

# The Revenge of the Databases

Antonio Badia  
abadia@louisville.edu

or:  
abadia@i2pinc.com

or:  
antonio\_badia@yahoo.com

# Overview of the Talk

- DB and IR Differences.
- DB and IR Integration.
- Fusion in Action: From Databases to Dataspaces.
- Fusion in Action: Keyword search in Databases.
- Fusion in Action: SQL search on Documents.
- Our Research.
- Why Should Yahoo! care?

# DB and IR: Differences

DB	IR
structured data	unstructured data
query language	keywords
exact answer	ranked answer
performance & optimization	no optimization
query & update	(relatively) static
soundness & completeness	probabilistic
no relevance	relevance
user smart?	user dumb?

# DB and IR Integration

- In the last few years, DBMS have incorporated documents using traditional IR techniques: Oracle's Text (10g), IBM's Content Manager.

# DB and IR Integration

- In the last few years, DBMS have incorporated documents using traditional IR techniques: Oracle's Text (10g), IBM's Content Manager.
- Is there more data *outside* the database than inside?

# DB and IR Integration

- In the last few years, DBMS have incorporated documents using traditional IR techniques: Oracle's Text (10g), IBM's Content Manager.
- Is there more data *outside* the database than inside?
- Yes: abundance of *documents* (email, manuals, reports) ...

# DB and IR Integration

- In the last few years, DBMS have incorporated documents using traditional IR techniques: Oracle's Text (10g), IBM's Content Manager.
- Is there more data *outside* the database than inside?
- Yes: abundance of *documents* (email, manuals, reports) ...
- ... and *the Web*.

# DB and IR Integration

- In the last few years, DBMS have incorporated documents using traditional IR techniques: Oracle's Text (10g), IBM's Content Manager.
- Is there more data *outside* the database than inside?
- Yes: abundance of *documents* (email, manuals, reports) ...
- ... and *the Web*.
- No: Most DB data now captured automatically, low level (sensors, clickstream, etc.).

# DB and IR Integration (Cont.)

- My Answer: this is irrelevant. It only counts *bytes*.

# DB and IR Integration (Cont.)

- My Answer: this is irrelevant. It only counts *bytes*.
- What kind of data do you have? How do you analyze it?  
What can you get out of it?

# DB and IR Integration (Cont.)

- My Answer: this is irrelevant. It only counts *bytes*.
- What kind of data do you have? How do you analyze it? What can you get out of it?
- Data in DB tends to be low level, but reliable, complete (w.r.t. domain).

# DB and IR Integration (Cont.)

- My Answer: this is irrelevant. It only counts *bytes*.
- What kind of data do you have? How do you analyze it? What can you get out of it?
- Data in DB tends to be low level, but reliable, complete (w.r.t. domain).
- Data in documents can be high and low level, incomplete, unreliable.

# DB and IR Integration (Cont.)

- My Answer: this is irrelevant. It only counts *bytes*.
- What kind of data do you have? How do you analyze it? What can you get out of it?
- Data in DB tends to be low level, but reliable, complete (w.r.t. domain).
- Data in documents can be high and low level, incomplete, unreliable.
- Analysis of structured data much more mature, easier; analysis of unstructured data harder, approximate.

# DB and IR Integration (Cont.)

- My Answer: this is irrelevant. It only counts *bytes*.
- What kind of data do you have? How do you analyze it? What can you get out of it?
- Data in DB tends to be low level, but reliable, complete (w.r.t. domain).
- Data in documents can be high and low level, incomplete, unreliable.
- Analysis of structured data much more mature, easier; analysis of unstructured data harder, approximate.
- *Best systems will integrate both* (for the Web this means the *Deep* or *Hidden Web*).

# DB and IR Integration (Cont.)

- So how do you put DB and IR together?

# DB and IR Integration (Cont.)

- So how do you put DB and IR together?
- Mix-up of models (dataspaces), query languages (keyword search in DBs, SQL on Web pages).

# DB and IR Integration (Cont.)

- So how do you put DB and IR together?
- Mix-up of models (dataspaces), query languages (keyword search in DBs, SQL on Web pages).
- Next step is to use Information Extraction to achieve integration in-depth...

# DB and IR Integration (Cont.)

- So how do you put DB and IR together?
- Mix-up of models (dataspaces), query languages (keyword search in DBs, SQL on Web pages).
- Next step is to use Information Extraction to achieve integration in-depth...
- ...because a *Venetian blind* and a *blind Venetian* are not the same (and *to be or not to be* should be indexed).

# From Databases to Dataspaces

- Challenge: integrate all types of data (structured, semistructured, unstructured), regardless.

# From Databases to Dataspaces

- Challenge: integrate all types of data (structured, semistructured, unstructured), regardless.
- Basic Data Model? Graph, of course.

# From Databases to Dataspaces

- Challenge: integrate all types of data (structured, semistructured, unstructured), regardless.
- Basic Data Model? Graph, of course.
- Capture data from applications, other sources.

# From Databases to Dataspaces

- Challenge: integrate all types of data (structured, semistructured, unstructured), regardless.
- Basic Data Model? Graph, of course.
- Capture data from applications, other sources.
- Query by keyword, by example,... (whatever it takes).

# From Databases to Dataspaces

- Challenge: integrate all types of data (structured, semistructured, unstructured), regardless.
- Basic Data Model? Graph, of course.
- Capture data from applications, other sources.
- Query by keyword, by example,... (whatever it takes).
- MIT's Haystack project, U of Washington's Semex.

# Keyword Search on Databases

- Lots of work in Keyword Search in Semistructured Data (XML).

# Keyword Search on Databases

- Lots of work in Keyword Search in Semistructured Data (XML).
- Recently, work on Keyword Search on Databases.

# Keyword Search on Databases

- Lots of work in Keyword Search in Semistructured Data (XML).
- Recently, work on Keyword Search on Databases.
- Keywords have become the *universal lowest common denominator*.

# Keyword Search on Databases

- Lots of work in Keyword Search in Semistructured Data (XML).
- Recently, work on Keyword Search on Databases.
- Keywords have become the *universal lowest common denominator*.
- Given some keywords, looks for them in all string-based attributes.

# Keyword Search on Databases

- Lots of work in Keyword Search in Semistructured Data (XML).
- Recently, work on Keyword Search on Databases.
- Keywords have become the *universal lowest common denominator*.
- Given some keywords, looks for them in all string-based attributes.
- Connect the tuples with the keywords as values (primary key-foreign key connection).

# Keyword Search on Databases

- Lots of work in Keyword Search in Semistructured Data (XML).
- Recently, work on Keyword Search on Databases.
- Keywords have become the *universal lowest common denominator*.
- Given some keywords, looks for them in all string-based attributes.
- Connect the tuples with the keywords as values (primary key-foreign key connection).
- Rank tuples by *tf-idf* of keywords in attribute, directedness of connection.

# Structured Queries over Documents

- Why would anyone want to run SQL over text?

# Structured Queries over Documents

- Why would anyone want to run SQL over text?
- SQL allows more complex queries (revenge), enables optimization (revenge), escalates (revenge),...

# Structured Queries over Documents

- Why would anyone want to run SQL over text?
- SQL allows more complex queries (revenge), enables optimization (revenge), escalates (revenge),...
- To make it work, we need *Information Extraction*.

# Structured Queries over Documents

- Why would anyone want to run SQL over text?
- SQL allows more complex queries (revenge), enables optimization (revenge), escalates (revenge),...
- To make it work, we need *Information Extraction*.
- Analyze text, put it into structured representation, then query.

# Our Research

- Assume you find page on the web and need to analyze.

# Our Research

- Assume you find page on the web and need to analyze.
- System works on single pages (for now).

# Our Research

- Assume you find page on the web and need to analyze.
- System works on single pages (for now).
- Lightweight SQL:  
`SELECT . . . WHERE . . .`

# Our Research

- Assume you find page on the web and need to analyze.
- System works on single pages (for now).
- Lightweight SQL:  
SELECT . . . WHERE . . .
- Algebra of *partial records*: partially analyzed strings.

# Our Research

- Assume you find page on the web and need to analyze.
- System works on single pages (for now).
- Lightweight SQL:  
SELECT . . . WHERE . . .
- Algebra of *partial records*: partially analyzed strings.
- Formally,  $\{A_1, \dots, A_n, ELSE\}$  with *ELSE* of string type,  $n \geq 0$ .

# Our Research

- Assume you find page on the web and need to analyze.
- System works on single pages (for now).
- Lightweight SQL:  
SELECT . . . WHERE . . .
- Algebra of *partial records*: partially analyzed strings.
- Formally,  $\{A_1, \dots, A_n, ELSE\}$  with *ELSE* of string type,  $n \geq 0$ .
- Intuitively,  $A_1, \dots, A_n$  are the *structured part*, items extracted from the *ELSE*.

# Our Research

- Assume you find page on the web and need to analyze.
- System works on single pages (for now).
- Lightweight SQL:  
SELECT . . . WHERE . . .
- Algebra of *partial records*: partially analyzed strings.
- Formally,  $\{A_1, \dots, A_n, ELSE\}$  with *ELSE* of string type,  $n \geq 0$ .
- Intuitively,  $A_1, \dots, A_n$  are the *structured part*, items extracted from the *ELSE*.
- A *collection* (list, sequence, table, set) of such records is present in many pages.

# Examples

**courier-journal.com**  
*Homes*  
The Courier-Journal Louisville, Kentucky

Home • News • Sports • Business • Features • Louisville Scene • Classifieds • Jobs • Cars • Homes • Marketplace • Contact Us • Search







## Property Transfers

- 1. Jefferson**  
40205  
Garry and Betty Terry to Jessica E. Causey, 1815 Gardiner Lane, UnitWE45, \$85,000, February 25, 2007  
Residential
- 2. Jefferson**  
40205  
Dung Nguyen and Ban Tran to Luis Suarez, 2727 Gardiner Lane, \$124,000, February 25, 2007  
Residential

Source of: [http://dbease.courier-journal.com/dbEase/cgi-bin/go\\_get.pl](http://dbease.courier-journal.com/dbEase/cgi-bin/go_get.pl) - Iceape

```
<li><b>
Jefferson</b><br>
40205<br>
Dung Nguyen and Ban Tran to Luis Suarez, 2727 Gardiner Lane, $124,000, February 25, 2007<br>
Residential<br><br>
```

# Examples

PRODUCT MATCHES		Results 1-20 of 1179 matches for 'car seat' in <b>Babies &amp; Kids</b>	
 <a href="#">enlarge</a>	<b><u>Britax Marathon Crocodile Convertible Car Seat</u></b> Multi-Facing, LATCH Britax - E9L06E0 - <a href="#">Convertible Car Seats</a> Rating: <a href="#">Write a Review</a>		from <b>\$269.99</b> (10 Sellers) <a href="#">Compare Prices</a> →
 <a href="#">enlarge</a>	<b><u>Britax Decathlon Tribeca Convertible Car Seat</u></b> Multi-Facing, LATCH Britax - E9L47A1 - <a href="#">Convertible Car Seats</a> Rating: ★★★★★ (1 Review)		from <b>\$239.99</b> (2 Sellers) <a href="#">Compare Prices</a> →
 <a href="#">enlarge</a>	<b><u>Britax Boulevard Onyx Convertible Car Seat</u></b> Multi-Facing, LATCH Britax - E9L5769 - <a href="#">Convertible Car Seats</a> Rating: ★★★★★ (1 Review)		from <b>\$299.95</b> (6 Sellers) <a href="#">Compare Prices</a> →
 <a href="#">enlarge</a>	<b><u>Fisher Price - Safe Voyage Deluxe Black Convertible Car Seat</u></b> Multi-Facing, LATCH - EF20BOA - <a href="#">Convertible Car Seats</a> Rating: <a href="#">Write a Review</a>	<div style="border: 1px solid gray; padding: 5px; text-align: center;">Albee Baby Carriage ★★★★★</div>	<b>\$129.99</b> + Tax & Shipping <a href="#">Shop Now</a> →
 <a href="#">enlarge</a>	<b><u>Cosco Alpha Omega Elite Convertible Car Seat</u></b> Multi-Facing, LATCH, Reclining Cosco - <a href="#">Convertible Car Seats</a> Rating: ★★★★★☆ (1 Review)	<div style="border: 1px solid gray; padding: 5px; text-align: center;">Target.com See All-Time Ratings</div>	<b>\$159.99</b> + Tax & Shipping <a href="#">Shop Now</a> →
 <a href="#">enlarge</a>	<b><u>Britax Roundabout Extra Cover with Belly Pad</u></b> Aloha Britax - R0402B0 - <a href="#">Accessories</a> Rating: <a href="#">Write a Review</a>	<div style="border: 1px solid gray; padding: 5px; text-align: center;">BestBuyBaby.com Seller Not Rated</div>	<b>\$54.99</b> + Tax & Shipping <a href="#">Shop Now</a> →

# Our Research (Cont.)

- Besides typical relational operators ( $\sigma$ ,  $\pi$ ), it uses  $\epsilon$ -*Extract* (IE tool as a black box).

# Our Research (Cont.)

- Besides typical relational operators ( $\sigma, \pi$ ), it uses  $\epsilon$  -*Extract* (IE tool as a black box).
- $\sigma, \pi$  work over the structured part.

$$\epsilon_B(\{A_1, \dots, A_n, ELSE\}) \rightarrow \{A_1, \dots, A_n, B, ELSE\}$$

(If  $B$  is not found,  $\epsilon$  puts a *null* value in the tuple).

# Our Research (Cont.)

- Besides typical relational operators ( $\sigma, \pi$ ), it uses  $\epsilon$  -*Extract* (IE tool as a black box).
- $\sigma, \pi$  work over the structured part.

$$\epsilon_B(\{A_1, \dots, A_n, ELSE\}) \rightarrow \{A_1, \dots, A_n, B, ELSE\}$$

(If  $B$  is not found,  $\epsilon$  puts a *null* value in the tuple).

- Query Plan: sequence of  $\sigma, \pi, \epsilon$  operators.

# Our Research (Cont.)

- Besides typical relational operators ( $\sigma$ ,  $\pi$ ), it uses  $\epsilon$  -*Extract* (IE tool as a black box).
- $\sigma$ ,  $\pi$  work over the structured part.

$$\epsilon_B(\{A_1, \dots, A_n, ELSE\}) \rightarrow \{A_1, \dots, A_n, B, ELSE\}$$

(If  $B$  is not found,  $\epsilon$  puts a *null* value in the tuple).

- Query Plan: sequence of  $\sigma$ ,  $\pi$ ,  $\epsilon$  operators.
- Every  $\sigma$  and  $\pi$  operators must be preceded by a  $\epsilon$  operation with arguments the attribute names mentioned in the operator.

# Our Research (Cont.)

- Besides typical relational operators ( $\sigma, \pi$ ), it uses  $\epsilon$  -*Extract* (IE tool as a black box).
- $\sigma, \pi$  work over the structured part.

$$\epsilon_B(\{A_1, \dots, A_n, ELSE\}) \rightarrow \{A_1, \dots, A_n, B, ELSE\}$$

(If  $B$  is not found,  $\epsilon$  puts a *null* value in the tuple).

- Query Plan: sequence of  $\sigma, \pi, \epsilon$  operators.
- Every  $\sigma$  and  $\pi$  operators must be preceded by a  $\epsilon$  operation with arguments the attribute names mentioned in the operator.
- $\pi$  cannot touch the *ELSE* field.

# Our Research (Cont.)

- Optimization: *minimize the number of Extracts done.*

# Our Research (Cont.)

- Optimization: *minimize the number of Extracts done.*
- Given query  
SELECT  $A_1, \dots, A_n$   
WHERE  $C_1 \theta \dots \theta C_m$   
need to extract  $\{A_1, \dots, A_n\} \cup \text{arg}(C_1) \dots \cup \text{arg}(C_m)$ .

# Our Research (Cont.)

- Optimization: *minimize the number of Extracts done.*
- Given query  
SELECT  $A_1, \dots, A_n$   
WHERE  $C_1 \theta \dots \theta C_m$   
need to extract  $\{A_1, \dots, A_n\} \cup \text{arg}(C_1) \dots \cup \text{arg}(C_m)$ .
- Idea: push down selections, conditions.

# Our Research (Cont.)

- Optimization: *minimize the number of Extracts done.*
- Given query  
SELECT  $A_1, \dots, A_n$   
WHERE  $C_1 \theta \dots \theta C_m$   
need to extract  $\{A_1, \dots, A_n\} \cup \text{arg}(C_1) \dots \cup \text{arg}(C_m)$ .
- Idea: push down selections, conditions.
- Basic property:

$$\epsilon_{B_1, \dots, B_n}(R) = \epsilon_{B_{i_1}}(\dots(\epsilon_{B_{i_n}}(R)\dots))$$

where  $i_1 \dots, i_n$  is a permutation of  $1, \dots, n$ .

# Our Research (Cont.)

- Pair up extractions with each (basic) condition.

# Our Research (Cont.)

- Pair up extractions with each (basic) condition.
- Complex conditions: done conditionally. Estimate which one has lower cost.

# Our Research (Cont.)

- Pair up extractions with each (basic) condition.
- Complex conditions: done conditionally. Estimate which one has lower cost.
- Heuristics: basic conditions preferred to complex ones, '=' is preferred to other comparisons.

$(A \wedge B) = \text{if } A \text{ then } B \text{ else False}$

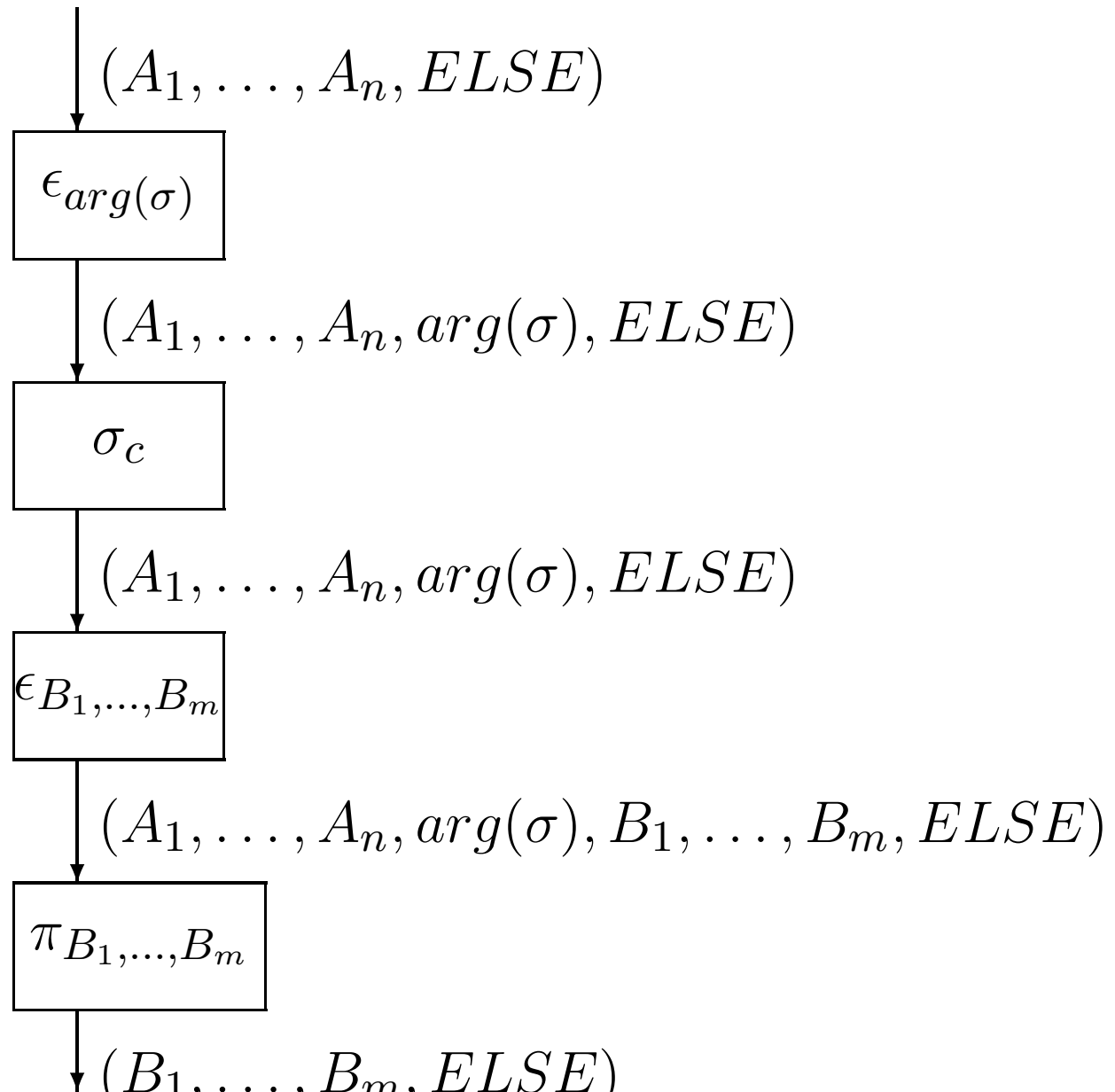
$(A \vee B) = \text{if } A \text{ then True else } B$

# Our Research (Cont.)

- Pair up extractions with each (basic) condition.
- Complex conditions: done conditionally. Estimate which one has lower cost.
- Heuristics: basic conditions preferred to complex ones, '=' is preferred to other comparisons.

$(A \wedge B) = \text{if } A \text{ then } B \text{ else False}$

$(A \vee B) = \text{if } A \text{ then True else } B$



# How does it extend to several pages?

- Distinguish between *linked* and *unlinked* pages.

# How does it extend to several pages?

- Distinguish between *linked* and *unlinked pages*.
- 2 types of linked pages: *continuations* and *references*.

# How does it extend to several pages?

- Distinguish between *linked* and *unlinked pages*.
- 2 types of linked pages: *continuations* and *references*.
- Continuations are divided lists (`Next`), treated with union.

# How does it extend to several pages?

- Distinguish between *linked* and *unlinked pages*.
- 2 types of linked pages: *continuations* and *references*.
- Continuations are divided lists (`Next`), treated with union.
- References are links on each record; treated as foreign keys (joins).

# How does it extend to several pages?

- Distinguish between *linked* and *unlinked pages*.
- 2 types of linked pages: *continuations* and *references*.
- Continuations are divided lists (`NEXT`), treated with union.
- References are links on each record; treated as foreign keys (joins).
- Unlinked pages bring the whole issue of *heterogeneous information integration* into play. Can be addressed with outer union or outer join, but not dealt with now.

# How does it scale to the Web?

- Analyze data in stages.

# How does it scale to the Web?

- Analyze data in stages.
- Each stage takes as input the output of previous stage, filters out some data.

# How does it scale to the Web?

- Analyze data in stages.
- Each stage takes as input the output of previous stage, filters out some data.
- Each stage can do more sophisticated analysis, as it handles less data.

# How does it scale to the Web?

- Analyze data in stages.
- Each stage takes as input the output of previous stage, filters out some data.
- Each stage can do more sophisticated analysis, as it handles less data.
- Original stage does keyword search over whole Web, like a search engine.

# How does it scale to the Web?

- Analyze data in stages.
- Each stage takes as input the output of previous stage, filters out some data.
- Each stage can do more sophisticated analysis, as it handles less data.
- Original stage does keyword search over whole Web, like a search engine.
- Final stage does in-depth analysis of a few pages.

# Why Should Yahoo! care?

- Querying is an important paradigm.

# Why Should Yahoo! care?

- Querying is an important paradigm.
- A priori analysis of data (PageRank, clustering, etc.) does not take user into account.

# Why Should Yahoo! care?

- Querying is an important paradigm.
- A priori analysis of data (PageRank, clustering, etc.) does not take user into account.
- A priori analysis of user (profiles) does not take data into account.

# Why Should Yahoo! care?

- Querying is an important paradigm.
- A priori analysis of data (PageRank, clustering, etc.) does not take user into account.
- A priori analysis of user (profiles) does not take data into account.
- The key is in *connecting user and data* (what's desired and what's available).

# Why Should Yahoo! care?

- Querying is an important paradigm.
- A priori analysis of data (PageRank, clustering, etc.) does not take user into account.
- A priori analysis of user (profiles) does not take data into account.
- The key is in *connecting user and data* (what's desired and what's available).
- Querying is one point where it happens: user expresses need *with respect to* some data/schema.

# Why should Yahoo! care? (Cont.)

- Traditional notion of query must be expanded; not a one-shot, but an *iterative, interactive* process.

# Why should Yahoo! care? (Cont.)

- Traditional notion of query must be expanded; not a one-shot, but an *iterative, interactive* process.
- Need to integrate structured and unstructured data (Deep/Hidden Web).

# Why should Yahoo! care? (Cont.)

- Traditional notion of query must be expanded; not a one-shot, but an *iterative, interactive* process.
- Need to integrate structured and unstructured data (Deep/Hidden Web).
- Example: this is your pipe on Yahoo; this is your pipe on steroids!

# Why should Yahoo! care? (Cont.)

- Traditional notion of query must be expanded; not a one-shot, but an *iterative, interactive* process.
- Need to integrate structured and unstructured data (Deep/Hidden Web).
- Example: this is your pipe on Yahoo; this is your pipe on steroids!

# Any questions?

abadia@louisville.edu  
abadia@i2pinc.com  
antonio\_badia@yahoo.com