Rapporto di Ricerca CS-2006-3
Giugno 2006

V. Del Fatto, L. Paolino, F. Pittarello

WebMGISQL 3D — Designing a Visual Environment for Querying GIS Databases
WebMGISQL 3D – Designing a Visual Environment for Querying GIS Databases

Vincenzo Del Fatto¹, Luca Paolino¹, Fabio Pittarello²

¹ Dipartimento di Matematica e Informatica, Università di Salerno
84084, Fisciano (SA), Italy
{vdelfatt, ipaolino}@unisa.it
² Dipartimento di Informatica, Università Ca’ Foscari
30172, Mestre (VE), Italy
pitt@unive.it


Abstract — The main goal of this work is to provide an advanced visual environment where users that are not skilled for what concerns the computer science domain may compose queries related to those geographical phenomena for which the third dimension is a relevant feature. Visual queries are composed in a 3D environment accessible from the web where the users manipulate 3D geographic objects, called 3D geometaphors. The geometaphors represent the operands of an underlying algebra characterized by a set of topological, directional and metrical operators; such operators are expressed in the query environment in terms of visual relations between the geographic objects. The introduction of the third dimension for querying the geographic databases has challenged the authors with a number of important issues related to the area of visualization, navigation and object manipulation. According to the principles of usability engineering, the authors have built different prototypes based on a client-server architecture that have been iteratively evaluated by experts and final users in order to discover drawbacks and to improve the quality of the proposal. The result is a coordinated user-friendly 3D visual metaphor for querying GIS on the web, where all the elements needed for composing a query have a visual, easy to understand, counterpart.

Index Terms — Advanced visual interfaces, geographic information systems, visual environment, visual query languages.

1. Introduction

Recent studies have shown that visual languages represent a promising means for allowing unskilled users to query geographic databases, and to interpret and possibly reuse recorded queries [2, 37].
Systems like *Cigales* [4], *GISQL* [5] and the *GIS Wallboard* [14] are mainly based on the definition of graphical representations for the spatial properties associated with the geographic data manipulated, and for the involved spatial operators. Spatial properties are referred to the geometry of the geographic data and to their topology, which describes the objects’ relative positions. When trying to add user-friendliness to GIS, the association of visual descriptions to such features seems to be quite a natural step. As a matter of fact, many systems have been primarily targeted at querying visually the spatial features of GIS data (see also [22, 39]).

The goal of the research that we have been carrying out in the last years has been to provide users with further intuition about the data processed, by means of a visual query language, which describes also the semantics of those data in the real world. A first step in that direction was the development of a visual environment, the Metaphor GIS Query Language (MGISQL), in which was introduced the concept of *geometaphor* to simultaneously capture the topological and the thematic components of geographic data [33].

In this paper we propose an extension of MGISQL to the 3D domain and to the web, called *WebMGISQL 3D*. The key features which have been added include the possibility to make queries about those phenomena where the third dimension is a relevant feature and the chance for the users to query and retrieve the spatial information by making remote requests on the web.

The first feature extends the usefulness and intuitiveness of visual queries to all the phenomena for which the third dimension is relevant. During the design phase the authors of this work have examined carefully the association of variables and relationships to their visual counterparts, for allowing the users to build more easily a mental model of the whole process.

The latter feature is particularly important for enabling the users to access seamlessly remote geographic databases using standard web technologies that require no configuration effort by the user; besides, the client-server approach is useful also for users that access information using systems with low computational performances (e.g., old computers and mobile systems).

The design process has taken advantage of techniques related to the usability engineering [25, 26], including iterative prototyping and a systemic approach to the evaluation of the different design phases in order to obtain a better result. In particular the evaluation process has involved in the different stages both expert evaluators and final users.

The paper is organized as follows: Section 2 analyzes systems related to the proposal (i.e., such systems have been divided in two categories, namely systems for making textual queries and systems for building visual queries); Section 3 gives to the reader an overview of the design methodology used for developing the visual query environment; Section 4 describes the user profiles, activities and scenarios that represent the starting points for the interface design; Section 5 focuses on the definition of the visual counterparts for the operands (i.e., the geomorphs) and for the spatial operators considered in this work; Section 6 presents the visual environment for composing queries: three different prototypes, iteratively developed and followed by their
evaluation, are described; Section 7 presents the underlying system architecture, based on a client/server model; Section 8 draws the final conclusions; finally, Appendix A illustrates background information related to the formal model underlying WebMGISQL 3D.

2 Related Work

The main research areas related to our work include textual languages and visual languages for GIS querying.

Concerning textual languages, Section 2.1 will offer a description of the different proposals and a justification for our choice of the OpenGIS SQL standard; the limits of the textual approach will be discussed too, illustrating to the reader the reasons for migrating to a visual solution.

Concerning visual languages, Section 2.2 will describe the previous proposals of visual languages and environments for performing GIS queries; such visual environments offered a significant starting point that our proposal improved in several respects, including:

− the ability to select in parallel all the categories of the spatial relationships (i.e., topological, directional and metrical);
− the extension of the directional relationship to the third dimension, allowing the users to build queries that include the zenith and nadir clauses;
− the ability to query remote geographical databases accessible from the web;
− the validation of the proposal, according to the techniques of the usability engineering and the user-centered design.

2.1 Textual Query Languages

In literature there are two main approaches for standardising the SQL framework in order to manage also spatial data: SQL3 [24] and OpenGIS SQL [28]. Other approaches for extending SQL [8, 16, 18, 32], although interesting, are not supported by implementations compliant with such proposals.

Both the SQL3 and OpenGIS SQL specifications have facilities for using abstract data types (ADTs). In particular OpenGIS SQL represents an extension to the SQL92 specification for incorporating geo-spatial ADTs [34], enhancing the features already available in the earlier SpatialSQL specification [8]. Currently it is considered as the standard for the manipulation of spatial data; that is the reason why we chose it as the target language for our proposal.

Spatial SQL syntax inherits the same problems associated to the textual query languages for traditional databases: typing commands can be tiring and lead to errors [10]; the syntax is tedious to use and to learn [9]; users may spend more time thinking about command tools to use than thinking about the task at hand [10]. Besides, Kaushik and Rundensteiner [20] state that while spatial relationships are often thought by the users in terms of images that depict the spatial positions, SQL translate
them into a non-spatial language; such translation represents therefore a further obstacle to the composition of the query.
Mark and Gould [23] suggest that composing queries using the natural language probably might be a significant opportunity for allowing the users to interact with a GIS; unfortunately this is not easy to achieve. Aufare and Trepied [1] add that although the natural language approach may appear as a better opportunity for the GIS users, query expressions can be verbose, difficult and characterized by unresolved ambiguities. That is the reason why we shifted towards a proposal including a visual translation of the operands and the operators that characterize the SQL syntax.

2.2 Visual Querying Languages

Visual languages are today being widely used as a means for reproducing the user’s mental model of the data-manipulated content [33]. A great deal of work has already been carried out to use such languages for traditional and object-oriented databases in order to address usability problems. Iconic, diagrammatic, graph-based and multimodal approaches are available. Some approaches related to the spatial domain have considered also sketch based interfaces for composing the query [9, 22]. Lee and Chin [22] have proposed an iconic language where icons are used to represent objects and processes. Queries are expressed by building up iconic sentences using a graphical interface. The difficulties related to such approach arise from the fact that the objects in the query expression need to be explicitly specified along with their associated class and attributes; unfortunately such approach makes the language cumbersome for the casual user. Besides, only a limited number of spatial relations can be used in the queries [9].
Jungert [19] has proposed a query language named GRAQULA. Simple queries are carried out as the user selects the objects on a map and executes functions between the objects themselves. Another graph-based language is described by Traynor and Williams [36] and Traynor [35]. In their approach, users place panels representing categories in a graphical diagram. In spite of that, users must still use textual commands to compose the query. Kaushik and Rundensteiner [20] have proposed a direct manipulation approach named SVIQUEL (Spatial Visual Query and Exploration Language). Users can specify topological and directional spatial relations between region objects using filters called S-sliders. This approach is limited in that it allows the users to compose a query using only objects belonging to the class region. Besides, although the interface is simple to use, there are some ambiguities in the query interpretation [13]. The system was extended [20] for allowing to query non-spatial data, even though this had to be carried out in a separate window. Non-spatial queries were made using sliders, limiting the approach only to simple queries.
Most sketch based languages adopt a query-by-example approach [40] where users sketch an example of the results that they would like displayed. In spatial databases sketches are usually referred to spatial configurations [9]. Sketch! [24] was one of the first sketch based languages for spatial databases, that used a syntax directed editor for composing the queries. The non-spatial parts of the database were queried using diagrams similar to the ER model. Haarslev and Wessel [16] argue that although...
formal semantics are used in Sketch!, there are no mathematical foundations to the spatial relations.

Cigales [4] is another example of a sketch based query language for GIS. Users express queries by clicking on a set of buttons for selecting features and operations; the system then activates the drawing of the queries. The main drawback of Cigales comes from the multiple interpretations and visual representations of the queries and also from the lack of logical operators [31].

LVIS [31] was defined as an extension to the Cigales language. The extension defined some new operators, attempting to resolve the interpretation ambiguities, and was integrated into a customisable visual environment. However, the attempt to resolve the ambiguities was limited [13].

In Spatial-Query-By-Sketch [3, 9] users build up queries by drawing spatial configurations on a touch sensitive screen. Users can augment or reduce the accuracy threshold for the drawing, enabling the browsing of exact and similar matches.

The Pictorial Query Language (PQL), proposed by [12], includes geographic features and it is characterized by an object-oriented environment. Users formulate queries by placing together configurations using symbolic features. In Ferri et al. [13], a syntactic and semantic correctness method is defined, in order to reduce the multiple interpretations of the queries.

In general sketch and drawing based approaches are suitable for expressing similarity-based queries. Such methods can become complex in a general context characterized by composite queries. Users need to know the target of their search and they do not get benefits when they only want to browse through the data sets [21]. Besides, sketch and drawing based approaches rely on the user’s ability to express spatial relationships in a sketch. Even if some approaches offer support to the user during the drawing phase, exact queries can be generally ambiguous due to the several possible interpretations of the visual configurations [31].

VISCO is a multimodal query language for defining approximate spatial constellations of objects [16, 39]. VISCO use a metaphor that is based on the semantics of everyday physical objects. It uses a combination of an iconic library and command line operators; queries are built by drawing in the query pane. A drawback of this language is that it can only query the spatial data in a GIS, and only simple thematic descriptors such as Lake or City can be typed onto the objects.

3 Design Methodology

As stated in the Introduction, the design methodology used for this work takes advantage of the concepts and the techniques related to the usability engineering [25, 26] and the user-centered design [27] in order to obtain a visual environment compliant with the user needs. As illustrated by Figure 1, users are an important part of the design process, starting from the early stages of the requirements’ definition to the final validation stages; together with the usability experts, they give an essential contribution both to the design and the validation of the different stages. Besides, the methodology takes advantage both of iterative design and evaluation of the different design phases; such techniques are two important features of the usability engineering, a
design methodology that aims at obtaining significant improvements of the final application through the systemic application and validation of usability principles.

Let us consider in detail the different steps of the design process: the process begins with the definition of a set of users profiles corresponding to different groups of users that may be interested in using the system; each profile is associated to a specific set of activities that the system should help to accomplish; finally, a set of scenarios consistent with the profiles and the tasks to be accomplished are designed. The user profiles, the related tasks and scenarios are the starting point for the design of the interface. In the final validation phase, representatives of the different user profiles will validate the design proposals, acting in the context of the scenario designed in the initial phase.

![Figure 1. A schematic representation of the design methodology](image-url)
The second phase includes the definition of visual counterparts for the geographical objects and their spatial relationships (i.e., topological, directional and metrical). While it is straightforward to find a good representation for the topological and the metrical relationships, defining a visual counterpart for the topological relationships is more complex. In our approach such definition is based on the 9-intersection model by Egenhofer and Herring, a comprehensive model for binary topological spatial relations that applies to objects of type region, line and point (see Appendix A). The visual representations derived from the 9-intersection model were submitted to a group of users for finding the best match between such visual objects and a set of textual labels related to the topological relationships defined by the OpenGIS Consortium [28].

The third phase, related to the definition of the visual environment for composing the queries, is the most complex and includes the iteration of the prototyping activity, coupled with two different validation stages. A preliminary prototypical environment for composing the queries is defined; in such environment the visual elements validated in the second design phase are visualized together with a set of supporting tools and aids that help the users to manipulate the spatial metaphors inside a 3D area called sensible board, receiving appropriate feedback. The prototypical environment is then validated by a small group of usability experts that consider the different features in relation to the well-known 10 usability heuristics [25, 26]. The results of such validation inform the design of a more advanced prototype taking into account the reviewers’ suggestions; a final validation stage, performed by representatives of the user profiles defined in the first design phase, concludes the process and leads to the third final prototype.

4 User Profiles, Requirements and Scenarios

We identified at the Computer Science department of the University of Salerno two different profiles of users that may be interested in using our system: the student and the database professor; such groups don’t include all the potential users of the system, but they represent two interesting profiles offering the additional advantage of permitting us to validate the system thanks to the constant collaboration of their representatives. Most of their suggestions, collected in the validation phase, are general enough to be useful also for other profiles that may be considered as additional targets for our proposal. The traits of the different profiles were identified interviewing a number of people belonging to such categories.

The student profile characterizes young women and men aged from 19 to 25, studying computer science. They like to travel and visit different places, including foreign countries. They would appreciate a web service accessible from the locations they visit to assist them in collecting information about such places, in order to minimize the time to discover new interesting places. Such service should have an easy interface and should enable them to ask information about locations of places (e.g., museums, restaurants, hotels, stations, etc.) or different associations (e.g., locations of hotels next
to a certain museums, etc.); beside such service should enable them to print maps containing summaries of the search results.

The database professor profile includes people aged from 35 and above, graduated in computer science and teaching database management systems in the public Italian university. The representatives of such profile are interested in exploring all the teaching opportunities to improve the knowledge of their students.

In particular they are interested in exploring new methods to teach spatial SQL that represents a challenge for educational activities: students have difficulties to understand the concepts and the relations between the different spatial operators; therefore the introduction of visual metaphors that represent textual queries may be useful to smooth the learning curve of their students.

4.1 User Requirements

Table 1 summarizes the activities for which the different categories of users would need support; each activity is briefly described and it is followed by two short notes derived from the users’ interviews about the importance attributed to it and to the frequency of the activity itself.

<table>
<thead>
<tr>
<th>student profile</th>
<th>Activity</th>
<th>Frequency</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access the service from any computer connected to the Internet, using standard web technologies that require no software installation</td>
<td>Sometimes</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Locate easily single categories (or instances) of places</td>
<td>Sometimes</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Locate easily places corresponding to spatial associations of different categories of locations</td>
<td>Sometimes</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>View maps describing the search results</td>
<td>Sometimes</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Print maps</td>
<td>Sometimes</td>
<td>Medium</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>database professor profile</th>
<th>Activity</th>
<th>Frequency</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Represent the visual counterparts of the operators and the spatial operands that characterize a spatial SQL</td>
<td>Frequently</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Compose in an intuitive manner a spatial query using such visual counterparts</td>
<td>Frequently</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Compare the visual query with the textual SQL query</td>
<td>Frequently</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Show the query results to improve further the learning process</td>
<td>Frequently</td>
<td>Medium</td>
<td></td>
</tr>
</tbody>
</table>

4.2 Scenarios

Two different scenarios, associated with the user profiles and consistent with the requirements defined above, were considered. Such scenarios were used in the valida-
tion phase where representatives of the two users groups were asked to test the system.

In the first scenario the user plays the part of a student going abroad for vacancy; when the student has arrived to her/his final destination, s/he decides to take advantage of a web based tourist information service, accessible from the hotel, in order to discover quickly interesting locations.

The student learns how to use the system by herself/himself. After a brief exploration of the capabilities of the system, the student takes advantage of the WebMGISQL interface to compose spatial associations of operands’ pairs (e.g., churches located at the city’s north side, one kilometer far from the hotel) in order to find the locations the best fit her/his interests. The student visualizes the search results on a map and finally prints them for using them as navigational aids in the exploration of the city.

In the second scenario the user plays the part of a database professor exploring the expressivity of the WebMGISQL visual interface to represent operands and spatial operators. The professor composes a number of queries selecting operands’ pairs and choosing different spatial operators.

5 Defining the Visual Counterparts for the Operands and the Spatial Operators

Following the traditional approach of visual languages, queries are expressed in terms of spatial composition of visual elements representing objects, operators, and functions. This Section will illustrate the association of visual counterparts to the geographical objects and to the related spatial relations; such association represents the first important design choice and is followed (see Section 6) by the definition of the visual environment for selecting and manipulating such objects.

Section 5.1 will describe the visual artefacts, named geometaphors, representing the geographical objects that can be selected in a query.

Section 5.2, 5.3 and 5.4 will consider the topological, directional and metrical relationships between the geometaphors’ pairs. As stated in the Introduction, while finding an appropriate visual association for the directional and metrical relationships was quite easy, associating textual labels identifying the topological relationships with the visual representations was more complex and included also a validation phase with the final users.

The concluding Section 5.5 will face the issue of representing in parallel all the spatial relationships described above.

5.1 Representing the Operands: the 3D Geometaphor

In our proposal, geographical objects are represented by a composite visual element, named 3D geometaphor, derived from previous research work [33] and represented in Figure 2; such visual element has been extended in order to take into account the role of the third dimension.
Different categories of geographical objects may be represented using the geometaphors, including Regions, Lines and Points.

Geometaphors are characterized by iconic and property components, as shown in Figure 2.

The first component is characterized by the physical description of the object, a textured 3D cube, with an associated meaning. The presence of labeled textures is an important feature in order to stimulate the visual and textual cognitive styles of the end users. In such a way, users understand the meaning at once and are immediately able to use them.

While 2D icons are effective for the retrieval of objects inhabiting a 2D space, they do not allow an exact definition of the spatial relationships where the third dimension is a relevant feature. That is the reason why we need to introduce 3D icons. Such introduction has been necessarily coupled with the definition of a 3D environment where such objects can be manipulated. In fact, as demonstrated by research in experimental and cognitive psychology, the mental processes of human beings simulate physical world processes. Computer-generated line drawings representing 3D objects are regarded by human beings as 3D structures and not as image features; as a consequence, humans imagine that spatial transformations, such as rotations or shifting, happen in the 3D space.

The second geometaphor’s component is divided into type, representing the object type used to store the data (e.g., a REGION for squares or a LINE for rivers), and source, indicating where the data should be retrieved. The source field can contain a table, a view name, an SQL query or a function. For example, the geographical object ARCHAEOLOGICAL OBJECTS, may be characterized by a physical description expressed by a 3D textured cube, a meaning defined by the words Archaeological Objects, a type Point and a source containing the query SELECT * FROM objects WHERE years = "200 b.c." (i.e., such query would retrieve all the archaeological objects instances dated around 200 b.c.).

Information related to the 3D geometaphor is stored using an XML description, validated by the DTD illustrated in Figure 3.
If we consider again the archaeological objects example, we can express the elements of such geographical object with the XML code shown in Figure 4, compliant with the DTD previously described.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE objects SYSTEM "objects.dtd">
<objects>
  <object type="Point">
    <name>Archaeological Objects</name>
    <description>...</description>
    <img>image\arobjects.gif</img>
  </object>
  ...
</objects>
```

The code is a fragment of an XML file that may contain more than one object, as declared by the DTD definition. The tag `object` has an attribute `type` for declaring the type of the object (i.e., Point) and contains all the other components: the tag `name`, for declaring the name of the element (i.e., archaeological objects), the tag `description` for declaring the source and the tag `img` for indicating the path to the image mapping the faces of the 3D icon.

After having defined the representation for the geographical objects, we will focus on the definition of visual counterparts for their relations. In particular, the following three subsections will focus on the topological, directional and metrical relations; at the end, a concluding subsection will focus on how to represent in parallel different categories of relations, giving to the user adequate feedback.

### 5.2 Modeling the Topological Relationships

The model for binding visual representations to topological relationships is derived from a survey we have performed with a group of potential users. The subjects were twenty males and females between the ages of 20 and 50; all the participants were usual users of PCs and were familiar with GIS technology and the notion of topological relationship. The test was conducted in a quiet classroom after courses.

We asked to such subjects to provide a resemblance value between pairs of geomorphs composed in a 3D abstract environment and textual labels expressing...
topological relationships.

Figure 5 represents the pairs of geometaphor and the visual compositions taken into account; each composition corresponds to a different valid Egenhofer 9-intersection matrix [11].

As discussed in [38], the representation of 3D objects on a 2D plane benefits from the implementation of a number of pictorial cues (e.g., perspective, shadow, etc.) that diminish the ambiguity of the recognition by the user. Accordingly to that research, we used a number of cues for easing the task of recognizing the spatial relations between the geometaphors pairs. The cues used included color, semi-transparency and motion (i.e. the ability for the user to change point of view in order to address occlusion issues). It is important to point out that the order the geometaphors are selected and composed was meaningful for the answer and that the subjects were enabled to distinguish the first selection from the latter. For the sake of the reader in Figure 5 the first selection is identified by the red color and labeled as R while the latter selection is identified by the blue color and labeled as B.

The topological relationships are defined according to the OpenGIS specification [28] and include: Equals, Disjoint, Touches, Within, Overlaps, Contains, Crosses and Intersects.

The scale for the resemblance parameter was set from 0 to 10, where 0 was associated to the value it does not correspond to the relationship and 10 was associated to the value it fully corresponds to the relationship.

The results we obtained from the interviews are summarized in Figures 6-11. Each visual composition is associated to a pie chart displaying the resemblance percentages resulting from the answers. The percentages are calculated by summing, for each relationship, the resemblance values assigned by all the subjects \(Y_j = \Sigma x_i\) where \(x_i\) is the value assigned by the user \(i\) and dividing the result by the overall sum of all the values assigned by the users to all the available labels \(Z = \Sigma Y_j\); then the results have been normalized, assigning the value 100% to \(Z\).

Therefore each figure represents, for a given visual composition, the relative importance of the different user interpretations, evidencing if there is any convergence on a specific textual label.
Figure 6 shows that users associate the representation Op1 to three relevant relations, namely Overlaps, Intersects and Crosses.

The result implies that the subjects’ mental model does not univocally associate the Op1 operation with a specific relationship. An additional filter is needed in order to select a single interpretation from the resulting set. For obtaining such result we take advantage of three rules that reduce the most part of the ambiguity:

- **Rule 1**: the order the geographical objects have been selected is relevant (e.g., for the Op4 composition only one of the following relations is true: $A$ Contains $B$ or $B$ Contains $A$).
- **Rule 2**: the Intersects relationship has a lower priority in relation to the other relationships; Intersects is a relaxed relationship that can be defined also as Not Disjoint; in other words it can be associated to all those situations where all the other relationships defined above (with the exception of Disjoint) exist; in this work, when ambiguity situations exist, the higher priority relationship is applied.
- **Rule 3**: some topological relations make sense only for specific categories of objects (e.g., the Point-Point geometry pair accepts only the Disjoint, Equals, Intersects and Overlaps relations).

Applying such rules to the Op1 visual composition we obtain a reduction of the ambiguity deriving from the subjects’ answers.

In the case of the Point-Point (P-P) and Region-Region (R-R) geometry pairs then only the Overlaps and Intersects relationships belong both the Egenhofer’s set and to the resemblance set (Overlaps, Crosses and Intersects). The Overlaps relationship has a priority greater than Intersects; therefore, applying Rule 2, the Overlaps relationship is selected.

In the case of the Line-Region (L-R), Point-Region (P-R) and Point-Line (P-L) pairs, only Crosses and Intersects belong both to the Egenhofer’s set and to the resemblance set. According to Rule 2, the Crosses relationship is representative of Op1 for such geometry pairs.

A more difficult case happens when the ge metaphors represent two lines (L-L). In
such situation both \textit{Crosses} and \textit{Overlaps} are admissible; the users, interacting with a system implementing our algorithm, will need to select the proper relationship at run-time, according to the specific situation.

As for the remaining geometry pairs, according to Rule 3, the \textit{Intersects} relationship is the only choice that may be applied. All the results related to \textit{Op1} are summarized in Table 4.

![Figure 7. Subjects’ interpretation of the visual composition Op2](image)

Figure 7 shows the resemblance result obtained for the \textit{Op2} operation. The most significant percentage is represented by the \textit{Disjoint} relationship (88%); a much smaller value (12%) is associated to the textual label identifying the \textit{Equals} relationship. Users motivated the latter association stating that the cubes had the same size and therefore could resemble the \textit{Equals} relationship; such answer motivated us to adopt a perspective view, instead of an orthogonal one, for the visual query environment where geomorphs are put in relation.

Concluding, the \textit{Op2} operation has been associated to the \textit{Disjoint} relationship for any pair built with the geometry types Point, Line and Region.

![Figure 8. Subjects’ interpretation of the visual composition Op3](image)

The results for the \textit{Op3} visual composition are summarized in Figure 8. Two relationships are particularly important: \textit{Touches}, which gains the 41% of the sample, and
Intersect which is represented by the 22% of the sample. According to Rule 2, we have associated the Touches relationship to any pair of geometries, with the exception of the Point-Point (P-P) pair; given that Touches is not defined for such category of objects (i.e. points), according to Rule 3 we have decided to apply the Intersect relationship.

Figure 9. Subjects’ interpretation of the visual composition Op4

As for the Op4 visual composition (Figure 9), the most important relationship is Contains, but significant results were obtained also for Intersect and Within. According to Rule 2, we have associated the Contains relationship for any pair of geometries that permits this relationship (i.e. all pairs where the first geometry has a greater dimension than the second one: (R-R), (R-L), (R-P), (L-L) and (L-P)). In the other cases we have applied the Intersect relationship that represents the second subjects’ choice. According to Rule 1, the Within relationship has not been considered; such relation seems to be related to a wrong interpretation by the subjects of the geomephors’ selection order.

Figure 10. Subjects’ interpretation of the visual composition Op5

Figure 10 shows the results for the Op5 operation: the Equals, Within and Contains relationships result as admissible hypotheses for the interpretation (i.e. their resemblance value varies from 19% to 20%). The interpretative model discriminates the proper relation on the basis of the geometry dimension (Rule 3). In particular:
• if both the geometries have the same dimension (e.g., Point-Point) the model returns an *Equals* relationship;
• if the dimension of the first geometry is strictly bigger than the second one the model returns a *Contains* relationship;
• if the dimension of the first geometry is strictly than the second one the model returns a *Within* relationship.

![Figure 11. Subjects’ interpretation of the visual composition Op6](image)

The last visual composition considered is *Op6* (Figure 11). Such visual composition is very similar to *Op4*; the only difference is related to the selection order of the operands. The consequence is that the results obtained represent a mirror of those one obtained for *Op4*. In fact, we note a significant predominance of the *Within* relationship, followed by *Contains* and *Intersects*.

According to Rule 2, the *Within* relationship is applied when the first geometry has an equal or smaller dimension than the second one. There is an exception to this rule when both the geometries belong to the type Points; in such situation *Intersects* is applied. Besides, *Intersects* is applied when the first geometry has a greater dimension than the second one.

According to Rule 1, the *Contains* relationship has not been considered because it seems related to a wrong interpretation by the subjects of the geometaphors’ selection order.

In conclusion, Table 2 summarizes the bindings among the different geometry pairs, visual compositions and labels describing spatial relationships.

<table>
<thead>
<tr>
<th>R/R</th>
<th>R/L</th>
<th>R/P</th>
<th>L/R</th>
<th>L/L</th>
<th>L/P</th>
<th>P/R</th>
<th>P/L</th>
<th>P/P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Op1</td>
<td>Overlaps</td>
<td>Intersects</td>
<td>Intersects</td>
<td>Crosses</td>
<td>Overlaps/Crosses</td>
<td>Intersects</td>
<td>Crosses</td>
<td>Crosses</td>
</tr>
<tr>
<td>Op2</td>
<td>Disjoint</td>
<td>Disjoint</td>
<td>Disjoint</td>
<td>Disjoint</td>
<td>Disjoint</td>
<td>Disjoint</td>
<td>Disjoint</td>
<td>Disjoint</td>
</tr>
<tr>
<td>Op3</td>
<td>Touches</td>
<td>Touches</td>
<td>Touches</td>
<td>Touches</td>
<td>Touches</td>
<td>Touches</td>
<td>Touches</td>
<td>Intersects</td>
</tr>
<tr>
<td>Op4</td>
<td>Contains</td>
<td>Contains</td>
<td>Contains</td>
<td>Intersects</td>
<td>Contains</td>
<td>Contains</td>
<td>Intersects</td>
<td>Intersects</td>
</tr>
<tr>
<td>Op5</td>
<td>Equals</td>
<td>Contains</td>
<td>Contains</td>
<td>Within</td>
<td>Equals</td>
<td>Contains</td>
<td>Within</td>
<td>Within</td>
</tr>
<tr>
<td>Op6</td>
<td>Within</td>
<td>Intersects</td>
<td>Intersects</td>
<td>Within</td>
<td>Within</td>
<td>Intersects</td>
<td>Within</td>
<td>Within</td>
</tr>
</tbody>
</table>
As stated at the beginning of the section, we mapped different categories of geometries (i.e., Points, Lines and Regions) with a unified representation, a 3D textured and labeled cube. Some subjects told us that further information about the category of geometries would have been useful to assign more precise bindings between the visual composition and the topological relationships. Because of that, we organized an additional test where the geometries were associated to three different representations (see Figure 12).

![Figure 12. An alternative approach for representing different geometries](image)

The results derived from the new test do not highlight significant improvements. In fact, the resulting resemblance diagrams are similar to the previous ones deriving from the unified representation of the objects; besides, in some cases, the new diagrams have shown a wider dispersion of data.

Figure 13 shows three distinct visual representations (Op 4.1, Op 4.2, Op 4.3); in each situation a blue object (i.e., respectively a Region, a Line or a Point) is represented within a red object representing a Region.

The different representations were shown in parallel to the subjects, asking them to associate the visual compositions to the textual labels. The results show that the choice of increasing the number of representation augments the confusion of the users. In all the situations, compared with the results obtained with the unified representation presented in Figure 9, there is an higher number of labels that have been associated to significant values with a lower variance among them.

We may infer from such results that a unified representation doesn’t vary significantly the users’ interpretation of a certain visual composition and in some cases may lead to better results in terms of convergence towards a unique association between representation and labels.
5.3 Modeling the Directional Relationships

In this subsection we will focus on how to model the directional relationships for representing them in a visual query language. There are several approaches to this issue: some works define directions approximating the objects in terms of their minimum bounding rectangles (MBRs) or as points; Goyal and Egenhofer [15] propose a
model that can be used also for crisp objects and that makes use of a Boolean 3 x 3 matrix for representing the cardinal directions.

However, all such models are not expressive enough to represent the relationships in a 3D space where the meaningful ones are not only North, South, East and West. Therefore we propose an extension for the 3D domain of the model based on the direction-relation matrix [15]. The extension takes into account two additional directions, defined by the zenith and the nadir: the first one is the point on the celestial sphere intersected by a line drawn from the center of the Earth through the observer’s location on the Earth’s surface (Figure 14), while the latter is the point opposite to the zenith.

Consider the minimum bounding parallelepiped (MBP, for short) of a 3D object located in the centre of the 3D space system: the infinite planes passing through the 6 faces of the MBP originate a 27-fold space partitioning (see Figure 15.a). Such 3D matrix(3x3x3, see Figure 15.b) can be used to represent the cardinal direction of a given object (the target object) with respect to the reference object placed into the centre of the matrix.

![Figure 14. Representation of the zenith in a 3D space system](image)

![Figure 15. (a) The space partitioning used for the directional relationships (b) The representation of the relationships in the 3D direction-relation matrix](image)

5.4 Modeling the Metrical Relationships

The model for binding visual representations to metrical relationships is very simple and doesn’t require sophisticated visual representations or complex interpretations of
the compositions of the visual elements. The value expressing the Euclidean distance, the metrical relationship considered for the operands’ pair, is visualized on the 3D scene between the two geometaphors.

5.5 Visualizing the Topological, Directional and Metrical Relationships in Parallel

The opportunity to represent in parallel the different spatial relationships allows to increase remarkably the visual language expressiveness. Without such capability, users could only describe queries referring to the single categories of operators, such as: A disjoint B (i.e., topological relationship), A North B (i.e., directional relationship) or A distance of B <= 100m (i.e., metrical relationship). It is evident that such samples represent only a part of all the sentences that can be generated using combinations of topological, directional and metrical operators.

The visual representations considered in the previous sections use the same artifacts (i.e., the geometaphors) for displaying different spatial relationships; only the metrical relationship adds an additional element to the representation (i.e., the numerical value between the geometaphors). In this section we add further elements to the already defined visual compositions for enabling the users to take advantage of a unified query environment where different relationships can be used in parallel; such elements help the users to distinguish clearly which operators they are currently using, avoiding ambiguity and errors in the query composition.

The following mapping between the spatial relationships selected by the users and the visualization on the 3D space was adopted:

- no selection of the spatial relationships: the cubes representing the geometaphors are displayed in an empty 3D space;
- selection of the topological relationships: the visual composition is enhanced by a simple plane textured with a grid; besides, a semi-transparent cube corresponding to the current bounding box of the geometaphors is added for helping the users to understand the relative positions of the geometaphors in the 3D space (see Figure 16.a);
- selection of the directional relationships: the visual composition is enhanced by a wind rose indicating the four cardinal directions and by a landmark indicating the zenith and the nadir; also in this case the semi-transparent cube corresponding to the current bounding box of the geometaphors is added for helping the users to understand the relative positions of the geometaphors in the 3D space (see Figure 16.b);
- selection of the metrical relationships: a label indicating the current Euclidean distance between the geometaphors pair is added (see Figure 16.c); the visualization is shifted upwards, in order to avoid occlusion problems; besides, the value is updated each time the users move any of the geometaphors.

Figure 16.d shows the solution, adopted in the final prototype (see Section 6.5), that permits to represent in parallel all the spatial relationships.
6 Defining the Visual Environment for Composing Queries

The third design phase, described in this section, is based on the results previously obtained. In particular, the visual counterparts for the geographical objects and the spatial relationships defined and validated in Section 5 are here considered in the context of a graphical interface that permits to manipulate them in order to compose a spatial query. The technique of the iterative prototyping, complemented with two different validation phases, is applied to the visual interface design for achieving a better result in terms of usability. The different phases, already summarized in Figure 1, are discussed in detail in the following subsections.

6.1 The First Prototype

The first prototypical interface was designed on the basis of the visual representations discussed in the previous sections and on a complementary brainstorming phase where different solutions for manipulating the geographical objects and selecting the spatial relationships were discussed. A special attention was reserved to the design of a set of tools for changing the point of view on the scene, in order to diminish the occlusion problems that characterize all the 3D representations.
Figure 17 shows the main 5 functional areas of the interface, evidenced also by different colors:

- the most important area is located in the centre of the user interface and contains the 3D scene where the queries will be composed; such area contains the geometaphors selected by the users and the other artifacts for giving appropriate feedback about the selected spatial operations. Figure 17 displays the content of the area after the selection of the directional relationships; according to Section 5.5 a wind rose and a landmark are displayed; the scene will be completed by a geometaphor pair after the user selection. The user interacting in this area can take advantage of direct manipulation techniques both for moving the geometaphors or for changing the point of view: in the first case the user can drag the different faces of the geometaphors cubes in order to move them along the three Cartesian axes; in the latter case the user can drag the mouse over the rest of the 3D area in order to rotate her/his point of view around the objects that compose the scene;
the area on the right of the 3D scene contains a repository of geometaphors named geometaphor dictionary; the user selects the geometaphor pair s/he’s interested in, dragging the icons and dropping them in the 3D scene in order to compose the visual query;

− the top grey area allows the users to select the spatial relationships they want to apply through a set of check boxes; the user can choose any combination of the available relationships (topological, directional and metrical); for each combination the system gives an appropriate visual feedback, as discussed in Section 5.5; the user may also leave all the three boxes unchecked; in this case the system will interpret such choice as a simple selection of the geographical elements involved;

− the light yellow area on the bottom contains a set of buttons that activate predefined viewpoints related to the four cardinal directions and to the azimuth; besides the fit view button allows to zoom in or zoom out the current viewpoint in order to visualize all the objects that compose the scene; finally, the initial view button, allows the users to get back to the default viewpoint on the scene in case s/he get lost moving through the scene; the introduction of such predefined views is suggested by a number of usability studies that show how the availability of guided navigation tools is a valuable support for users [17]; it offers to the users an alternative option for the direct manipulation of the viewpoint;

− the dark yellow area contains the Query button that triggers the execution of the algorithm for interpreting the scene content as a spatial SQL query; the textual query is visualized in the text area located below the button.

6.2 Heuristic Evaluation

While the validation performed with a consistent group of subjects represent the best methodology for discovering the usability problem related to a given interface, such technique can be very expensive and time consuming. That is the reason why we reserved such methodology for a more advanced stage of the interface implementation, considering an alternative technique for the early stage of the evaluation process.

The heuristic evaluation is a technique that requires the involvement of a small group of usability experts; while such technique doesn’t guarantee the same results of a controlled experiment with a group of users, it can contribute greatly to discriminate the main problems of a given application. According to Nielsen [25, 26], a group of 5 expert evaluators can discover about the 75% of the application’s usability problems.

The experts have been supported in their work by the Nielsen-Molich 10 usability heuristics [26], a set of general principles that can be used both as a guide for the design or for the validation of a user interface.

The results of the heuristic evaluation are shown in Table 3: each guideline is followed by the list of the problems and of the solutions identified by the 5 evaluators; besides, each problem has been assigned a rating, according to the experience of the evaluators. The scale for such ratings, presented in Table 4, corresponds to the Nielsen proposal [25].
Table 3. Results of the heuristic evaluation for the first prototype

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Problem</th>
<th>Solution</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Visibility of the system status</td>
<td>1.1 The SQL string corresponding to the visual composition is visualized only at the end of the query composition process.</td>
<td>Display always the SQL string.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.2 In some circumstances there is no clear visual feedback in the 3D scene about the set of spatial relationships currently selected.</td>
<td>Improve the visual feedback in the 3D scene for all the combinations of the spatial relationships.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.3 There is no visual feedback when the user drags a geometaphor in the 3D scene.</td>
<td>Display an image representing the selected geometaphor under the mouse pointer during the drag operation.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.4 The SQL query string displayed to the users is not always consistent with the visual composition.</td>
<td>Update the SQL query string displayed to the users after each change in the 3D scene or a different selection of the spatial relationships.</td>
<td>3</td>
</tr>
<tr>
<td>2) Match between the system and the real world</td>
<td>2.1 Some interface labels don’t correspond to the users’ language.</td>
<td>Use natural language terms.</td>
<td>2</td>
</tr>
<tr>
<td>3) User control and freedom</td>
<td>3.1 It is not possible to delete the geometaphors dragged in the 3D scene.</td>
<td>Add a trash metaphor to the interface.</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>3.2 After the scene deletion, it is not possible to restore it.</td>
<td>Add a function for saving the current visual composition.</td>
<td>2</td>
</tr>
<tr>
<td>4) Consistency and standards</td>
<td>No problem.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5) Error prevention</td>
<td>5.1 Even though the spatial relationships are defined only between geometaphors’ pairs, the user can insert more than two geometaphors into the scene.</td>
<td>Disable the insertion mechanism after the second geometaphor.</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5.2 The text field for showing the SQL can be edited by the users.</td>
<td>Don’t allow the users to edit the text field.</td>
<td>4</td>
</tr>
<tr>
<td>6) Recognition rather than recall</td>
<td>No problem.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Flexibility and efficiency of use

7.1 The users can’t delete the current scene. Add a button for clearing the scene.

Aesthetic and minimalist design

No problem.

Help users recognize, diagnose, and recover from errors

No problem.

Help and documentation

10.1 Some technical terms like drag, scene, topological, metrical and directional are not clear to users that are not familiar with the GIS domain. Insert links to popup windows for helping the users to understand the meaning of the technical terms.

Table 4. The rating scale for the heuristic evaluation.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not an usability problem.</td>
</tr>
<tr>
<td>1</td>
<td>Cosmetic problem only: no need to fix it unless extra time is available for the project</td>
</tr>
<tr>
<td>2</td>
<td>Minor usability problem: low priority for fixing it</td>
</tr>
<tr>
<td>3</td>
<td>Major usability problem: high priority for fixing it</td>
</tr>
<tr>
<td>4</td>
<td>Usability catastrophe: highest priority for fixing the problem</td>
</tr>
</tbody>
</table>

As shown by Table 3, the most important problems evidenced by the evaluators are related to heuristics number 1, 3, 5 and 10, referred respectively to the visibility of the system status, the user control, the error prevention and the documentation. The exam of the interface in relation to the other heuristics reveals only minor problems or the absence of specific drawbacks.

The observations of the evaluators were considered for the second user interface implementation, according to the following criteria: the problems associated to ratings between 3 and 4 were resolved following the evaluators’ suggestions; the problems associated to lower ratings were resolved only if the solution didn’t require a high implementation effort.

6.3 The Second Prototype

The second prototype, shown in Figure 18, takes advantage of the suggestions derived from the heuristic evaluation.

Concerning the catastrophic usability problems (i.e., rating value 4) the following solutions have been implemented:
problem 3.1: a scissor icon has been added to allow the removal of the geometaphors from the scene (Figure 18, on the bottom);

problem 5.1: the dictionary area is disabled after the second geometaphor has been dragged to the 3D scene;

problem 5.2: the editable text area where the SQL string appeared has been replaced by a not editable text area (Figure 18, below the 3D scene); the content of such area appears when the second geometaphor is dragged into the 3D area and it is dynamically updated each time the user moves any geometaphor or changes the selection of any spatial relationship; therefore such improvement represents a solution also to problems 1.1 and 1.4;

Concerning the major usability problems (i.e., rating value 3), the prototype has been improved as follows:

- problem 10.1: a set of links has been placed in the different areas of the interface in order to give contextual help in relation to the terminology and to the different functions of the prototype.

Concerning the remaining lower level usability problems, the following improvements have been considered:

- problem 1.2: the visual feedback on the 3D scene has been improved in order to let the users to understand more clearly which spatial relationships are currently used; in order to reach such aim, some of the complementary artifacts that characterize the different spatial relationships (i.e. the grid plane and the rose wind) have been transformed into semi-transparent surfaces for letting the users to see when they are both present into the scene;

- problem 2.1: some interface labels have been changed for adhering to the users’ language.

Finally, the second prototype includes two new features for improving its flexibility:

- the possibility for the users to select the geometaphors representing specific instances of geographical objects instead of generic classes (e.g., the geometaphor Garda Lake instead of the generic Lake); the mouse-click on a generic geometaphor of the dictionary labeled as Layer activates a secondary dictionary labeled as Feature, related to the available instances for such class (see Figure 18 on the left); the user can choose to drag classes and/or instances into the 3D scene for composing the query;

- the possibility to save the results of the query in a new layer of the GIS, associating to it an identification icon and a set of user definable colors for representing the results on the map (Figure 18, on the right).
6.4 The Usability Test

The second prototype was presented for evaluation to a group of users, in order to receive further feedback about the application. The users were representatives of the profiles defined at the beginning of the design process and were asked to play the scenarios defined in Section 4.2; their contribution was very useful to lead to the final refined version of the interface. The test included both quantitative and qualitative measures, in order to get information both in terms of performance and subjective satisfaction.

6.4.1 Methodology

Participants
We selected twenty participants, divided into two groups composed by 10 subjects, which were totally unfamiliar with our application.

Group A was composed by people corresponding to the student profile; they were males and females between the ages of 19 and 25, novices in relation to the GIS/DBMS domain.
Group B was composed by subjects belonging to the database professor profile; they were between the ages of 35 and 50, skilled in the GIS/DBMS domain.

**Apparatus**
All the tests were organized at the Department of Matematica e Informatica of University of Salerno, in a quiet classroom after courses. A laptop computer working both as client and server was used. Full details about the client-server architecture can be found in Section 7. In order to record the interaction on the screen and the users’ talking aloud we used the Camtasia Studio™ software by Techsmith ™.

### 6.4.2 Scenarios and Tasks
Both the groups of the participants were asked to play scenarios compliant with those ones defined in Section 4.2, in order to allow the survey team to capture their performance and subjective satisfaction. Each scenario was described by a script that was given to the users:

**Script 1: Scenario A for the student profile.**
You play the part of a student going abroad for vacancy. You are a lover of eighteenth century paintings and you know that Salerno’s churches are particularly full of this kind of pictures. You stay in an hotel located in the Mercatello quarter which is quite far from the historical centre of the city, where the churches are concentrated. You have rented a car for parking it on the Concordia square because the hotel’s receptionist told you that it is very near to the centre. You have not much time for visiting and therefore you decide to take advantage of the WebMGISQL 3D system, accessible from the hotel, for satisfying your information needs. After a brief self training you have to solve two different tasks.

**Script 2: Scenario B for the database professor profile.**
You play the part of a database professor teaching information systems and you want to explain the SQL language taking advantage of the visual metaphors offered by WebMGISQL 3D for permitting the users to understand more clearly the concepts of operands and spatial relationships. In particular you have to compose visually two queries concerning the city of Salerno and involving different spatial operators; at the end you have to visualize the corresponding SQL code and show the results on a map.

While the groups performed their work in the context of different scenarios, we assigned the same tasks to both of them, in order to make a comparison of the results obtained:

- **Task 1.** Look for all the roads located at Northwest of Concordia square and distant less than 1000 meters from the square itself.
- **Task 2.** Display the map showing all the churches within the historical centre quarter.

Both the tasks assigned allowed to test a number of significant functionalities of the prototype, including: selection and composition of geometaphors, selection of spatial
relationships, SQL code generation and visualization of results.

6.4.3 Procedure

The survey team was present at the test and included a test assistant, an observer/note-taker and an observer/time-keeper. At the beginning of the session, the test assistant read to each participant the appropriate script and illustrated the two tasks to be accomplished. Participants were not trained, but each subject was allowed to explore the functionalities of the interface by herself/himself before beginning the tasks. The observer/note-taker recorded the participants’ actions, the navigational errors, and the verbal comments made during the task execution (i.e. talk aloud technique). The observer/time-keeper kept track of the time needed by each participant to complete the task and of her/his ability to perform the task correctly. After the completion of all the tasks, participants were asked to answer to a brief questionnaire about the features of the interface, evidencing both the positive features and the drawbacks.

6.4.4 Discussion of the Results

The aim of our test was to verify the application potentialities on two main groups principally composed by novice users (group A) and GIS/DBMS experts (group B) and to validate the design choices that characterize the user interface.

The results obtained, summarized in Tables 5-8, are encouraging. In particular:

- Table 5 shows that all the GIS/DBMS experts were able to complete the assigned tasks using the visual interface. The percentage of success decreases for the novice users; such result can be explained with the basic knowledge related to the GIS domain that characterized the subjects belonging to the group A; in spite of that the results obtained are encouraging and vary from 75% to 87.5% of completed tasks (i.e., due to the higher difficulty of the task 2, a minor percentage of users were able to complete the task).

- The influence of the knowledge of the domain is underlined also by Table 6 that shows the time spent by the different groups for completing the tasks (i.e., only the completed tasks were considered). The difference is particularly relevant for the first task: the novice users completed the task in 13’02” while the GIS/DBMS experts needed only 8’56”. The table shows also that the experience acquired performing the first task was a great help for diminishing the performance time during the execution of the second task; such result shows that the interface is not characterized by a steep learning curve and allows the users to perform significantly better even after the execution of a single task. In particular the results related to the task 2 show that the novice users obtain the best reduction of the execution time; besides, the mean time of their performance is similar to that one of the experts users (i.e., 4’35” vs. 4’04”).

- Table 7 and 8, concerning the subjective satisfaction of users, confirm the appreciation for the overall interface and the visual composition of the queries. The vis-
ual query composition got the highest satisfaction values, while the appreciation for the overall interface, although positive, was slightly lower due the identification of some drawbacks; such drawbacks were taken into account for the development of the final prototype (see Section 6.5). In both cases, novice users assigned higher satisfaction values, because the interface allowed them to overcome their inability to compose the queries using alternative methods, such as the SQL string composition.

Table 5. Percentage of participants who completed correctly the assigned tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Group A - novices</th>
<th>Group B – experts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>87.5%</td>
<td>100%</td>
</tr>
<tr>
<td>Task 2</td>
<td>75%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 6. Mean time for completing the assigned tasks (the reported value includes also the time spent by the users on reading the task script, which is estimated in 30" for each task).

<table>
<thead>
<tr>
<th>Task</th>
<th>Group A - novices</th>
<th>Group B – experts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>13’02”</td>
<td>8’56”</td>
</tr>
<tr>
<td>Task 2</td>
<td>4’35”</td>
<td>4’04”</td>
</tr>
</tbody>
</table>

Table 7. Ease of use - Overall interface

<table>
<thead>
<tr>
<th></th>
<th>Group A – novices</th>
<th>Group B – experts</th>
</tr>
</thead>
<tbody>
<tr>
<td>(scale from 1 to 5, where 1=very difficult and 5=very easy)</td>
<td>3.31</td>
<td>2.93</td>
</tr>
</tbody>
</table>

Table 8. Ease of use - Visual query composition

<table>
<thead>
<tr>
<th></th>
<th>Group A - novices</th>
<th>Group B - experts</th>
</tr>
</thead>
<tbody>
<tr>
<td>(scale from 1 to 5, where 1=very difficult and 5=very easy)</td>
<td>4</td>
<td>3.79</td>
</tr>
</tbody>
</table>

6.5 The Final Prototype

The analysis of the observations during the task execution and of the answers recorded after the test completion led to develop the final refined version of the user interface, shown in Figure 19. In particular the following improvements were introduced:
the map of the results is displayed on the right of the visual query environment on a unified interface panel and it is updated each time the user presses the Update Map button;

the set of widgets for saving the results of the query was moved to a pop-up windows that can be recalled after the visualization of the results; the user is no more obliged to specify the parameters for saving the query (i.e. the name of the new GIS layer, the identification icon and other options) before visualizing the results, speeding up the process;

the top of the interface displays the sequence of steps for making a visual query; each label appears on the top of the related interface widgets and it is associated to a short explanation; such adjustment was necessary because the correct step sequence was not always intuitive for all the users;

the text SQL string located on the bottom side of the interface was replaced by a natural language query representation; the SQL representation is still visible clicking over the show SQL code button, added on the top of the interface;

the set of the checkboxes for choosing the spatial relationships was moved near to the natural language query representation, in order to let the users to understand better the influence of the spatial operators on the query composition;

a new link, named Get started, was added in order to help the users to get acquainted with the application; such link triggers an help page containing information about the application aim, the steps for performing the query and the meaning of the different interface objects;

a Delete button was added to the interface for allowing the users to delete the 3D scene; a similar widget was already available in the previous interface, but its meaning was not understood by all the user because of its representation as a scissor;

the labels associated to the selection of the classes and the instances of the geometaphors (layer and feature) were replaced by descriptions (category and object) more familiar to the end users;

the objects dictionary displays simultaneously all the instances; in the older prototype such repository displayed only the instances corresponding to the class selected by the user in the category dictionary; such behavior was not understood by some users that consequently were not able to find the objects they needed;

finally, the supplementary textual labels associated to the geometaphors in the second prototype were cancelled because they generated occlusion problems when also the metrical relationships were selected (see Figure 18); for avoiding additional occlusion problems the metrical label was shifted upwards (see also Figure 16.d).
7 The Architecture

The 3D WebMGISQL environment presented is based on a client-server architecture, described in Figure 20.

The client side is used for composing the queries and visualizing the corresponding results whereas the server side consists of a GIS application server devoted to the query processing.

On the client side the Interaction Components, evidenced by the dotted line, re-
ceive the user input and communicate the visual feedback during the query composition. The user chooses the appropriate geomeaphors from the Dictionary and selects the spatial relationships using the Interpretation Selector. The 3D Scene Component materializes the geomeaphors in the 3D environment and provides the user with visual feedback about the selected spatial relationships. The user manipulates the chosen geomeaphors in the 3D space for completing the query; besides, the user is enabled to change the initial selection of the spatial relationships, receiving an appropriate visual feedback. The Visual Query Translator component transforms the visual query into an SQL-like statement according to the model defined in Section 5. Finally, the SQL string is sent to the server for being processed.

On the server side the SQL code is received by the Web MGISQL 3D server module which triggers the Postgis and Mapserver servers in order to generate the results. The results are converted to a raster format and sent back for the visualization on the client.

The visual query system can be easily applied without any modification to existing GIS databases compliant with Mapserver. The only part that needs to be generated for each database is the set of geomeaphors, according to the available data.

All the technologies used for the implementation on the server side (i.e., Apache, PHP, Mapserver and Postgis) are open-source and freely distributable. The client has been built using widely diffused web technologies. The visual query environment can be accessed by any Windows XP computer connected to the Internet and running the Microsoft Internet Explorer web browser. All the environment components (i.e. the VRML ActiveX for the 3D scene and the XHTML/ Javascript components for the other interaction widgets) can be accessed and automatically installed from the network, giving to the system a significant flexibility and avoiding complex setup procedures for the users.

8 Conclusion

In this paper we presented the design of WebMGISQL 3D, a visual spatial query language for geographic information systems.

The design process took advantage of techniques related to usability engineering, including iterative prototyping and a systemic approach to the evaluation of the different design phases in order to obtain a result compliant with the users’ needs. To obtain such result, the evaluation process involved in the different stages both expert evaluators and final users.

The first design phases considered, from a user centered perspective, the association between visual representations (i.e., geomeaphors) representing geographical objects and a complete set of spatial relationships. The association between a set of topological relationships compliant with the OpenGIS standard and visual counterparts was the most complex and included an experimental survey with a group of users. While the results of the survey often showed a convergence towards specific associations, in other cases users indicated a set of associations complementary to the main choice. Therefore, in order to reduce the ambiguity, the answers to the survey
were filtered with a set of rules taking into account the priority of the relationships, the type of the geomeorphors and the order they were selected.

The result was an interpretation model of the visual compositions representing the topological relationships; such model was then complemented with the definition of visual counterparts for the directional and metrical relationships.

The visual mapping of the geometrical objects and of the spatial relationships represented a primary element for the design of a visual environment for querying geographical information systems.

Two different prototypes were iteratively developed and evaluated, leading to the implementation of a final visual environment. Such environment represents a significant contribution for the implementation of visual query systems taking into account as a primary requisite the users’ mental model, in order to augment the overall usability of the systems themselves.

Preliminary and partial versions of the themes discussed in this work are available in [6, 7, 29, 30].

References


Appendix A

The 9-Intersection Model

The formalism underlying MGISQL 3D is based on the 9-intersection model by Egenhofer and Herring [11]. It is a comprehensive model for binary topological spatial relations and applies to objects of type Region, Line and Point. It characterizes the topological relation $r$ between two point sets, A and B, by the set of intersections of A’s interior ($A^\circ$), boundary ($\partial A$), and exterior ($A^-$) with the interior, boundary and exterior of B. With each of these nine intersections being empty ($\emptyset$) or non-empty ($\neg\emptyset$), the model has 512 possible topological relations between two point sets, some of which can never occur, depending on the dimensions of the objects, and the dimensions of their embedding space. For example, let us consider the topological relation adjacent (between two regions A and B). It can be represented as follows.

A adjacent B ⇔

\[
\begin{array}{ccc}
A^\circ & \partial A & A^- \\
B^\circ & \emptyset & \emptyset & \neg\emptyset \\
\partial B & \emptyset & \neg\emptyset & \neg\emptyset \\
B^- & \neg\emptyset & \neg\emptyset & \neg\emptyset \\
\end{array}
\]

Figure 1. The Egenhofer’s 9-intersection model for pairs of cubes
The spatial relationships are used as a basis for the definition of the *MGISQL 3D* spatial operators. The Egenhofer’s model can be used for describing the spatial compositions between 3D objects. Figure 1 shows all the possible combinations between two cubes (i.e., in this paper cubes are used for representing different typologies of geographical objects) and the association with an Egenhofer’s matrix.

**The OpenGIS Relationships**


Table 1. Description of the topological relationships defined in the OpenGIS SQL specification

<table>
<thead>
<tr>
<th>Group</th>
<th>Relationship</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topological</td>
<td>Touches(g1, g2)</td>
<td>Returns TRUE if the only points in common between g1 and g2 lie in the union of the boundaries of g1 and g2.</td>
</tr>
<tr>
<td></td>
<td>Crosses(g1, g2)</td>
<td>Return TRUE if the intersection between g1 and g2 results in a value whose dimension is less than the maximum dimension of g1 and g2 and the intersection value includes points interior to both g1 and g2, and the intersection value is not equal to either g1 or g2.</td>
</tr>
<tr>
<td></td>
<td>Within(g1, g2)</td>
<td>Returns TRUE if g1 is completely contained in g2.</td>
</tr>
<tr>
<td></td>
<td>Contains(g1, g2)</td>
<td>Returns TRUE if g1 totally contains g2.</td>
</tr>
<tr>
<td></td>
<td>Overlaps(g1, g2)</td>
<td>Returns TRUE if the intersection of g1 and g2 results in a value of the same dimension as g1 and g2 that is different from both g1 and g2.</td>
</tr>
<tr>
<td></td>
<td>Disjoint(g1, g2)</td>
<td>Returns TRUE if the intersection between g1 and g2 corresponds to the empty set.</td>
</tr>
<tr>
<td></td>
<td>Equals(g1, g2)</td>
<td>Returns TRUE if both Contains and Within are TRUE</td>
</tr>
<tr>
<td></td>
<td>Intersects(g1, g2)</td>
<td>Return TRUE if one of the Crosses, Within, contains, Overlaps or Equals relationships are TRUE.</td>
</tr>
<tr>
<td>Metric</td>
<td>Distance(g1, g2)</td>
<td>Returns the minimum Euclidean distance between g1 and g2.</td>
</tr>
</tbody>
</table>
This specification treats several aspects about geographical data, such as the DBMS architecture, data types, implementation, relationships, SQL according to SQL92 specification, etc.

After six years, the OpenGIS specification has become the standard for all the companies and research groups dealing with geographical data management. In particular, the OpenGIS definition of spatial relationships has become a standard for every tool or language which needs to query GIS. Basically, the definition includes topological and metrical relationships. The former expresses a qualitative property between two geometric objects while the latter adds quantitative information.

Table 1 describes the topological and metrical relationships as defined by OpenGIS consortium for the geometries $g_1$ and $g_2$. 