QUERYING AND VISUALIZING THE INTERNET

- Introduction
  - An overview of the Internet
  - Accessing information on the Internet
- The World-Wide Web
  - Hypertext
  - The http protocol
  - The HTML language
  - Finding information in the Web
- Querying the Web
  - Resource discovery
  - Global information systems
  - Web query languages
- Visualizing the Web
  - Large graph visualization
  - The Hy+ visualization system

What is the Internet?

- **Def**: An internet is a federation of computer networks that speak the TCP/IP protocols
- **Def:** The Internet is a global internet that connects homes, schools, universities,corporations, and governments in over 60 countries.

Motivation:

- Data communication has become a fundamental part of computing
- No single network hardware is good for all applications
- Users desire universal interconnection

History

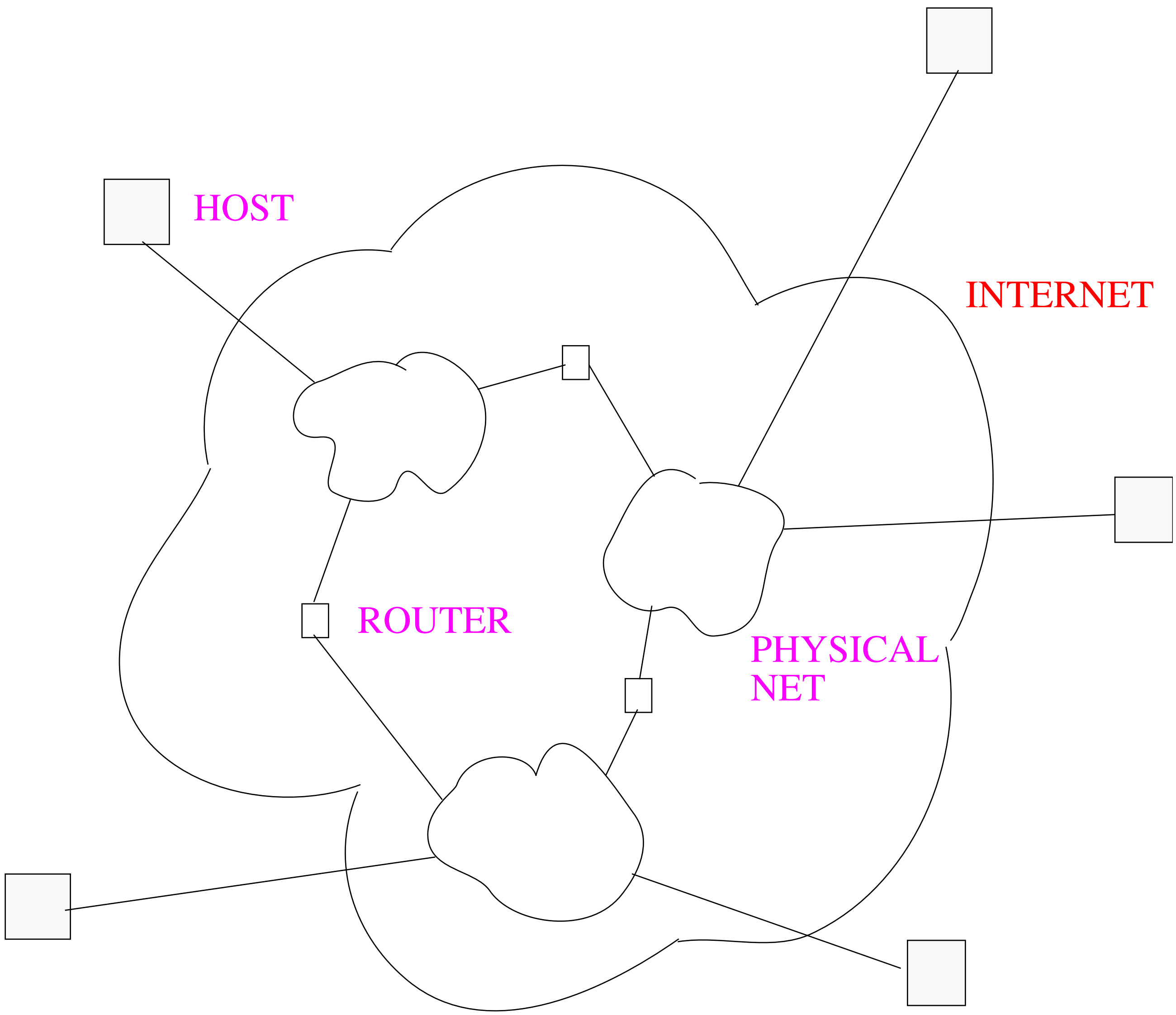
•1970's	Packet switching
	ARPANET
•1980	Conversion to TCP/IP
	TCP/IP bundled with Berkeley Unix
•1985	NSFNET, 56 Kbps backbone
	Domain Name System
•1992	Internet Society
	ANSNET, 45 Mbps backbone

Internet Architecture

Cooperative interconnection of networks

Networks communicate through routers

All networks are equal



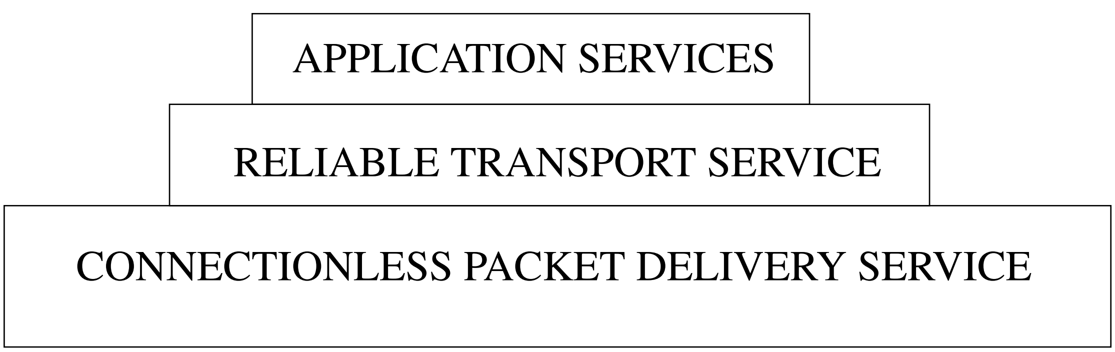
Physical Networking

- Packet-switching (connectionless)
- Circuit-Switching (connection oriented)
- LANs:
  - Ethernet
  - Token Ring
  - FDDI
  - ATM

The TCP/IP Protocols

Transmission Control Protocol/Internet Protocol

Other protocols in suite: UDP, ARP, RARP, ICMP, IGMP, ...



IP: Connectionless packet delivery

- Datagram delivery
- Connectionless
- Routing
- Unreliable (best-effort)

IP ADDRESS:

Hosts have unique 32-bit IP addresses

Multihome hosts have one address per network connection

```
db.toronto.edu128.100.2.100
hplabs.hp.com15.255.176.47
math.tau.ac.il132.67.64.4
cs.toronto.edu128.100.1.105,
128.100.1.13,
128.100.2.81, 128.100.2.15
```

TCP: Reliable Stream Transport

- Stream oriented
- Virtual circuit connection between ports
- End-to-end Reliability by ack and retransmission
- Full Duplex

Getting on the Internet

Three roles:

- Information Provider:

  Makes information available for users

- User:

  Consumes information and service

- Connection Provider:

  Provides the network connection

Types of Connections

- Full-time IP connection

- Dial-up access to a networked computer

- Dial-up IP link

Internet Services

- Electronic Mail
- File transfer: ftp
- Remote login: telnet
- File navigation: gopher
- Hypermedia: World Wide Web

FTP

```
twist> ftp ftp.apple.com

Connected to bric-a-brac.apple.com.

220 bric-a-brac.apple.com FTP server (Version 4.11
Thu Jun 4 10:17:00 PDT 1987) ready.

Name (ftp.apple.com:mendel): anonymous

331 Guest login ok, send ident as password.

Password:

230 Guest login ok, access restrictions apply.

ftp> cd pub/standards

ftp> get read_me_stds.txt

200 PORT command successful.

150 Opening ASCII mode data connection for
read_me_stds.txt (1673 bytes).

226 Transfer complete.

local: read_me_stds.txt remote: read_me_stds.txt

1708 bytes received in 0.0055 seconds (3.1e+02
Kbytes/s)

ftp> bye
```

ARCHIE

- Index of ftp archives

```
twist> archie scriptx

Host ftp.apple.com

 Location: /dts/multimedia
 DIRECTORY drwxrwxr-x 512 Sep 27 1994 scriptx

twist>
```

GOPHER

- Navigation of information resources

```
Internet Gopher Information Client 2.0 pl11

 Root gopher server: gopher.utoronto.ca

 1. About the UTORinfo Gopher service/
 2. News, Events, Weather/
 3. Phone Books/
 4. List of Gopher sites at U of T/
 5. Worldwide Gopher and Information Servers/
```

Hypertext/Hypermedia

- Non-sequential reading and writing
- Interlinked pieces of text or other information
- Common in on-line help/documentation systems

Hypertext History

Memex, Vannevar Bush, 1945

> *"a device in which an individual stores his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility."*
>
> *"...associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another ... The process of tying two items together is the important thing."*

Augment/NLS, Doug Engelbart, 1962-1976

> First work in office automation and document processing. Invented the mouse, among other things.

Xanadu, Ted Nelson, 1965-present

> Coined word "hypertext." Repository of everything ever written.

Experimental systems in the 80's: KMS, Intermedia, NoteCards

HyperCard, Bill Atkinson, 1987

> Free with every Macintosh 1987-1992

First ACM Conference on Hypertext, 1987, European Conference, 1989

World Wide Web, Tim Berners-Lee, CERN, ~1990

Mosaic, Marc Andressen, NCSA, 1993

World Wide Web

- Hypertext/Media over the Internet
- Documents written in HTML with embedded links
- HTTP protocol for client/server communication
- Client browsers:NETSCAPE, Mosaic, Lynx, HotJava,...
- Single client can access

  HTML documents

  Gopher

  ftp

  WAIS

  Usenet news

  SMTP mail

  local files

HTML

Hypertext Markup Language: a form of SGML

Text marked up with tags defining logical units:

- Title
- Heading
- Paragraph
- List
- etc.

Logical units mapped to specific display properties by browsers.

Example:

```
<h1> This is a heading </h1>
```

Will be displayed *centered, bold, large.*

```
<HTML>

<HEAD> <TITLE>The Hy+ System Project Members</TITLE>

</HEAD>

<BODY>

<H2> Hy+ Project Members </H2>

<HR> <A HREF = “people/consens/consens.html”>

<IMG ALIGN = BOTTOM SRC = “pictures/mar.gif”> Mariano
Consens </A>

<HR> <A HREF = “people/hasan/hasan.html”> <IMG ALIGN =
BOTTOM SRC = “pictures/mas.gif”> Masum Hasan </A>

<HR> <A HREF = “people/mendelzon/aom.html”> <IMG ALIGN =
BOTTOM SRC = “pictures/alb.gif”> Alberto Mendelzon </A>

<HR> <A HREF = “people/noik/Welcome.html”> <IMG ALIGN =
BOTTOM SRC = “pictures/noi.gif”> Manny Noik </A>

<HR> <A HREF = “people/vista/vista.html”> <IMG ALIGN =
BOTTOM SRC = “pictures/dim.gif”> Dimitra Vista </A>

<HR> <P>

<A HREF=”Hy.html”>

<IMG SRC=”gifs/back-hand.gif”> Back to Hy+ Home page </
A>

</BODY>

</HTML>
```

Document Head

Meta-information about the document

In the future, servers will support accessing document head only

Elements:

`TITLE`	Title of document
`ISINDEX`	Document is searchable
`BASE`	Original URL of the document
`NEXTID`	Unique document identifier
`LINK`	Relationship between documents
`META`	Other meta-information

LINK Element

```
<HEAD>
<Title> The Book of Everything </Title>
<LINK HREF="chapter1.html" REL="subdocument">
</HEAD>
```

`REL` attribute: describes type of relationship

`REV` attribute: same relationship in reverse direction

(not widely supported, not standardized)

Examples:

- `annotation`
- `reply`
- `embed`
- `precedes`
- `subdocument`
- `made`
- `owns`
- `supports`
- `approves`
- `refutes`

Anchor elements

Specify destination of a hypertext link or mark text as the target of a link

```
<A> ... </A>
```

- Specify destination:

```
<A HREF="http://db.toronto.edu:8020/Hy.html">

The Hy+ Page

</A>
```

Can also take `REL` and `REV` attributes

- Mark target:

```
<A NAME="people"> List of project members </A>
```

Can be referenced by:

```
<A HREF="#people> The people </A>
```

or:

```
<A HREF="http://db.toronto.edu:8020/
Hy.html#people>

The Hy+ People

</A>
```

Universal Resource Locators

URL syntax: (Internet RFC 1738)

<scheme>:<scheme-specific-part>

Current schemes supported:

`ftp`	File Transfer protocol
`http`	Hypertext Transfer Protocol
`gopher`	The Gopher protocol
`mailto`	Electronic mail address
`news`	USENET news
`nntp`	USENET news using NNTP access
`telnet`	Reference to interactive sessions
`wais`	Wide Area Information Servers
`file`	Host-specific file names
`prospero`	Prospero Directory Service

HTTP URL's

`http://<host>:<port>/<path>?<searchpart>`

Examples:

```
http://db.toronto.edu:8020/Hy.html
http://www.w3.org/WWW/Protocols/Overview.html
http://www2.infoseek.com/Titles?qt=Maradona
```

HTTP PROTOCOL

Hypertext Transfer Protocol

Stateless, 8-bit communication (ISO Latin 1)

Connection has four stages:

- Open connection
- Send request
- Receive response
- Close connection

```
Request:
Method URI ProtocolVersion
[Header Lines]
[Data Lines]
```

```
Response:
StatusLine
[Header Lines]
[Data Lines]
```

Client Request

```
Method URI ProtocolVersion

[Header Lines]

[Data Lines]
```

Example:

```
GET /people/Mendelzon.html HTTP/1.0
```

- Methods:

```
GET/PUT, HEAD, CHECKIN/CHECKOUT, SHOWMETHOD,
DELETE, POST,LINK/UNLINK, TEXTSEARCH, SPACEJUMP
```

- Request Headers:

  Description of the client's capabilities, e.g.

  ```
  Accept: text/plain
  Accept: image/jpeg
  ```

- Data:

  Usually results of filling out a `FORM`

Server Response

```
StatusLine
[Header Lines]
[Data Lines]
```

- Example:

```
HTTP/1.0 200 OK
Date: Thursday, 13-Jul-95 14:28:37 GMT
Server: NCSA/1.3
MIME-version: 1.0
Content-type: text/html
Last-modified: Friday, 7-Jul-94 18:00:05 GMT
Content-length:
<blank line>
<html>
<head><title> A test file </title></head>
<body>
<h1>This is a test</h1>
</body>
</html>
```

The CGI Interface

Common Gateway Interface

Communication between servers and server-side gateway programs

Some uses:

- Database access
- Text retrieval
- User registration
- Transaction processing

Example:

`http://some.site.edu/cgi-bin/lookup?some_query`

Querying the Internet

Three cultures

- Resource Discovery --building the access methods

  Gathering: net robots

  Indexing and Searching

  Examples: *Lycos, Web Crawler, OpenText, WWW Worm*

  The *Harvest* Architecture

- Global Information Systems --designing and using views

  The Information Manifold

  TSIMMIS

- Web Query Languages --expressing and processing queries

  W3QS

  Weblog

Resource Discovery

Culture: Networking/Information Retrieval

IRTF Research Group on Resource Discovery and Directory Service

Bowman *et al*:

Scalable Internet Resource Discovery: Research Problems and Approaches

CACM 37:8, August 1994

- Information interface layer

  Presenting information to users

- Information dispersion layer

  Replicating, distributing, and caching

- Information gathering layer

  Collecting and correlating

Scalability Problems

Dimension	Layer	Problems	Research Focus
Data volume	Interface	Information overload	Topic specialization, visualization, query
User base	Dispersion	Insufficient replication, manual distribution topology	Massive, adaptive replication; object caching
Data diversity	Gathering	Data extraction; low data quality	Operation mapping; data mapping

Data Diversity

- Information in Internet repositories is:

  heterogeneous

  inconsistent

  incomplete

- Data mapping

  generate aggregate repository from multiple sources

  (materialized view)

  example: `Netfind` extracts user directory information from DNS, finger, and SMTP.

- Operation mapping

  construct "gateway" between existing systems

  (non-materialized view)

  example: `savvysearch` sends query to multiple index servers

User Base

- Server instrumentation

  servers should monitor their query streams to decide where to place replicas, when to create specialized sub-services, and to detect looping clients

- Massive replication

  tools that search information spaces that are hard to partition (e.g. Archie) need massive replication: to get 5-second response at peak times, with current Internet size, would require 1800 servers instead of 30

- Object caching

  hierarchical caching scheme would reduce network load, improve response time, and protect against looping clients that retrieve the same object repeatedly

- Search service customization

  tracking search history

  use context informationin search, e.g. who is the user

  customize ranking, e.g. by quality, reliability, cost

Data Volume

- Multimedia will cause orders of magnitude increase in volume...

- But size and number of “searchable items” will not grow as fast

  raw sensor data will not be searchable

  searching image or audio files desirable, but still too slow to support

- Organizing/browsing breaks down, searching needed

  organizing done by someone else: e.g. *Cavalry* is at the same level as

  Mathematical Sciences

  browsing depends heavily on the quality of the organization

- Searching requires indexing

  trade index granularity vs. size

  is it feasible to build low granularity indexes of the whole Internet?

  specialize: 99% of data is of no interest to 88% of users

Information Gathering

- **Robots** (Wanderers, Spiders, Crawlers,...)

  A program that traverses the Web graph by retrieving a document and recursively retrieving all referenced documents

  Static, does not really "crawl"

  Applications:

  -Resource discovery and index creation

  -Statistical analysis

  -Link maintenance

  -Mirroring

  High bandwidth and server load

  Not everyone likes them: *robot exclusion* standard

  Examples:

  WWW Worm, Lycos, WebCrawler, InfoSeek, OpenText

Information Gathering (cont.)

- Summary-based Gathering

  IETF Templates:

```
Template-Type: SERVICE

Title: The ArchiePlex Archie Gateway

URL: /public/archie/archieplex/archieplex.html

Description: A Full Hypertext interface to Archie.

Keywords: Archie, Anonymous FTP.


Template-Type: DOCUMENT

Title: The Perl Page

URL: /public/perl/perl.html

Description: Information on the Perl Programming
Language. Includes hypertext versions of the Perl 5
Manual and the latest FAQ.

Keywords: perl, programming language, perl-faq
```

- Gatherers like ALIWEB (Archie-Like Web Searching) access the summaries only, reducing the load

Information Gathering (cont.)

- HARVEST

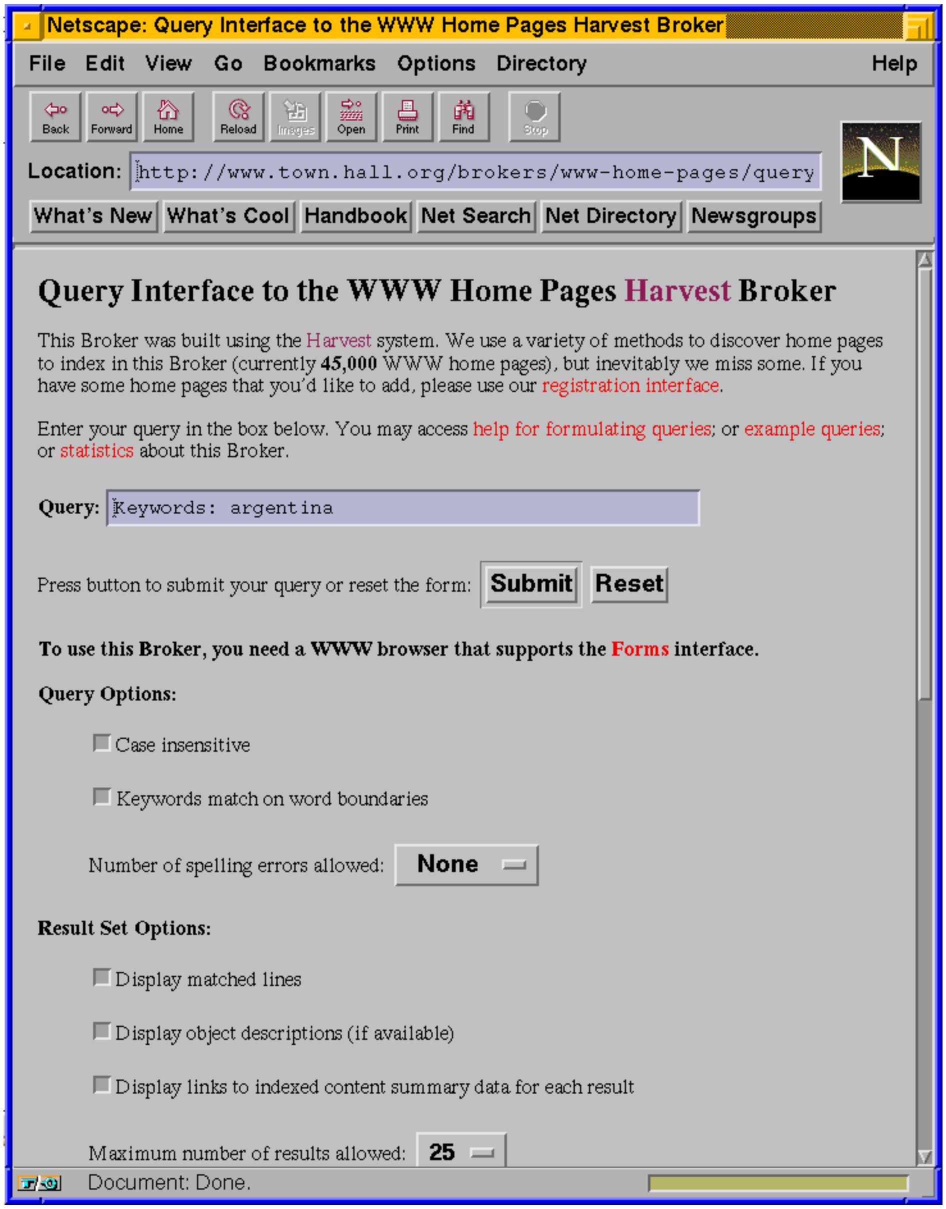
  Gatherer/Broker Architecture:

  gatherer can run at provider site, feed multiple brokers

  Summary Object Interchange Format :

  IETF/IAFA format augmented with BIBTEX

Information Interface



Global Information Systems

Culture: Databases/Knowledge Representation

TSIMMIS (Stanford), Garlic (IBM Almaden), Information Manifold (AT&T)

Differences with traditional heterogeneous database work:

- Large number of data sources
- Structured, semi-structured, unstructured data sources
- Human interaction

The Information Manifold

Levy, Sagiv, and Srivastava, AT&T Bell Labs

Exploit KB technology to represent *descriptions* of information sources.

Example: information about papers written by colleagues

Conventional tools: store list of FTP addresses

KB approach: store properties of the colleague (area of expertise, institution, etc.)

Query: What are the FTP sites of query optimization researchers who had a paper in SIGMOD this year?

- Site description knowledge base in CLASSIC
  - Concepts, roles, relations
  - Horn rules
  - Constraints
- Query language
  - Horn clauses + description logic
- Determine minimal sets of sites for a given query

Example

Multiple sources of airline flight and phone number information

- KB concepts:

  customer, divided into disjoint *business* and r*esidential*

  Business customers have *business_type* role

  travelAgent defined by

  (business $\wedge$ *(fills business_type "Travel"))*

- KB relations:

  q*uote(Agent,AirlineSrc,Dst,Price,Date)*

  dir(Cust,Area,PhoneNumber)

- Integrity Constraints:

  quote.Agent is a travelAgent

  quote.Price is nonnegative

  etc

- Site Descriptions:

  Air Canada site stores relations AC_schedule and AC_quote

  AC_schedule(Flt,Src,Dest) $\wedge$ *AC_quote(Agent,Flt,Price,Date)* $\Rightarrow$

  quote(Agent,'AirCanada',Src,Dest,Price,Date)

Query Processing

Find agents in the 416 area code who sell tickets from Toronto to Rome for under \$1000, and their phone numbers.

$$quote(Agent,\ \text{'Air Canada'},\ \text{'YYZ'},\text{'FCO'},\ Price,\ Date) \wedge$$
$$dir(Agent,\ 416,\ Number) \wedge Price < 1000 \Rightarrow$$
$$ans(Agent,\ Number)$$

- Determine relevant portions of the domain relations
- Determine minimal set of portions of the information sources
- Formulate subqueries to each information source
- Assemble the answers to the subqueries

Theorem: The relevant portions of the source relations can be determined in time polynomial in the size of the knowledge base.

- Completeness and Redundancy

  Suppose Air Canada flights are stored at more than one site; how do we avoid looking at all the sites?

  Extend the language to allow *completeness* statements:

  $$quote(Agent,\text{'AirCanada'},Src,Dst,Price,Date) \Rightarrow$$
  $$AC\_schedule(Flt,Src,Dest) \wedge AC\_quote(Agent,Flt,Price,Date)$$

Web Query Languages

Culture: Databases/Graph Queries

- Previous work:
  - Querying structured documents
  - Graph queries
  - Hypertext queries
- New challenges:
  - No structure, or unknown structure
  - “Infinite” database
  - Cost constraints
- Requirements:
  - SQL-like
  - High expressive power, including cost constraints
  - Combine structural and content-based queries
  - Optimizable
  - Take advantage of existing index infrastructure

W3QLS

Konopnicki and Shmueli, VLDB 1995

Find Latex articles written by A. Einstein. Use indexes listed in file `ImportantIndexes.url`.

```
SELECT cat n2 > result;

FROM n1,l1,n2;

WHERE

        n1 in ImportantIndexes.url;

        FILL n1.form AS IN ImportantIndexes.fil
WITH

            keyword = "A. Einstein";

        SQLCOND (n2.format = Latex) AND

            (n2.author = "A. Einstein");
```

- Graph pattern `n1,l1,n2`
- Form filling from template file `ImportantIndexes.fil`
- `SQLCOND` for content/attribute conditions

W3QL (cont.)

Maintain list of pointers to Latex articles written by Einstein.

```
SELECT CONTINUOUSLY SQLPRINT n2.URL;
FROM n1,l1,n2;
WHERE
        n1 in ImportantIndexes.url;
        FILL n1.form AS IN ImportantIndexes.fil
WITH
            keyword = “A. Einstein”;
        RUN learnformat IF n1.form UNKNOWN;
        SQLCOND (n2.format = Latex) AND
            (n2.author = “A. Einstein”);
EVALUATED EVERY week;
```

W3QL (cont.)

Find articles cited in the first chapter of the book stored at `BookIndex.html`.

```
SELECT cat art > result;

FROM ind,l1,chap,l2,ref,l3,art;

WHERE

        SQLCOND (ind.url =

            "http://cs.technion.ac.il/
BookIndex.html")

        AND (chap.url = /.Chapter-1.html/)

        AND (l2.HREF = /.\#$l3.NAME/);

USING BFS
```

- Book file points to Chapter file points to References (for whole book).
- Chapter 1 file points to References with a link of the form `URL#label`, where label is the name of the article cited.
- Match the `#label` with the link from the References file to the Article file
- Breadth-First Search specified

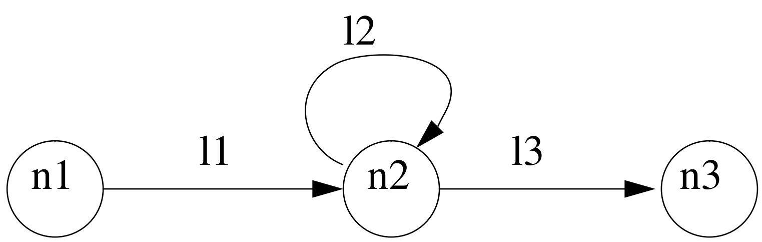
W3QL (cont.)

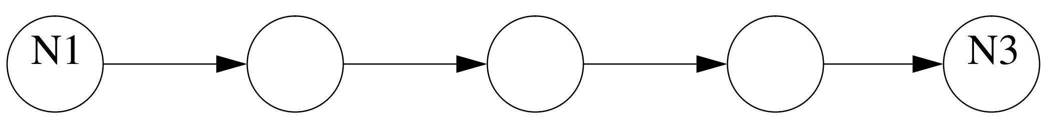
- SQLCOND: A separate program that evaluates conditions on attributes. Each file format has its own attributes:

- Graph patterns

    Used in the FROM clause

n1,l1,(n2,l2),l3,n3





Weblog

Mendelzon and Milo, forthcoming

- Based on structured document query language
- Regular expression patterns
- Two-part queries: context + retrieval
- Context may include cost bounds
- Theory: measuring *query locality*

Querying Structured Documents

Christophides, Abiteboul, Cluet and Scholl, SIGMOD 1994

Motivation: bring benefits of OODB technology to document management:

- recovery
- concurrency control
- high-level queries

Approach:

- Map SGML DTD's to OODB schemas
- Extend O_2SQL query language to permit:

  Text pattern matching - `contains` predicate

  Retrieval without precise knowledge of structure - `PATH` and `ATT` sorts

Other work:

Gonnet and Tompa, VLDB 1987

Bertino, Rabitti and Gibbs, ACM TOIS, 1988

Güting, Zicari, and Choy, ACM TOIS, 1989

Beeri and Kornatzky, ECHT 1990

Minohara and Watanabe, FODO 1993

An SGML Document Type Definition

```
1.<!DOCTYPE    article[
2.<!ELEMENT    article --
               (title,author+,abstract,section+)>
3.<!ELEMENT    title        (#PCDATA)>
4.<!ELEMENT    author       (#PCDATA)>
5.<!ELEMENT    abstract     (#PCDATA)>
6.<!ELEMENT    section

((title,body+)|(title,body*),subsectn+)))>
7.<!ELEMENT    subsectn     (title,body+)>
8.<!ELEMENT    body         (figure|paragr)>
9.<!ELEMENT    figure       (picture,caption?)>
10.<!ATTLIST   figure       label ID #IMPLIED>
11....
```

Example Schema

```
class Article tuple (title: Text,
        authors:list(Text), abstract: Text,
        sections: list(Section))


class Section union
         (a1: tuple(title: Text, bodies:
list(Body)),
          a2: tuple(title: Text, bodies: list(Body),
             subsectns: list(Subsectn)))


class Subsectn tuple (Title: Text, bodies:
list(Body))

class Body union (figure: Figure, paragr: Text)

class Figure tuple(picture: Bitmap, caption: Text,
             label: list(Object))
```

Queries

- String matching

Find title and first author of articles with a section title containing “Query” and “Web”

```
select  tuple(t: a.title, f: first(a.authors))
from    a in Articles, s in a.sections
where   s.title contains (“Query” and “Web”)
```

- Union types

Find all subsections of articles containing the term “hypermedia”

```
select  ss
from    a in Articles, s in a.sections,
        ss in s.subsectns
where   text(ss) contains (“hypermedia”)
```

Note *implicit selection* on sections with subsections

Path and *Attribute* Sorts

- A value of type *PATH* denotes a path through complex objects

`.sections[0].subsectns[0]`

- A value of type *ATT* denotes an attribute (of a tuple or union)

`title, authors`

- *PATH* Variables:

Find all titles in article `art1`:

```
select  t
from    art1 PATH_p.title(t)
```

The expression `art1 PATH_p.title(t)` returns a set of pairs `<P,T>` such that the title `T` can be reached from art1 by following path `P` (i.e. title of the article, title of any section, subsection, etc.)

- *ATT* Variables:

Find attributes of art1 whose value contains "Ted Nelson"

```
select  name(ATT_a)
from    art1 PATH_p.ATT_A(val)
where   val contains ("Ted Nelson")
```

Querying the Web

- Path expressions
- Local vs. global links
- Context and query
- Path semantics
- Query locality

Path Regular Expressions

Find titles of all sections and subsections of article `art1`.

```
select  t
from    art1 (.(section | subsectn[_]))* .title(t)
```

Local vs. Global links

Find all local documents reachable from an article with a given URL

```
select  d
from    A (.llink)+ -> d
where   A.url = 'http.//www.microsoft.com'
```

Find all non-local documents reachable from article A that point back to A

```
select  d
from    A (.glink)+ -> d
where   d.glink->A and
        A.url = 'http.//www.microsoft.com'
```

Local vs. Global links (cont.)

Follow chains of length up to 3, ignoring anchors labeled “Back” or “Home”

```
select  d
from    art1 PATH_p -> d
where   PATH_p in .glink .glink? .glink? and
        not (contains(PATH_p, ATT_a) and

contains(label(ATT_a),"Back"|"Home"))
```

Select a document and all its subdocuments

```
select  d
from    art1 PATH_p -> d
where   PATH_p in .glink* and
        contains(PATH_p, ATT_a) =>
                REL(att_A)="subdocument"
```

Context and detail

- Separate the *gathering* from the *searching*
- Separate query in two parts: *context* and *detail*
- Context is the scope of the search

  only North-American sites

  only sites indexed by Lycos

  only sites reachable directly from some local document
- Detail is the detailed query specification

  documents containing clickable maps that contain a pointer to my home page
- Query is a pair of expressions (context,detail)
- Context expression is restricted to be efficiently executable

  "...I share your concern about what could become a very fragile future if we don't build in mechanisms for constraining the scope of these mobile agents or dealing with the potential resources they might consume. A nightmare situation is that you pose a query and it turns out to be the $64,000 question and you get the bill."

  Vinton Cerf

Path semantics

- The language for structured documents restricts path variables:

  A concrete path (instance of a path variable) cannot traverse two objects of the same class.

- Consequence: since the schema is fixed, paths are of bounded length.

- For hypertext and Web queries, must relax the semantics.

  unrestricted paths

  or

  simple paths (do not visit the same object twice)

- Consequence: paths are of unbounded length

  Simple paths: finitely many

  Unrestricted paths: infinitely many

  Simple paths are good for queries, but computationally expensive

Query Locality

How expensive is a query?

One measure: what fraction of the web must be searched in the worst case

O(n) Too expensive - can't do without indices

•*Find all documents containing 'Ted Nelson'*

•*Find all documents reachable from here*

•*Find all documents directly reachable from here*

(if document size is arbitrary)

O(1) Can do by probing

•*Find documents pointed to by "Next" and "Previous"*

•*Find all documents directly reachable from here*

(if document size is bounded)

O(log n) ?

•*Find a DNA sequence in a list of n URL's of DNA sequences, sorted in lexicographic order of the sequences*

Locality Levels

Say two graphs *agree* on a common set of vertices W if they both contain the subgraph induced by W and in both graphs this subgraph is linked to the rest of the graph in the same way.

(Formally, W and its immediate neighborhood in one of the graphs is isomorphic to W and its immediate neighborhood in the other graph).

Given a query Q, a class of graphs Γ, and a graph G in Γ, we say Q is *W-local for G and* Γ if for every graph G' that agrees with G on W, $Q(G')=Q(G)$.

Say Q *depends on W for G and* Γ if Q is W-local and W is minimal. W is a *window* for Q in G.

The *locality* of query Q wrt Γ is the maximum window size over all G of size n, as a function of n.

Theorem Computing a window for a query is no more expensive than computing the query.

Research Issues in Querying

- Infrastructure
  - Indexing and caching
  - View maintenance
- Data Model
  - Relational, OO, IR?
- Language Design
  - Expressive power
  - Style: logic, procedural, algebraic, SQL, IR
- Implementation
  - Reuse of views (indices, caches)
  - Query rewriting
  - Access plan selection
- User Interaction
  - Answer formatting
  - Integration of browsing and querying
  - Visualization

General References

The Internet

Douglas E. Comer, *Internetworking with TCP/IP Volume I,* Prentice-Hall, 3d. edition, 1995.

Cricket Liu et al, *Managing INTERNET Information Services,* O'Reilly & Associates, Inc., 1995.

Special Issue on Internet Technology, *CACM* 37:8, 1994.

Special Issue on Digital Libraries, *CACM 38:4, 1995.*

The World-Wide Web

Ian S. Graham, *HTML Sourcebook,* John Wiley, 1995.

Proc. WWW '94 Conference, `http://www1.cern.ch/WWW94/PrelimProcs.html.`

Special issue on 4th. Int'l WWW Conference of *Computer networks and ISDN systems*, 27:6, 1995.

Hypertext

Jakob Nielsen, *Multimedia and Hypertext: The Internet and Beyond,* Academic Press, 1995.

Special issue on Hypermedia, *CACM* 37:2, 1994.

Proc. Hypertext '87,'89,'91,'93, ACM Press.

VISUALIZING THE WEB

Uses of visualization:

- User orientation
  - Overview diagrams
  - Navigation histories
  - Landmarks
  - Server maps
- Log analysis
- Query answer display

Taxonomy:

- Hierarchical vs. graph
- On-line vs. off-line
- Structural vs. content-based

Some work

- *Navigational Web Builder*, Mukherjea, Foley, and Hudson, CHI '95

  Hierarchical, static, content+structure

- *WebViz,* Pitkow and Bharat, `http://www1.cern.ch/PapersWWW94/pitkow-webvis.ps.`

  Graph, static, structure

- *WebOOGL,* Munzner et al, `http://www.geom.umn.edu/docs/weboogl/weboogl.html.`

  Hierarchical, dynamic, structure

- Hy+, Hasan et al, http://www.db.toronto.edu:8020/Hy.html.

  Graph, dynamic, structure

VISUALIZING LARGE GRAPHS

- Layout
- Queries
- Abstraction and clustering
- Presentation emphasis

Non-graph Visualizations

- *Information Visualizer*, Card et al, Xerox
- *Seesoft*, Eick, AT&T Bell Labs
- *Treemaps*, Johnson, University of Maryland