We Don’t Need No Education: From Building for Coders to Building for Users

Sourav S Bhowmick
Nanyang Technological Univ
Singapore
World of Graphs

Security analyst

biologist

Techies

Financial analyst

ecologist

journalist

Pharmacist/chemist

social scientist
DB Approach to Graph Interaction

```prefix wp: <http://vocabularies.wikipathways.org/wp#>
prefix dcterms: <http://purl.org/dc/terms/>
prefix foaf: <http://xmlns.com/foaf/0.1/>

select (str(?organismName) as ?organism) ?page ?gene1 ?gene2 ?interaction where {
  ?gene1 a wp:GeneProduct ;
  ?gene2 a wp:GeneProduct ;
  ?interaction wp:source ?gene1 ;
  wp:target ?gene2 ;
  a wp:Conversion ;
  dcterms:isPartOf ?pathway .
  wp:organismName ?organismName .
  FILTER (?gene1 != ?gene2)
} ORDER BY ASC(?organism)
```
# QL – Command Language Style of Interaction

<table>
<thead>
<tr>
<th><strong>Pros</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Flexible</td>
</tr>
<tr>
<td>• Appeals to “power” users</td>
</tr>
<tr>
<td>• Allows convenient creation of user-defined macros</td>
</tr>
<tr>
<td>• Domain of expert frequent users</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Cons</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Error rates are high</td>
</tr>
<tr>
<td>• Substantial training is necessary</td>
</tr>
<tr>
<td>• Retention may be poor</td>
</tr>
</tbody>
</table>
Are We Nice to End Users?

World’s First “Self-Driving” Database
- No Human Labor
- Half the Cost
- No Human Error
- 100x More Reliable

Oracle Autonomous Database
Shneiderman’s View

### Evolution of Users

In the first decades of computer-software development, technically oriented programmers designed text editors, programming languages, and applications for themselves and their peers. The substantial experience and motivation of these users meant that complex interfaces were accepted and even appreciated. **Now, the user population for mobile devices, instant messaging, e-business, and digital libraries is so vastly different from the original that programmers’ intuitions may be inappropriate. Current users are not dedicated to the technology; their background is more tied to their work needs and the work tasks they perform.**

---

Ben Shneiderman & Catherine Plaisant  
*Designing the User Interface (5th Ed, 2010)*
Thirty years of research on query languages can be summarized by: we have moved from SQL to XQuery. At best we have moved from one declarative language to a second declarative language with roughly the same level of expressiveness. It has been well documented that end users will not learn SQL; rather SQL is notation for professional programmers.

The Lowell Database Research Self-Assessment, Communication of the ACM (May 2005)
Today’s data consumers may not know how to formulate a query at all—e.g., a journalist who wants to “find the average temperature of all cities with population exceeding 100,000 in Florida” over a structured data set. Our community’s challenge is to make it possible for such people to get their answers themselves, directly. This requires new query interfaces, e.g., interfaces based on multitouch, not just console-based SQL interfaces. We need interfaces that combine visualization, querying, and navigation.

The Beckman Report on Database Research, Communication of the ACM (Feb 2016)
What Do Practitioners Think?

53 practitioners (36 from IT)

The overwhelming majority of the emails and issues were routine engineering tasks, such as users asking how to write a query...

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Total</th>
<th>R</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability (i.e., software that can process larger graphs)</td>
<td>45</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>Visualization</td>
<td>39</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td>Query Languages / Programming APIs</td>
<td>39</td>
<td>18</td>
<td>21</td>
</tr>
<tr>
<td>Faster graph or machine learning algorithms</td>
<td>35</td>
<td>19</td>
<td>16</td>
</tr>
<tr>
<td>Usability (i.e., easier to deploy, configure, and use)</td>
<td>25</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Benchmarks</td>
<td>22</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Extract &amp; Transform</td>
<td>20</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>More general purpose graph software (e.g., that can process offline, online, and streaming computations)</td>
<td>20</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>Graph Cleaning</td>
<td>17</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Debugging &amp; Testing</td>
<td>10</td>
<td>2</td>
<td>8</td>
</tr>
</tbody>
</table>

“You’ve got to start with the customer experience and work back toward the technology – not the other way around”

Steve Jobs
Direct-Manipulation Interfaces

Pros

• Visually present task concepts
• Allows easy learning
• Allows easy retention
• Allows errors to be avoided
• Encourages exploration
• Appealing to novice, is easy to remember for intermittent users, and can be rapid for frequent users
Classical Direct Manipulation-based Approach

Query formulation

Query processing

time

HCI

DB

Graphs
Taking Advantage of Direct-Manipulation Interaction Style

VISUAL QUERY INTERFACE → QUERY ENGINE

Shallow Integration

VISUAL QUERY INTERFACE → QUERY ENGINE

Deep Integration
Benefits of Deep Integration

- Query suggestions and feedback [ICDE 19, VLDB J 17, VLDB 15, CIKM 15]
- Faster query response time [SIGMOD 10, ICDE 12, SIGMOD 13, CIDR 13, VLDB J 14, SIGMOD 18]
- Interactive search and exploration [ICDE 19, VLDB 17]
- Interactive visualization of results [VLDB 17, VLDB J 14]
Faster Query Response Time

Why Wait?

Query formulation

Query processing
Non-traditional Challenges

- Partial query-aware indexing schemes
- Materialization of intermediate results
- Selectivity-free query processing
- Focus on waiting time of users
Graph Query Processing

- **Online**: Adaptive On-the-fly index
- **Offline**: Precomputed Index

**Steps**:
1. Candidate matches
2. Candidate matches
3. Final Results
An Example

Is *C. elegans* a suitable animal model for studying apoptosis in human?

Bob (biologist, a non-programmer)

Bob’s Biological Insights

*C. elegans* is likely to be a suitable model if genes related to apoptosis behave similarly to those that are in human.

Due to evolution, models behave differently.

In conserved biological processes, interacting gene pairs are in close proximity.
An Example

C. elegans apoptotic pathway

Apoptotic stimuli

EGL-1
CED-9
CED-4
CED-3
SUN-1
CSP-3
CSP-2
CSP-1

Apoptosis

query graph topology

query graph vertex label

Data graph

Human apoptotic pathway

Apoptotic stimuli

BID
MAPK8
BCL-2
BAD
BK
BCL2L1
BAX
MYC
APAF1
HSPA4
BARD1
XIAP
CASP3
CTNNB1
PML

Apoptosis

Bob’s Biological Insights

edge bounds

[1,2]
[1,3]
[1,2]
[1,3]
[1,3]
Bounded 1-1 P-hom (BPH) Query

1-1 P-homomorphism (Fan et al., 2010)

G is **1-1 P-homomorphic** to G’ if there exists a 1-1 injective P-homomorphism from G to G’
- Distinct nodes have **distinct** matches
- **Edges** in G mapped to **paths (arbitrary length)** in G’
- Match vertices based on **vertex similarity**.

W. Fan et al., Graph homomorphism revisited for graph matching. VLDB, 2010
Given a BPH query Q visually constructed on a visual query interface and a data graph G=(V,E,L), the goal of **visual bounded 1-1 p-hom search problem** is to retrieve all bounded 1-1 p-hom matches of Q in G by **interleaving (i.e., blending)** formulation and processing of Q.
BOOMER In Action

https://www.youtube.com/watch?v=zRYqo-F4xcg&feature=youtu.be
Exploratory Search on Graph DB

Visual Exploratory Search

- Open ended, evolving in nature
- Search state is **ambiguous** in the beginning
- **Multiple and iterative** query formulation and execution
- Initial query may often grow in size
- Visual query interface to facilitate exploration

Let $\mathbf{A}$ be an exploratory action sequence (i.e., sequence of `add()`, `modify()`, `run()` actions) undertaken by a user on a visual interface for exploring a graph database $D = \{g_1, g_2, ..., g_n\}$. Then the goal of visual exploratory subgraph search (VESS) problem is to retrieve all the graphs $g_i \in D$ with $\text{dist}(g_i, q) \leq \delta$ for each $\text{run}(q) \in \mathbf{A}$ where $\delta$ is the subgraph similarity distance threshold.
https://www.youtube.com/watch?v=_ogvGKRROY&feature=youtu.be
Summary

Usability

Direct-Manipulation Interface

Performance

Query response time

Visual query interface

Query feedback, Exploratory search

Functionality
Open Problems

- Rethinking in a distributed environment
- On knowledge graphs
- Multi-faceted exploration and visualization
- Expanding the paradigm to other data types
- Direct Manipulation-driven analytics
Final Remarks

Human-Graph Interaction

• Non-progammers are typically not going to invest resources in educating themselves with GQLs
• Usable graph search systems by exploiting direct manipulation-based interaction
• HCI, Cognitive Psychology & Data Management

HINT Project