Team Members:

Mariano Consens	Isabel Cruz
Frank Eigler	Sergio Faria
Masum Hasan	Yanni Jew
Christine Knight	Alberto Mendelzon
Carlos Mendioroz	Manny Noik
Toomas Toomsalu	Dimitra Vista
Peter Wood	Annie Yeung

Sponsors:

NSERC
IRIS
ITRC
IBM Canada

Outline:

1. Data visualization and visual queries

2. The Hy+ system

3. Hy+ in:

   - software engineering
   - network management
   - distributed debugging

$$\text{DB} \xrightarrow{\text{query}} \text{db}$$

Problems:

- `DB` too complex
- `db` hard to understand
- `query` hard to formulate

Solution: *Visualize*

Visual `DB`, `db`	*data visualization*
Visual `query`	*visual queries*

Related Areas

- Scientific Visualization
  - Mapping 3D time-varying continuous phenomena into images.
  - *Khoros, apE, SGI Explorer.*

- Visual Programming
  - Using diagrams and images to specify computations.
  - *LabView, Prograph*

- Program Visualization
  - Displaying program structure and behaviour.
  - *Sorting out Sorting, Balsa, Pecan.*

Visualization is like the surgeon's knife: properly used it quickly cuts to the essentials; improper use brings colorful but disastrous results.

Barton Miller

Darrell Raymond
Characterizing Visual Languages
IEEE Workshop on Visual Languages, 1991

	dense syntax	discrete syntax
dense domain	**visual language**	
discrete domain		

Hy$^+$ System

- Data visualized as directed labelled graphs and **hygraphs**.

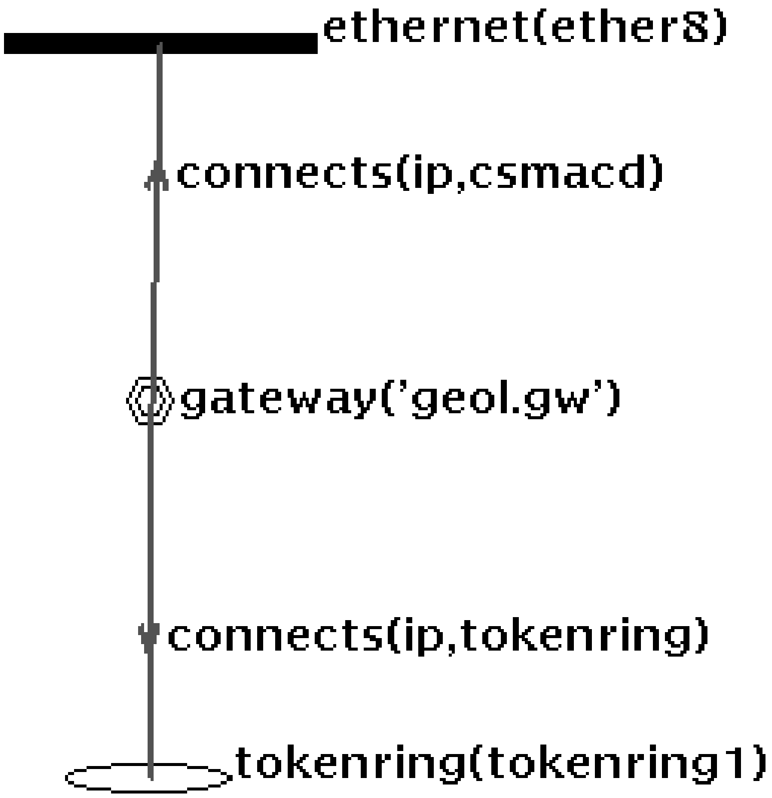
- First-order fact or relational tuple: $calls(function(f12), function(f17), line(135))$ mapped to:

  * Nodes labelled with first-order logic ground terms $function(f12), function(f17)$

  * Edges labelled with positive literals $calls(line(135))$

connects*(gateway('geol.gw'), ethernet(ether8), ip, csmacd).*

connects*(gateway('geol.gw'), tokenring(tokenring1), ip, tokenring).*



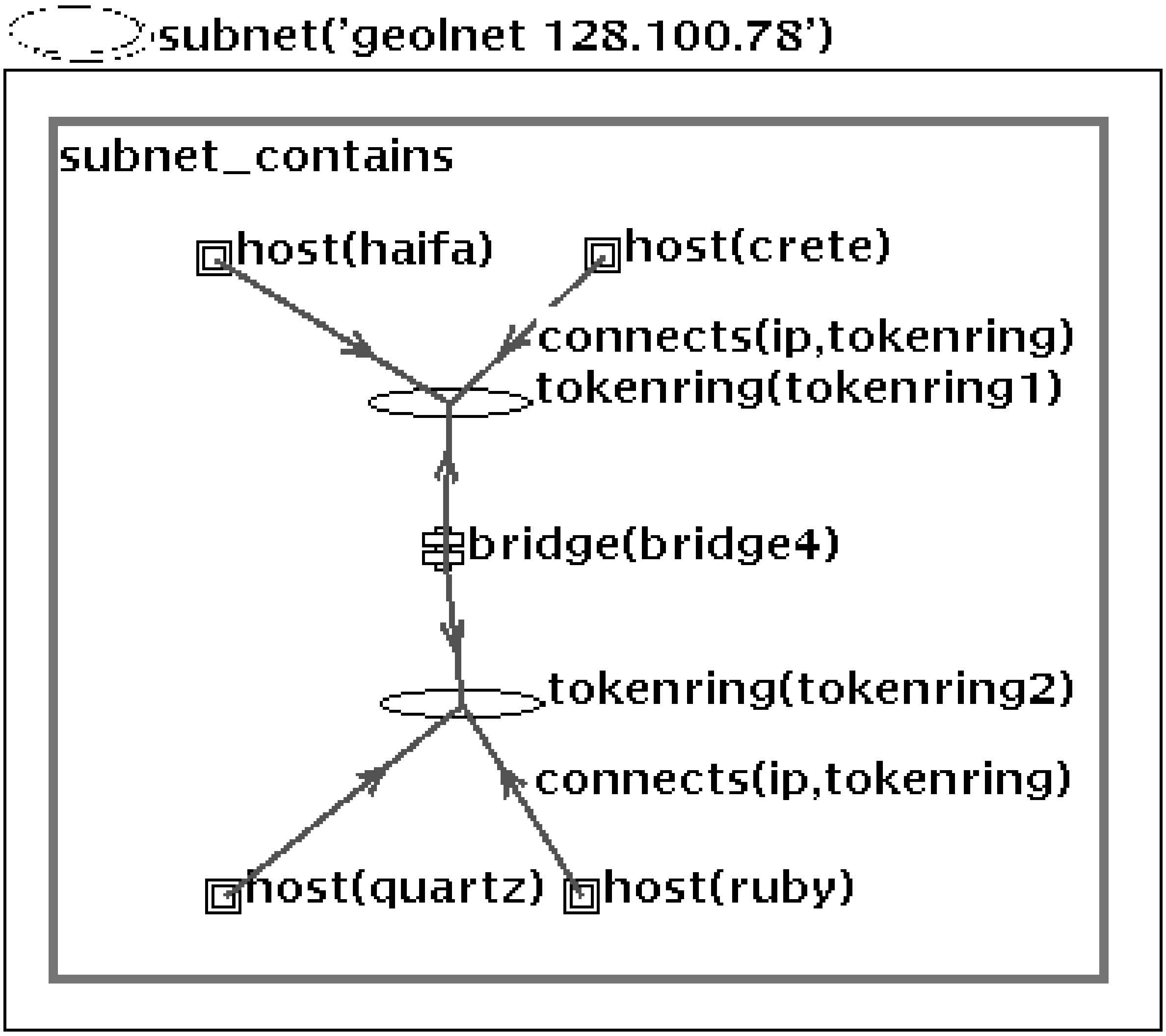
Visualizing Tuples

Hygraph

- Hygraph $\rightarrow$ graph augmented with **blobs**.

- Blob relates a containing node with a set of contained nodes.

- Visually, blob is represented as a rectangular region.



Visualizing a Hygraph

GraphLog Query Language

- GraphLog queries are graph patterns to be matched against database:

  * nodes labelled by first-order terms;

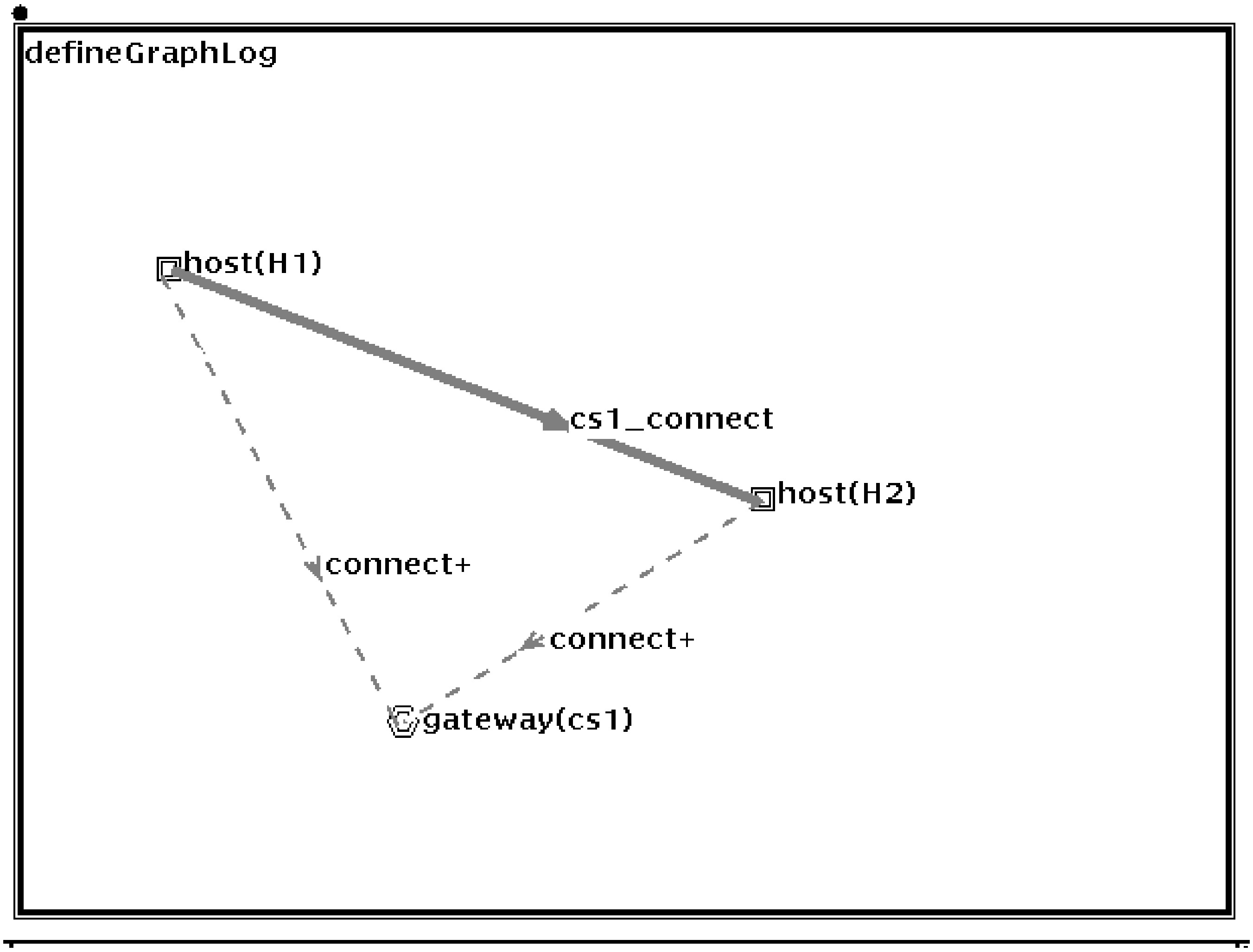
  * edges by *path regular expressions* on literals.

  * $E \leftarrow < lit >; (E)^+; -(E); \hat{}(E); (E|E); (E.E)$

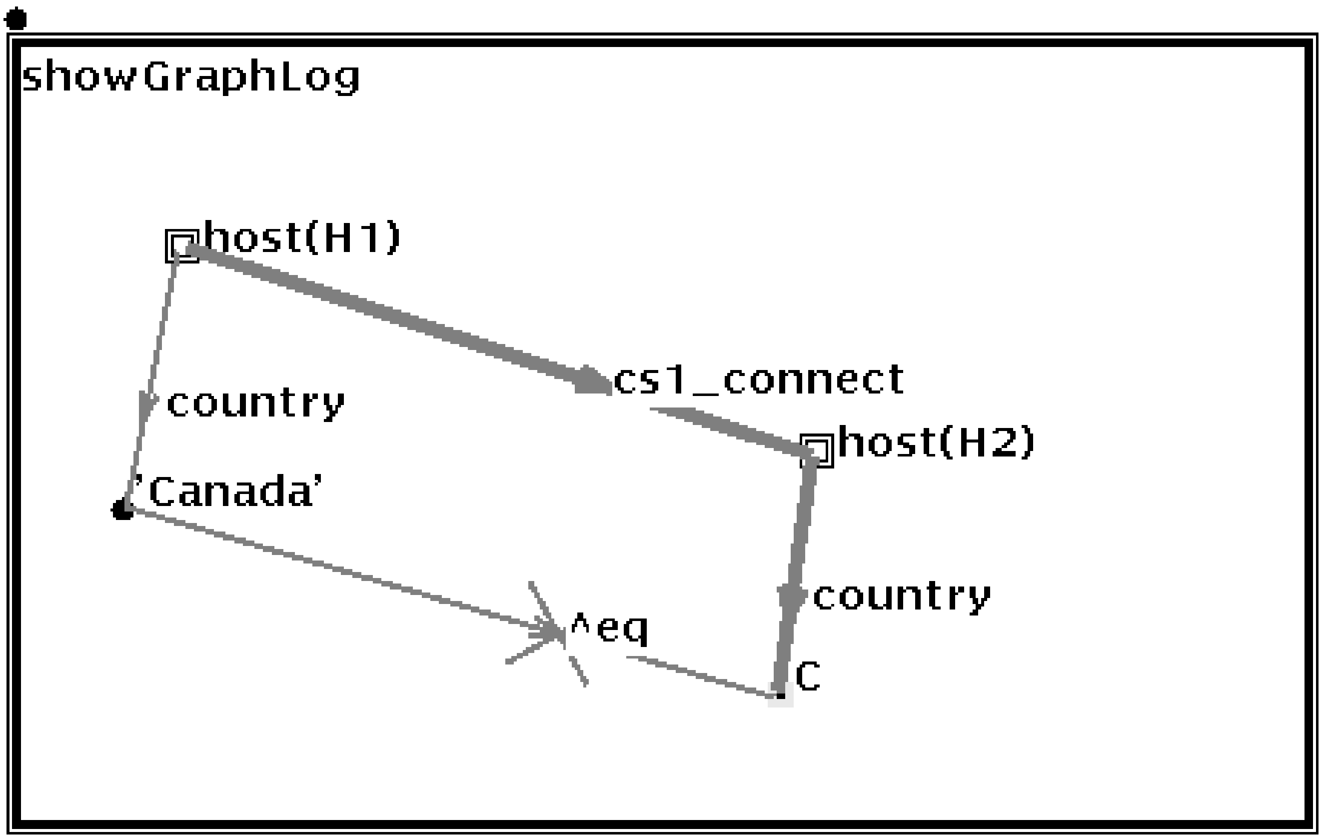
- Two kinds of queries:

  * **Define**: create new relationships

  * **Show**: filter out and show matched patterns.

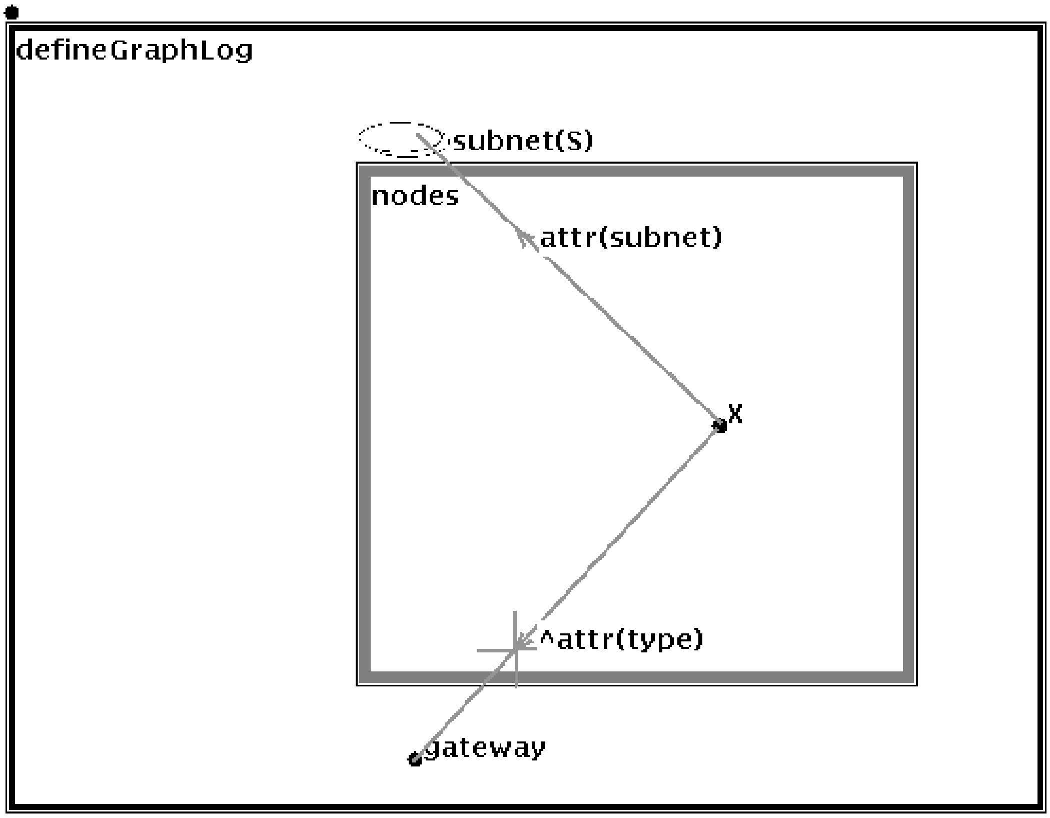


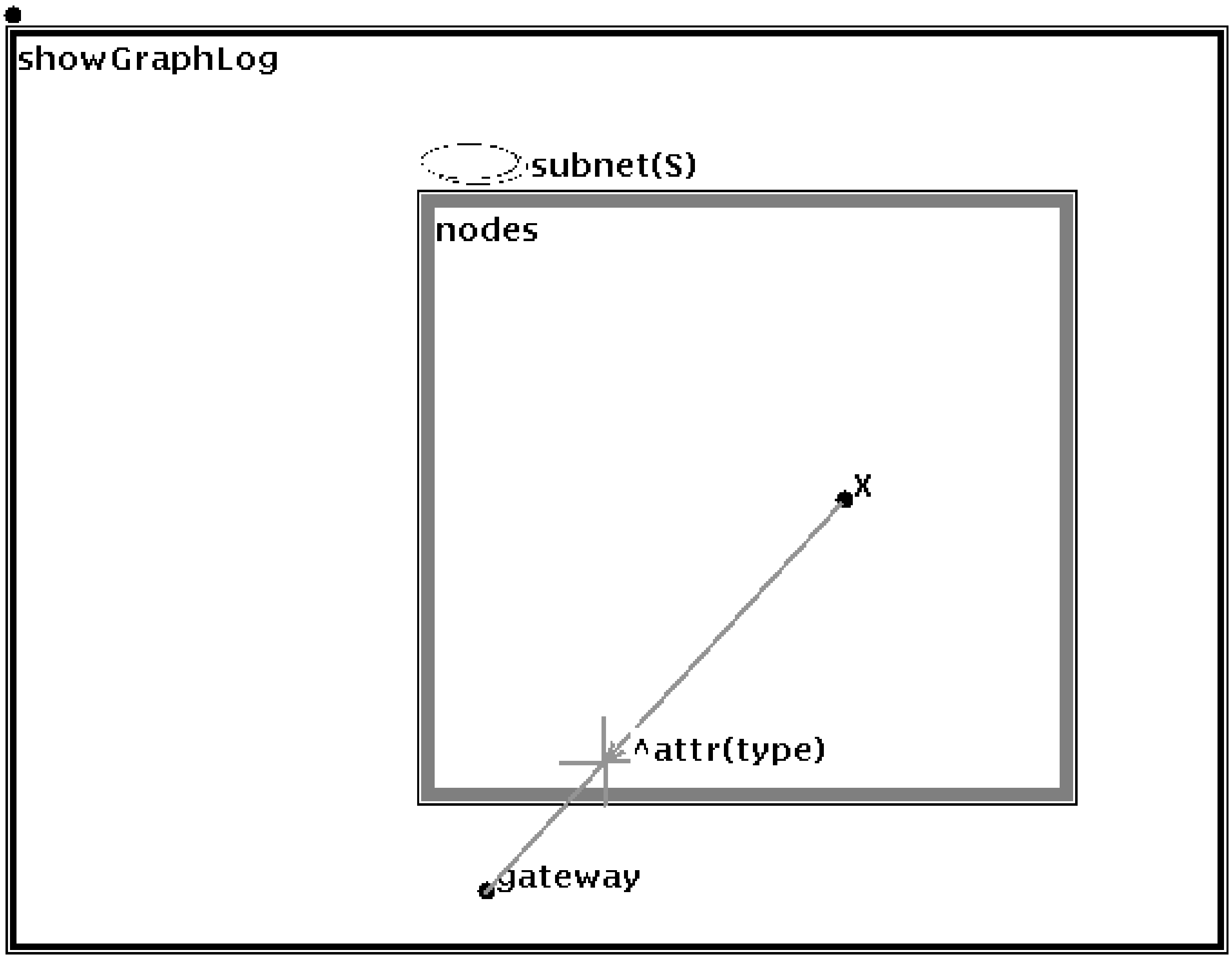
A Define query


showGraphLog
host(H1)
country
cs1_connect
host(H2)
'Canada'
country
^eq
C

A Show query

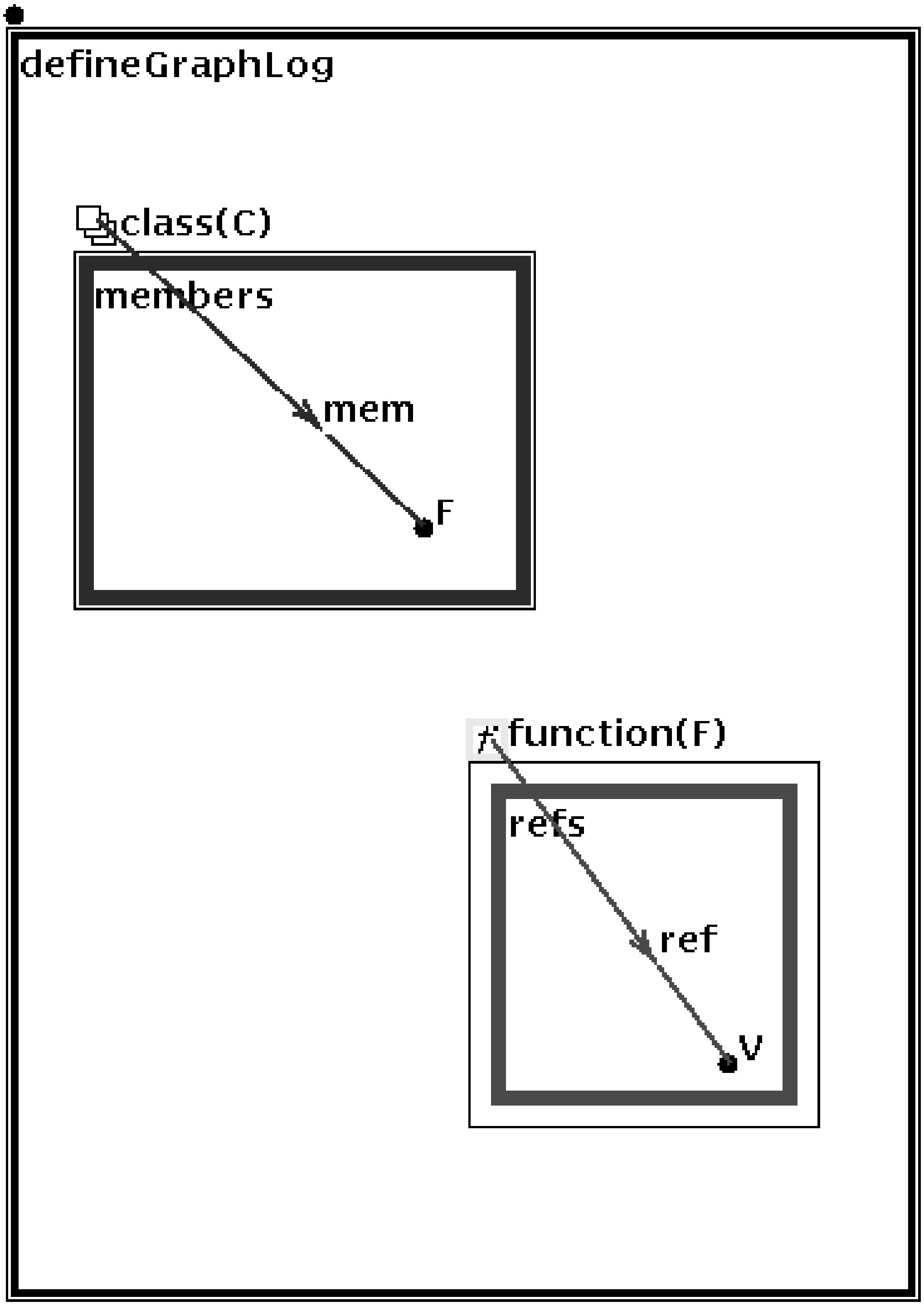
Defining Blobs with a Query

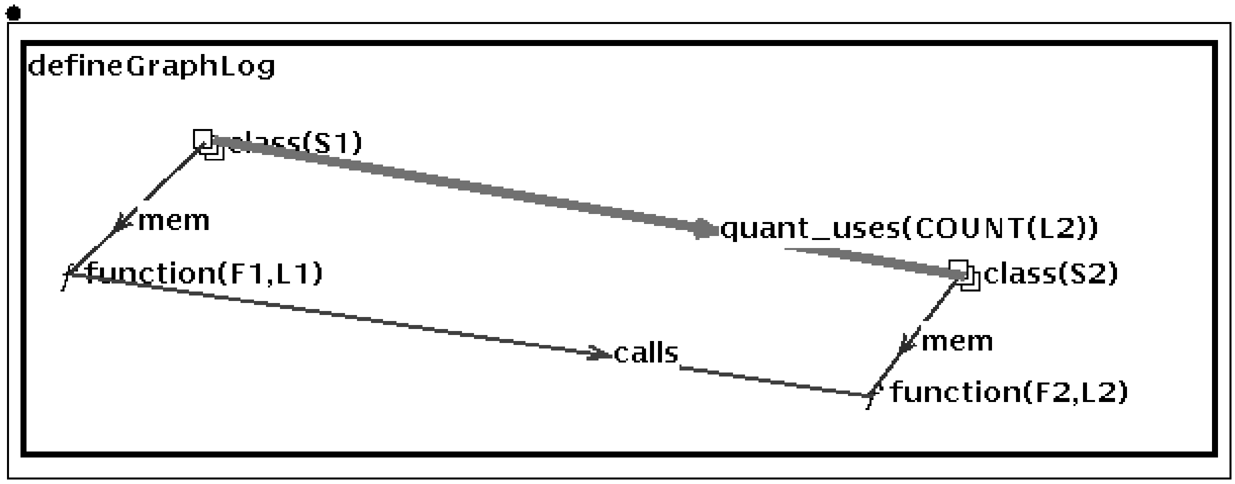


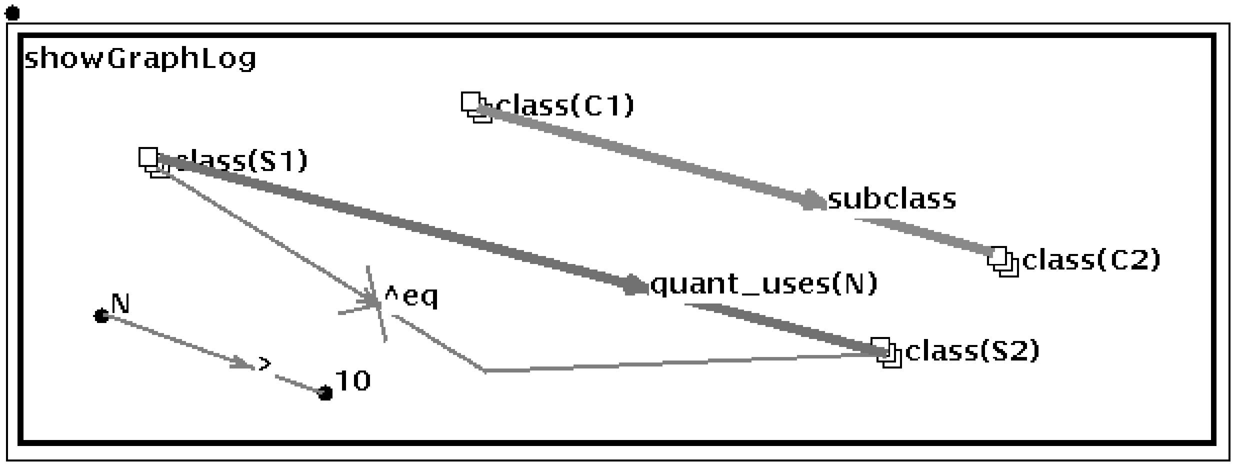


Showing a Blob

Defining Nested Blobs







Aggregation

Expressive Power of GraphLog:

Equivalent to:

- Linear *Datalog* with stratified negation
- Relational Algebra + transitive closure
- Non-deterministic LOGSPACE

With aggregation:

Strictly contains SQL.

Querying and Visualizing Software Objects

Objective:
Develop visual tools to support design and exploration of large software systems

Related Projects

AT&T Software Information Base

ITHACA

Arcadia Consortium

Software Discovery Tool

Requirements:

- Visualize
- Explore
- Analyze

information about a software system.

Limitations of Browsing

Browsers are critical to the success of OOP.

One group, which used Eiffel, stated that the lack of a good browser to locate existing reusable software meant that many classes were redundantly created. Another group, using the Smalltalk browser, felt it did not have the flexibility required by sophisticated programmers. There is a need for both browsing in the large (getting the big picture) and in the small (inspecting the internal nuts and bolts.)

Holt, Stanhope & Lausman, *Object Oriented Computing.*

Tools to:

"...select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, refine, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, sort, pick over, group, pigeonhole, integrate, blend, average, filter, lump, skip, smooth, chunk, inspect, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glan ce into, leaf through, skim, list, glean, synopsize, winnow wheat from chaff, and separate the sheep from the goats."

Edward R. Tufte
Envisioning Information

Visualizing

Software is invisible and *unvisualizable*.

The reality of software is not inherently embedded in space. Hence, it has no ready geometric representation in the way that land has maps. silicon chips have diagrams, computers have connectivity schematics. As soon as we attempt to diagram software structure, we find it to constitute not one, but several, general directed graphs superimposed one upon another.

In spite of progress in restricting and simplifying the structures of software, they remain inherently unvisualizable, and thus do not permit the mind to use some of its most powerful conceptual tools.

Fred Brooks, *No Silver Bullet.*

HY+ as a Software Discovery Tool

- Visualize

  graphs/hygraphs, colours, icons, user-specified views

- Explore

  zooming, scrolling, overviews

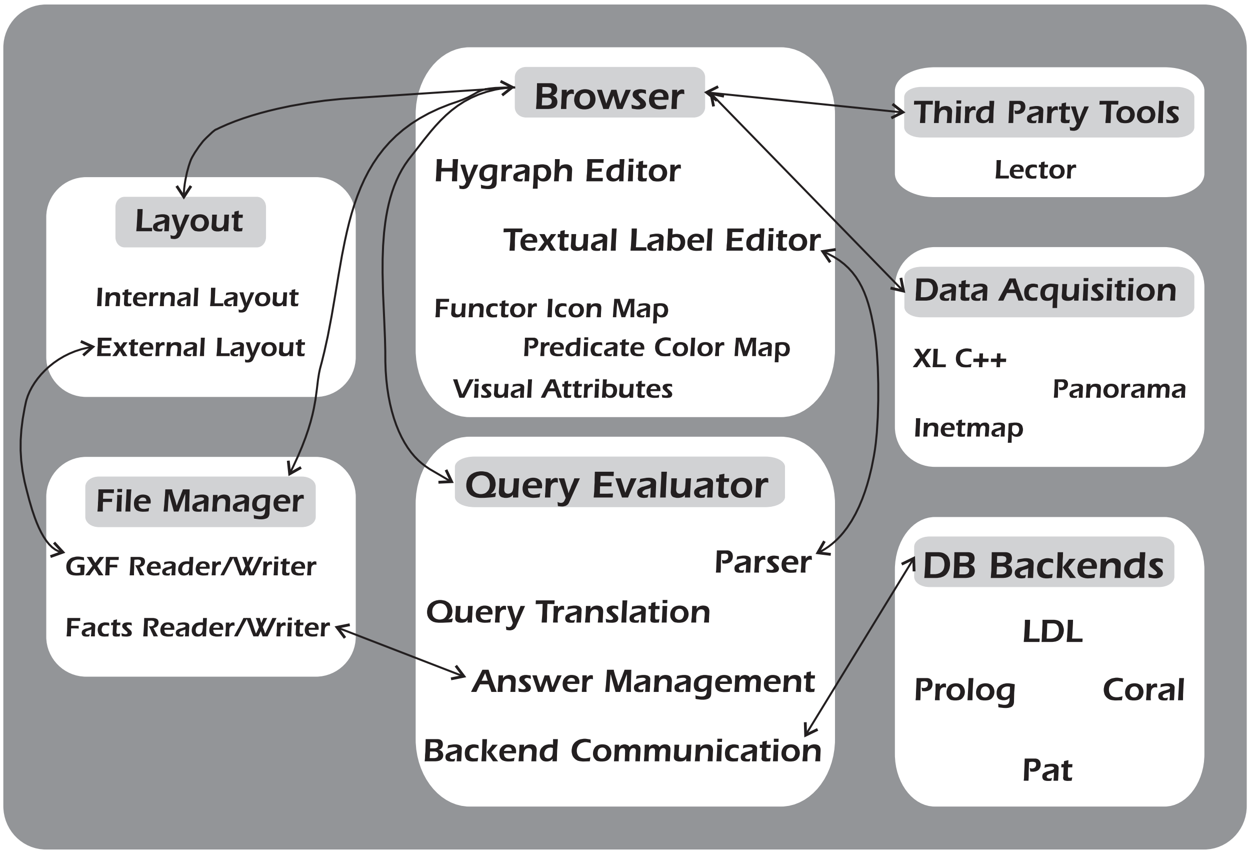
  GraphLog queries

- Analyze

  GraphLog aggregation/summarization

  Interface to other tools

VIDEOTAPE



Hy^{+} Architecture

Work in progress:

- *Deconstruction* of Hy^+

  broadcast-based architecture (*Softbench* BMS)

  autonomous display, query, layout, update components

- Query processing

  Query optimization, incremental queries

  Interfaces to SQL/OODBMS

- Layout and Presentation

  Fish-eye Views

  Declarative Layout

  Incremental Layout

Hy$^+$ for Network Management

- **The network management problem:** Monitor, analyze, and control network behaviour and status.

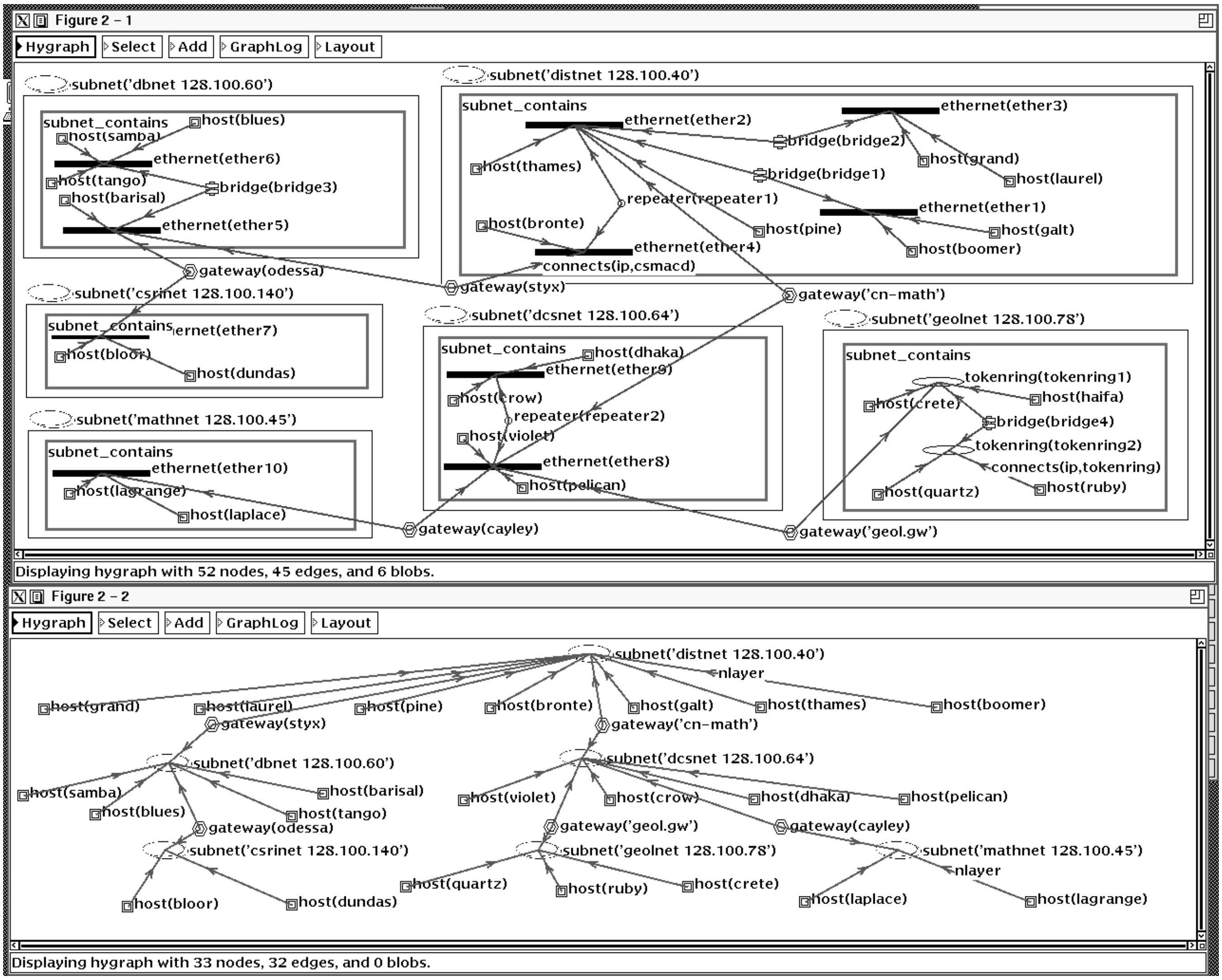
- **Typical scenario:** *Management station* communicates with network through management protocol (e.g. *SNMP*) and interfaces with manager.

Requirements:

- **Abstraction**
  * Cluster network objects into abstract entities
  * Reduce volume of information displayed.
- **Filtering**
  * Focus on relevant portion of data.
- **Visualization**
  * Make behaviour of the network easily perceivable.
  * Control layout for clarity.

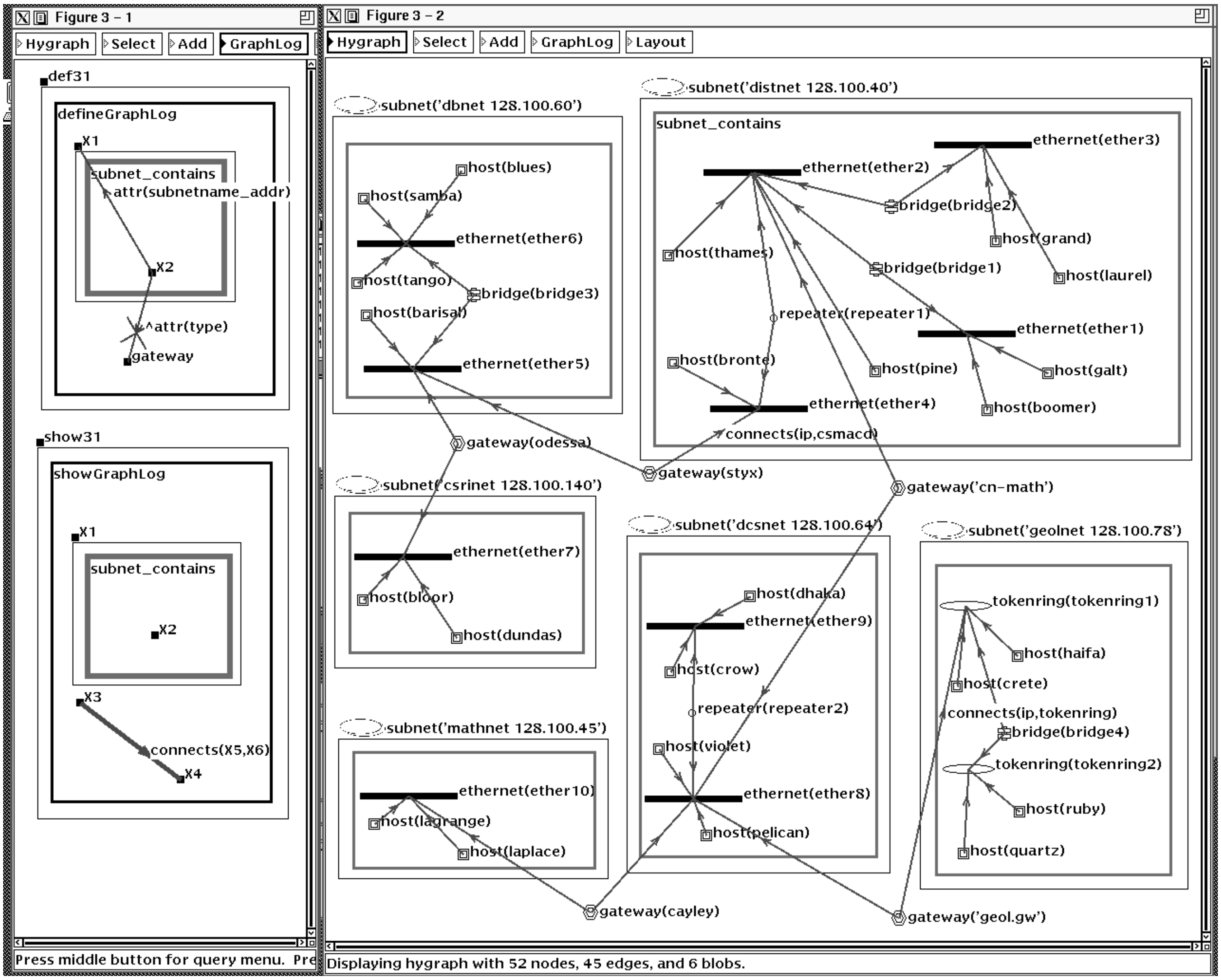
Network Abstraction

- Geographical
- Administrative
- Protocol layer



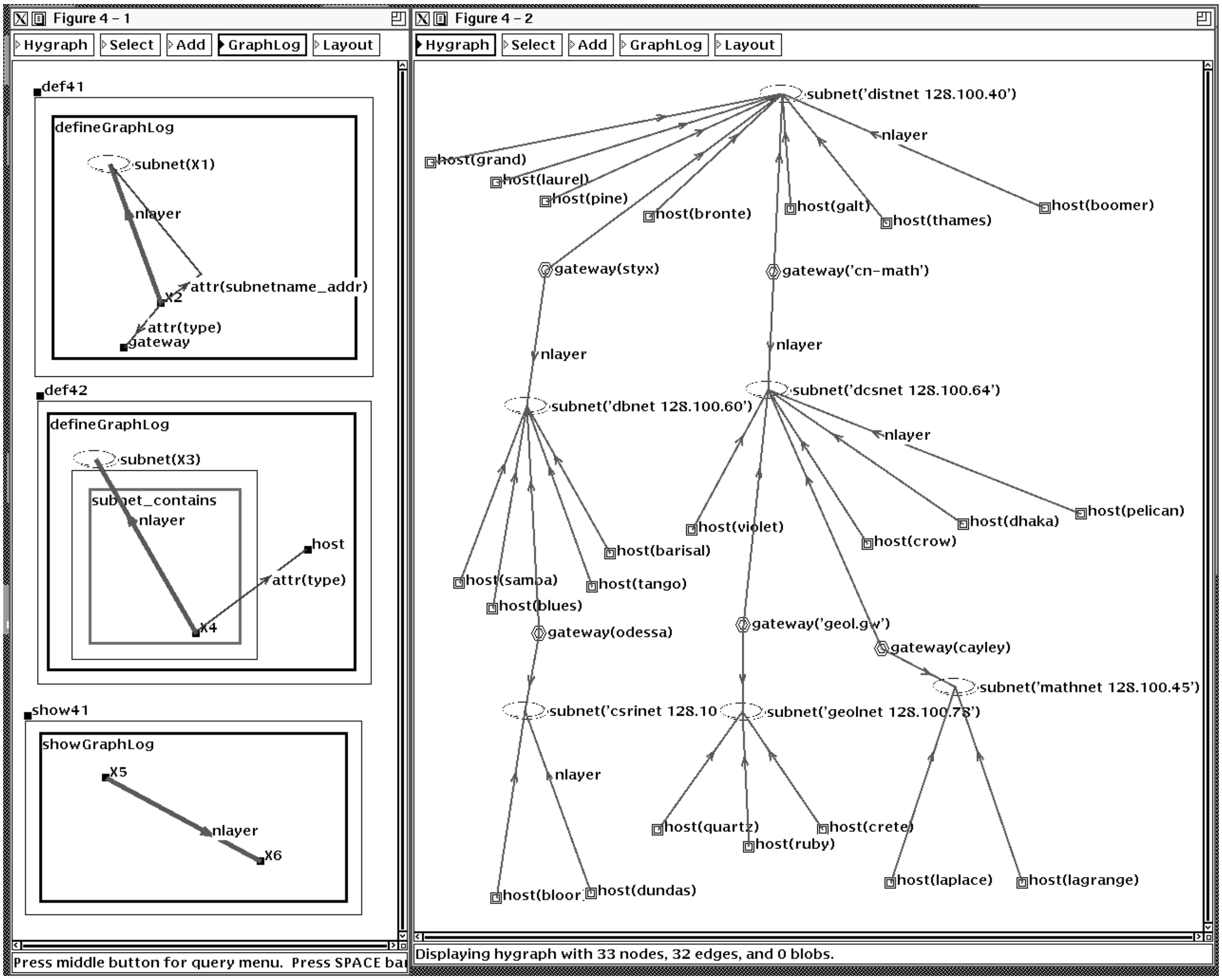
- Abstractions of network topology.

- Physical and Logical Network topology Map.



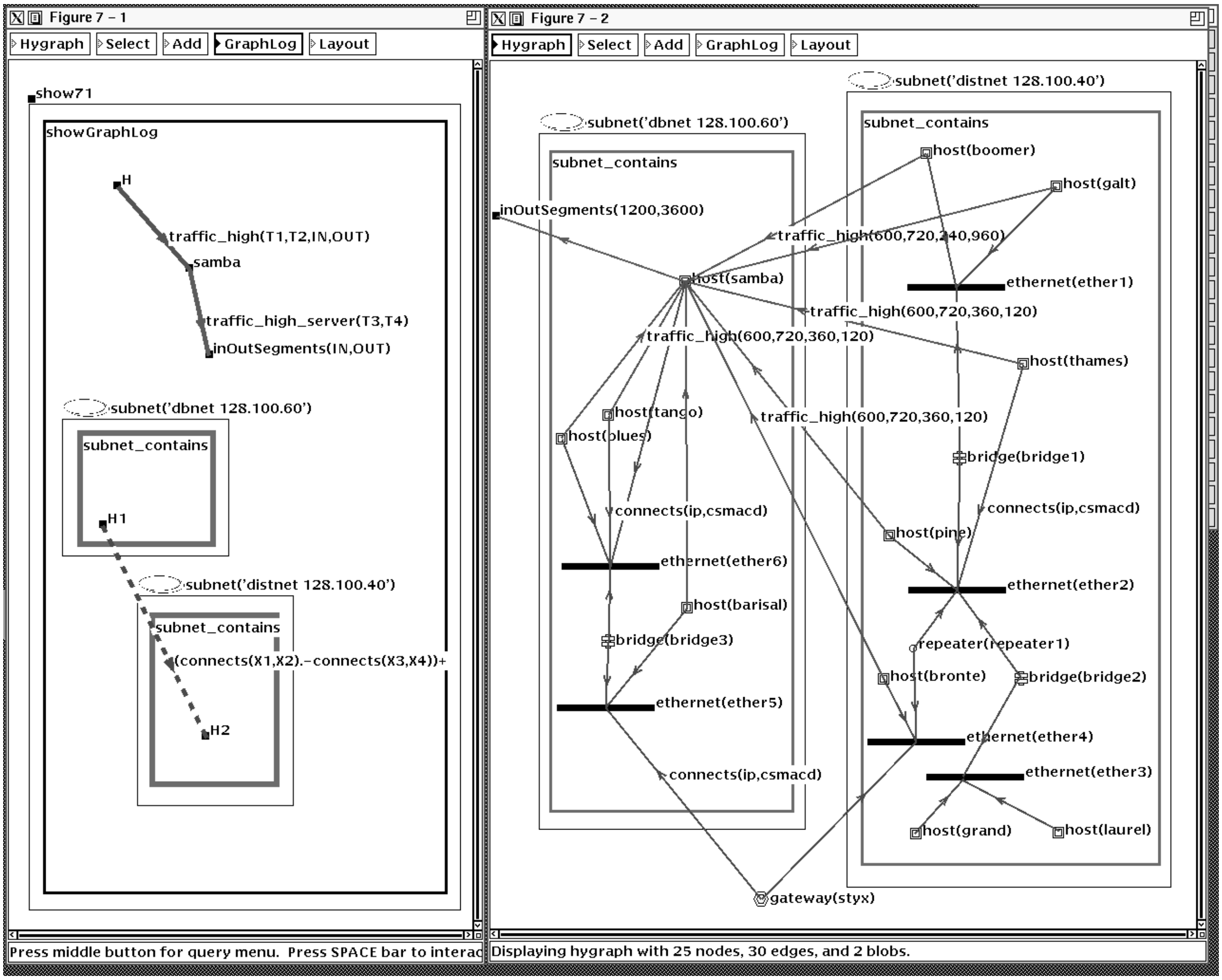
- Physical Map.

Abstraction



- Logical Map from Physical map (gateways and hosts shown only).

Filtering



- Show only `dbnet` and `distnet`

- Show high traffic links from/to server `samba`

Future work: Dynamic Visualization

- Visual Representations of Dynamic Systems

- Incremental Visualization

- Behavioural Queries

Hy^+ for Distributed/Parallel Debugging

- **The debugging problem:** Monitor, diagnose, and correct the behaviour of a set of interacting processes.

- **Typical scenario:** Programmer uses debugger to analyze trace of a computation.

- **Problem:** Too much information, fixed level of detail.

Visualizing Process Behaviour:
"Bart Miller stated that he likes visualization but sees some problems. He gave two examples in which nice patterns emerged in pictures only when the layout of processors or processes was carefully arranged. Such visualization requires intimate knowledge of the relation among processors and processes and is problem-specific ... He believes that for visualization to gain wide acceptance, it ... must allow views at different levels of abstraction ... He claimed that we don't want to visualize a large parallel program; we want to focus on the part of the program which has the problem. This requires visualization with analysis, not just visualization alone."

Panel on Visualization, ACM/ONR 1991 Workshop on Parallel and Distributed Debugging.

Using Hy$^+$:

- Automatic/flexible layout
- Hygraphs for variable levels of abstraction
- GraphLog queries for focusing

Conclusion

- Non-traditional applications: software engineering, network management, distributed debugging

- Deductive queries + Visualization = Hy^+