

# A Survey on Visual Query Systems in the Web Era

Jorge Lloret-Gazo<sup>(✉)</sup>

Dpto. de Informática e Ingeniería de Sistemas, Facultad de Ciencias,  
Edificio de Matemáticas, Universidad de Zaragoza, 50009 Zaragoza, Spain  
jlloret@unizar.es

**Abstract.** As more and more collections of data are becoming available on the web to everyone, non expert users demand easy ways to retrieve data from these collections. One solution is the so called Visual Query Systems (VQS) where queries are represented visually and users do not have to understand query languages such as SQL or XQuery. In 1996, a paper by Catarci reviewed the Visual Query Systems available until that year. In this paper, we review VQSs from 1997 until now and try to determine whether they have been the solution for non expert users. The short answer is no because very few systems have in fact been used in real environments or as commercial tools. We have also gathered basic features of VQSs such as the visual representation adopted to present the reality of interest or the visual representation adopted to express queries.

## 1 Introduction

In recent years, and mainly because of the arrival of the web, more and more collections of data are becoming available to everyone in fields ranging from biology to economy or geography. One of the consequences of this fact is that end users, but not experts in Computer Science, demand easy ways to retrieve data from these collections.

Beginning in 1975 with Query By Example (QBE) [39] there have been many proposals in this direction, that is, to facilitate the work of the final user. In [8], the authors reviewed the so-called Visual Query Systems (VQS) from 1975 to 1996 defined as “systems for querying databases that use a visual representation to depict the domain of interest and express related requests”.

In this paper, we extend the review from 1997 to date, concentrating our efforts on visual queries to structured information, for example, queries to underlying relational or XML databases. We do not consider the typical search on semistructured documents such as web pages through search engines like Google. Although they are also a good solution for end-users, in this survey we do not take into account natural language interfaces for database query formulation.

The main goal of this survey is to answer the following question: To what extent have the VQS been the solution for novel users for querying databases?

---

The author would like to thank Rafael Bello for making the initial collection of papers for this review.

To answer this question, we have studied two features: web availability of and validation undergone by the systems. The first feature indicates that the system was designed to be reached easily by novel users simply by means of a web browser, without the burden of installation and with universal availability. The second feature indicates the widespread use of VQSs in practice. Thus, the more systems commercially available, the greater the extension reached by VQSs.

The short answer to the question is that, as far as we know, there is only one system commercially available and designed for the web: Polaris [34].

Moreover, we have included two basic features extracted from the paper [8]: the visual representation adopted to present the reality of interest and the visual representation adopted to express queries. With respect to web features, we have also considered relevant whether the prototype deals with data formatted for the web, that is, XML data or RDF data.

The rest of the paper is organized as follows. In Sect. 2 we state the method followed for elaborating the survey and we briefly describe the values of the relevant features included in the paper. Finally, in Sect. 3, we have drawn several conclusions about the VQSs.

## 2 Statement of the Method

A survey about a particular object must determine the relevant features of the object with respect to a particular purpose. Once the features have been determined, the next step is to find the possible values of these features. Finally, we have to determine the best combinations of the pairs (feature, value) for the particular purpose.

Usually, we can extract the relevant features and their possible values from published papers about the object, by assuming features in their entirety or by adapting them to new perspectives appearing after the papers have been published. Moreover, we can add features detected by ourselves which were not previously included in any paper.

The survey develops through several steps, which are usually interspersed. In the first step, a complete search of sources determines the candidate papers that deal with the object. In the second step, the relevant features of the object with respect to the particular purpose are determined.

Our object in this survey are the visual query systems with the purpose of facilitating querying databases to non expert in Computer Science users.

The survey [8] reviews up to 80 references from 1975 until 1996 used for querying traditional databases. For this survey, we have searched for papers related with VQS from 1997 to date and we have found 194 candidate papers. Next, we have discarded papers about query languages but without visual part (122) and papers about natural language query languages (8) because they deserve a separate survey. In the remaining 64 works, we have determined sets of ‘similar papers’ and we have discarded all but one paper in each set. A set of similar papers is composed of several papers built on different aspects of the same idea for a VQS. They also include preliminary versions of the VQS which were later

on subsumed by more complete journal publications. We have found 30 similar papers. So, we have discarded  $122 + 8 + 30$  papers, that is, 160 papers. As a result, the number of papers reviewed in this survey is 34.

As for relevant features, we have extracted the following from the survey of Catarci [8]: Visual representation adopted to present the reality of interest and visual representation adopted to express queries. The values of these features have been determined from the work [8] and from other papers, such as [11], where the faceted option appeared. For answering the question of this paper, we have added the following features: Web orientation and validation.

Let us explain briefly each of the features as well as their values.

## 2.1 Visual Representation Adopted to Present the Reality of Interest

This feature has been borrowed from the work of Catarci [8]. The reality of interest is modeled by a designer by means of a data metamodel as, for example, the entity/relationship metamodel or a graph data metamodel. As a result of the modelization process, a data model is obtained and it is presented to the user so that (s)he formulates queries on it.

The ways the data model is presented to the user are briefly described next and a more detailed explanation of some of the papers is given in [20].

**Diagram-based.** Data metamodels come with an associated typical representation for their elements. For example, in the entity/relationship metamodel, there are many representations available and one of them consists of drawing rectangles for the entity types, diamonds for the relationship types and ovals for the attributes. In the diagram-based option, the user has available a diagrammatical representation of the data model elaborated with the typical graphical representation for the elements of the metamodel.

**Icon-based.** Unlike the diagram-based approach, in this representation there are only iconic representations of some elements of the data model, but the user does not have available the complete data model. According to Catarci [8], ‘these VQS are mainly addressed to users who are not familiar with the concepts of data models and may find it difficult to interpret even an E-R diagram’. The aim of the icons is to represent a certain concept by means of its metaphorical power. The problem of these systems is how to construct them in such a way that they express a meaning which is understandable without ambiguity to the users.

**Form-based.** The typical forms of web pages serve for presenting the extensional database. This occurs in papers such as [34].

**Faceted.** The data are modeled as faceted classifications which organize a set of items into multiple, independent taxonomies. Each classification is known as a facet and the collection of classification data is faceted metadata. The specific category labels within a facet are facet values. For example, the set of items can

be architectural works. For these items, the facets are the architect, the location or the materials. The facet values for materials are stone, steel, etc.

**Unknown.** As the data model always exists, this option refers to the case where the data model is unknown. For example, the data model may be presented in a paper in textual form but there is no explanation about the way it is presented to the user. For example, paper [26] hides the database and tries to guess the paths for the query from the entities chosen by the user.

## 2.2 Visual Representation Adopted to Express the Queries

This feature has been borrowed from the work of Catarci [8] and we have adapted it to the object of the survey by adding the Faceted value.

The ways the queries are formulated are briefly described next and a more detailed explanation of some of the papers is given in [20].

**Diagram-based.** The diagram-based option means that the query is expressed on a diagrammatic representation of the data model.

**Icon-based.** The icon-based option includes two cases. In the first case, the system offers icons for representing the elements involved in the query. For building a query, the user drags and drops the appropriate icons into a canvas. The second case is the same as in [8], where the icons ‘denote both the entities of the real world and the available functions of the system’.

**Form-based.** Another way to facilitate the query is the form option where the user composes the query by completing options of different elements of a form. The drawback is that the query logic of the end-user does not always fit into a form.

**Faceted.** We have added as a new value ‘Faceted’ for describing a system which includes data and metadata in the same page. There, the user specifies the query by clicking on the appropriate links. We have found this situation only in one paper [11].

## 2.3 Web Orientation

For the web orientation, we have selected two features which are not mutually orthogonal. The first feature is whether the prototype is working on the web or has been conceived to be used in local mode. For the first situation, the value is *Available on the web* and this means that the final user can query the database by means of a prototype which is working on the web. The two values are: There is no web orientation and Available on the web. The second feature indicates whether the user can query data formatted for the web and the values are: Data not formatted for the web, Query XML data, Query RDF data. The values are not orthogonal. So, a paper can have the two values. This is the case, for example, of paper [7].

## 2.4 Validation

The validation of an idea can be done from several points of view. Regarding query systems, there are, at least, two dimensions: usability and performance.

For example, paper [10] focuses on performance and explains query rewriting techniques that improve the query evaluation performance so that the query execution time is reduced. However, in this paper we concentrate on the usability dimension, that is, the experiments made with users in order to determine the ease of use of the proposed prototype. For this feature, the list of values is: Only prototype, Prototype tested with users, Prototype tested in a real environment, Commercial tool.

Next, we describe briefly each value of this feature. The option *only prototype* means that a prototype has been built but no test has been made with users. The value *prototype tested with users* means that several experiments have been carried out in order to determine the usability of the prototype. The value *prototype tested in a real environment* means that it has been used for real tasks in a particular setting, for example in a department of a university. Finally, the option *commercial tool* means that the VQS has been fully implemented, offered to the public and is in real use in diverse installations.

**Table 1.** Visual query systems (1997–2003)

Cite	Database	Query	Web	Validation
[2]	Unknown	Icon	No	Only prototype
[31]	Unknown	Form	No	Tested with users
[5]	Diagram	Diagram	No	Only prototype
[9]	Diagram	Diagram	No	Tested with users
[21]	Diagram	Diagram	No	Only prototype
[32]	Diagram	Diagram	No	Only prototype
[24]	Icon	Icon	No	Only prototype
[3]	Diagram	Diagram	No	Tested with users
[12]	Unknown	Form	Query XML data	No
[13]	Icon	Icon	Available on the web	Tested with users
[28]	Diagram	Diagram	No	Only prototype
[33]	Icon	Icon	No	Tested with users
[27]	Diagram	Form	Query XML data	Only prototype
[25]	Icon	Icon	No	Tested with users
[23]	Unknown	Icon	No	Only prototype
[4]	Unknown	Icon	Query XML data	Only prototype
[26]	Form	Form	No	Tested with users
[1]	Unknown	Form	Query XML data	Only prototype
[14]	Unknown	Form	Query XML data	Only prototype

### 3 Discussion

The arrival of the web brought with it more facilities for users to query databases. As a consequence, users expect to access easily through the web databases situated anywhere in the world.

For expert users, one solution is to express queries in query languages such as SQL or XQuery. However, for novice users whose main concern is to extract data from the database but not the query languages themselves, learning SQL or XQuery is a huge task that is very far from their main concern.

One solution for novice users is to hide the complexity of query languages behind a visual scenery where it is supposed that the complexity is softened with the aid of visual metaphors. This is the idea of Visual Query Systems (VQS) defined in [8] as “systems for querying databases that use a visual representation to depict the domain of interest and express related requests”.

In this paper, we have reviewed basic features of Visual Query Systems, such as the representation of databases and the representation of queries. We have also considered the feature of accessing data formatted for the web. Finally, we have reviewed two features we consider relevant to determine whether the VQSS ease querying for novel users: web availability and validation. Next, we discuss the results for each of these features.

**Table 2.** Visual query systems (2004–2015)

Cite	Database	Query	Web	Validation
[7]	Diagram	Icon	Available on the web; Query XML data	Only prototype
[22]	Diagram	Diagram	Query XML data	Tested with users
[15]	Unknown	Icon	Query RDF data	No
[29]	Diagram	Diagram	No	Tested in a real environment
[16]	Unknown	Form	Query XML data	Tested in a real environment
[36]	Unknown	Form	No	Tested in a real environment
[30]	Diagram	Icon	Query XML data	Only prototype
[34]	Form	Form	Available on the web	Commercial tool
[11]	Diagram	Faceted	Available on the web	Only prototype
[17]	Unknown	Diagram	Query RDF data	Only prototype
[37]	Diagram	Icon	No	No
[6]	Diagram	Diagram	Available on the web	Tested with users
[18]	Unknown	Diagram	No	Tested with users
[35]	Diagram	Icon	No	Tested with users
[38]	Unknown	Icon	No	No
[10]	Diagram	Diagram	Query XML data	Only prototype

The majority of papers offer a diagrammatic representation of the database, only four papers an iconic one [2, 13, 25, 33] and one paper with form representation [34]. For several reasons, there are many papers whose database representation is unknown. For example, paper [26] hides the database and tries to guess the paths for the query from the entities chosen by the user.

With respect to the query representation, the distribution is more balanced between the icon (12 papers), the diagram (11 papers) and the form (8 papers) representation. A special form of query, the faceted one, appears only in one paper [11].

Regarding the data format, there are 9 papers [1, 4, 7, 10, 12, 16, 22, 27, 30] out of 34 which query XML data and only two papers which query RDF data [15, 17]. The rest of the papers do not query web data.

The rest of the features we have identified deal with the main question we have formulated in this paper, that is, to what extent have the VQS been the solution for novel users for querying databases?

For answering this question with respect to the web availability, we can distinguish two periods. From 1997 to 2003 (see Table 1), when the web usage was beginning to spread, there was only one paper oriented to the web [13]. This was very understandable because of the time needed for reorienting the research into the new web setting. In the period 2004 to 2015, only papers [6, 7, 11, 34] propose a web implementation (see Table 2). Although the number of web oriented papers in this period is greater than in the 1997–2003 period, the low number of papers indicates that web orientation has scarcely been taken into account.

For the validation feature, we have found a great number of papers which have only a prototype or have been tested with users in reduced experiments. Only three prototypes have been tested in real environments [16, 29, 36] and we have found only one commercial tool [34]. So, few papers go beyond testing the prototype with a few users.

As a conclusion of these two features, very few papers are web oriented and also very few papers offer a prototype which has been tested in a real environment. In fact, the combination of both features is only found in paper [34]. Then, although the visual query systems seem to be a great idea for easing the query process for novice users, the reality is that very few papers describe real implementations.

So, the answer to the main question of the paper is that, for the moment, VQSs have not been a widely accepted solution for novel users. From this observation a new, more general question arises: Is there any solution for easing the specification of queries?

If the answer is no, novel users have to learn by themselves query languages or they have to ask computer experts for the specification of queries. In the latter case, no new research would be needed in this field. If the answer is ‘we do not know’, then new research is required in order to find simple visual query languages which help novice users.

We strongly believe that the idea of VQSs is a good one and that the research should continue in this direction. Recent papers such as [19] also support the idea that a solution for naive users is not available but is necessary in this world

in which the use of databases is democratized. The paper proposes as a solution visual systems in which the user writes examples of queries and the system extracts and specifies the desired query in the corresponding query language.

## References

1. Abraham, R.: Foxq-xquery by forms. In: Proceedings of the 2003 IEEE Symposium on Human Centric Computing Languages and Environments, pp. 289–290. IEEE (2003)
2. Balkir, N.H., Sükan, E., Özsoyoglu, G., Özsoyoglu, Z.M.: Visual: a graphical icon-based query language. In: Su, S.Y.W. (ed.) ICDE, pp. 524–533. IEEE Computer Society (1996)
3. Benzi, F., Maio, D., Rizzi, S.: Visionary: a viewpoint-based visual language for querying relational databases. *J. Vis. Lang. Comput.* **10**, 117–145 (1999)
4. Berger, S., Bry, F., Schaffert, S., Wieser, C.: Xcerpt and visxcerpt: from pattern-based to visual querying of xml and semistructured data. In: Proceedings of the 29th International Conference on Very Large Data Bases, vol. 29, pp. 1053–1056. VLDB Endowment (2003)
5. Bloesch, A.C., Halpin, T.A.: Conceptual queries using conquer-ii. In: Embley, D.W. (ed.) ER 1997. LNCS, vol. 1331, pp. 113–126. Springer, Heidelberg (1997)
6. Borges, C.R., Macías, J.A.: Feasible database querying using a visual end-user approach. In: Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 187–192. ACM (2010)
7. Braga, D., Campi, A., Ceri, S.: XQBE (xquery by example): a visual interface to the standard xml query language. *ACM Trans. Database Syst.* **30**(2), 398–443 (2005)
8. Catarci, T., Costabile, M.F., Levialdi, S., Batini, C.: Visual query systems for databases: a survey. *J. Vis. Lang. Comput.* **8**, 215–260 (1997)
9. Catarci, T., Santucci, G., Cardiff, J.: Graphical interaction with heterogeneous databases. *VLDB J.* **6**, 97–120 (1997)
10. Choi, R.H., Wong, R.K.: VXQ: A visual query language for XML data. *Inf. Syst. Front.* **17**(4), 961–981 (2015)
11. Clarkson, E., Navathe, S.B., Foley, J.D.: Generalized formal models for faceted user interfaces. In: JCDL, pp. 125–134 (2009)
12. Cohen, S., Kanza, Y., Kogan, Y.A., Nutt, W., Sagiv, Y., Serebrenik, A.: Equix easy querying in xml databases. In: WebDB (Informal Proceedings), pp. 43–48 (1999)
13. Cruz, I.F., Leveille, P.S.: As you like it: personalized database visualization using a visual language. *J. Vis. Lang. Comput.* **12**, 525–549 (2001)
14. Erwig, M.: Xing: a visual xml query language. *J. Vis. Lang. Comput.* **14**(1), 5–45 (2003)
15. Harth, A., Kruk, S.R., Decker, S.: Graphical representation of rdf queries. In: WWW, pp. 859–860 (2006)
16. Jagadish, H.V., Chapman, A., Elkiss, A., Jayapandian, M., Li, Y., Nandi, A., Cong, Y.: Making database systems usable. In: SIGMOD Conference, pp. 13–24 (2007)
17. Jarrar, M., Dikaiakos, M.D.: Querying the data web: the mashql approach. *IEEE Internet Comput.* **14**, 58–67 (2010)
18. Jin, C., Bhowmick, S.S., Xiao, X., Cheng, J., Choi, B.: Gblender: towards blending visual query formulation and query processing in graph databases. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 111–122. ACM (2010)



19. Li, F., Jagadish, H.V.: Usability, databases, and hci. *IEEE Data. Eng. Bull.* **35**(3), 37–45 (2012)
20. Lloret-Gazo, J.: A survey on visual query systems in the web era (extended version). <http://www.unizar.es/ccia/articulos/VQSCompleto.pdf>
21. Madurapperuma, A.P., Gray, W.A., Fiddian, N.J.: A visual query interface for a customisable schema visualisation system. In: *IDEAS*, pp. 23–32 (1997)
22. Meuss, H., Schulz, K.U., Weigel, F., Leonardi, S., Bry, F.: Visual exploration and retrieval of xml document collections with the generic system x2. *Int. J. Digit. Libr.* **5**(1), 3–17 (2005)
23. Morris, A.J., Abdelmoty, A.I., El-Geresy, B.A.: A visual query language for large spatial databases. In: *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI 2002*, pp. 359–360. ACM, New York (2002)
24. Murray, N., Paton, N.W., Goble, C.A.: Kaleidoquery: a visual query language for object databases. In: *AVI*, pp. 247–257 (1998)
25. Narayanan, A., Shaman, T.: Iconic sql: ractical issues in the querying of databases through structured iconic expressions. *J. Vis. Lang. Comput.* **13**, 623–647 (2002)
26. Owei, V.: Development of a conceptual query language: adopting the user-centered methodology. *Comput. J.* **46**(6), 602–624 (2003)
27. Papakonstantinou, Y., Petropoulos, M., Vassalos, V.: Qursed: querying and reporting semistructured data. In: *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pp. 192–203. ACM (2002)
28. Poulouvasilis, A., Hild, S.G.: Hyperlog: a graph-based system for database browsing, querying, and update. *IEEE Trans. Knowl. Data Eng.* **13**(2), 316–333 (2001)
29. Rontu, M., Korhonen, A., Malmi, L.: System for enhanced exploration and querying. In: *AVI*, pp. 508–511 (2006)
30. Sans, V., Laurent, D.: Ifox: interface for ordered xquery an algebraic oriented tool for ordered xquery visualization. In: *SAC*, pp. 1252–1257 (2008)
31. Sengupta, A., Dillon, A.: Query by templates: a generalized approach for visual query formulation for text dominated databases. In: *Proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries, ADL 1997*, pp. 36–47. IEEE (1997)
32. Shin, D.-G., Grajewski, W., Chu, L.-Y.: An epistemological display query interface. In: *AVI*, pp. 286–288 (1998)
33. Silva, S.F., Catarci, T., Schiel, U.: Formalizing visual interaction with historical databases. *Inf. Syst.* **27**, 487–521 (2002)
34. Stolte, C., Tang, D., Hanrahan, P.: Polaris: a system for query, analysis, and visualization of multidimensional databases. *Commun. ACM* **51**(11), 75–84 (2008)
35. Störrle, H.: Vmql: a visual language for ad-hoc model querying. *J. Vis. Lang. Comput.* **22**(1), 3–29 (2011)
36. Terwilliger, J.F., Delcambre, L.M.L., Logan, J.: Querying through a user interface. *Data Knowl. Eng.* **63**, 774–794 (2007)
37. Varga, V., Sacarea, C., Takacs, A.: Conceptual graphs based representation and querying of databases. In: *2010 IEEE International Conference on Automation Quality and Testing Robotics (AQTR)*, vol. 3, pp. 1–6. IEEE (2010)
38. Zongda, W., Guandong, X., Zhang, Y., Cao, Z., Li, G., Zhiwen, H.: Gmql: a graphical multimedia query language. *Knowl.-Based Syst.* **26**, 135–143 (2012)
39. Zloof, M.M.: Query by example. In: *Proceedings of the May 19–22, 1975, National Computer Conference and Exposition*, pp. 431–438. ACM (1975)