DB ⊗ HCI: Towards Bridging the Chasm Between Graph Data Management & HCI

Sourav S Bhowmick
Nanyang Technological University
Singapore
www.ntu.edu.sg/home/assourav
assourav@ntu.edu.sg, sourav@mit.edu
The World has Changed

Then

- Data is generated in companies
- Resides in companies
- Used by companies
- DB-literate users

Now

- Data is generated by everyone
- Resides everywhere
- Used by everyone
- Non-DB literate users
“The old computing was about what computers could do; the new computing is about what people can do”

Ben Shneiderman (2002)
Users are willing to share information online

What Happens in an Internet Minute?

- 6.578.800 GB of global IP data transferred
- 135 Solnet infections
- 1.300 New mobile users
- 100+ New LinkedIn accounts
- 47.000 App downloads
- 61.141 Hour of music
- 583.000 In sales
- 204 million Emails sent
- 20 million Photo views
- 6 million Facebook views
- 2+ million Facebook status updates
- 320+ New Twitter accounts
- 320 New Twitter accounts
- 1.0 million Video views
- 30 Hours of video uploaded
- 2.075.000 Facebook logins

And Future Growth is Staggering

The Road Ahead

- DB in the changing world
- HCl in the changing world
- The chasm!
- HCl-aware data management
- Conclusions
Complexity has Increased

Parallel & Distributed DB

Cloud data management

Data streams

DB+ IR

mobile data management

Object-Oriented Model

<table>
<thead>
<tr>
<th>Object 1: Maintenance Report</th>
<th>Object 1 Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>01-12-01</td>
</tr>
<tr>
<td>Activity Code</td>
<td>23</td>
</tr>
<tr>
<td>Route No.</td>
<td>1.85</td>
</tr>
<tr>
<td>Daily Production</td>
<td>2.5</td>
</tr>
<tr>
<td>Equipment Hours</td>
<td>6.0</td>
</tr>
<tr>
<td>IATL Hours</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object 2: Maintenance Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Code</td>
</tr>
<tr>
<td>Activity Name</td>
</tr>
<tr>
<td>Production Unit</td>
</tr>
<tr>
<td>Average CHP Production Rate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>Product ID</th>
<th>Price ($)</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>120</td>
<td>0.7</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>80</td>
<td>0.3</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>110</td>
<td>0.6</td>
</tr>
<tr>
<td>d</td>
<td>b</td>
<td>90</td>
<td>0.4</td>
</tr>
<tr>
<td>e</td>
<td>c</td>
<td>140</td>
<td>0.5</td>
</tr>
<tr>
<td>f</td>
<td>c</td>
<td>110</td>
<td>0.3</td>
</tr>
<tr>
<td>g</td>
<td>c</td>
<td>100</td>
<td>0.2</td>
</tr>
<tr>
<td>h</td>
<td>d</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

semistructured data management

graph data management

Probabilistic DB
DB Community’s Love Affair

SPEED, SPEED, SPEED
SCALE, SCALE, SCALE
FUNCTIONALITIES!!
Assumptions Made By (Most) DM Systems

- Navigation capabilities of a fly
- Stamina of a camel
- Memory of an elephant
- Skills of a monkey
- Eyes of an eagle

"The Perfect User"
The Real User

DB Query languages are hard to use
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
PREFIX rss: <http://purl.org/rss/1.0/>  
PREFIX dc: <https://purl.org/dc/elements/1.1/>  

SELECT ?title ?known_name ?link 
FROM http://planetrdf.com/index.rdf 
FROM NAMED <phil-toaf.rdf> 
WHERE { 
  GRAPH <phil-toaf.rdf> { 
    ?me foaf:name "Phil McCarthy". 
    ?me foaf:knows ?known_person . 
    ?known_person foaf:name ?known_name . 
  }.
  ?item dc:creator ?known_name .
  ?item rss:link ?link .
  ?item dc:date ?date .
} 
ORDER BY DESC(?date) LIMIT 10
Reality

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)
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?
DB Research Since 2006

- Function- alities
- Usability
- Performance
Next..

- The World of HCI
What is Human Computer Interaction (HCI)?

“concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them”

ACM SIGCHI (1992)
Studying and improving usability is part of Human-Computer Interaction (HCI).

Usability and good UI design are closely related.
“The interface is the system.”

**View**
- Interface provides/conveys the *only* view of the “underlying” system

**Perception**
- Usable software sells better

**Superficiality**
- Users blame themselves for UI failings
- People who make buying decisions are not always end-users
“Always should have “good” interfaces. Computing time (power) is getting cheaper but users’ time isn’t.”

“If the user can’t use it, it doesn’t work”
The World of Cool Interfaces!

Role of visual interfaces

“A picture is worth a thousand words. An interface is worth a thousand pictures”

Ben Shneiderman, 2003

Easy-to-use “dummy” visual interfaces are key to the spread of data management tool to wider community.
Lessons from HCl: 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.

How do these “rules” impact data management?
"A USER INTERFACE IS LIKE A JOKE. IF YOU HAVE TO EXPLAIN IT, IT’S NOT THAT GOOD"

Anonymous, LinkedIn
The Chasm
Our Affinity to Visual Languages

- Cave painting
- Cuneiform
- Hieroglyphics
- Coats of arms
- Emoji
- Visual query language
Separate Ways for 40 years

HCl

DB
Data Management Research
Querying Graph Databases

Query Formulation

• Formal query language
  • GraphQL, SPARQL, PQL

Query Processing

• Efficient algorithms and optimization techniques to process queries “quickly”
  • FG-Index, Tree-PI, SSI, C-Tree, TALE
Classical Visual Query Interface Construction

GraphDB
Classical Visual Query Interface Construction
Classical Visual Querying Paradigm

Query formulation

Query processing

time

GraphDB
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.
HCI-aware Data Management
Next..

- HCI-aware data management
Broad Goals

- Data-driven construction of visual query interface
- Interleave visual actions on the GUI and query processing
- Data-driven feedback and guidance to users centered around usability
- Minimize users' time
Functional Architecture

Data-driven Visual Query Interface Manager

Visual Action-aware Query Performance Simulator

Visual Action-aware Query Processing Engine

Visual Action-aware Indexer

Graph DB
Goal 1

- Data-driven construction of visual query interface
Challenges

Panel 2

• Alphabetical, semantic, unordered
• How do we place the items for querying?
  • Search time to the same target position increases as the number of items in the panel increases [Byrne et al., CHI 99]
  • Semantic and alphabetical is faster [Halverson et al., CHI 08]
  • Users are faster at selecting targets that are closer to the top [Cockburn et al, CHI 07]
  • Last item effect [Bailly et al, CHI 14]
Palette Selection

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

Issues
- “Data-driven” selection
- Size of the palette
- Frequent and infrequent patterns
- Maximally covers the DB
- Minimal redundancy among patterns
- Dynamically maintained
Characteristics

• Node/edge-at-a-time or subgraph-at-a-time approach
  • Size of the query fragment increase by 1 or $k$
• Query can be modified at any time
  • Size may not increase monotonically
• Different formulation sequences
• At any step, the partial query is either frequent or infrequent
  • The chance of a fragment to remain frequent diminishes as the size of the query grows
A novel paradigm

• Why wait for the complete visual query to be constructed before initiating query evaluation? How can we blend these two steps?
• By initiating query processing "early", can we significantly reduce the user’s waiting time?
## Non-traditional Challenges

### Action-aware Indexing

- Prune irrelevant results even when partial query graph is known
- Efficient traversal from $g'$ (Step i) to $g''$ (Step i+1) where $g' \subseteq g''$ and $|g''| = |g'| + k$
- Support of pruning graph-structured frequent and infrequent fragments
- **Smaller-sized** graphs should be efficiently indexed
- Minimize candidate verification of partial results

### Intermediate results materialization

- Materialization of all partial candidate graphs matching the query fragment is necessary
- Unreasonable in traditional DB
- Challenging due to computational hardness of subgraph matching
Bounds on materialization time

\[ T(m) + T(s) + T(d) + T(e) \]
Computing time (power) is getting cheaper but users’ time isn’t..”

Non-traditional Challenges

Selectivity-free query processing

- Selectivity-based query processing is impractical
- Query fragments are drawn in any arbitrary sequence
- Complete query is not available during query evaluation due to the paradigm
  - “Push-down” highly selective fragment is not possible
  - Classical approach of physical query plan generation is ineffective

Focus on waiting time of users

- Quick formulation of queries
- System response time (SRT) matters more than backend processing cost!
- SRT should be robust to different query formulation sequences.
A Video of Visual Action-aware Query Processing
Related Efforts

- QUBLE [SIGMOD 13, VLDBJ 14]
  - For large networks
- MUSTBLEND [DASFAA 13, ICDE 09, ICDE 06]
  - For XML data
Goal 2

- Side-effects of intermediate results materialization
- Data-driven feedbacks and notifications

What kind of feedback?

- Aiding query construction by making appropriate suggestions
  - Given a query fragment, which top-k patterns are most likely to be added to the query?
- Empty results detection
Empty Results Detection
Goal 3

- Interruption-sensitive notifications

Lessons from HCI & Cognitive Psychology

- Interrupting users engaged in tasks by delivering notifications inopportune can negatively impact task completion time, lead to more errors, and increase user frustration [Bailey et al., JCSB 2006].
- Breakpoint-based notifications
Notification in the data management world

- Overly-aggressive!
- Immediate notification regardless of the cognitive impact on the user

Solution

- Seamlessly integrate deferred-to-breakpoint strategy [Iqbal et al., CHI 2007, 2008] with query formulation
- Defer notification to optimal breakpoints
Bounds on feedback generation time

\[ 0 < T(\text{opt}) < T(m) + T(s) \]
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?
Next..

- DB in the changing world
- HCl in the changing world
- The chasm!
- HCl-aware data management
- Conclusions
Blending more Complex Queries

- Homeomorphic queries
- Distance-constraint queries
- Multi-attribute queries
- Graph simulation
HCl-aware Data Management in the Big Data Era
Kuhn’s Paradigm

1) Establish a paradigm

2) Mature science

3) Anomalies appear

Published in 1962
Kuhnian Example

1700: Classical (Newtonian) mechanics

1800
Experiments with light show anomalous behavior

1900

1905: Relativity
Kuhnian Paradigm for Data Management

1970s: Query Formulation \(\rightarrow\) Query Processing

2000s: Visual query formulation \(\leftrightarrow\) query processing

- **Blends** data management and HCI
- Three key components
  - Data-driven visual interface management
  - Visual action-aware query processing
  - Visual action-aware performance simulation
- Can be extended to wide variety of data types

**Broad goal**

- Pervasive desire to stimulate a cultural shift in our thinking by bringing together HCI and data management to “work” together
Acknowledgements

Byron Choi
Hong Kong Baptist Univ

Changjiu Jin

Shuigeng Zhou
Fudan Univ, China

Ba Quan Truong

Ho Hoang Hung