What Happened When Database Researchers Met Usability

Tiziana Catarci
Dipartimento di Informatica e Sistemistica
Università di Roma “La Sapienza”
Roma, Italy

catarci@dis.uniroma1.it
http://www.dis.uniroma1.it/~catarci/
Outline

• The initial problem
• QBD*
• Usability
• User Trials
• More General Approaches
• Open Problems
The initial problem…

Information = “data which serve a purpose”

How to find it?

Is it the right one?

How to manipulate it?
Query Languages

A **Query Language** is a set of formally defined operators that, suitably composed, allow one to express requests to a database. A Query Language works on a **database** defined in terms of a **data model**.

**Basic Components**

- Data Model
- Data Representation
- Query Model
- Query Representation
List the names of all the persons, who made a transaction on May 25th

SQL QUERY

```
SELECT Name
FROM Customer
WHERE Type='Person' AND
Customer# IN
(SELECT Customer#
FROM Account
WHERE A/C# IN
(SELECT A/C#
FROM Transaction
WHERE Date='May 25th'))
```

QBE QUERY

```
Customer# Address Name Type Sex Capital
X F Person
```

GRAPHICAL QUERY

```
CUSTOMER
ADDRESS
NAME
PERSON
SEX
ORGANIZATION
ACCOUNT
A/C# BALANCE
CAPITAL
TRANSACTION
TRANSACTION#
DATE = 'May 25th'
AMOUNT
```

ICONIC QUERY

```
PERSON
TRANSACTION
25
NAME
May
```
QBD*

QBD* is based on four main ideas:

• to graphically represent the database (ER schema)
• to define a textual query language on the ER model
  – relationally complete
  – including a significant class of recursive queries (transitive closure)
• to design a fully graphical user interface
  – each instruction of the textual language is expressed through a set of
    visual constructs, allowing the user to graphically formulate the query on
    the ER schema
• to enrich the system with several additional facilities to simplify the
  user interaction
Navigation within QBD*

a)

b)

2.2 a

2.2 b
VQSs

VISUAL LANGUAGES

LANGUAGES SUPPORTING VISUAL INTERACTION
VISUAL PROGRAMMING LANGUAGES
VISUAL INFORMATION PROCESSING LANGUAGES
ICONIC VISUAL INFORMATION PROCESSING LANGUAGES

VISUAL QUERY LANGUAGES

Chang's Classification
VQS Taxonomic Criteria

Visual Representation
  – for expressing the query
  – for visualizing the result

Expressive Power

Query Strategies
  - Understanding
  - Query Formulation
  - Testing
Usability

"The extent to which a product can be used with efficiency, effectiveness and satisfaction by specific users to achieve specific goals in specific environments" (N. Bevan & M. Macleod).

Usability has multiple components (J. Nielsen):

- **Learnability**: system easy to learn so that the user rapidly starts getting some work done.
- **Efficiency**: system efficient to use, so that a high level of productivity is possible.
- **Memorability**: system easy to remember, so to facilitate the return to it after a while.
- **Errors**: low error rate.
- **Satisfaction**: system pleasant to use.

System usability can be achieved through a human-centred system design.
Human-centred Design

- Making systems more human-centred has substantial economic and social benefits.
- More usable systems meet user and organisational needs better and:
  - are easier to understand and use, thus reducing training and support costs;
  - reduce discomfort, stress and improve user satisfaction;
  - improve the productivity of users and the operational efficiency of organisations;
  - improve product quality, aesthetics and impact and can provide competitive advantage
- The complete benefits of human-centred design come from calculating the total life-cycle costs of the product including conception, development, implementation, support, use and maintenance.
General Principles

Human-centred design is characterised by:

● the active involvement of users and a clear understanding of user and task requirements

● an appropriate allocation of function between user and system

● the iteration of design solutions

● multi-disciplinary design

It is oriented towards achieving the “quality in use” for interactive computer-based systems
General Principles

- Understand and specify the context of use
- Specify the user and organisational requirements
- Produce design solutions
- Evaluate designs against requirements
Usability Evaluation

What to Evaluate:

- **Effectiveness**: it refers to the extent to which the intended goals of the system can be achieved;
- **Efficiency**: it is the time, the money, and the mental effort spent to achieve these goals;
- **Satisfaction**: it depends on how comfortable the users feel using the system.
For VQSs...

- The effectiveness can be evaluated by relating the goals or subgoals of using the system to the accuracy and completeness that these goals can be achieved with.
- In the case of VQSs the main goal is to extract information from the database by performing queries, and the accuracy in achieving such a goal is generally measured in terms of the accuracy of query completion.
- Measures of efficiency relate the level of effectiveness achieved at the expense of various resources.
- The user's efficiency is most frequently measured in terms of the time spent to complete a query.
How to Evaluate Usability

- Empirical studies
- Analytical studies
- Information foraging theory
- FADIVA benchmarks
User-based Methods

- User testing with real users through controlled experiments in usability labs.
- Observation
- Questionnaires and Interviews
- Focus Groups
- Logging actual use
- User feedback
- Performance measures
- Thinking aloud
Evaluator-based Methods

- Heuristic evaluation (systematic inspection of a user interface design for usability)
- Pluralistic usability walkthrough
## Summary of Usability Methods
*(from J. Nielsen, Usability Engineering, 1993)*

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Lifecycle Stage</th>
<th>Users Needed</th>
<th>Main Advantages</th>
<th>Main Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic evaluation</td>
<td>Early design, &quot;inner cycle&quot; of iterative design</td>
<td>None</td>
<td>Finds individual usability problems. Can address expert user issues</td>
<td>Does not involve real users, so does not find &quot;surprises&quot; related to their needs.</td>
</tr>
<tr>
<td>Performance measures</td>
<td>Competitive analysis, final testing</td>
<td>At least 10</td>
<td>Hard numbers. Results easy to compare</td>
<td>Does not find individual usability problems.</td>
</tr>
<tr>
<td>Thinking aloud</td>
<td>Iterative design, formative evaluation</td>
<td>3-5</td>
<td>Pinpoints user misconceptions. Cheap test.</td>
<td>Unnatural for users. Hard for expert users to verbalize.</td>
</tr>
<tr>
<td>Observation</td>
<td>Task analysis, follow-up studies</td>
<td>3 or more</td>
<td>Ecological validity; reveals users' real tasks. Suggests functions and features.</td>
<td>Appointments hard to set up. No experimenter control.</td>
</tr>
<tr>
<td>Questionnaires</td>
<td>Task analysis, follow-up studies</td>
<td>At least 30</td>
<td>Finds subjective user preferences. Easy to repeat.</td>
<td>Pilot work needed (to prevent misunderstanding s).</td>
</tr>
<tr>
<td>Interviews</td>
<td>Task analysis</td>
<td>5</td>
<td>Flexible, in-depth attitude and experience probing.</td>
<td>Time consuming. Hard to analyze and compare.</td>
</tr>
<tr>
<td>Focus groups</td>
<td>Task analysis, user involvement</td>
<td>6-9 per group</td>
<td>Spontaneous reactions and group dynamics.</td>
<td>Hard to analyze. Low validity.</td>
</tr>
<tr>
<td>Logging use</td>
<td>Final testing, follow-up studies</td>
<td>At least 20</td>
<td>Finds highly used (or unused) features. Can run continuously</td>
<td>Analysis programs needed for huge mass of data. Violation of users' privacy.</td>
</tr>
<tr>
<td>User feedback</td>
<td>Follow-up studies</td>
<td>Hundreds</td>
<td>Tracks changes in user requirements and views</td>
<td>Special organization needed to handle replies.</td>
</tr>
</tbody>
</table>
QBD* Trials

■ The sample users: undergraduate students, secretaries and professionals (104 people). They were classified in three categories:
  – Non expert users (58 people)
  – Intermediate users (30 people)
  – Expert users (16 people)

■ The query classes:
  ■ Semantic distance of the path
    – CLOSE
    – FAR
    – VERY FAR
  ■ Cycles in the path
    – CYCLIC
    – UNCYCLIC
  ■ Close-Uncyclic (CU) --> all users
  ■ Far-Uncyclic (FU) --> all users
  ■ VeryFar-Uncyclic --> intermediate & expert
  ■ Far-Cyclic --> expert only
Results (Accuracy)

- Naive
- Intermediate
- Expert
Results (Time)

The graph shows the results for different levels of expertise (Naive, Intermediate, Expert) using two query languages, SQL and QBQ. The x-axis represents different query types (CU, FU, VU, VFU, FC), and the y-axis represents time. The bars indicate the performance of SQL (light blue) and QBQ (purple) for each query type across the different expertise levels.
Discussion

The hypothesis that a graphical query language would perform better than SQL was confirmed by the experiment results.

QBD* shows good results both on the accuracy and on the response time with respect to the semantic distance of the paths involved in the query expression. On the other hand, whenever queries contain cycles, expressing them using QBD* is slightly less accurate than in SQL.
Discussion (2)

- The reason why QBD* users perform better when expressing queries characterized by a high semantic distance could very well be that paths in QBD* are displayed on the screen as part of the database schema, and s/he has only to follow them.

- Considering the queries containing cycles, we can say that whenever a query involves two attributes belonging to the same concept the QBD* users get a bit confused, perhaps because of the duplication on the screen of such attributes (distinguished by different numbers)
More Experiments

QBD vs QBI: Need of a multiparadigmatic approach!
Multiparadigmatic Approaches

• VENUS
• DARE
VENUS Environment

OBJECTIVE

Adaptive visual interface, offering the possibility of switching between different visual representations.

THROUGH

Common underlying model, i.e. Graph Model, powerful enough to represent the databases expressed in the most common data models.

Graph Model Data Bases queried by means of a multiparadigmatic interface.

Semantics of the query operations uniformly defined in terms of the Graphical Primitives.

Definition of the "atomic query", i.e. the minimal portion of a query that can be transferred from one interaction paradigm to another and be processed by the system.

System support to the user by suggesting the more appropriate interaction modality as well as the visual representation, with reference to the user model.
The user is provided with several visual representations and interaction mechanisms, and can consistently switch among them.

Databases expressed in different models are treated in a homogeneous way.

The visual representation most appropriate for her/his skill and needs is proposed to the user.
The final goal

• To provide general frameworks for automatic (or semi-automatic) generation of correct, complete, and effective visualizations (given any data, users, tasks)
Visualization which is neither correct nor complete

<table>
<thead>
<tr>
<th>Town</th>
<th>People #</th>
<th>Position</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rome</td>
<td>4,000,000</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Milan</td>
<td>1,800,000</td>
<td>North</td>
<td>600</td>
</tr>
<tr>
<td>Naples</td>
<td>1,500,000</td>
<td>South-East</td>
<td>200</td>
</tr>
<tr>
<td>Pisa</td>
<td>150,000</td>
<td>North-West</td>
<td>350</td>
</tr>
<tr>
<td>Pescara</td>
<td>200,000</td>
<td>East</td>
<td>220</td>
</tr>
</tbody>
</table>
Complete but not correct

- Pisa
- Rome
- Naples
- Milan
- Pescara

Distances:
- 350 Km
- 220 Km
- 600 Km

People #:
- >2,000,000
- From 1,000,000 to 2,000,000
- From 500,000 to 1,000,000
- <500,000
Correct & complete

- Milan
  - Pisa: 600 Km
  - Rome: 350 Km
- Rome
  - Pescara: 220 Km
  - Naples: 200 Km
- People #:
DARE

- General theory for establishing the adequacy of a visual representation, once specified the database characteristics
- DARE (Drawing Adequate REpresentations) system, which implements such a theory and works in two modalities, namely:
  - *Representation Check* - checking the adequacy of visual representations proposed by the user. The adequacy is expressed in terms of *completeness* and *correctness* of a visual representation wrt a database. A visual representation is complete if the user can perceive from it all the database information content, and it is correct if s/he can perceive only this.
  - *Representation Generation* - automatically associating to any database the most effective visual representation. Such a visual representation has to be not only adequate, but it has also to convey some database features specified by the designer (e.g., that some concepts are the most relevant).
Knowledge Base

• The DARE system is based on a knowledge base containing different kinds of rules:
  – *Visual rules*: characterize the different kinds of visual symbols.
  – *Data rules*: specify the characteristics of the data model, the database schema, and the database instances.
  – *Mapping rules*: specify the link between data and visual elements.
  – *Perceptual rules*: tell us how the user perceives a visual symbol, relationships between symbols, and which is the perceptual effect of relevant visual attributes such as colour, texture, etc.

• The knowledge base contains predicates concerning different levels of the knowledge used by the system (e.g., instance, schema, and metaschema levels).

• It is partitioned into two layers: the predicate layer and the object layer.

• The predicates of the knowledge base are classified into three categories:
  – the predicates concerning those aspects related to the data;
  – the predicates concerning the visual symbols;
  – the predicates concerning the links between the data and the visual symbols.
Example

DataType(class).
DataType(binrel).
IsOfType (x, binrel) --> HasArg(x,2).
IsOfType (x, class) --> HasArg(x,1).

IsOfType(person, class).
IsOfType(city, class).
IsOfType(lives, binrel).
HasAttr(person, age).
HasAttr(person, weight).
HasAttr(city, name).

Rep(triangle, class).
Rep(thickline, binrel).
Rep(redtriangle, person).
Rep(bluetriangle, city).
Rep(texture, income).
Rep(size, age).

GoodVis(triangle, class).
GoodVis(size, age).
GoodVis(size, weight).
Example (cont.d)

- Redtriangle and bluetriangle have to be defined as visual sub-categories of triangle.

- The representation is not complete, since lives has not been associated with a visual category: add the assertion Rep(thickline,lives).

- No perceptual rules
First implementation

• Simple (but important) problem: handling a large amount of data representing the result of a query (on a data model or on the web)

• Simple (but frequent) query: selection (or a conjunctive query in which the user is not allowed to change the join condition to restrict the final result)

• Few attributes involved in the selection

• Few used visual symbols and attributes
Medical application

• Data about hemodialyses from Nefrology Center of Vigevano hospital
• Data collected to evaluate the success trend in hemodialyses
• Three relational tables containing data about summaries of hemodialyses, detailed measures taken during hemodialyses, overall hemodialysis evaluation
Concise data about hemodialyses
Overall hemodialysis evaluation
Open Problems & Conclusions

• There are not many serious studies on the fundamentals of the user-DB interaction

• Several new systems, intended to be friendly for the final user, are:
  – just ideas, neither implemented nor tested
  – variations on well known approaches, such as using diagrams, clicking, pointing, etc.
  – one-to-one graphical mappings with well-known formal languages (no user-oriented interaction techniques, simply boxes and arrows instead of relations and keys)

• The concept of usability, and the contribution of psychologists, ergonomists, human factor experts are still largely ignored

• There are a few formal studies on the new approaches.
Open Problems & Conclusions

- Need to address new issues coming from new applications: data mining, data warehousing, WWW, electronic commerce, etc.
- Huge amount of data, need of finding new visualizations
- Improvements in user-modeling, adaptation, personalization, usability
- Handling of different types of knowledge
- Handling of several data types
- Extension of the query language expressive power

Many open issues for research!