

TOWARDS A NEW VISUAL QUERY LANGUAGE FOR GIS

Haifa Elsidani Elariss
Faculty of Computing, Information
Systems and Mathematics,
Kingston University London
Kingston upon Thames KT1 2EE,
United Kingdom
K0437051@kingston.ac.uk

Souheil Khaddaj
Faculty of Computing, Information
Systems and Mathematics,
Kingston University London
Kingston upon Thames KT1 2EE,
United Kingdom
s.khaddaj@kingston.ac.uk

Ramzi Haraty
Division of Computer Science
and Mathematics,
Lebanese American University
Mme Curie Street – Beirut,
Lebanon
rharaty@lau.edu.lb

ABSTRACT

In recent years, many non-expert user applications have been developed to query Geographic Information Systems (GIS) that are used to browse and view data about space and time thus naming them spatio-temporal databases. Our research aims at designing and developing an International Visual Query Language (IVQL) that allows users to query databases based on their location. The databases are installed on a GIS server computer. The queries are sent to the server from a mobile phone through the Short Messages System (SMS). With the emerging Globalization of user interfaces, IVQL is meant to have a global and international user interface that could be understood by all users worldwide who are from different countries with different cultures and languages. We propose a user interface consisting of smiley icons that are used to represent and build an international query language. Smiley icons enable the users to access data and build queries easily and in a user-friendly way. The visual query language, its user interface and its model are explained. The query formulation is illustrated using a sample GIS system for tourists. IVQL can be applied to fields such as Management Information Systems and tourism. We then conclude about our future work.

KEY WORDS

Visual query language, database and mobile phones

1. Introduction

During the last decades, computer applications have been deployed to manage spatial data with Geographic Information Systems. GIS are able to store, represent, manipulate, analyze and visualize a large volume of geographic information. They play a major role in many domains such as urban and environmental planning, transportation, utility mapping and tourism. They can be applied in different fields such as Bio-informatics, Air traffic, Location Based Services (LBS), Management Information Systems (MIS), Customer Relationship Management (CRM), Decision Support Systems (DSS), problem solving and many others [1].

GIS deal with spatial data objects and relations that are often represented by a visual approach. The spatial objects are stored in the form of maps. To execute

operations that analyze spatial data, structured query languages (SQL) are used to integrate the analysis with the GIS. One problem is that SQL query languages possess complex textual syntax, textual input and output. Another problem with these textual query languages is the need to know the structure of the database schema before writing a query [2]. Most of GIS users are expected to be non experts in the geographic domain. Since it is difficult and inappropriate for casual users to learn query languages, the need emerges to design and develop visual, user-friendly and ease-of-use query interfaces.

A major challenge nowadays is the visualization of query languages. The visual query languages purpose is to enhance the GIS effectiveness and usability. They facilitate the process of query formulation and the manipulation of data analysis functions. We hereby propose a framework for an International Visual Query Language (IVQL). The approach addresses the visual query formulation through the use of smiley icons that represent geographic features, relations and functions such as, themes, objects and operators.

With the emerging Globalization of user interfaces, the use of smiley icons is proposed in order to build an international query language. IVQL is an International query language. It can be expanded as much as the user needs. It can include as many icons as needed whether they represent themes or objects. It is expected to manage dynamic query processing. The query formulation is hereby illustrated using a sample GIS system for tourists who need to query the database based on their current location from a mobile phone. The databases are installed on a GIS server computer. The queries are sent to the server from a mobile phone through the Short Messages System (SMS). We also suggest the implementation of a prototype and the evaluation of the framework and the IVQL user interface.

Globalization refers to the production of products that are used internationally and worldwide. With this emerging global information society, the need to develop international user interfaces becomes a crucial issue in visual query languages. Marcus, in his paper [3], defines globalization in user-interface design and demonstrates why globalization is vital to the success of computer-based communication products.

A smiley is a graphical representation that displays a smiley face. Smiley faces have a common name: Emoticon or Smiley. They are used to express emotions and convey facial expressions such as, happy, sad or bored. They can be easily understood by all peoples in all countries worldwide because they are universal signs that can be well transferred from one culture to the other. Smiley icons have a high level of expressiveness. Pang, in his thesis [4] uses smiley icons to express the operator's feelings in a nonverbal interface for the McDrive system developed for the McDonald chain. The smiley faces are displayed on the screen to show the right facial expression. In our research, smiley icons are used to represent themes, objects and operators. They are used to formulate queries in a global language.

Section 2 deals with the state of the art visual languages that have been proposed for querying spatial or spatio-temporal databases. In section 3, we describe the user interface and explain how smiley icons represent themes, objects and operators. Section 4 contains a description of the IVQL architecture and how queries are processed after being formulated. Section 5 describes the model. Section 6 includes the conclusions drawn from this paper and discusses suggestions for future work.

2. Visual Query Languages

The IVQL language is based on the state-of-the-art visual query languages. Each state-of-the-art visual query language has its own advantages and limitations. IVQL is built on a combination of the advantages of some of the state-of-the-art visual query languages. In this section we briefly describe some of the state of the art visual query languages that have combined to build the IVQL language.

2.1 The LVIS Language

LVIS [5] is a visual query language for Geographic Information Systems (GIS) and for spatio-temporal databases. It uses Metaphors for Querying Visually Spatio-Temporal Databases [6]. It uses Geographic pictures to represent spatial objects (object types) and Geometric shapes to represent relations among them (operators). It uses Balloons and Anchors placed on a temporal axis to describe spatio-temporal criteria. The user builds a visual query by selecting a combination of the icons that represent the pictures, shapes, and temporal axis [7]. The temporal queries are built inside balloons. The balloons may be minimized in form of anchors for better visualization of the query under development.

The interface of LVIS contains a working area where the user builds the visual queries. Once a query is ready, it is validated then transferred to the query area either for execution or for further refinement. The query is then translated to a pilot language that is independent from the host query language of the target GIS platform. The pivot language is compatible with the standardized Spatial-SQL [8] for spatial operators and with TSQL [9] for temporal

operators and uses the Allen temporal intervals [10]. Then, the visual query is translated from the pivot language into the GIS target language and the result of the query is visualized on a map. LVIS is an extension of the CIGALES [11] and [12] which is a visual query language for spatial databases where temporal aspects are not considered and the set of operators is very limited. LVIS interface is based on object types, objects and operators. An icon, a geometric shape and a color are used to visualize an object. Textual elements can be added to reinforce the visual representation. Balloons and anchors can also be added to represent temporal queries.

2.2 The PHENOMENA Language

PHENOMENA, described in [13] and [14], is a visual query language for Continuous Fields developed for Geographic Information Systems and databases. It manages both continuous fields and discrete objects in a uniform manner. Continuous fields represent real-world events and phenomena that are related to the environment and its resources, such as temperature, pressure and electromagnetism. Phenomena are measured by distinguishing what varies and how smoothly. Discrete objects represent points, lines or areas. In the visual interface Phenomena, the user is provided with the capability to capture some features of a scenario by selecting an area of interest and handling the events involved.

The Phenomena visual language allows the user to select portions of continuous fields and formulate queries based on spatial conditions. Continuous fields and spatial conditions are visualized using a set of suitable metaphors called *geometaphors*. When building a query, the user chooses first a continuous field that is represented as geographic data and then he chooses a function that describes its behavior. Combining continuous fields with spatial conditions provides the user with the capability of querying those phenomena that vary in space, such as the temperature on a region, the density of population in a city, the pollution in a river, a surface elevation or the point where the wind has the maximum power. Thus, Phenomena is intended to be used by domain experts such as geologists, meteorologists, archeologists and sociologists.

The graphical user interface of Phenomena is made up of two areas. The area on the left contains the icons that represent continuous fields such as temperature and the icons that represent spatial objects such as river. The area on the right contains the icons that represent the aggregate functions such as minimum. The user drags and drops the icons into the central tabbed panel in order to formulate the query that follows the SQL like Select-From-Where scheme. Once a query is formulated, it can be either executed or saved for later loading and refinement. Phenomena is used to define an extension of the standard *OpenGIS SQL* that is meant to manage continuous fields and to integrate them with discrete data.

2.3 The *GeoQA* Language

GeoQA [15] is a visual query and analysis tool that has been proposed as an integral component of the GiniNT which is an Object-Oriented geographic information system framework used for GIS applications development. GeoQA has an interactive visual interface used for formulating and processing spatial, non-spatial and combined queries. It provides the user with the capability of performing spatial, thematic and statistic analysis. It provides the user with graphical presentation of query results for an easier spatial analysis of specific geographic situations. The GeoQA tool may be applied in GIS applications that are used for urban planning, natural resources management, weather modeling, vehicle navigation, prevention of pollution and natural disasters, etc. It has been applied on the GIS application GeoTT which is developed for monitoring, maintaining, inventory managing and analyzing the Telegraph-Telephone cable network in the Republic of Serbia, Yugoslavia as described in [2] and [16].

The visual graphical user interface of GeoQA is a WYSIWYG interface made up of icons, toolbars and a panel in which maps are displayed each in a separate window. Maps may be moved, panned and zoomed. The GeoQA GUI is supported with dialog boxes, forms, pop-up menus and direct object manipulation. Two sets of Icons are displayed in toolbars based on their function and representation. The first set contains icons representing all spatial entities. The second set contains icons representing spatial relations (topological, geometric and direction) and spatial operators (point, region, window). The user uses the mouse and the icons to manipulate directly and interactively the maps and objects that are visible on the screen. Dialog boxes and text input are used in formulating queries related to spatial queries including spatial relations. GeoQA provides the user with a variety of query facilities such as *point-query*, *region-query*, *query composer*, and *SQL query builder*. Queries can be saved for later loading and refinement.

The output result of a query is displayed in a window in the panel with or without the background map based on the user's choice. GeoQA has a fast query processing. It uses the *Mediator* [17], which is a mediator for an OO front-end to RDBMSs, to process the non-spatial queries. It uses geometric filters, spatial indexing (grid structure) and computational geometry algorithms to process spatial queries.

2.4 Other Visual Languages

The above described state-of-the-art visual query languages are some of the research work developed. There are also a number of other approaches to Visual Query Languages that are hereby described. Kaleidoquery [18] is a visual query language for object databases that depicts the query as a filter flow. The Filter-flow [19] and [20] is a visual query language and interface for large spatial databases. The various types of queries are expressed by Basic Filters that are represented by icons in

order to construct query diagrams between data input and data output elements. The Exploratory Database View Constructor (EDVC) [21] and [22] is a visual query interface especially experimented to support Link Analysis (LA) that is used to understand data collected in criminal or terrorist threat analysis. It uses rectangular icons to visualize objects such as people and locations. It uses directional lines to represent the connections and links between the objects. Special Interactive Visual Tools have been developed, as described by [23], to explore the analysis of spatial-Temporal data. The interface environment provides the user with time-control icons located in a time-control panel. These icons allow the user to specify the beginning and the end of a certain period of time during which the changes have occurred on a spatial location.

2.5 Advantages and Limitations of the Current Visual Query Languages

The above described state-of-the-art visual query languages have demonstrated a considerable improvement in the field of visual query languages. However, each one of them has some limitations. We hereby discuss the advantages and limitations of the above described visual query languages. We also explain how our IVQL language combines their advantages and how it aims at finding a solution to their limitations.

The LVIS visual query language that is described by [7] has a major advantage which is the idea of using iconic Metaphors to depict, represent and visualize objects locations and operators. The PHENOMENA visual query language that is described by [13] and [14] has a major advantage which is the use of a uniform visual environment. The uniform visual environment provides the user with iconic Metaphors that represent operators. These operators can be applied simultaneously to spatial and continuous fields. Thus, allowing the user to manage continuous fields as actual spatial data. The GeoQA visual query language that is described by [15] has a major advantage which is the flexibility in the options in building queries. In GeoQA, the user can TEXT INPUT a query, build a query by using a combination of TEXT INPUT and icons, and the possibility of building complex queries by using only icons (iconic query building).

A combination of the advantages of the above state of the art visual query languages has been used to build the IVQL visual query language. The use of iconic metaphors to depict object, locations and operators has been used from the LVIS visual query language by using the smiley icons that have a more expressive power. The idea of applying the same operators to continuous fields as well as spatial fields in the PHENOMENA visual query language has been used in the IVQL visual query language. The operator *< less than* can be applied to spatial and temporal fields such as, find locations that fall within a certain distance and to find events that fall within a certain time interval. The IVQL visual language was also built on a feature of the GeoQA visual query

language. The GeoQA visual query language provides the user with the possibility to either text input a query or to build a query by using only icons. The IVQL user interface also provides the user with the possibility to either text input his location with the destination location or select them from list boxes. The user's current location could be automatically generated by any GPS, LBS or RADAR system in the near future.

The described visual query languages do not provide the non-expert mobile phone user with the possibility to exploit information about locations surrounding his current position, such as formulating simple queries about geographic area of interest and locations. None of the above languages sends the query to the server through SMS messages which is cost effective compared to WAP and GPRS services.

To find a solution to the above mentioned limitations, our proposed framework of IVQL can be effectively applied in various GIS information systems such as, fire department system, emergency departments, tourist information systems and management information systems. In the next section, we describe the visual user interface and the model used in IVQL related to the Location Based Services which are considered part of spatio-temporal systems.

3. IVQL User Interface

IVQL has a global user interface that consists of smiley icons. The use of smileys is intended for users from different countries with different cultures and languages. They enable the users around the world to access data and build queries easily and in a user-friendly way. Smiley icons are used as the basis to form and represent an international query language. Smiley icons are expected to have a high level of expressiveness. They express the same knowledge that can be represented by natural languages. They are text-free. Their rich set increases the accessibility to the user and his comprehension of the queries. Each iconic query expresses a sentence in natural language. Hence, it is equivalent to a sentence which means that it could replace it and be replaced by it without changing the information it represents. In terms of information content, the iconic query is more compact than any SQL query but still able to express the same content. Moreover, the language is attractive.

3.1 Themes, Objects, Locations and Instances

In IVQL, a theme represents a category, a type or a group of objects such as, entertainment, transportation and tourist. Each object represents a group of locations such as restaurant, hospital and hotel. A location is an instance of an object such as XXX restaurant, YYY hospital and ZZZ hotel. IVQL is extensible and can accommodate as many themes, objects and instances as needed.

Figure 1 shows the basic visual elements of the IVQL visual user interface.

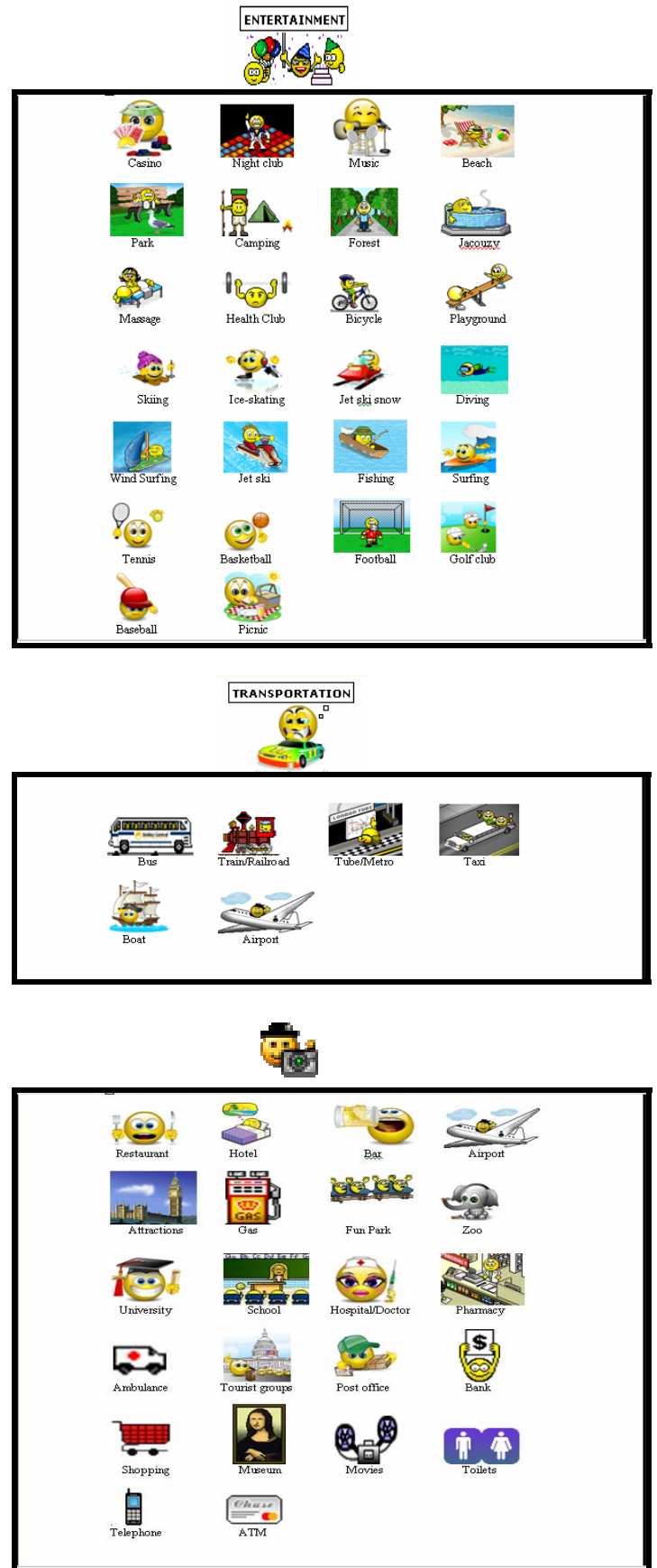


Figure 1: Basic elements of the IVQL visual user interface.

3.2 Visual Representations of Operators and Queries

The queries are formulated using the smiley icons that are expected to demonstrate a high level of expressiveness. The smiley icons are intended to be used by casual and non-expert users who have no prior knowledge of GIS information systems and query languages to databases. They can be easily read and understood which makes them the basis of an international language. No text is used in the language. Thus, it can be considered as a huge step towards the globalization of languages. The use of smiley icons makes the process of formulating a query easier. The operators that are used in the IVQL visual query language represent and depict actions and instructions selected by the user in order to find the shortest path between two locations, the nearest neighbor, or all locations of a certain type within a certain distance. The smiley icon in figure 2 depicts the command: *FIND*.



Figure 2: The smiley icon that visualizes the FIND command.

Figure 3 shows the icon that represents the command: *FIND THE SHORTEST PATH*.



Figure 3: The icon that represents the command: FIND THE SHORTEST PATH.

To find the nearest location of a certain type such as, find the nearest hospital or find the nearest restaurant, the user clicks the icon shown in figure 4.



Figure 4: The icon that represents the command FIND THE NEAREST.

To find all locations of a certain type such as, all restaurants or all hotels, etc. that are located within a certain distance from the user's current location, the user clicks the smiley icon shown in figure 5. A list box is displayed containing incrementing numerical values that are multiples of 100. This list box allows the user to choose and select the distance in meters within which he wants to find all locations of a certain type.



Figure 5: The smiley icon that represents the FIND WITHIN A DISTANCE option.

In case the user wants to build complex queries that include a combination of simple queries, the joining command *AND* may be used. The user formulates the first simple query, clicks the *AND* operator icon, then formulates the second simple query. If more than two simple queries need to be formulated, the user follows the same steps explained above and clicks the *AND* operator between every two simple queries. The maximum number of characters that an SMS message can contain is 160. So, the user can not build complex queries that are made up

of more than 160 characters. Figure 6 shows the depiction of the *AND* operator.



Figure 6: The smiley icon that depicts the AND operator.

3.3 Visual User Interface

IVQL provides the user with two major toolbars, one horizontal and one vertical. It has a middle area where objects are displayed and a query formulation area at the bottom of the interface. Each toolbar displays a set of expressive smiley icons that are easily understood by any user. The horizontal toolbar contains the smiley icons that represent proximity relations and operations that should be applied on icons that are displayed in the vertical toolbar. The vertical toolbar displays icons that represent themes and spatial object types. The middle area displays the theme objects on which the spatial operation is to be applied.

The query formulation is done first by selecting a smiley icon from the horizontal toolbar. The icon is automatically moved to the query formulation area which appears at the bottom of the interface. Second, the user selects a theme of interest from the vertical toolbar. A group of all objects that belong to the selected theme are displayed in the middle area of the interface. Finally, the user selects the object needed. The selected object is then moved to the query formulation area. The user interface of IVQL is shown in figure 7.

IVQL		ENTER SOURCE / YOUR LOCATION ADDRESS: _____	
FIND SHORTEST PATH		ENTER THE DESTINATION ADDRESS: _____	
ENTERTAINMENT			
TRANSPORTATION			
BACK			SELECT

Figure 7: The IVQL user interface.

The first two rows of the IVQL user interface are used to allow the user to text input the source and destination addresses. He can input his current location and input the address of the destination then query the shortest path between these two locations by selecting the "SHORTEST PATH" icon. In the first row, the location of the user could be automatically determined by a GPS, LBS (Location Based Services), RADAR, etc. in the near future. So, there will be no need to include the Text Input of the user's location. The same applies to the second row. With the possibility of having an open-line SMS in

the future, the user can query a list of hospitals and get back an output showing their locations on the map as well as a list box that includes the name and address of each hospital. In such a case, the user can select the hospital from the list box and use it as an input to the destination and then ask for the shortest path. Hence, there will be no need to Text Input the destination. To formulate complex queries, the user selects the “AND” icon and repeats the steps explained above. The length of complex queries is limited to the length of the SMS messages. Today, SMS messages can afford 160 characters only. IVQL does not allow the user to formulate queries that are longer than the SMS length. Figure 8 shows the IVQL user interface where the query FIND WITHIN 500M ALL RESTAURANTS is formulated.

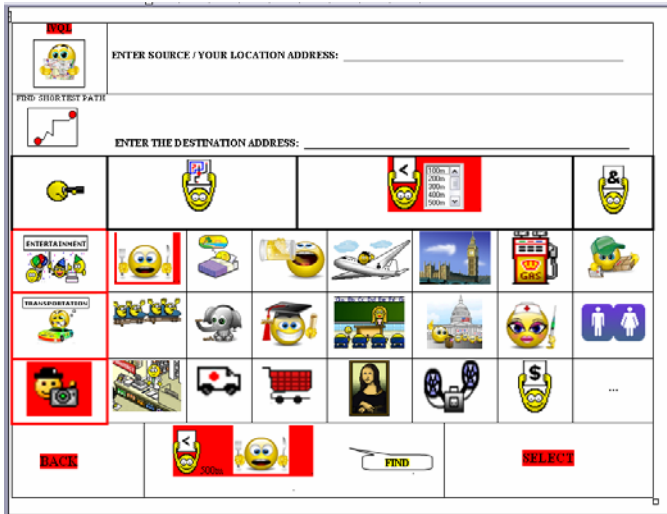


Figure 8: The smiley icons representing the tourist theme elements with a formulated visual query.

IVQL is an extensible visual query language. Each of the sets of icons that represent themes and objects can be extended to include as many icons as the user defines. If the area of a panel is not enough to display all the icons, the icon “more” “...” appears as the last icon in the list to inform the user that there are more icons that could be displayed. The user selects it and a new list of the next group of icons is displayed in the panel.

3.4 System Messages

Smiley icons are used to inform the user the status of their options and queries. If the user chooses a wrong icon, the smiley icon 🙄 is displayed. If the user chooses to execute a query, the smiley icon 🤖 is displayed. If the user formulates a correct query, the smiley icon 😊 is displayed. While the SMS message is being sent, the smiley icon 📶 is displayed.

4. Architecture of Query Processing

After building a query, the user clicks the FIND icon in order to instruct IVQL to process it. IVQL converts or translates the smiley icons visual query into normal text.

To do so, each visual icon has been assigned a TEXT name, such as



IVQL uses the icon name to formulate the Text Query. An example of the Text Query is: ‘nearest restaurant’. The Text query is displayed as an SMS message in the SMS environment of the mobile phone. The user chooses the ‘send’ option and inputs the phone number that is assigned to the GIS server by any provider, such as Alfa and Vodafone. The SMS message is received by the GIS server where the GIS Database is located. All SMS messages that are received by the GIS server are saved in a special file sequentially.

The query interpreter that is installed on the GIS server processes the SMS messages sequentially using the FIFO philosophy. Each message processed is first translated into a query that follows the Select-From-Where SQL query statement and then executed on the GIS database. The resulting map of the query is sent to the user as an MMS message to his mobile phone number. Figure 9 shows the structure of IVQL environment and the process through which a smiley icon query undertakes in order to be executed.

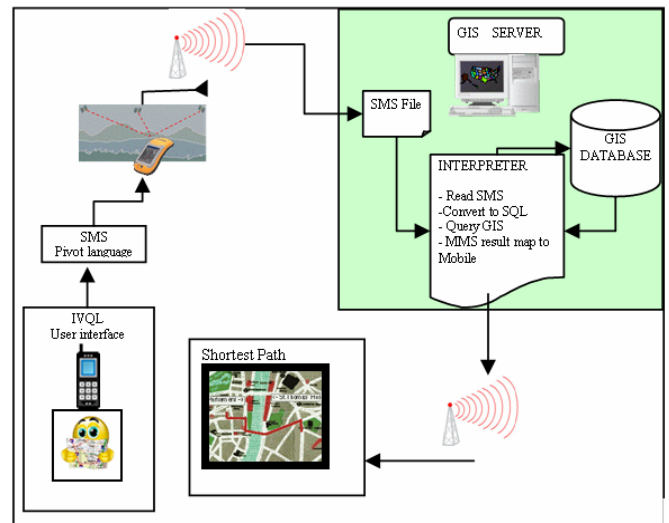


Figure 9: The architecture of the visual query processing.

The IVQL user interface is implemented for mobile phones by using the J2ME platform installed on the Symbian operating system. The IVQL query interpreter is implemented on the server computer to query the spatio-temporal database such as the ArcGIS. IVQL is expected to have a high level of cost effectiveness. Some of the queries that can be formulated are presented below. Figure 10 represents the query: Find the nearest golf club and display the shortest path between my location and the golf club found. This query is text-free.



Figure 10: Find nearest golf club and display the shortest path.

Figure 11 represents the query: Find all restaurants within 500m. This query expresses a sentence in natural language.



Figure 11: Find all restaurants within 500m.

Figure 12 represents the query: Find all universities and schools within 500m. This query is equivalent to a sentence in terms of information content.



Figure 12: Find all universities and schools within 500.

Figure 13 represents the query: Find the shortest path from my location to a specific address *destination address*. One icon only is used to summarize the sentence: find the shortest path.

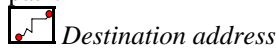


Figure 13: Find the shortest path between my location and a destination address.

Figure 14 represents the query: Find all bus stations within 500m and all train stations within 700m. The iconic query has an expressive power.



Figure 14: Find all bus stations within 500m and all train stations within 700m.

5. IVQL Model

The model that is suggested for IVQL has been developed based on the fact that its elements are spatial geographic features and taking into consideration the relations that exist between them. Accordingly, a spatial data model and a relations model have been designed.

5.1 Data Model

The data model of IVQL is based on the object-oriented approach. Its major components are feature objects, feature classes, attributes, inheritance and aggregation. Figure 15 shows the hierarchy of spatial classes that represent geographic spatial features.

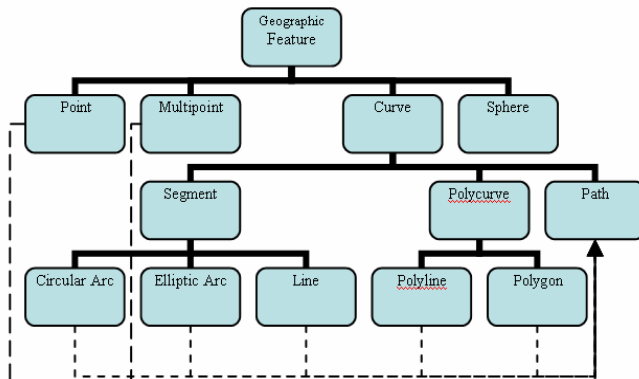


Figure 15: The hierarchy of geographic feature class.

The highest-level class *Geographic Feature* contains attributes and methods that are common to all other classes below it. The lowest-level classes such as *Circular Arc*, *Elliptic Arc*, *Line*, *Polyline* and *Polygon* are used to construct the *Path* class. Each of the lowest-level classes possesses an internal identifier (oid), a spatial location and a thematic identifier. The spatial location class contains the x-coordinates and y-coordinates of the geographical object. The thematic class contains the type of the object such as *line* and *polyline* whereas the thematic class contains the attributes of objects such as name and address. The *Path* class is used for network analysis. It contains various attributes and methods used to find the shortest path between two locations, the nearest neighbor/facility to a network location and all network elements within a given distance from a network location. The Geographic Feature Class hierarchy is extensible in the sense that the user can add classes and subclasses according to specific real-life applications. He can define new attributes and new relations to existing and new subclasses.

5.2 Relations Model

Our model consists of geometric relations that are used to define metric properties of spatial objects such as *length* & *distance*. Other major network relations can be derived from them such as the length of a path, minimum distance and within a distance. These relations play a major role in proximity and network analysis such as querying objects within a certain range, finding the nearest neighbor of a certain feature class and finding the shortest path between two defined objects. Figure 16 shows the three basic network relations that we defined namely *within a distance*, *nearest neighbor* and *shortest path*.



Figure 16: The models of the network relations.

5.3 Implementation of Relations

The above mentioned network relations are declared in the *Path* class. The skeleton of their definition and their results are explained.

The *IPointToEID* object finds the nearest element to a given point in a geometric network. It returns the element EID nearest to the input point and the physical point location that corresponds to the EID. Its major method is: *GetNearest*(in Point, out nearestEID, out Location).

The *ITraceFlowSolver* object contains a set of trace routines developed for path queries. Its major method is the *FindPath* which determines the optimal path between two or more locations: *FindPath*(in FlowMethod, out junctionEIDs, out edgeEIDs, out segmentCosts).

The *NetSolver* object finds all features of a certain type within a distance. Its filters specify the attribute values or ranges to use when tracing. Example, a trace might be done on Hotels within a range of 500m. Its major method is: *SetFilterRanges*(in Element Type, in fromValues, in toValues, out Locations).

6. Conclusion

In this paper we propose a visual approach to querying databases from mobile phones. The approach aims at designing and implementing an international visual query language that has an international user interface. We present IVQL, an international visual query language and we present its model. This model includes the definition of themes, objects and locations. It describes the operators that are used to represent functions. IVQL is aimed to be used by casual and non-expert users with no prior knowledge of query languages to databases. The use of smiley icons in the user interface is intended to simplify and facilitate the query formulation. Smiley icons are introduced to show how they represent themes, objects and operators. The formulation of queries is explained to show the expressiveness of the smiley icons.

The architecture of the project, its model, and its user interface are described. Some examples of visual queries are illustrated concerning a tourist information system. The tourist can formulate queries to find the shortest path between his current location and another location, the shortest path to the nearest object, and a display on a map of all the locations of a certain object within a certain distance. IVQL could be used in other information systems such as urban and environmental planning, transportation, utility mapping, Bio-informatics, Location Based Services, Management Information Systems, Customer Relationship Management, problem solving and many others.

In future work, IVQL will be implemented and tested after including in it dynamic query processing. Also, an evaluation will be conducted to assess the potential of icons to be used in visual query formulation.

References

[1] E. Clementini, P. De Felice, & P. Van Oosterom, A small set of formal topological relationships for end-user interaction. *Proc. Advances in Spatial Databases. 3rd International Symp.*, Springer-Verlag, Singapore, 1993, 277-295.

[2] S. Djordjevic-Kajan, Functions and contents of digital maps in process of building GIS in Serbia PTT. *Proc. in GIS/LIS*, Budapest, Hungary, 1996, 82-91.

[3] A. Marcus, Globalization of user-interface design for the Web. *Proc. 5th Conf. on Human Factors and the Web: The Future of Web Applications*, Gaithersburg, Maryland, USA, 1999.

[4] Y. Pang, Multimodal McDrive System. *Master Thesis*, Delft University of Technology, 2002.

[5] C. Bonhomme, C. Trepied, M.A. Aufaure, & R. Laurini, A visual language for querying spatio-temporal databases. *Proc. Advances in Geographic Information System*, Kansas, USA, 1999.

[6] C. Bonhomme, C. Trepied, & M.A. Aufaure, Metaphors for visual querying spatio-temporal databases. *Proc. 4th International Conf. on Visual Information Systems*. Springer Verlag, 2000, 140-153.

[7] C. Bonhomme & M.A. Aufaure, Mixing icons, geometric shapes and temporal axis to propose a visual tool for querying spatio-temporal databases. *Proc. Advanced Visual Interfaces*, Trento, Italy, 2002.

[8] ISO/IEC JTC1/SC21/N10441, *SQL Multimedia and Application Packages, Part3: Spatial*, p.172, 1996.

[9] R. Snodgrass, *The SQL2 temporal query language* (Kluwer Academic Publishers, 1995).

[10] J.F. Allen, Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11), 1983, 832-843.

[11] M.A. Aufaure, A high-level interface language for GIS. *Journal of Visual Languages and Computing Academic Press*, 6(2), 1995, 167-182.

[12] A. Lbath, M.A. Aufrane-Portier, & R. Laurini, Using a visual language for the design and query in GIS customization. *Proc. 2nd International Conf. on Visual Information Systems*, San Diego, USA, 1997, 197-204.

[13] R. Laurini, L. Paolino, M. Sebillio, G. Tortora, & G. Vitiello, Phenomena – A visual query language for continuous fields. *Proc. ACMGIS 2003 Association for Computing Machinery*, New Orleans, Louisiana, USA, 2003.

[14] L. Paolino & R. Laurini, Dealing with geographic continuous fields – the way to a visual GIS environment. *Proc. Advanced Visual Interfaces*, Gallipoly, Italy, 2004.

[15] Z. Stojanovic, S. Djordjevic-Kajan, & D. Stojanovic, Visual query and analysis tool of the object-relational GIS framework. *Proc. Advanced Visual Interfaces*, McLean, VA, USA, 2000.

[16] Z. Stojanovic, S. Djordjevic-Kajan, & D. Stojanovic, Query language in telecom GIS. *Proc. GISPLANET*, Lisbon, Portugal, 1998.

[17] L. Stoimenov, A. Mitrovic, D. Mitrovic, & S. Djordjevic-Kajan, Bridging objects and relations: a Mediator for an OO front-end to RDBMSs. *Information and software technology*, 41(2), 1999, 57-66.

[18] N.S. Murray, N.W. Paton, & C.A. Goble, Kaleidoquery: A visual query language for object databases. *Proc. Advanced Visual Interfaces, L'Aquila, Italy*, 1998.

[19] A.J. Morris, A.I. Abdelmoty, D.S. Tudhope, & B.A. ElGeresy, A filter-flow visual query language and interface for spatial databases. *GeoInformatica*, 8(2), 2004, 107-141.

[20] A.J. Morris, A.I. Abdelmoty, D.S. Tudhope, & B.A. ElGeresy, Design and implementation of a visual query language for Large spatial databases. *Proc. 6th International Conf. on Information Visualisation*, 2002.

[21] M.N. Smith, Enhancing database interface support for link analysis. *PhD Thesis*, Birkbeck College, University of London, 2004.

[22] M. Smith & P. King, A database interface for Link Analysis. *Journal of Database Management, Idea Group Publishing*, 16(1), 2005, 60-74.

[23] N. Andrienko & G. Andrienko, Interactive visual tools to explore spatio-temporal variation. *Proc. ACM Advanced Visual Interfaces*, Gallipoly, Italy, 2004.