Data ring: Turning the network into a database

Serge Abiteboul
INRIA-Futurs and Univ. Paris 11

Outline

1. Introduction – the data ring
2. Web support for distributed information management
3. Logical for distributed data management
4. Algebra for distributed data management
5. Self administration
6. Conclusion

Introduction – the data ring

Success stories after the Internet bubble

Google: management of Web pages

Mapquest: management of maps

Amazone: book catalogue

eBay: product catalogue

Napster (emule, bearshare, etc.): music database

Flickr: picture database

Wikipedia: dictionary

del.icio.us: annotations

In France:

Meetic: dating database

Kelkoo: comparative shopping

They are all about
publishing some database

The trends: peer-to-peer and interactivity

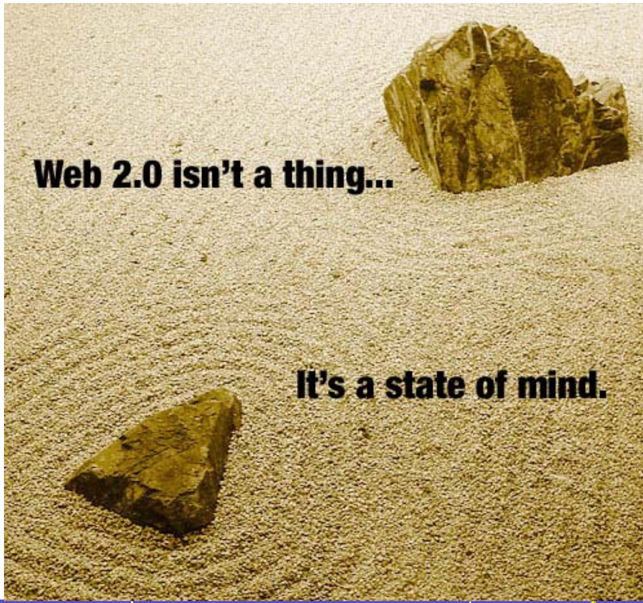
Switch from centralized servers to communities and syndication

Peer-to-peer: *A large and varying number of computers cooperate to solve some particular task without any centralized authority*

seti@home; kazaa; cabal

Interactivity and Web 2.0

Motivations: Social, organizational



Information management in a P2P network

Private terminology: **data ring**

Management of information: Data + knowledge

- Data: relations, documents, multimedia, services
- Knowledge: Meta-data, ontologies, view definitions
- Physical data: Indices and materialized views

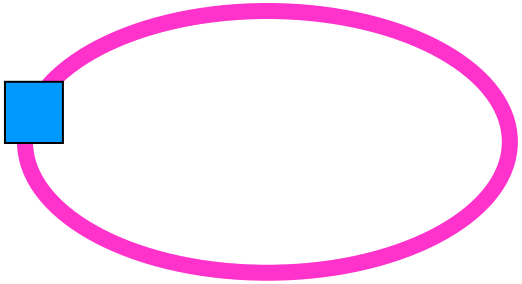
Information is heterogeneous, distributed, replicated, dynamic

Peers are heterogeneous, autonomous and possibly mobile

Typically very large number of peers

Variety of requirements: QoS, performance, security, etc.

What is a peer?



A mainframe database

A file system

Web server

A PC

A PDA

A telephone

A sensor

A home appliance

A car

A manufacturing tool

A telecom equipment

A toy

Another data ring

Any connected device or software with some information to share

Why P2P?

It is easy to get access to lots of processing power

- Cpu, disk, memory, network
- Hardware is cheap
- Lots of available hardware that is not used most of the time

What can we do with this processing power?

- Simulate life (cell, heart, gene, etc.), climate, etc.
- Build new services with all the information available on the net

Advantages of P2P

- Performance
- Scalability
- Availability
- Cost

Disadvantages

- Complexity
- Updates and transactions
- Quality of Services
- Access rights

Examples

Content sharing community

- *A group of users that share and query information within some domain*
- Flickr, an association, a company, a consortium

Personal data management

- Pda, phone, pc, home appliance, car, tv…

Scientific data management

- Experiments and simulations generate huge quantity of data

Google search in P2P

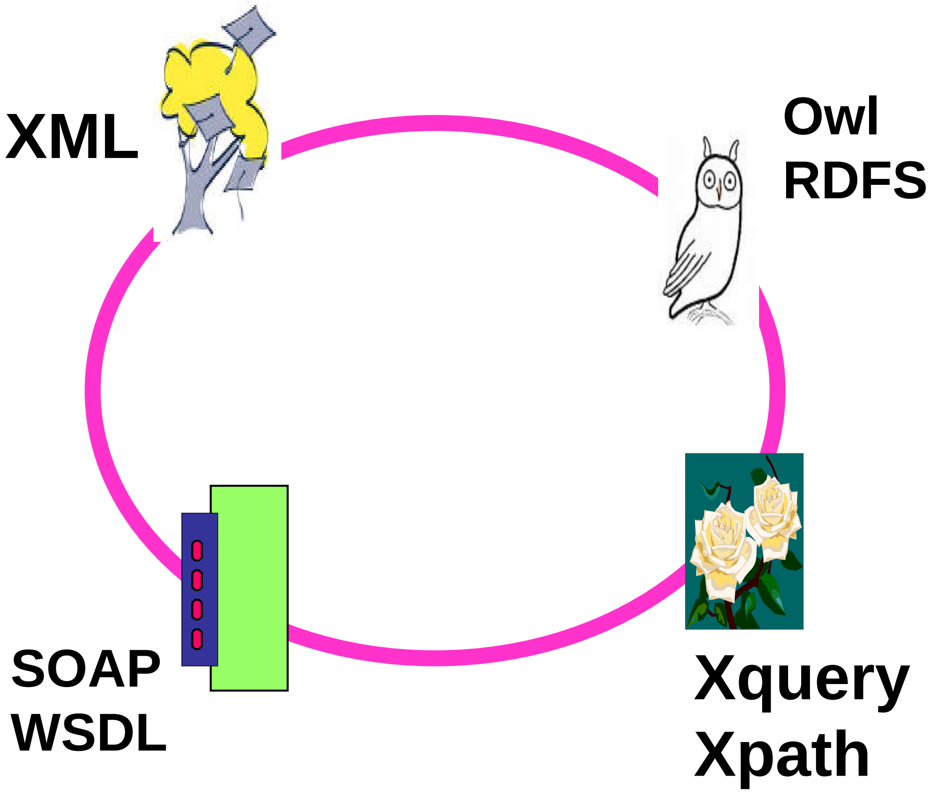
Taxonomy

- Volume of information & number/volatility of peers
- Quality of service

Web support for distributed data management

INRIA

Crash course on Web standards



Information used to live in islands but it is changing

Different formats: relational, metadata, documents, text, DXF

- A Web standard for data exchange, XML, is fixing it
- XML captures all kinds of information over a wide spectrum
- XML comes with a family of emerging standards: XML schema, XSL/T, Xquery, domain specific schemas…

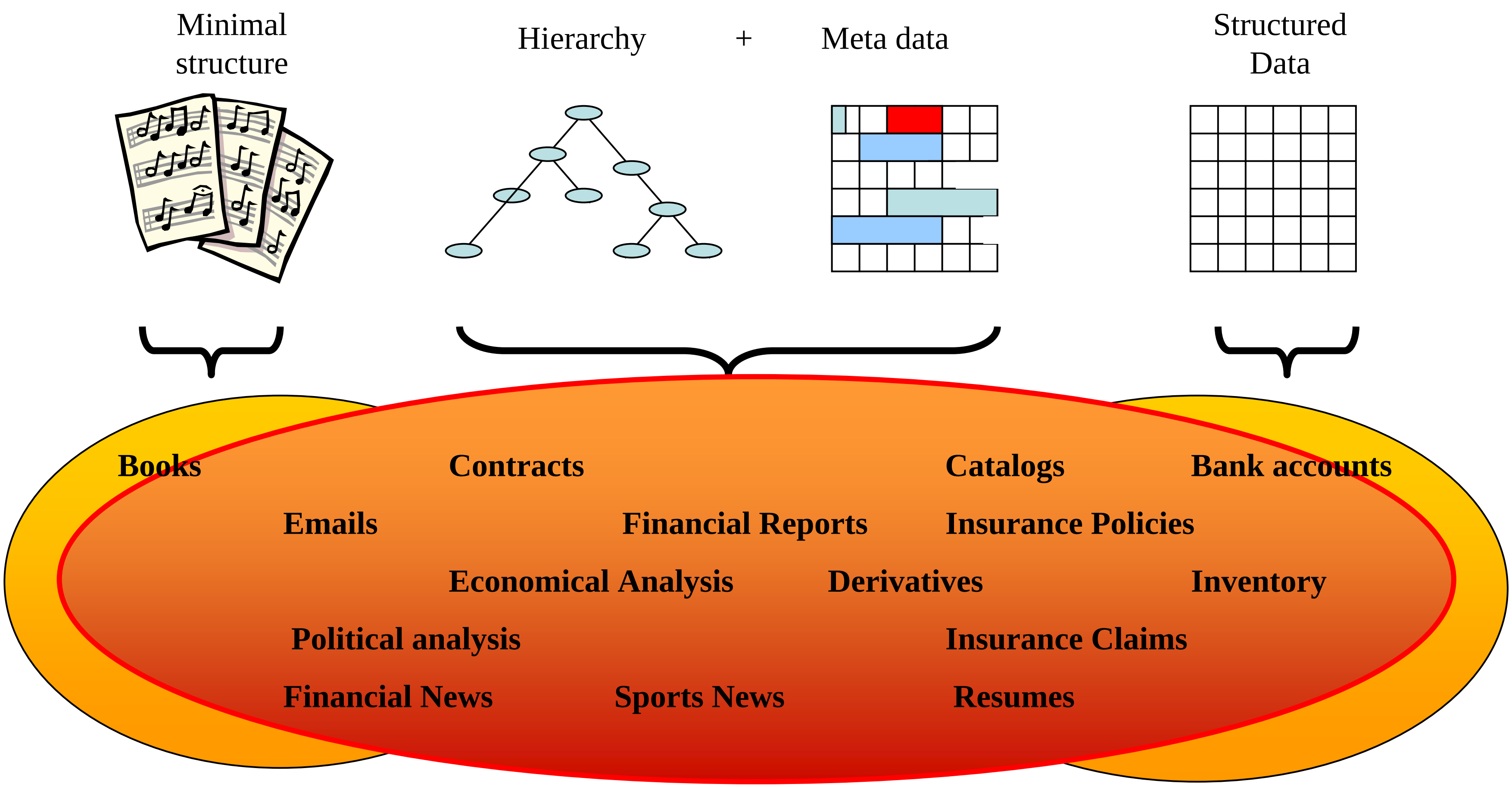
Different computers, platforms, languages, applications

- A standard for Web services, SOAP, is fixing it
- SOAP allows ubiquitous computing on the Internet
- SOAP comes with a family of emerging standards: WSDL, UDDI

Do you like these standards?

- I find them sometimes too complicated
- But this is just not the right question

The information spectrum and XML

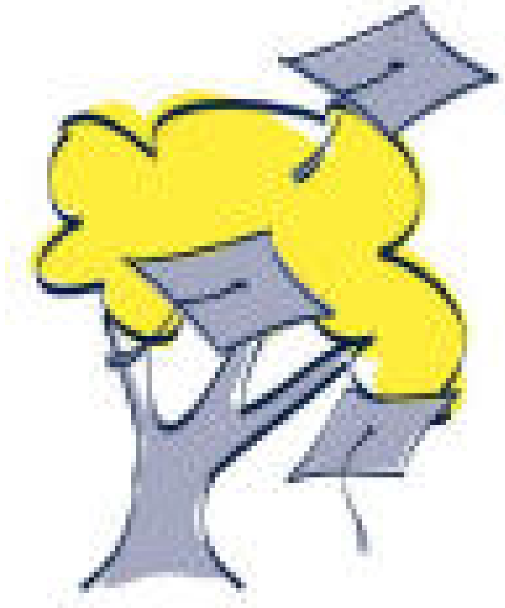


A standard for information exchange: XML

Labeled ordered trees where leaves are text

Marriage of document and database worlds

Marriage of full text indexing and structure indexing



Applications need typing

- XML typing: DTD and XML schema
- **Tree automata**

Semantics and structure are in tags and paths

- product-table/product/reference
- product-table/product/price

Standards for distributed computing: Web services

Possibility to activate a method on some remote Web server

(something like Corba in less elaborate)

Exchange information in XML: input and result are in XML

Ubiquitous XML distributed computing infrastructure

With XML and Web services, it is possible

- To get information from virtually anywhere
- To provide information to virtually anywhere

A family of standards: SOAP, WSDL, UDDI, BPEL

A standard for XML queries: Xquery

A “logic” for labeled, ordered, unranked tree

– a declarative language

Inspired by SQL: standard for relation data

Inspired by OQL: standard for object databases

- Functional as OQL
- Not as clean

Mixes structure and content – information retrieval

- Give me the documents where the word XML appears in title
- Some full-text extension is coming

Also an update language

Some emerging standards for semantics

Owl and RDF

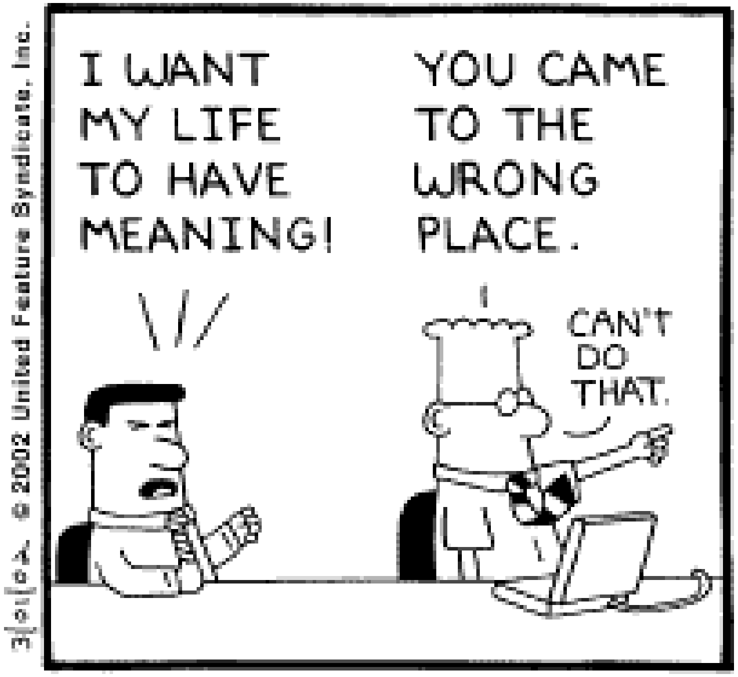
To describe Web resources

- Domain specific ontologies
- Web page & service meaning

Semantic Web

Standards: still unclear

Personal opinion: too complex



Summary of crash course

Data exchange format

- XML
- **Labeled, unranked, ordered trees**

Distributed computing protocol

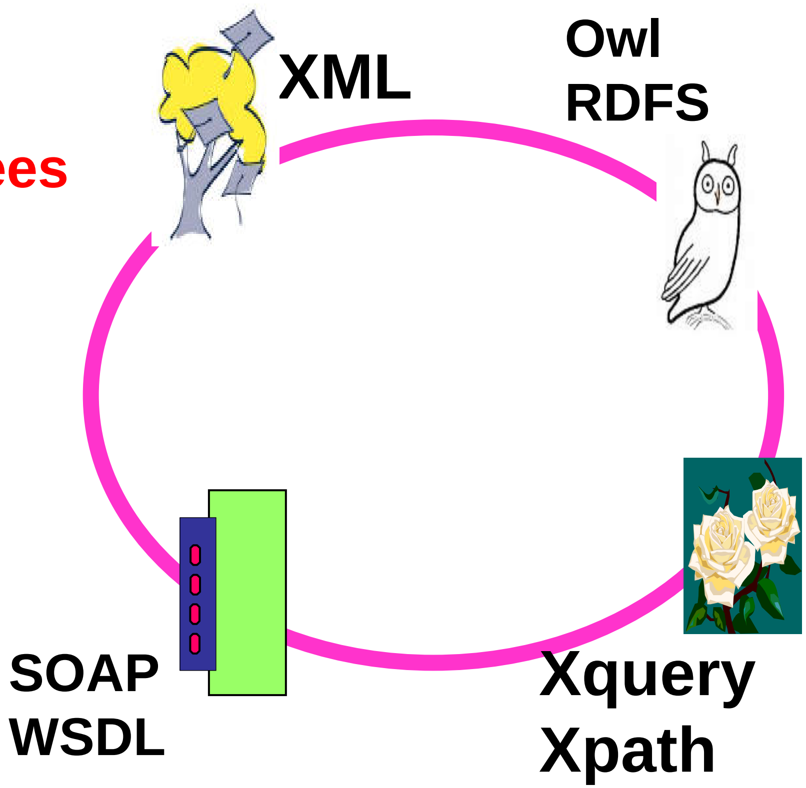
- Web services

Query languages

- XPath and XQuery

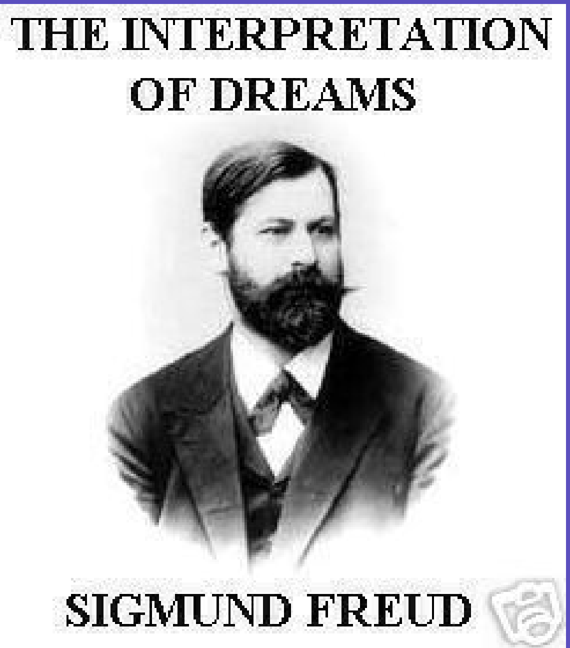
Knowledge representation

- Owl or RDF/S



Uniform access to information…

…the dream for distributed data management



To do what? Answer queries precisely

Query: what is the email of the director of INRIA Sophia?

Yesterday's Web: a human asks the query, gets a list of pages and browse them to find the answer

Tomorrows Web:

To: ? the director of INRIA Sophia ?

my Webmail finds Gerard.Giraudon@inria.fr

How: with more semantics

- The web site of INRIA Sophia should specify the meaning of web pages and services

Challenging problems

Data management

Knowledge management

Natural language processing

- The main source of information is text

Distributed system

Data mining

Personal bias: data management and distributed systems

- A logic for distributed information management
- An algebra for distributed information management
- Self administration

The success of databases

Main impact of mathematical logic in computer science

Slogan: First-order logic on the everybody's desk

A huge industry (Oracle server, IBM DB2, MS Access…)

Crux: specify declaratively your needs, not by some complicated code

- Easier to specify
- Cleaner code
- Optimizable queries

First-order logic
Tarski/Coddd's algebraïzation
Rewrite-based optimization

Relational systems

Thesis: do the same for distributed information management!

The success of the relational model, i.e., of 2D-tables on a server :

1. A logic for defining tables
2. An algebra for describing query plans over tables

By analogy, we need for trees in a P2P system

1. A logic for defining distributed tree data and data services
2. An algebra for optimizing queries over trees/services

XQuery is fine for local XML processing and publishing but not for distributed data management

On-going work – ActiveXML –

Guidelines for logic and algebra

Manage trees in a distributed setting

- Mention explicitly the topology if desired
- Ignore it if preferred

Support for streams

- Essential for subscription services
- Also necessary to support recursion

Handle both extensions and intensions

- Extensional information: e.g., documents and xml pages
- Intensional information (views): web services
- Seamless transition between them
  - What is the email of the director of INRIA Sophia?
  - Find the name in Web page and call a service to obtain email

Active XML:
a logic for distributed data management

Joint work with
Omar Benjelloun (Google) and Tova Milo (Tel Aviv)

INRIA

The basis

AXML is a declarative language for distributed information management and an infrastructure to support the language in a P2P framework

Simple idea: ***XML documents with embedded service calls***

Intensional data

- Some of the data is given explicitly whereas for some, its definition (i.e. the means to acquire it when needed) is given

Dynamic data

- If the data sources change, the same document will provide different information

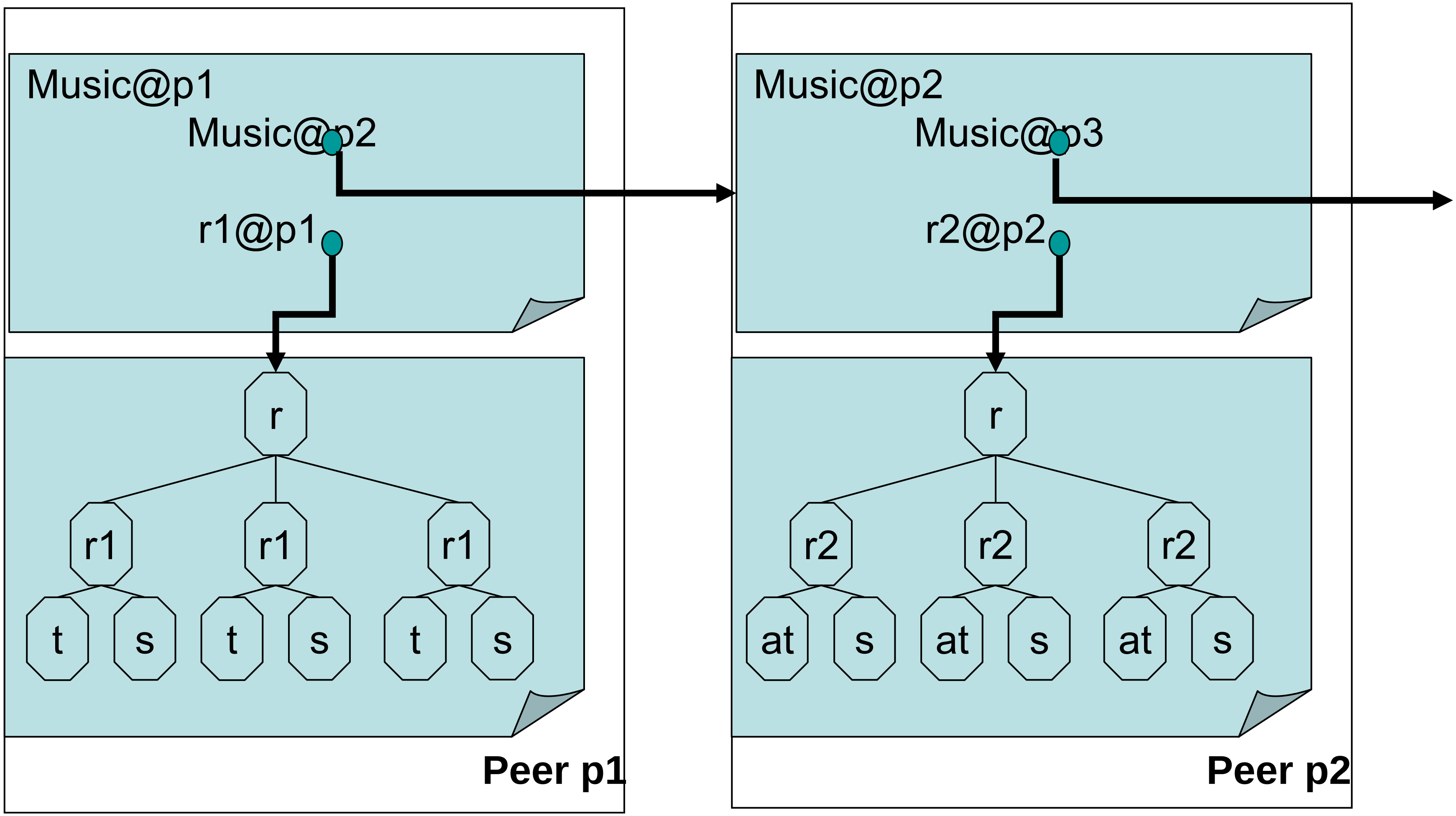
Example
(omitting syntactic details)

```
<resorts state='Colorado'>
   <resort>
      <name> Aspen </name>
      <sc> Unisys.com/snow("Aspen") </sc>
      <depth unit="meter">1</depth>
      <hotels ID=AspHotels > ….
      Yahoo.com/GetHotels(<city name="Aspen"/>)
      </hotels>
   </resort>   …
</resorts>
```

May contain calls
to any SOAP web service :
- e-bay.net, google.com…

to any AXML web services
- ***to be defined***

ActiveXML: XML documents with embedded service calls



Marketing $\vee$ Philosophy

Active answer = intensional and dynamic and flexible

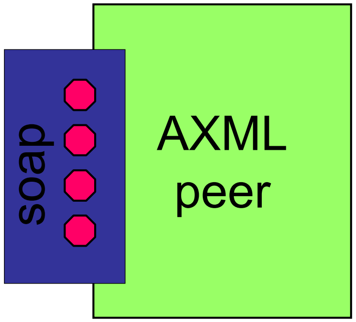
Embedding calls in data is an old idea in database

Manon: What's the capital of Brazil?
Dad: Let's ask Wikipedia.com!

Manon: How do I get a cheap ticket to Galapagos?
Dad: Let's place a subscription on LastMinute.com!

Manon: What are the countries in the EC?
Dad: France, Germany, Holland, Belgium, and hum… Let's ask YouLists.com for more!

Active XML peer



Peer-to-peer architecture

Each Active XML peer

- **Repository**: manages Active XML data
- **Web client**: calls the services inside a document
- **Web server**: provides (parameterized) queries/updates over the repository as web services

Exchange of AXML instead of XML

What is an AXML peer?

Any connected device or software with some information to share

A key issue: call activation

When to activate the call?

- Explicit pull mode: active databases
- Implicit pull mode: deductive databases
- Push mode: query subscription

What to do with its result?

How long is the returned data valid?

- Mediation and caching

Where to find the arguments?

- Under the service call: XML,XPATH or a service call

Another key issue: what to send?

Send some AXML tree t

- As result of a query or as parameter of a call

The tree t contains calls, do we have to evaluate them?

- If I do, I may introduce service calls, do we have to evaluate all these calls before transmitting the data?

Hi John, what is the phone number of the Prime Minister of France?
• Find his name at whoswho.com then look in the phone dir
• Look in the yellow pages for deVillepin's in phone dir of www.gov.fr
• (33) 01 56 00 01

Active XML
cool idea – complex problems

- Blasphemous claim:
  - Active XML is the proper paradigm for data exchange!
  - Not XML + not XQuery
- Brings to a unique setting
  - distributed db, deductive db, active db, stream data
  - warehousing, mediation
- This is unreasonable? Yes!
- Plenty of works ahead… to make it work
- But first, the algebra

Active XML algebra for distributed data management

Joint work with Ioana Manolescu (INRIA-Saclay)

INRIA

Motivation

Relational model: **centralized tables**

optimization: algebraic expression and rewriting

Active XML model: **distributed trees**

optimization: algebraic expression and rewriting

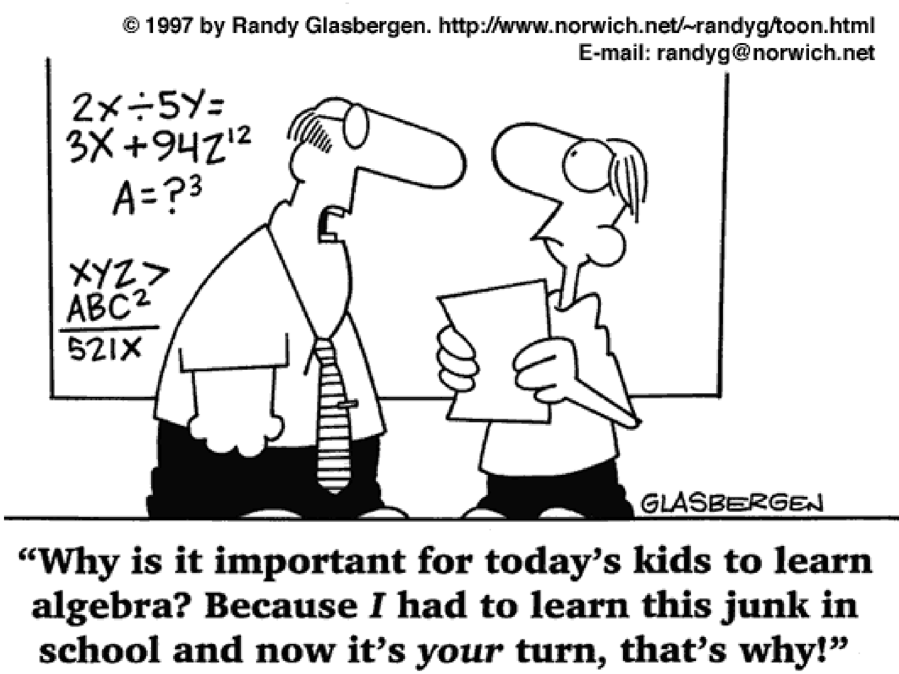
Distributed query optimization based on algebraic rewriting of Active XML trees

Based on experiences with AXML optimization

ActiveXML algebra

Why an algebra?

- Specify a query declaratively
- Compile it into a distributed query plan
- Optimize the query plan in a distributed manner
- Exchange query plans between peers



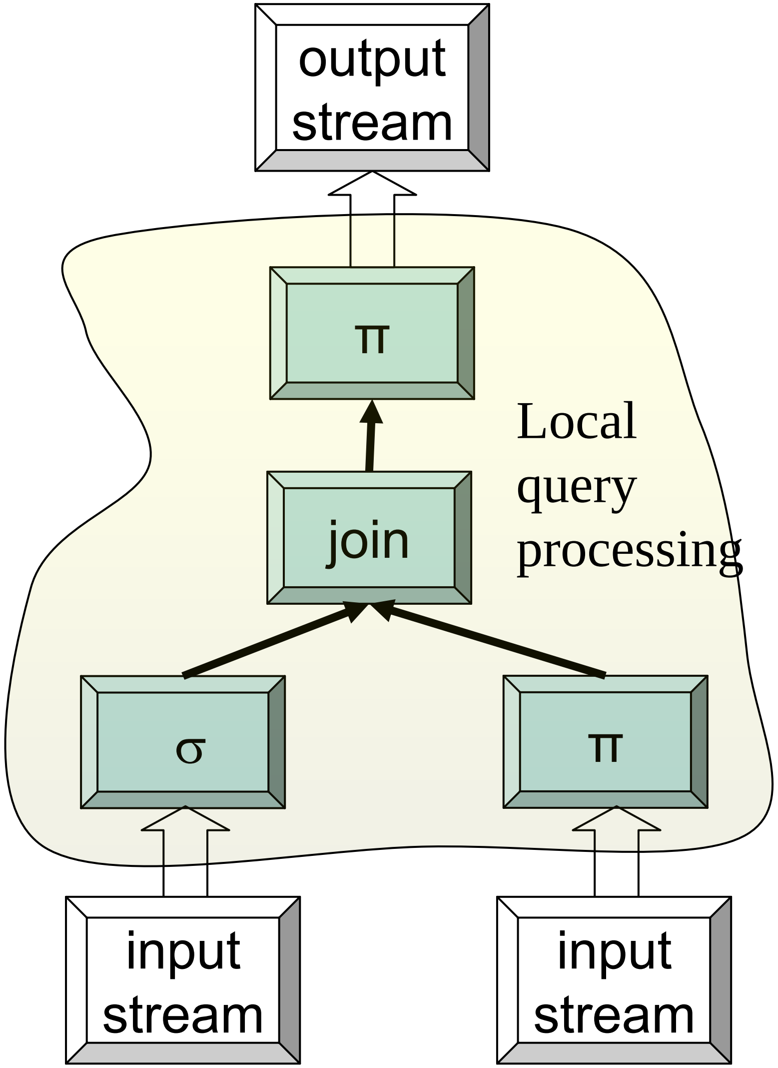
Example: title of songs by Carla Bruni?

Active XML peers

We focus on positive AXML
- Set-oriented data
- Positive/monotone services

Services = tree-pattern-query-with-join queries

Services produce streams
- Optimized by a local query optimizer
- Evaluated by a local query processor
- Out of our scope



The problem

An AXML system

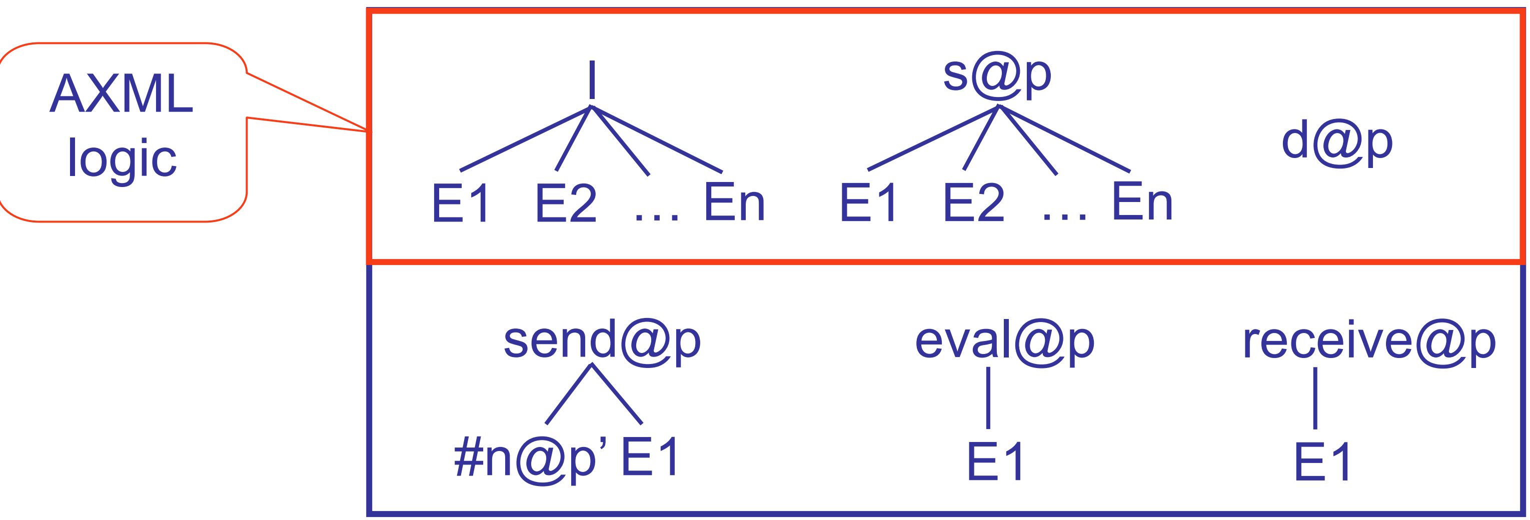
- A set of peers
- For each peer a set of documents and services
- Extensional data is distributed
- Intensional data (knowledge) is distributed
  - Defined using query services (TPQJ queries)
  - These services are generic: any peer can evaluate a query

A query q to some peer

Evaluate the answer to q with optimal response time

AXML algebra

(AXML) algebraic expressions:



Each such expression lives at some peer
Includes the AXML trees

The AXML algebra – conclusion

Captures distributed XML query processing/optimization

Based on a communication model *a la* CCS

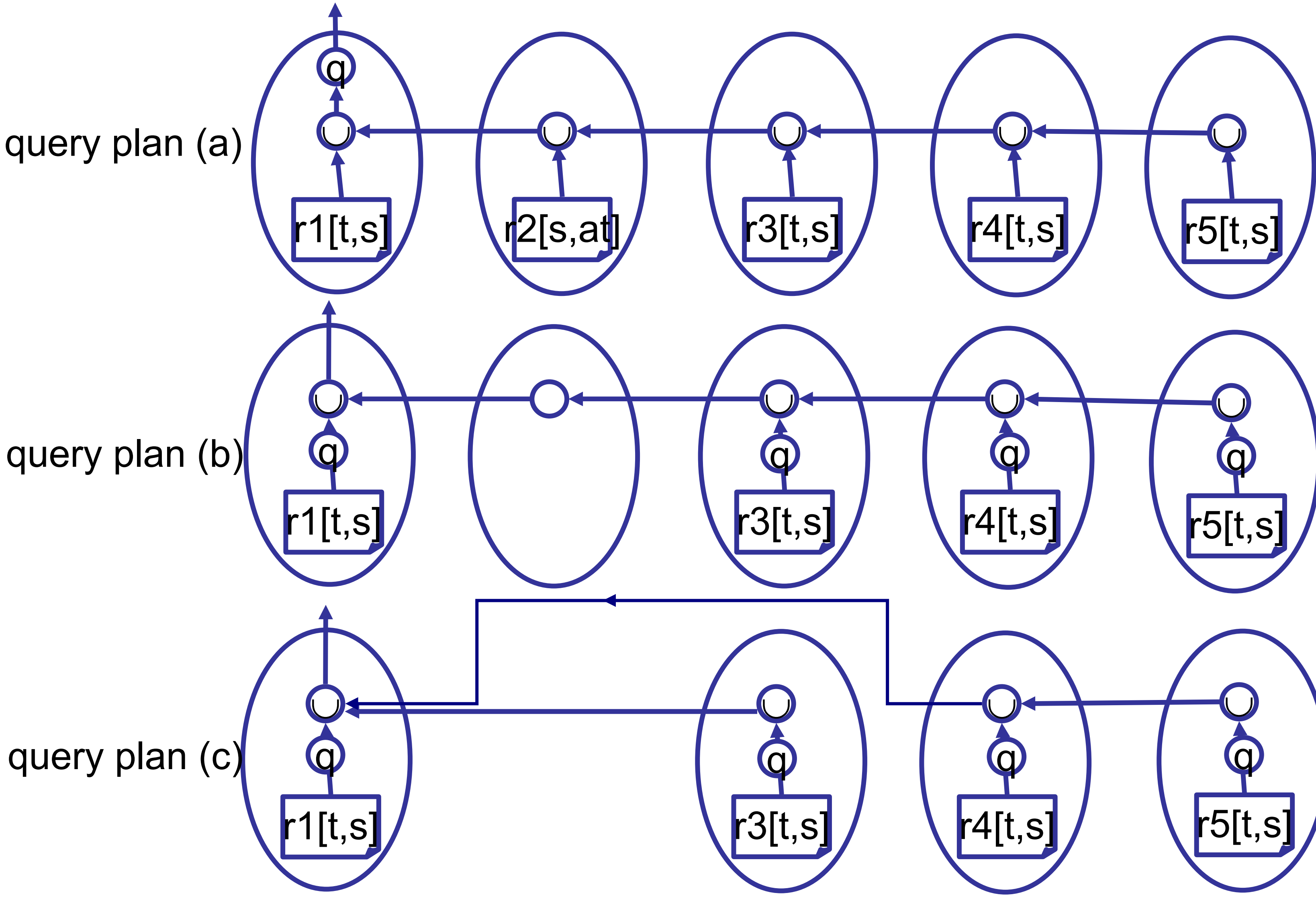
Algebraic – stream-oriented

Orthogonal to the local XML query optimizer

Orthogonal to the network support (DHT, small world etc.)

What is not yet available? A cost model and heuristics

$q = \prod_t \sigma_{s=\text{"Bruni"}} \cup (r_i)$ where $\cup$ = outer join



Self-administration

Joint work with Alkis Polyzotis (UCSC)

INRIA

The data ring is self administered

No experts

- The users of the system, e.g., scientists, are not experts

No central authority that can be responsible for administration

No centralized servers

Requirements

- Ease of deployment (zero-effort)
- Ease of administration (zero-effort)
- Ease of publication (epsilon-effort)
- Ease of exploitation (epsilon-effort)
- Participation in community building notably via annotations

Happy info admin

What should be made automatic

Self-statistics from the monitoring of the data ring

- Logs and statistics on system operation
- Models of system performance

Self-tuning based on the self-statistics

- Enrichment of physical layer with access structures
- Decide to install access structures: indexes, views, etc.
- Control replication of data and services

Self-healing

- Recovery from peer and network failures
- Recovery from unexpected anomalies

And automatic file management

Monitoring and surveillance

Essential aspects

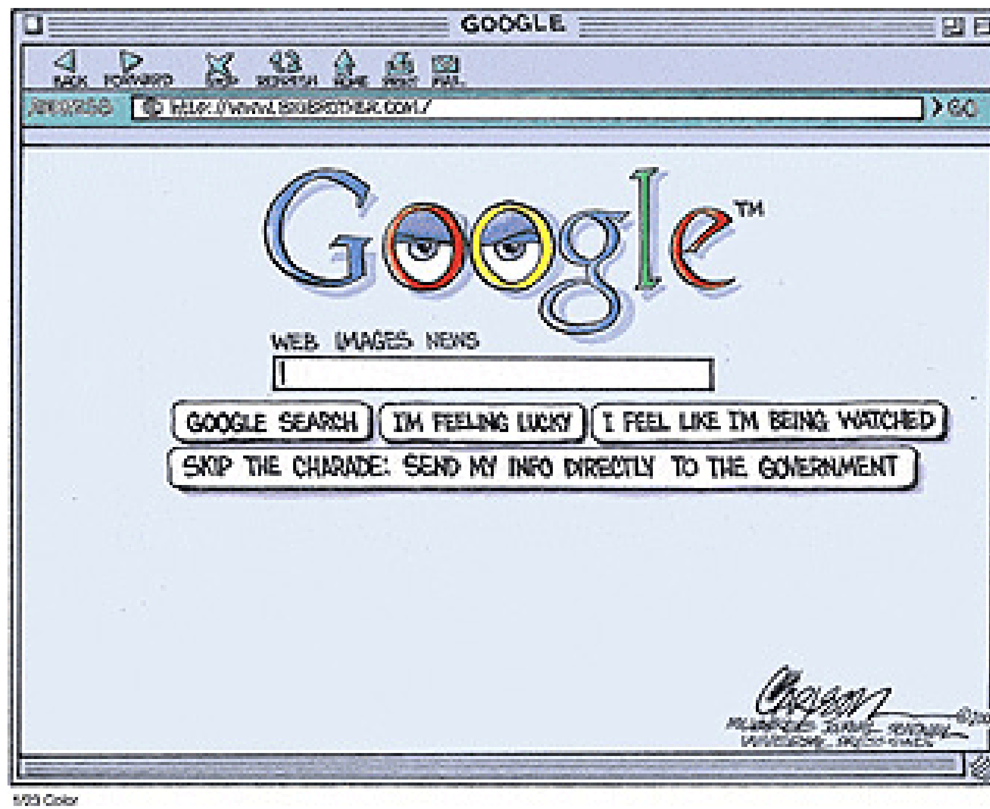
Monitor web service calls, database updates, RSS feeds

Produces a stream of events

Query subscriptions

Info-surveillance

Self-statistics and tracing



Any hope?

Technology exists (database self-tuning, machine learning, etc.)

But self-tuning for databases has advanced very slowly

Why can this work?

1. There is no alternative (for db, this was just a cool gadget)
2. KISS (keep it simple stupid!)
3. The power of parallelism

   This is assuming lots of machine have free cycles (true) and bandwidth is generous (not always true)

6. Conclusion

The data ring

Several challenging and interesting problems

Leverage existing technology

Need to tackle lots of open issues

Related systems

Structured P2P nets: Pastry, Chord

Content delivery net: Coral, Akamai

XML repositories: Xyleme, DBMonet

Multicas systems: Avalanche, Bullet

File sharing systems: BitTorrent, Kazaa

Pub/Sub systems: Scribe, Hyper

Distributed storage systems: OceanStore, GoogleFS

Etc.

Lots of system developments but fundamental research is left behind

Personal on-going work

Logic and algebra for distributed data management

Algebraic optimization

P2P indexing

On-line tuning

Imprecise data and exploitation of web services

Projects

- EC Project Edos on Linux distribution
- ANR platform webContent on Web surveillance
- ACI docFlow on P2P monitoring and data-centric workflow

Many open issues

P2P

- query optimization
- Monitoring
- semantic integration
- P2P inference
- system analysis and verification…

More

data, semantics, peer, dynamicity, interactivity

Very complex problem

be modest and have fun

Many open issues

Pick your favorite problem for data or knowledge management and study it in a P2P setting

- With gigabytes of data and thousands of peers

If you find it boring, consider it

- With terabytes of data and/or millions of peers

Merci