

# AN EVALUATION OF A VISUAL QUERY LANGUAGE FOR INFORMATION SYSTEMS

Souheil Khaddaj, Haifa Elsidani Elariss

*Faculty of Computing, Information Systems and Mathematics, Kingston University, Penrhyn Road, London, UK*  
[K0437051@kingston.ac.uk](mailto:K0437051@kingston.ac.uk), [skhaddaj@kingston.ac.uk](mailto:skhaddaj@kingston.ac.uk)

Ramzi Haraty

*Division of Computer Science and Mathematics, Lebanese American University, Mme Curie Street, Beirut, Lebanon*  
[rharaty@lau.edu.lb](mailto:rharaty@lau.edu.lb)

**Keywords:** Intelligent user interface, Human computer interaction, Intelligent query language, Spatio-temporal databases, Location-Based Services, Information Systems, Smiley icon, Globalization, Mobile phone applications.

**Abstract:** In the recent years, many non-expert user applications have been developed to query Geographic Information Systems (GIS). GIS are also being used to browse and view data about space and time thus naming them spatio-temporal databases. Many approaches to querying spatio-temporal databases have been recently proposed. The Location Based Services (LBS) which are considered as part of the spatio-temporal field, concern the user who asks questions related to his current position over a mobile phone. Our research aims at designing and developing an Intelligent 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 using 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. A query is formulated by means of selecting the smiley icon that represents the operation to be executed then selecting the theme or object to be found. IVQL is an expandable language. It can include as many icons as needed whether they represent themes or objects. The visual query language and its user interface are explained. The IVQL model is described. The query formulation is illustrated using a sample GIS system for tourists. The IVQL user interface and query formulation can be applied to other fields such as Management Information Systems specifically in Customer Relationship Management (CRM), air traffic and bioinformatics. We then conclude about our future work.

## 1. INTRODUCTION

During the last decades, computer applications have been deployed to manage spatial data with Geographic Information Systems (GIS). 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. They 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 analyze spatial data, GIS support one of the query languages such as SQL. One problem is that SQL-like 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. 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 Intelligent 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 intelligent 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. The query formulation is hereby illustrated using a sample GIS system for tourists who need to query the database based on their current location (LBS) over the 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.

In section 2, we define the concept of Smiley Visual Icons and their use in human-computer interaction. We also explain their importance in the process of Globalization. Section 3 deals with the state of the art visual languages that have been proposed for querying spatial or spatio-temporal databases. In section 4, we describe the user interface and explain how smiley icons represent themes, objects and operators. Section 5 contains a description of the IVQL model and its environment. It explains the architecture of IVQL and how queries are processed after being formulated. Section 6 provides some examples of IVQL queries. Section 7 includes the conclusions drawn from this paper and discusses suggestions for future work.

## **2. THE VISUAL SMILEY ICONS IN GLOBALISATION**

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 [MARC99], 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 [PANG02] uses smiley icons to express the operator's feelings in a nonverbal interface for McDrive system developed for the McDonald chain. They smiley faces are displayed on the screen to show the right facial expression.

## **3. 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.

### **3.1. The *LVIS* Language**

LVIS [BONH99] is a visual query language for Geographic Information Systems (GIS) and for spatio-temporal databases. It uses Metaphors for Querying Visually Spatio-Temporal Databases [BONH00]. 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 [BONH02]. 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 [ISO196] for spatial operators and with TSQL [SNOD95] for temporal operators and uses the Allen temporal intervals [ALLE83]. 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 [AUFA95 and LBAT97] 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.

### 3.2. The *PHENOMENA!* Language

PHENOMENA, described in [LAUR03] and [PAOL04], is a visual query language for Continuous Fields developed for Geographic Information Systems (GIS) 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.

### 3.3. The GeoQA Language

GeoQA [STOJ00] is a visual query and analysis tool that has been developed 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 [DJOR96] and [STOJ98]. The visual graphical user interface (GUI) 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* [STOJ99], 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.

### **3.4. Other Visual Query 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 [MURR98] is a visual query language for object databases that depicts the query as a filter flow. The Filter-flow [MORR04] and [MORR02] 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) [SMIT04] and [SMIT05] 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 [ANDR04], 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.

### **3.5. The Present Visual Query Languages and their Contribution in the Current IVQL**

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 to find a solution to their limitations.

The LVIS visual query language that is described by [BONH02] 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 [LAUR03] and [PAOL04] 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 [STOJ00] 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 expressiveness 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.

Some limitations of some of the above state of the art visual query languages are hereby summarized:

- The visual languages can not be used by mobile phone users.
- None of the above visual languages sends the query to the server through SMS messages.
- The feature of finding the shortest path between two locations is not available.
- The concept of finding the nearest location of a certain type is not included.
- The concept of displaying the shortest path between the user's location and the location of the nearest object is not taken into consideration.
- The aspect of showing or displaying on a map all locations of a certain type within a certain distance is not considered.

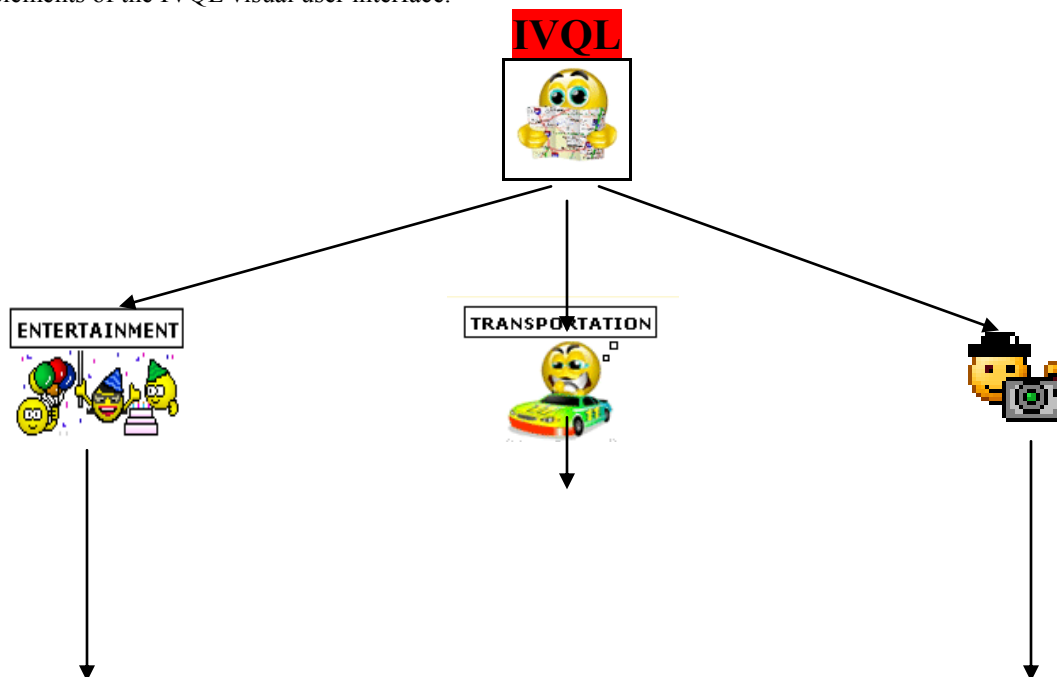
The described visual query languages do not provide the non-expert user with the possibility to exploit information about locations surrounding his current position, such as formulating simple queries about geographic area of interest and locations. To find a solution to the above mentioned limitations, we hereby propose a framework of a new Intelligent Visual Query Language (IVQL) that can be effectively applied in various GIS information systems such as, fire department system, tourist information systems and management information systems. The IVQL visual query language uses the smiley icons as the basis of its user friendly interface. We introduce the visual representation of the user interface. We describe the model used in IVQL based on the Location Based Services (LBS) systems which are considered as a part of spatio-temporal systems.

## 4. IVQL USER INTERFACE

IVQL has a global user interface that consists of smileys 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.

### 4.1. Themes, Objects 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 expandable and can accommodate as many themes, objects and instances as needed. Figure 1 shows the basic visual elements of the IVQL visual user interface.



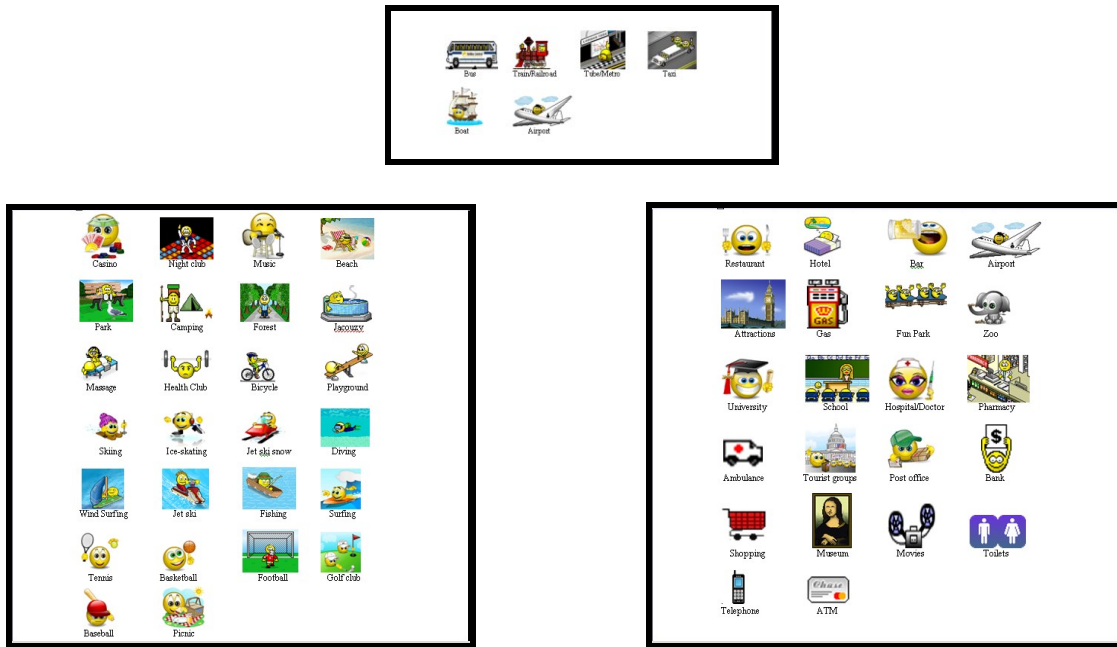


Figure 1: Basic elements of the IVQL visual user interface.

## 4.2. 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 with 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 or to find all locations of a certain type within a certain distance.

The smiley icon in figure **A** depicts the command: *FIND*. The user should select it before building his query.



Figure **A**: The smiley icon that visualizes the *FIND* command.

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

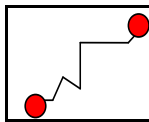


Figure **B**: The icon that represents the command: *FIND THE SHORTEST PATH*.

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 **C**.



Figure C: 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 has to click the smiley icon shown in figure D. 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 D: 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 160 characters. The IVQL visual user interface gives the user an error message if the complex queries exceed 160 characters. Figure E shows the depiction of the *AND* operator.



Figure E: The smiley icon that depicts the *AND* operator.

### 4.3 Visual User Interface

The user interface of the IVQL visual query language is shown in figure F.

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

Figure F: 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.

The third row consists of the icons that represent the operators. The user selects first which operation he wants to apply on the themes or objects. He selects the icon that represents the operation. This icon is moved to the query area which is located in the bottom line. The theme icons are displayed vertically in the left panel. The user selects the icon of the chosen theme. IVQL displays all the objects of the selected theme in the middle panel of the interface. The user selects the chosen object icon. This icon is moved to the query panel. To execute the query, the user has to select the “FIND” icon. When the user queries the nearest hospital, the system produces the output on a map including the shortest path. When the user queries all the hospitals within a certain distance such as 500m, the system might find many hospitals. It displays them all on the resulting map. 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 G shows the IVQL user interface where the query “FIND WITHIN 500M” ALL RESTAURANTS is formulated.

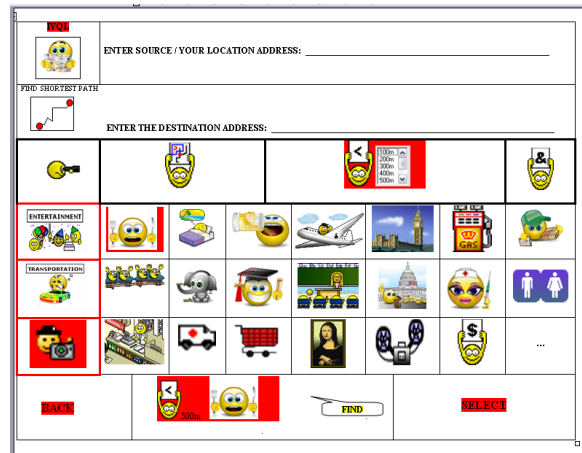






Figure G : The smiley icons representing the tourist theme elements with a formulated visual query.

IVQL is an expandable visual query language. Each of the sets of icons that represent themes and objects can be expanded 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.

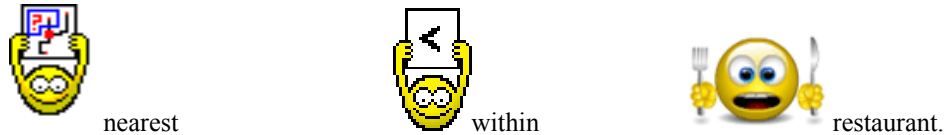
#### 4.4. Visual Representations of the System Messages

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.



## 5. MODEL, STRUCTURE AND ENVIRONMENT OF IVQL

After building a query, the user clicks the EXECUTE 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 Intelligent 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 X shows the structure of IVQL environment and the process through which a smiley icon query undertakes in order to be executed.

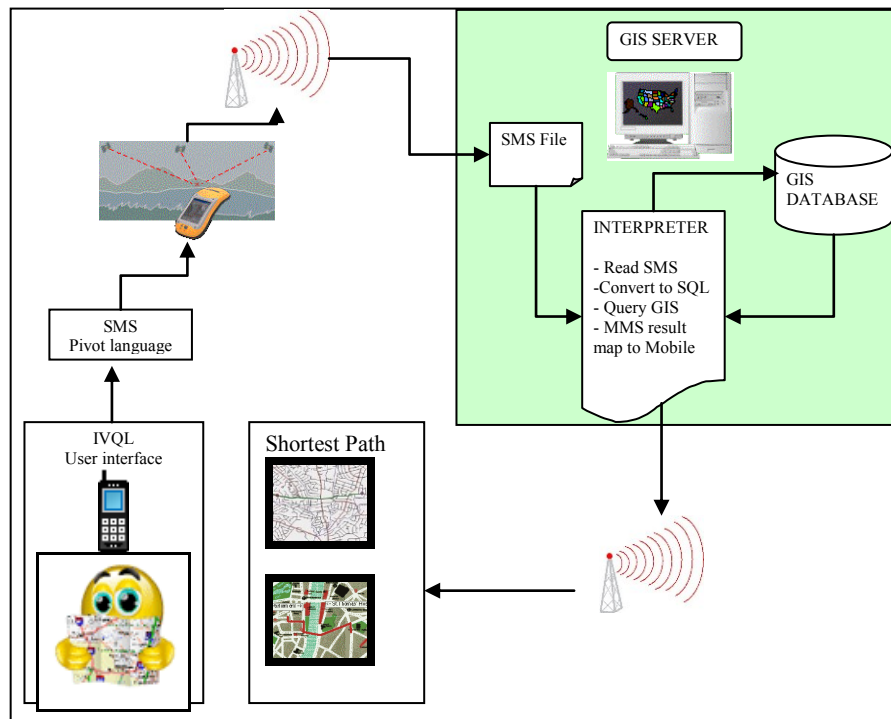


Figure X: The structure and framework of IVQL showing the flow of a query. (General architecture of model).

The IVQL user interface is implemented for mobile phones by using the J2ME platform installed on the Symbian operating system. The IVQL intelligent 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.

## 6. EXAMPLE QUERIES

Query 1: Find nearest golf club and display the shortest path



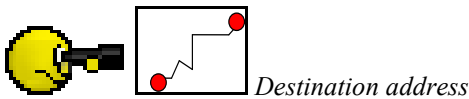
Query 2: Find all restaurants within 500m



Query 3: Find all universities and schools within 500m



Query 4: Find shortest path from my location to the *destination address*



Query 3: Find all bus stations within 500m and all train stations within 700m



## 7. CONCLUSION

In this paper we propose a visual approach to querying databases from mobile phones. The approach aims at designing and implementing an intelligent visual query language that has an international user interface. We present IVQL, an intelligent 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 aims at 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 and the user interface of the language 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, 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, Air traffic, Location Based Services (LBS), Management Information Systems (MIS), Customer Relationship Management (CRM), Supplier Relationship Management (SRM), Decision Support Systems (DSS), problem solving and many others. The evaluation of the intelligent visual query language will be considered in future work.

## REFERENCES

- [ALLE83] J.F. Allen. Maintaining knowledge about Temporal Intervals. *Communications of the ACM*, Vol. 26, n°11, pp 832-843.
- [ANDR04] N. Andrienko and G. Andrienko. Interactive Visual Tools to Explore Spatio-Temporal Variation. In *Proceedings of ACM AVI'04, Advanced Visual Interfaces*. May 25-28, 2004. Gallipoly (LE), Italy.
- [AUFA95] M.A. Aufaure. A High-Level Interface Language for GIS. *Journal of Visual Languages and Computing*. Academic Press, Vol. 6, n°2, pp 167-182.
- [BONH02] C. Bonhomme and M.A. Aufaure. Mixing Icons, Geometric Shapes and Temporal Axis to Propose a Visual Tool for Querying Spatio-Temporal Databases. *Advanced Visual Interfaces (AVI'2002)*, Trento, Italy.
- [BONH00] C. Bonhomme, C. Trepied and M.A. Aufaure. Metaphors for Visual Querying Spatio-Temporal Databases. In: *Advances in Visual Information Systems*, edited by R. Laurini, *Proceedings of the 4<sup>th</sup> International Conference on Visual Information Systems*. Springer Verlag, Lecture Notes in Computer Science, pp. 140-153.
- [BONH99] C. Bonhomme, C. Trepied, M.A. Aufaure and R. Laurini. A Visual Language for Querying Spatio-Temporal Databases. In *Proceedings of ACM GIS'99, 7<sup>th</sup> ACM Symposium on Advances in Geographic Information Systems*. November 5-6, 1999. Kansas City, USA.
- [DJOR96] S. Djordjevic-Kajan. Functions and contents of digital maps in process of building GIS in Serbia PTT. *Proceedings in GIS/LIS*, Budapest, Hungary, June 10-14, 1996, GIS/LIS Editions, 82-91.
- [ISO196] ISO/IEC JTC1/SC21/N10441, SQL Multimedia and Application Packages, Part3: Spatial, p.172.
- [LAUR03] R. Laurini, L. Paolino, M. Sebillio, G. Tortora, G. Vitiello. Phenomena – A Visual Query Language for Continuous Fields. In *Proceedings of ACMGIS 2003 on Association for Computing Machinery*, New Orleans, Louisiana, USA 2003.
- [LBAT97] A. Lbath, M.A. Aufrane-Portier, R. Laurini. Using a Visual Language for the Design and Query in GIS Customization. *VISUAL'97, 2<sup>nd</sup> International Conference on Visual Information Systems*, San Diego, USA, pp. 197-204.
- [MARC99] A. Marcus. Globalization of User-Interface Design for the Web. *Fifth Conference on Human Factors and the Web: The Future of Web Applications*. June 3, 1999. Gaithersburg, Maryland, USA.
- [MORR04] A. J. Morris, A. I. Abdelmoty, D. S. Tudhope, and B. A. ElGeresy. N. S. A Filter-flow Visual Query Language and Interface for Spatial Databases. *GeoInformatica* Vol. 8(2), p. 107-141.
- [MORR02] A. J. Morris, A. I. Abdelmoty, D. S. Tudhope, and B. A. ElGeresy. Design and Implementation of a Visual Query Language for Large Spatial Databases, *Proceedings of the Sixth International Conference on Information Visualisation (IV'02)*.
- [MURR98] N. S. Murray, N. W. Paton, and C. A. Goble. Kaleidoquery: A Visual Query Language for Object Databases. In *Proceedings of Advanced Visual Interfaces, L'Aquila, Italy, May 25-27, 1998*.
- [PANG02] Y. Pang. Multimodal McDrive System. Master Thesis. Delft University of Technology. Faculty of Information Technology and Systems. August 2002.
- [PAOL04] L. Paolino and R. Laurini. Dealing with Geographic Continuous Fields – the Way to a Visual GIS Environment. In *Proceedings of ACM AVI'04, Advanced Visual Interfaces*. May 25-28, 2004. Gallipoly (LE), Italy.
- [SMIT05] M. Smith, P. King. A Database Interface for Link Analysis. *Journal of Database Management*. Idea Group Publishing, Vol. 16, n°1, pp 60-74, Jan-March 2005.
- [SMIT04] M. N. Smith. Enhancing Database Interface Support for Link Analysis. PhD Thesis, Birkbeck College, University of London. Submitted November 2003.
- [SNOD95] R. Snodgrass (Ed.). *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers.
- [STOI99] L. Stoimenov, A. Mitrovic, D. Mitrovic and S. Djordjevic-Kajan. Bridging objects and relations: a *Mediator* for an OO front-end to RDBMSs. *Information and software technology* 41(2), pp.57-66.
- [STOJ00] Z. Stojanovic, Slobodanka Djordjevic-Kajan and D. Stojanovic. Visual Query and Analysis Tool of the Object-Relational GIS Framework. In *Proceedings of ACM AVI'00, Advanced Visual Interfaces*. CIKM, 2000. McLean, VA USA.
- [STOJ98] Z. Stojanovic, S. Djordjevic-Kajan and D. Stojanovic. Query language in Telecom GIS. *GISPlaNET'98 Conference*, 7-11.9.1998, Lisbon, Portugal.