basis of the requirements of the specific application. It follows that the efficient management of mixed queries (i.e. queries involving both descriptive and spatial/geometrical properties of the objects) is obtained for the queries foreseen by the designer. In reality, even for restrict and well specified geographical applications the set of general queries not foreseen by the designer can be very large and the inefficiency can become a crucial matter when such mixed queries have to be answered. Owing to their complexity is in fact more difficult to foresee all the parameters that characterize such kind of queries. Though apparently simple queries as "returns the name of the rivers intersecting the square region I draw on the screen" can be formulated designing a suitable language, current geographical information systems can answer it, whenever it can, often after a very time and space consuming query and data processing. In fact anserwing mixed queries requires a strong interaction between the two subsystems in order to extract the data and information relavant for answering the query. Furthermore, the needed information, also in non mixed queries, often have to be dynamically obtained, and not simply extracted, from the databases. This is a reasonable thing if we consider that it should be widely space consuming to explicitely represent in the database the topological relationships existing among the objects.

3) **Data Independence** As already mentioned in the first point, physical independence of the data is difficult to guarantee because it doesn't exist a well defined frame for the logical representation of the objects. This is a direct consequence of an architecture that proposes physical connection as main vehicle to map the user view into the physical implementation. Furthermore, because the definition of a query language naturally turns out from the user view of the real world, it is straightforth depending of the particular application. As a consequence, the lack of a logical model that allows independence between query language and physical implementation results in a little flexible overall system.

On purpouse to overcome the drawbacks, the more significant of which we have just outlined, of the current geographical information systems, a great deal of interest is recently developed for the realization of more sophisticated architectures (Fig. b) that

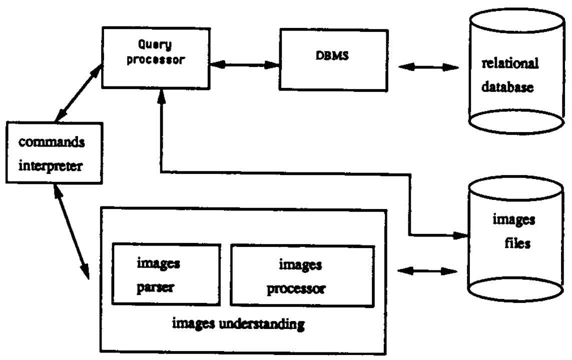

Fig. a
A system architecture of a typical geographical DBMS's

are based on the integration, rather then the coopling, of distinct systems technologies [MUL89], [RFS88], [ASG86].

A step towards this aim can be done through the definition of a suitable logical model. The introduction of an integrated logical data model in fact allows to overcome such problems since it both defines a reference schema for the specification of the integration between the subsystems and provides the user with a high level data manipulation language. The model therefore guarantees to the user a uniform view of geographical entities, still allowing him to refer to them through both the descriptive and the geometrical/spatial aspects. Neverthless, the design of file structures, that efficiently (i) represent geographical information and (ii) support the resolution of geometrical and mixed queries, is a crucial problem above all in geographical applications dealing with very intensive image data processing, and a largely open research area.

Since the query language, by means of the logical model, doesn't rest on the physical level but on the logical model, the independence between integrated physical system and query language can be guaranteed and the overall system as a consequence results more flexible both during the designing and the interactive management phase. There is therefore need of a suitable logical model that

1) guarantees to the user a uniform and expressive view of geographical entities, still allowing him to refer to them through both descriptive and geometrical/spatial characteristics.
2) supports the definition of high level data manipulation language and access paths and file structures for the efficient answering of mixed queries.
3) provides to the designer a formal methodology for the correct specification of the geographical information at the logical level.

In the next section we introduce the logical model we propose for the representation of geographical information. It satisfies the above requirements and so it represents a first step towards the defintion of integrated geographical DBMS's.

3. An extended relational model to represent geographical entities in integrated way.

In according to [GNT88], [S&V89] we can consider a geographical object as an entity of the realty of interest consisting of two components: a descriptive one and a geometrical one. The former provides descriptive characteristics and properties of the object, the latter its spatial extension. For example, the descriptive component of an entity "River" can be a collection of

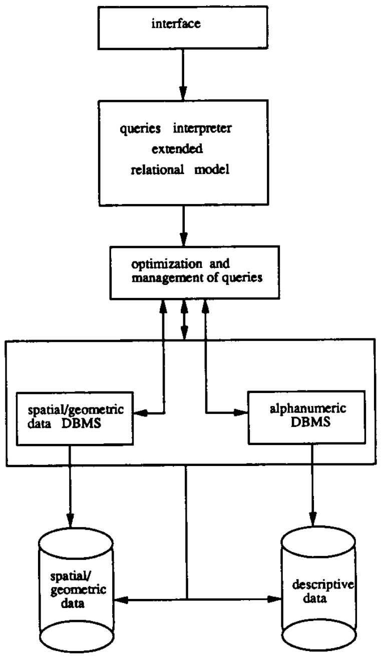
interface
queries interpreter extended relational model
optimization and management of queries
spatial/geometric data DBMS
alphanumeric DBMS
spatial/ geometric data
descriptive data

Fig. b
An overall proposal of system architecture

attributes as its name, flow etc., while the geometrical one can be its shape and location. From now on we refer to a geographical object simply as an object.

From the point of view of data modeling, the choice of a relational model leads to representing the logical structure of data in tables. In such a context, a descriptive attribute of a geographical entity (e.g. the flow of a river) can correspond in a direct way to a single valued attribute whose underlying domain is traditional (e.g. alphanumeric). On the other side, attributes defined on traditional atomic domains can not adequately represent the geometrical component of such objects. In fact, using traditional domains naturally leads to represent an object by distributing its shape and location on a relation as in [C&F80],[Gro84] so obtaining a fragmented logical data representation, far from the way the user looks at the reality of interest. In order that the logical structure of data reflects the way the user looks at data themselves, the logical model must therefore be capable of supporting non atomic domains for representing the geometrical component of an object.

In the next subsection we define a non atomic domain, SHAPES. It results in an algebrical structure with respect to the operations in it defined (SHAPES is closed under such operations). Since a single element of SHAPES univocally can define the shape and location of an object, the geometrical extension of the object itself can be fully represented at the logical level by a single valued attribute on such domain. The logical view of the database can thus straightforth support the user view of the reality of interest, becouse the logical model esplicitly takes into account that the user looks at the gemetrical component as an atomic concept. In subsection 3.2 the extended relational model, ERM, is defined and the way objects can be represented in ERM is shown.

3.1 Definition of the SHAPES algebra.

In defining the domain for the representation of the geometrical component of objects and the relative operations (the SHAPES algebra), we implicitly refer to representation schema [Gun88] based on a discrete decomposition of the space [Gun88] from which the objects themselves are drawn. We focus on the case in which the geometrical extension of an object is described by sets of points belonging to a raster decomposition of the plane (or raster plane). This choice provides a more direct understanding of the proposed model in this paper and it is not a restrictive choice because a straightforward generalization [GNT89] makes the model introduced in this paragraph applicable to representation schema based on a general classe of space discrete decomposition [GNT89].

Let us now formally define what we mean by raster plane. Let $\mathbb{I}$ denote the set of integers.

Def.1: Given two closed intervals I_1, I_2 of $\mathbb{I}$, we call raster plane, denoted by **R**, the cartesian product of I_1, I_2, i.e. $\mathbf{R} = I_1 \times I_2$.

Notation. Given a non empty finite set S, H_S denotes the set $H_S = P(P(S))$ where P(S) denotes the powerset of S (hence H_S is the set of all sets of sets of elements in S).The symbols $\cup$ and $\cap$ denote, respectively, the union and intersection operations in the way they are used in set theory. Finally, note that H_R denotes the P(P(**R**)) set of the raster plane, **R**, above introduced.

Def.2: Given a raster plane **R**, we call SHAPES algebra on **R** the 5-tuple $\mathbf{H_R}=(H_R, \cup, \cap, \cap^*, \text{geo})$, and we denote it symply by $\mathbf{H_R}$, where $\cap^*$ and geo are the two operators defined in H_R, for all A, B $\in H_R$, as follows:

$$A \cap^* B = \{ c \in P(R) : (\exists a)(\exists b)\ a \in A \wedge b \in B \wedge c = a \cap b \}.$$

$$\text{geo}(A) = \{ X \}, \text{ where } X = \{ y \in R : (\exists a)\ a \in A \wedge y \in a \}.$$

An element of $\mathbf{H_S}$ is called a **Shape**. If a given **Shape** contains exactly one element of P(S) it is called a simple **Shape**. If a given **Shape**, A, contains more than one element, each element is called an A-component. The geometrical extension of an object can be represented by means of a **Shape** or a simple **Shape**. Figures (1a),..,(1e), explain the operators and informally show how the geometrical extension of the objects can correspond to any non connected subset of the plane and can be represented by means of a **Shape** or a simple **Shape** of the SHAPE algebra. Unformally, the $\cap^*$ operator, when it is applied to two **Shape**'s A and B, returns a **Shape** whose components contain the points belonging to the intersections of the A components with the B components; the **geo** operator applied to a **Shape** A containing more than one component returns a **Shape** containing only one component (i.e. a simple **Shape**) obtained as the union of the points of every A-component. If it is applied to a single **Shape**, A, it returns the **Shape** A itself.

3.2 The extended relational model

In this section the extended relational model, ERM, is defined and the way objects can be represented in ERM is shown. The model allows the integrated representation of geographical entities by the support of SHAPES valued attributes (i.e. attributes whose value is an element of SHAPES).

Let **U** be a given set of names and let **T** be a finite set of type symbols $\mathbf{T=\{T_1,\ldots,T_n,\ G\}}$. The elements of **U** are called "attributes".

Let $\mathbf{H_R}$ be the SHAPES algebra on **R** introduced in the previous section. To any A$\in$U a domain of values of type t$\in$**T**, called the domain of A, can be associated and it is denoted by **dom**(A). Given A$\in$U, **type**(A) returns the symbol type of the elements of **dom**(A). For all A$\in$U, **type**(A)=G if and only if **dom**(A)=$\mathbf{H_R}$.

A relation schema $\mathbf{R_n = X_1,\ldots,X_n}$ (where n is the degree of the relation schema R) is a non empty subset of **U**. If $\mathbf{type(X_i)=T_i}$, $\mathbf{X_i}$ is called descriptive attribute of **R**; if

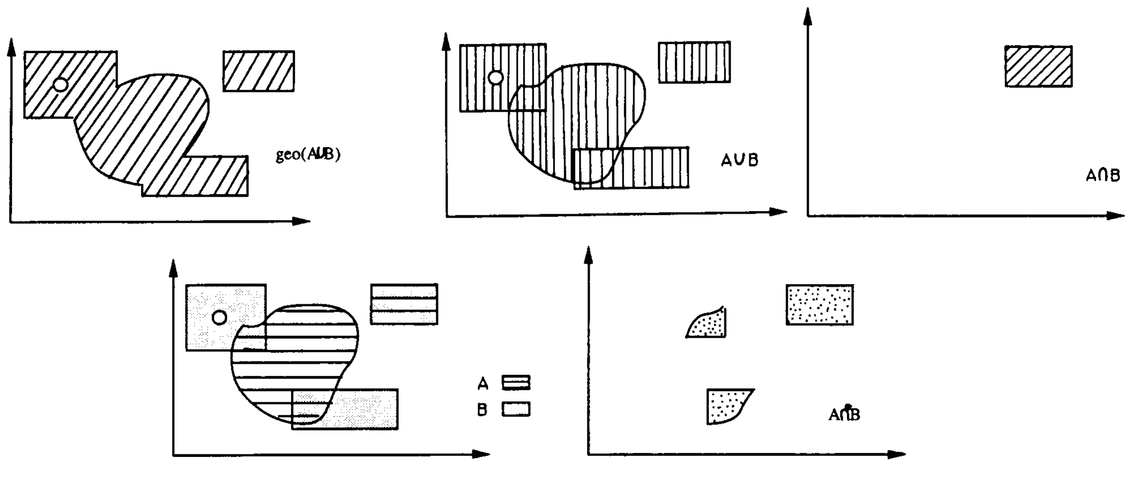

Fig. 1a,..,1e
Informal explanation of the SHAPE algebra operators

$type(X_i)=G$, X_i is called geometric attribute of **R**. We briefly write $\mathbf{R}(A_1,\dots,A_n)$ to denote a relation schema $\mathbf{R}_n$ with attributes $A_1,\dots,A_n$ or simply **R** when confusion is not generated. An instance **r** of a relation schema R is a set of n-tuples each having a component for each attribute of R. Given a relation schema R, an instance of R is also denoted by R(I). Let r be an instance of R and let X∈R be a descriptive attribute. We denote with $\mathbf{f_X}$ any function returning a simple value obtained from the X attribute of the set of tuples of r and we will refer to it as an "aggregate function".

As an example of the above introduced definitions, let us consider an example of a geographical survey database with relation schemes, where each underlined attribute is a candidated key for the relation:

RIVER SEGMENT(NAME, CROSS.REGION, LENGHT, <u>SEGMENT</u>)
LAKES(<u>NAME</u>, CROSS.REGION, AREA, <u>LAKE SHAPE</u>)
REGIONS(<u>NAME</u>, AREA, DENSITY OF POP., <u>REGION SHAPE</u>)

We focus on the relation RIVER SEGMENT. Analogous considerations hold for the relation shemas LAKES and REGIONS. We have

type(name)=string	dom(name)={Strings}
type(C.Reg.)=character	dom(C.Reg.)={Char}
type(Lenght)=real	dom(Lenght)=$\mathbb{R}$
type(Segment)=G	dom(Segment)=P(P($\mathbb{I}^2$))

RIVER SEGMENTS			
Name	C. Reg.	Length	Segment
river 1	A	2	{ {riv_{11}} }
river 1	B	4	{ {riv_{12}} }
river 2	B	2.5	{ {riv_2} }
river 3	A	4.7	{ {riv_{31}} }
river 3	H	5	{ {riv_{32}} }
river 3	B	6	{ {riv_{33}} }
river 4	C	1	{ {riv_4} }
river 5	B	10	{ {riv_5} }

Fig. 3
(Relation R_1)

A tuple of R_1 represents information about rivers. For each river, the crossing region, the segment individuated by its intersection with the crossing region and the length of such a segment are given. Note that a geometrical extension of a river segment is a set of pairs of integers (set of points of the raster decomposition). In the sake of concision, we have denoted it by "riv_{ij}" indicating by this espression the jth segment of the river "riv i". The crossing region is represented and accessed by its name. If we had interest in, we could choose to represent the crossing region by its shape as the relation R shows (here obviously is type(Cross.region)=G and an element of C.Reg. has been denoted by "regX" anaugously to "riv_{ij}"). This example outlines that there are no restrictions on the number of geometric attributes in a relation schema. Properly, any relation is permitted to be interpretated only on the basis of the functional constraints expressed in the database, as in the traditional relational approach. It follows that the relation schema structure only depends on database design criteria.

RIVER SEGMENTS			
Name	C. Reg.	Length	Segment
river 1	{ {regA} }	2	{ {riv_{11}} }
river 1	{ {regB} }	4	{ {riv_{12}} }
river 2	{ {regB} }	2.5	{ {riv_2} }
river 3	{ {regA} }	4.7	{ {riv_{31}} }
river 3	{ {regH} }	5	{ {riv_{32}} }
river 3	{ {regB} }	6	{ {riv_{33}} }
river 4	{ {regC} }	1	{ {riv_4} }
river 5	{ {regB} }	10	{ {riv_5} }

Fig. 4
(Relation R)

4. An informal approach to the extended relational algebra, ERA.

We extend the relational algebra essentially by defining three new algebric operators that are capable to directly manipulate SHAPES valued attributes. Due to the underlying formal frame introduced by SHAPES algebra, the usual relational operators of the relational algebra are straightforth extended to deal with SHAPES valued attributes as it can be seen in the Appendix. In this section we informally describe, through examples on relational tables, only the main geometrical operators G-Compose, G-Decompose and G-Fusion (where G stands for geometrical). The formal definition of ERA can be found in the Appendix.

(1) **G-Compose.** This operation groups the values of a specified geometrical attribute on the basis of the equality of an other specified attribute, either geometrical or descriptive. Referring to relation R_1, let us suppose that we want to group the river segments of those rivers that cross a same region. The following expression of ERA can be issued:

$$\text{G-Compose}_{\text{C.-Reg.}}(\text{Segment})(R_1)$$

Rivers classified by region	
C. Reg.	Segment
A	{ {riv_{11}},{riv_{31}} }
B	{ {riv_{12}},{riv_2},{riv_5} ,{riv_{33}} }
C	{ {riv_4} }
H	{ {riv_{32}} }

Fig. 5
(Relation R_2, an example of G-Compose operator)

A value of the geometric attribute (Segment) in R_2 is thus the set of the river segments belonging to the same region.

(2) **G-Decompose.** This operation ungroups a specified geometrical attribute with respect to another specified attribute, either geometrical or descriptive. Fig. 5.a shows this operator when the following ERA expression

$$\text{G-Decompose}_{\text{Cr.Reg.}}(\text{Segment})(R_2)$$

is issued. In this relation, the value of the geometric attribute is the simple **Shape** of a river segment that crosses a given region. Note that the relation R_2 is been somewhat normalized by the G-Decompose operator.

River segments and belonging region	
C.Reg.	Segments
A	{ {riv_{11}} }
A	{ {riv_{31}} }
B	{ {riv_{12}} }
B	{ {riv2} }
B	{ {riv5} }
B	{ {riv_{33}} }
C	{ {riv_4} }
H	{ {riv_3} }

Fig. 5.a
(Relation R_{2a}, an example of G-decompose operator)

But note also that the value of the geometric attribute is always a single element of SHAPES. Roughly we can say that this operator decomposes each **Shape** in its components (remember that an element of a **Shape** is called component).

The G-compose operator has been introduced to support "temporal" (reversible) aggregation, from which it is possible to go back by means of the application of the G-Decompose operator. From the user point of view, it is useful to aggregate a group of objects sharing a same property (e.g. the same kind of coltivation in relation R_5) that can be manipulated as a whole (e.g. identified by pointing only one of them on the screen), but disaggregated after manipulations have been performed and desired results obtained.

(3) **G-Fusion.** This operation "fuses" the values of a specified geometrical attribute (e.g. Segments in relation R_1) on the basis of the equality of an other specified attribute, either geometrical or descriptive (e.g. Name in relation R_1). An aggregate function (e.g. +) is applied to a third attribute (e.g. length in R_1). Relation R_4 in Fig. 6 is obtained by applying

$$\text{G-Fusion}_{\text{Name}}(\text{Segments})(+,\text{Length})$$

to R_1 of Fig.1 (here "+" stands for the addition operation).

Rivers		
Name	Tot. Length	River Shape
river 1	6	{ {riv_{11},riv_{12}} }
river 2	2.5	{ {riv_2} }
river 3	15.7	{ {riv_{31},riv_{32},riv_{33}} }
river 4	1	{ {riv_4} }
river 5	10	{ {riv_5} }

Fig. 6
(Relation R_4, an example of G-Fusion operation)

Note that a value of the geometrical attribute is a simple **Shape**; it follows that the resulting relation can not be disaggregated in the original components.

Example - An application: thematic maps.

In this example we show how our model supports the representation and the manipulation of thematic maps. We also want to underline that the model can represent maps by means of relations in which every tuple corresponds to a region defined by a set of descriptive attributes and by one geometric attribute as in [Gut88] and [S&V89]. In addition to that, as there are no restrictions on the number of geometric attributes in a relation schema, a relation with two geometric attributes G_1, G_2, does not require to be interpreted as a relation with only a single geometric attribute (the result of the intersection of the two G_1, G_2) as in [S&V89].

Consider a new relation R_5 in Fig. 7, representing coltivation lands. Here "ch." stands for chemicals, "colt." for coltivation and "Q" for quantity.

Coltivation lands				
coltivation	ch.	Q.colt.	Q.ch.	Area
corn	A	20000	13	{ {A1} }
potatoes	B	10050	17	{ {A2} }
tobacco	R	14500	23	{ {A3} }
corn	C	37850	30	{ {A4} }
potatoes	B	15070	20	{ {A5} }

Fig. 7
(Relation R_5)

Applying $\text{G-Fusion}_{\text{Coltivation}}(\text{Area})(+,\text{Q. colt.})$ to R_5, the relation R_6 in Fig 7.a is obtained.

Coltivation map		
coltivation	Q.colt.	Area
corn	57850	{ {A1,A4} }
potatoes	25120	{ {A2,A5} }
tobacco	14500	{ {A3} }

Fig. 7.a
(Relation R_6)

Relation R_7 in Fig 7.b is obtained by applying $\text{G-Fusion}_{\text{Coltivation}}(\text{Area})(+,\text{Q. chem.})$ to R_5.

Chemical map		
chemical	Q.chem	Area
A	13	{ {A1} }
B	37	{ {A2,A5} }
R	23	{ {A3} }
C	30	{ {A4} }

Fig. 7.b
(Relation R_7)

It is important to note that the aggregate function provides new values of Quantity of Coltivations and Quantity of Chemicals that correspond to the new areas obtained by the Fusion of the areas having the same kind of coltivation in the relation R_5. As we have already noted, the new values for the geometric attribute obtained as a result of the application of the G-Fusion cannot be disaggregated in the original components as they are simple Shapes (section 3.1). In this sense, the G-Fusion operator is not reversible. The thematic maps or "views" on the database thus created present a sort of "permanent" aggregation of geometric attribute values.

The examples of this section have only the purpose of explaining the semantics of the introduced operators. The applicabilty of ERA as a data manipulation and querying language cannot be extensively shown in this paper. However it is easy to realize that the ERA here defined can constitute the basis of high level geographical database languages. In fact, general geographical queries (i.e. queries involving predicates on both geometric and descriptive properties of geographical data) can be formulated by means of a suitable expression of ERA. As an example, the query: "Return the name of all the rivers whose segments intersect tobacco coltivation lands." can be formulated by means of the following expression of ERA:

$$\text{Project}_{\text{Name}}(\text{Select}_{\text{coltivation='tobacco'}}(\text{Join}_{\text{Segment}\cap^*\text{Area}}(R_1,R_5)))$$

5. Conclusion and further research.

In this paper we have presented an extended relational model, ERM, and an extended relational algebra, ERA, that together allow the representation and manipulation of geographical data in the context of an integrated geographical database.

ERA forms the basis for the definition of high level data manipulation and definition languages. Furthermore, it represents a high level procedural language that makes it possible to deal with geometrical component of geographical objects as if it was an atomic element. In such a way, the access and manipulation of non zero size objects do not imply any knowledge of low level information depending on physical implementation, as it happens in well known approaches based on extensions of the relational model [C&F80, C&F81]. The overall model here presented provides an effective and uniform representation of the gegraphical information at the user level. Besides, it introduces a formal frame that leads towards the definition of a formal methodology for the correct specification of the geographical information at the logical level.

We believe that in such way a step towards the realization of integrated geographical databases have been done. In addition, we are working to the definition of more general frameworks for dealing with pictorial information. In this context, it is our opinion that a logical model that supports different logical views and different physical representation schema have to be defined. In such a way one can obtain both a direct correspondence between logical definition and physical implementation and a full independence between the logical and the physical level. At this regard recall that the logical model ERM relies on the definition of a non atomic domain SHAPES. Its definition rests on discrete decompositions of the plane but it can easily extended to support a very general class of data representation schema. It turns out, in fact, that a straigthforward extension [GNT89] of the logical model here introduced take advantage of a large class of data structure that efficiently support (i) the implementation of the operators defined at the logical level [A&D89] (ii) and the representation of the geometrical component of complex objects. That states that the applicability of our model is very general and encourages us to purse the integration issue on the basis of our approach.

Acknowledgements: we are indebted with Maurizio Talamo for his scientific contribution and his encouragement during this research. We thank Giorgio Gambosi and Domenico Sacca' for their helpful comments.

Appendix

In this section, the set **E** of ERA expressions is formally defined. Here, advantage is taken of the formal framework introduced in section 3.1 in defining the primitive operations of the extended relational algebra for the manipulation of geographical entities. Borrowing notation from [OOM87], let us consider a database schema $\mathbb{R}=(R_1,..,R_N)$, where R_i is a relation schema. Let $Attr(\mathbb{R})$ the set of all the attribute names in $\mathbb{R}$.

(1) **Literals.** For any $c\in \bigcup_{A\in U} dom(A)$, $\{c\}\in\mathbf{E}$, has degree 1 and $\{c\}(I)=\{c\}$.

(2) **Relations.** For each R_j in $\mathbb{R}$, $R_j \in \mathbf{E}$ and $R_j(I) = r_j$.

(3) **Projection.** Let $e_n \in \mathbf{E}$, X be a subset of the attributes of e_n, then $e_n[X] \in \mathbf{E}$ its degree being equal to length(X), where length(X) is a function that returns the number of elements in X, and:
$e_n[X]\ (I) = \{t[X] \mid t\in e_n(I)\}$.

(4) **Cross Product.** Let a_n, $b_m \in \mathbf{E}$. Then $(a_n x b_m)\in \mathbf{E}$, its degree is n+m and (here $\circ$ denotes concatenation)
$(a_n x b_m)(I) = \{t_1\circ t_2 \mid t_1\in a_n(I) \wedge t_2\in b_m(I)\}$.

(5) **Restriction.** Let $e_n \in \mathbf{E}$, X and Y two attributes of e_n, where X and Y are both descriptive or both geometric. If $\Theta_d =\{<, =\}$ and $\Theta_g = \{\equiv, \subset, \cap, \cap^*\}$ then $e_n[X\theta Y]\in\mathbf{E}$, where $\theta\in\Theta_d$ (if X and Y are descriptive attributes) or $\theta\in\Theta_g$ (otherwise).

(6) **Union, difference.** Let a_n, $b_n \in \mathbf{E}$. Then $a_n \cup b_n$ and $a_n - b_n$ are in **E** and both have degree n.
$(a_n \cup b_n)\ (I) = \{ t \mid t\in a_n(I) \vee t\in b_n(I)\}$
$(a_n - b_n)\ (I) = \{t \mid t\in a_n(I) \wedge t\in b_n(I)\}$.

(7) **G-Compose.** Let $R_n\in \mathbf{E}$, X a non empty subset of descriptive attributes of R_n and G a geometric attribute of R_n. Then, $\text{G-Compose}_x(G) \in \mathbf{E}$ and has degree equal to length(X)+length(G). Semantically, for each tuple $p\in R_n[X](I)$, let v_p be the tuple defined as follows:

$v_p[X] = p$
$v_p[G] = \bigcup_{t\in R(I) \wedge t[X]=p} t[G]$.

Then,
$\text{G-Compose}_x(G)(R)(I) = \{ v_p \mid p\in R_n[X](I) \}$

(8) **G-Decompose.** Let $R_n\in \mathbf{E}$, X a non empty subset of the descriptive attributes of R_n, G a geometric attribute of R_n. Then, $\text{G-Decompose}_x(G) \in \mathbf{E}$ and has degree length(X)+1. Semantically, for each tuple $t\in R_n(I)$, let U_t be the set defined as follows

$U_t=\{ b \mid (\exists y)(y\in t[G]) \wedge b[G]=\{y\} \wedge b[X] =t[X]\}$

Then

$$\text{G-Decompose}_x(G)(R)(I) = \cup_{t \in R(I)} U_t$$

(9) **G-Fusion.** Let $R_n \in E$, X and Y descriptive attributes of R_n, $Y \neq X$, G a geometric attribute of R_n and f_y an aggregate function. Then, $\text{Fusion}_x(G)(f_y,Y)R_n \in E$ and has degree equal to length(X)+length(Y)+length(G). Semantically, for each tuple $p \in R_n[X](I)$, let v_p be the tuple defined as follows:

$v_p[X] = p$
$v_p[G] = \text{geo}(\cup_{t \in Rn(I) \wedge t[X]=p} t[X])$
$v_p[Y] = f_y\{t[Y] \mid t \in R_n(I) \wedge t[X]=p\}$.

Then,

$$\text{G-Fusion}_x(G)(f_y,Y)(R_n)(I) = \{ v_p \mid p \in R_n[X](I) \}$$

References

[A&D89] Arcieri F. and Dell'Olmo P., "A Data Structure for the Efficient Treatment of Topological Join " Fourth International Symposium on Computer and Information Sciences, Turkey (1989) .

[A&K89] Abiteboul S., Kanellakis C., "Object Identity as a Query Language Primitive" Proc. of the 1989 ACM-SIGMOD, SIGMOD RECORD Vol.18, N. 2, (June 1989).

[ASG89] Abiteboul S., Shool M., Gardarin G., Simon E., "Towards DBMSs for Supporting New Applications" Proceedings of the Twelfth International Conference on Very Large Data Bases, Kyoto, August, (1989) .

[Ban88] Bancilhon F.,"Object-Oriented Database Systems". In SIGMOD-SIGACT Conference on PODS (1988).

[B&K89] Bancilhon F., Khoshafian S., "Object-Oriented Database Systems". Journal of Computer and System Science 38, (1989), 326-340.

[C&F80] Chang N.S. and K.S. Fu, "A Relational Database System for Images". In: S.K. Chang and K.S. Fu (eds.), Pictorial Information Systems, Springer, 1980, 288-321.

[C&F81] Chang N.S. and K.S. Fu, "Picture Query Languages for Pictorial Data-Base Systems". Computer 11 (1981), 23-33.

[Cod70] Codd E.F., "A relational model for large shared data banks". Comm. A.C.M. 13, 6 (1970), 377-387.

[GNT89] Gargano M., Nardelli E., Talamo M.,"Representing Geometric Objects in an Extended Relational Model", IASI Technical Report (1989), in preparation.

[Gun88] Gunther O., "Efficient Structures for Geometric Data Management". In Lecture Notes in Computer Science Vol. 337 Springer-Verlag 1988.

[Gut88] Guting R.H., "Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems". In Conference on Extending Database Technology (EDBT 88).

[GNT88] Gambosi G., Nardelli E., Talamo M., "A conceptual model for the representation of statistical data in geographic information systems". In Proceedings of the IV International Working Conference on Statistical and Scientific Database Management (1988).

[Gro84] Grosky W.I., "Toward a Data Model for Integrated Pictorial Databases", Computer Vision, Graphics, and Image Processing 25, 371-382 (1984).

[LRV88] Lecluse C., Richard P., and Velez F., "O_2, an object oriented data model". In Conference on Extending Database Technology (EDBT 88).

[L&J88] Lorentzos N. A. , and Johnson R.G., "An Extension of the Relational Model to Support Generic Intervals", In Conference on Extending Database Technology (EDBT 88).

[M&K88] Mhoan L. and Kashyap R.L., , "An Object-Oriented Knowledge Representation for Spatial Information", IEEE Transaction on Software Engineering, Vol 14, No. 5, May 1988, 675-681.

[M&O86] Manola F. Orestein J.A., "Toward a General Spatial Data Model for an Object-Oriented DBMS", Proceedings of the twelth International Conference on Very Large Spatial Data Bases, Kyoto1986, 328-335.

[MUL89] " MULTIDATA: Integrazione tra Basi di Dati Tradizionali e Basi di Dati Multimediali: Metodo e Strumenti di Progettazione e Interrogazione", Plan of the Research project MULTIDATA

[O&M88] Orestein J.A. and Manola F.A, "Probe Spatial Data Modeling and Query Processing in an Image Database Application", IEEE Transaction on Software Engineering, Vol 14, No. 5, May 1988, 611-629.

[OOM87] Ozsoyoglu G., Ozsoyoglu M. and Matos V., "Extending Relational Algebra and Relational Calculus with Set-Valued Attributes and Aggregate Functions", ACM Transaction on Database Systems, Vol 12, No. 4, December 1987, 566-592.

[RFS88] Roussopoulos N., Faloutsos C., Sellis T. G., "An Efficient pictorial Database System for PSQL", IEEE Transaction on Software Engineering, Vol 14, No. 5, May 1988, 639-651.

[Sam84] H.Samet, "The quadtree and other related hierarchical data structures", Computing Surveys, Vol.16, no.2, June 1984

[S&V89] Scholl, M., Voisard A., "Modeling of thematic maps: an application to geographic databases", Technical Report, INRIA , (1989).