Emergence of Network Data

Social network

Ecological network

Human PPI network

WWW

The emergence of network maps:

- Movie Actor Network, 1998
- World Wide Web, 1999
- C. elegans neural wiring diagram 1990
- Citation Network, 1998
- Metabolic Network, 2000
- PPI network, 2001
Fifth Generation Data Management

Data management has democratized

**End Users:** Generator, processor, and consumer

DB illiterate

Increasingly complex data and computation

Resides everywhere

Querying Graphs

- A large set of small/medium-sized graphs
- A large graph/network
- Massive graph

Query Formulation

- Formal query language
- SPARQL, Cypher

Query Processing

- Efficient algorithms and optimization techniques to process queries “quickly”
Querying Graphs: The First Generation Approach

```sparql
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 .
  ?interaction wp:source ?gene1 ;
      wp:target ?gene2 ;
      a wp:Conversion ;
      dcterms:isPartOf ?pathway .
  ?pathway foaf:page ?page ;
  wp:organismName ?organismName .
  FILTER (?gene1 != ?gene2)
} ORDER BY ASC(?organism)
```
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)
Who Wants to Use GQLs?

- biologist
- Pharmacist/chemist
- social scientist
- ecologist
- security analyst
- financial analyst
- journalist
- techies
- pharmacist/chemist
- social scientist
Example: PubChem
Example: DrugBank
Usability Matters!

Usability

Data-driven Software

Performance

Functionality
Usability [Preece et al.]

What is it?
How well users can use the system’s functionality

Dimensionality

- **Learnability**: is it easy to learn?
- **Efficiency**: once learned, is it fast to use?
- **Memorability**: is it easy to remember what you learned?
- **Errors**: are errors few and recoverable?
- **Satisfaction**: is it enjoyable to use?
The overwhelming majority of the emails and issues were routine engineering tasks, such as users asking how to write a query…

53 practitioners (36 from IT)

Less Action, More Talk
Usability and good UI design are closely related.

Studying and improving usability is part of Human-Computer Interaction (HCI).
Different Worlds

DB World

Functionalities
Performance

Usability

HCl World

Functionalities
Performance

Usability
The Chasm for 40+ Years
Graph Querying Meets HCI
Lessons from HCI: Schneiderman’s 8 Golden Rules

- Strive for consistency.
- Give shortcuts to the user.
- Offer informative feedback.
- Make each interaction with the user yield a result.
- Offer simple error handling.
- Permit easy undo of actions.
- Let the user be in control.
- Reduce short-term memory load on the user.

Cartoon: Your user requirements include four hundred features. Do you realize that no human would be able to use a product with that level of complexity? Good point. I’d better add “easy to use” to the list.
Tutorial Overview

- Visual Interfaces for Graph Queries
- Visual Graph Query Formulation & Guidance
- Visual Action-aware Graph Query Processing
- Query Results Exploration & Visualization
Visual Interfaces for Graph Queries
Visual Graph Query Interfaces

Manual

Data-driven

Functionalities vs Aesthetics
Manual Visual Graph Query Interfaces

Search Options

- Similarity
- Substructure
- Exact

Similarity threshold

0.7

Molecular Weight Filter

e.g. 100 to e.g. 250

Maximum Results

100

Drug Types (default all):
- Approved
- Vet approved
- Nutraceutical
- Illicit
- Withdrawn
- Investigational
- Experimental

Search

MarvinJS Tutorials
Load example
Manual Visual Graph Query Interfaces
Classical Approach of Construction

- Chemical compounds
- Hardcoded labels, patterns
- Limited variety
- Manual maintenance
- Not portable
Data-driven Visual Interface Construction & Maintenance

Graph Repository

Auto

Diverse content
Portable
Auto maintenance
Data-driven Construction

Content selection

- Which patterns should be in the palette?
  - Formulate query easily and faster
  - Give shortcuts

- Issues
  - Size of the palette
  - Maximally covers the DB
  - Minimal redundancy among patterns
  - Aesthetics-aware
Data-driven Maintenance

Content Maintenance

• How do we maintain the labels and patterns as underlying data changes?

• Issues
  • Real-time maintenance
  • Batch vs Incremental
  • Enhance usability (gain in coverage and reduction in redundancy)
  • Leverage usage patterns and query workload (if available)
DAVINCI: Initial Effort
[ICDE 15, VLDB 16]

Online

Closure graphs

Canned patterns

Offline

GraphDB

Large set of small/medium sized graphs

Graph set clustering

Topologically-similar partitions

Closure graph set generation
Visual Graph Query Formulation & Guidance
## Opportune Query Feedback

### Modeling feedback
- An alert or notification for a secondary task when a user is working on a primary task

### Needs
- Detect efficiently
- Notify **opportunistically**
  - Ineffective to notify at the end of query formulation

---

Delivering notifications inopportunistically can negatively impact task completion time, lead to more errors, and increase user frustration.
When to notify?

Breakpoint

- The moment of transition between two observable, meaningful units of task execution, and reflects a change in perception or action [Newton, 1973]
- Coarse, Medium, and Fine
- Best moment to interrupt a user is on breakpoints between tasks [Iqbal & Bailey, CHI 2008]
  - Defer the notification to appear in the next breakpoint

Adopt defer-to-breakpoint-based strategy for interrupting query formulation tasks

Reduction of Interruption cost and frustration

React faster to notifications

Task-relevant notifications should be delivered at Medium or Fine breakpoints
Modeling Optimal Notification Time

Deliver notification before the construction of the succeeding query condition is finished (Optimal breakpoints)

How do we estimate the time available for deliver of notification at optimal breakpoint?
HCI-Inspired Quantitative Model

\[ T_m = a + b \log_2 \left( \sqrt{\frac{D}{W}} + \eta \frac{D}{H} + 1 \right) \]

\[ T_s = m + n \times (\log_2(p + 1)) \]

\[ 0 < T_{opt} < T_m + T_s \]

[Accott & Zhai, CHI 03]

[Ahlstrom, CHI 05]
The iSERF Framework

- **Interrupted-Sensitive Notification Module**
  - Compute movement time $T_m$
  - Suspend notification by $T_m$ time

- **Cursor moving towards Schema Panel**
  - Compute movement time $T_m$
  - Suspend notification by $T_m$ time

- **Cursor in Schema Panel**
  - Compute selection time $T_s$
  - Suspend notification by $T_s$ time for item to be selected

- **Notification delivery**
  - Deliver appropriate notification identifying condition(s) for empty result
More on the Feedback Module

- Query Autocompletion
- Action Guidance
Query Autocompletion Demo

## Autocompletion Comparisons

<table>
<thead>
<tr>
<th></th>
<th><strong>Keyword Search</strong></th>
<th><strong>Visual Graph Query</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Action</strong></td>
<td>keystroke</td>
<td>Drag, click</td>
</tr>
<tr>
<td><strong>Atomic Unit</strong></td>
<td>char: 'a', 'b', 'c', ...</td>
<td>edge: C-C, C=C, ...</td>
</tr>
<tr>
<td><strong>Logical Unit</strong></td>
<td>keyword: &quot;world&quot;, &quot;clock&quot;, ...</td>
<td>subgraphs</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>concatenated keywords</td>
<td>graphs</td>
</tr>
<tr>
<td></td>
<td>&quot;world clock&quot;, &quot;world cup&quot;, ...</td>
<td>C=C-C=C-C=C, ...</td>
</tr>
</tbody>
</table>
The AutoG Framework

Client

① Submit query $q$ and user intent $\vec{u}$

$q$: $\vec{u} : (\alpha, \beta)$

⑤ Review suggestions $Q'_{topk}$

Server

② Decompose query $q$ as a set of embeddings of c-prime features $M_q$

$M_q$

③ Derive all candidate suggestions $Q' = \{q'_1, \ldots, q'_{n}\}$ using $M_q$ over FDAG

④ Rank top-$k$ candidate suggestions $Q'_{topk} = \{q'_1, \ldots, q'_k\}$

FDAG
User Preference / Intent

User intent value of a query (suggestion) set

$$\text{util}(Q') = \alpha \times \frac{1}{k} \sum_{q' \in Q'} \text{sel}(q') + \beta \times \frac{1}{k(k-1)} \sum_{q'_i, q'_j \in Q', i \neq j} \text{dist}(q'_i, q'_j)$$

(MCCS) Distance between two graph suggestions

$$\text{dist}(g_1, g_2) = 1 - \frac{|\text{cs}(g_1, g_2)|}{\max\{|g_1|, |g_2|\}}$$

Property: $\text{util}$ is submodular $\rightarrow$ greedy
Optimizations

The FDAG index
- Index c-Prime features and their pairwise compositions
- Prune automorphic suggestions (redundant suggestions) early

Online ranking
- Approximate selectivities of query suggestions
- Prune empty suggestions early
- Optimize diversity computation
  - trimming the common parts between suggestions
More on the Feedback Module

Query Autocompletion

Action Guidance
Orion

Overview

- Interactive visual query builder with suggestions
- Iteratively suggest edges based on their relevance to the user’s query intent, according to the partial query graph so far
  - Edge ranking: query-specific random decision paths
- The use of statistics based on data graph, query logs, and so on.

Suggestions: Grey nodes/edges

- Accepted by users: Positive edges (become blue)
- Reject or ignored by users: Negative edges

User’s intent can be derived from these edges
Orion GUI

Dynamic list of all possible user actions at any given moment

Control panel for various settings and tips
Orion Implementation

- **Prototype**
  
  http://idir.uta.edu/orion

- **Video Introduction**
  
Visual Action-aware Graph Query Processing
Classical Visual Querying
Paradigm: Shallow Integration

40+ years old query paradigm!
Towards Deep Integration of HCI & DB

Query formulation

Query processing

“Computing time (power) is getting cheaper but users’ time isn’t..”
What are the Benefits of Not Waiting??

<table>
<thead>
<tr>
<th>Query Suggestions &amp; Guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Based on partial query formulated by a user, we can give opportune suggestions &amp; feedbacks during visual query formulation</td>
</tr>
<tr>
<td>• AutoG [VLDB J 17, VLDB 15], iSERF [CIKM 15]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Faster System Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>• By starting to process a visual query early, we can finish its execution earlier</td>
</tr>
<tr>
<td>• GBLENDER [SIGMOD 10], PRAGUE [ICDE 12, SIGMOD 13, CIDR 13], QUBLE [SIGMOD 13, VLDB J 14], BOOMER [SIGMOD 18]</td>
</tr>
</tbody>
</table>
What are the Benefits of Not Waiting??

Interactive Search & Exploration

- Supports exploratory search
- Do not need to process a partially formulated query from scratch again and again!
- PICASSO [VLDB 17]

Efficient and Interactive Visualization

- Human actions on the visual interface determines what to visualize
Graph Query Processing

Online

Adaptive On-the-fly index

Candidate matches

Offline

Precomputed Index

Final Results
Non-traditional Challenges

- Partial query-aware indexing schemes
- Materialization of intermediate results
- Selectivity-free query processing
- Focus on waiting time of users
Motivating 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**.
Motivating Example

C. elegans apoptotic pathway

Apoptotic stimuli

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

Apoptosis

Human apoptotic pathway

Apoptotic stimuli

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

Apoptosis

Query graph
topology

Query

Data graph

Bob's Biological Insights

edge bounds

[1,2]  [1,3]  [1,3]
BID  BCL-2  CASP3
[1,2]  APAF1

[1,2]  [1,2]  [1,3]

query graph vertex label
BPH Query

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

Human apoptotic pathway

Data graph

Apoptosis

BAD
BID
MAPK8
BCL-2
BAK
BCL2L1
BAX
MYC
APAF
HSPA4
BARD1
XIAP
CASP
CTNNB1
PML

BID
[1,2]
[1,3]
[1,3]

BCL-2
[1,2]

APAF
[1,2]

CASP
[1,2]

edge bounds

Bob’s Biological Insights
1-1 P-homomorphism (Fan et al., 2010)

G is \textbf{1-1 P-homomorphic} to G’ if there exists a \textbf{1-1 injective} P-homomorphism from G to G’

- Distinct nodes have \textbf{distinct} matches
- \textbf{Edges} in G mapped to \textbf{paths (arbitrary length)} in G’
- Match vertices based on \textbf{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 Framework

Online

- Visually constructed BPH query

- Constructs CAP index as BPH query is constructed iteratively

Offline

- Data Graph

- Pruned landmark labelling (PML) index

- Fast exact shortest-path distance computation
- Based on 2-hop cover

Ave edge processing time ($t_{avg}$)

- To prioritize BPH edges and to determine when to process them
- Ave. processing time of 1M randomly selected dist. queries bet. vertex pairs

Ave. processing time of 1M randomly selected dist. queries bet. vertex pairs

Pruned landmark labelling (PML) index

Visually constructed BPH query

CAP index

Akiba et al., Fast exact shortest-path distance queries on large networks by pruned landmark labeling. SIGMOD, 2013
## Challenges for Performance Study

### Large-scale performance study

- **Traditional approach**
  - Randomly extract subgraphs of different size and execute them
- **Doesn’t work in this paradigm!**

### Why?

- Queries need to be visually constructed by users
- GUI latency is critical for performance study

### Challenge

- Users are expensive!
- How do we simulate visual query formulation?
VISUAL [ICDE 15, TKDE 17]

C1: Test Subgraph Query Generator

C2: Quantitative Model of Query Formulation

C3: Visual Subgraph Query Simulator

Index

Graph Repository

Test queries

Data graphs

Model
Quantitative Model for Query Formulation Time

$T(m) + T(s) + T(d) + T(e)$

$T_m = a + b \log_2 \left( \sqrt{\frac{D}{W}} + \eta \frac{D}{H} + 1 \right)$

$T_s = m_1 + n_1 \times (\log_2(p + 1))$

$T_e = 2n_c \times t_c + T_m$

[Accott & Zhai, CHI 03]

[Cockburn et al, HCI 09]
Query Results
Exploration & Visualization
Query Results Exploration

Two Categories

- Very few efforts!
- Large set of small graphs vs large networks

Large set of small graphs

- Typically a decision problem
- Highlight a subgraph that matches the query
- [SIGMOD 10, ICDE 12]
- PICASSO [VLDB 17]
Query Results Exploration

Large Networks

- Summarization-based (SLQ [SIGMOD 14])
- Supergraphlet-at-a-time (QUBLE [VLDBJ 14, SIGMOD 13])
- Feature-based (R2DB [ICDE 12])
Conclusions
Bridging Usability, Performance, Functionality

- Visual query interface
- Query response time
- Query feedback, Exploratory search

Usability
Software
Performance
Functionality
Shifting Traditions

1990-2015: Visual query interfaces are constructed manually

2015: Automatic, data-driven construction of visual graph query interface

1970s-2005s: Query Formulation → Query Processing

2006s: Visual query form. ↔ Query Processing

1990s - 2015: Visual query performances are carried out manually

2015: Automated query construction and performance benchmarking
Open Research Problems

- More complex graph queries: Multi-attribute queries, graph simulation
- Visually querying massive graphs
- How can we extend data-driven GUI construction to be aesthetics-aware?
- Multi-faceted exploration and visualization of query results
- HCl-awareness with other types of data?
Final Words

**HCI-aware Data Management**

- Towards **usable** data management systems
- Making visual query interface design **data-driven**
- Making query formulation & processing **HCI-driven**
- Novel area of research

**Multi-disciplinary effort:**

Data management  
**HCI**  
Cognitive psychology

**Broad goal**

Stimulating a cultural shift in our thinking by **HCI**, cognitive psychology and data management to “work” together
Visual > Text
Grab the Book!

Human Interaction with Graphs
A Visual Querying Perspective

Sourav S. Bhowmick
Byron Choi
Chengkai Li

SYNTHESIS LECTURES ON DATA MANAGEMENT

H.V. Jagadish, Series Editor