Information Visualization and Knowledge Management

Dr. Margaret-Anne Storey
Dept of Computer Science, UVic

Trees and Networks

London underground (before Beck's idea)

FIGURE 8.2
The London Underground map before Beck's idea was adopted

Source: *London Underground Map designed by H.F. Stingemore (1927) © London Transport. Reproduced by kind permission of London Transport*

Beck's original map of London underground

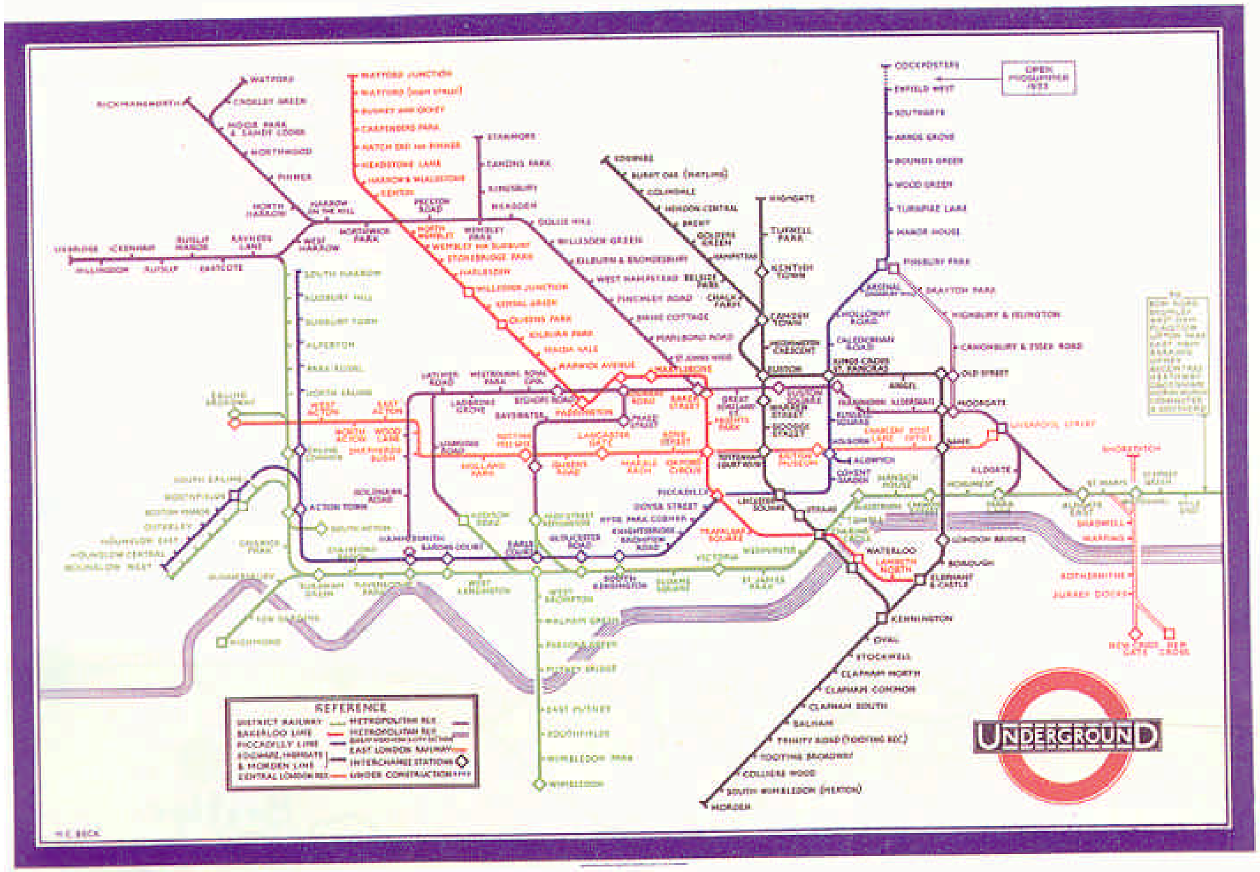


FIGURE 8.1
Harry Beck's original map of the London Underground system

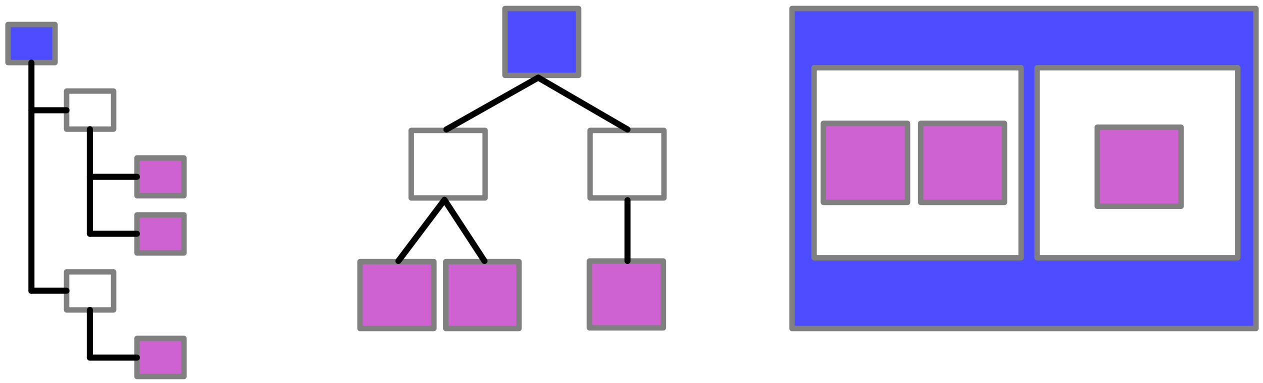
Source: *London Underground Map designed by Harry Beck (1953) © London Transport. Reproduced by kind permission of London Transport*

Visualizing structures

- Node-link diagrams can be used to encode relationships between data
- Space is always a big issue
  - Wasted space for many tree layouts (enclosure layouts tend to be more space efficient)
  - Never enough for large trees and networks
- Difficulties navigating
  - “Lost in space”
    - Can use context and detail views
    - Distortions
- Position is usually very important for tree structures (and sometimes networks)
  - Importance of “Preserving the Mental Map”
- Let’s look at two types of structures: trees and networks

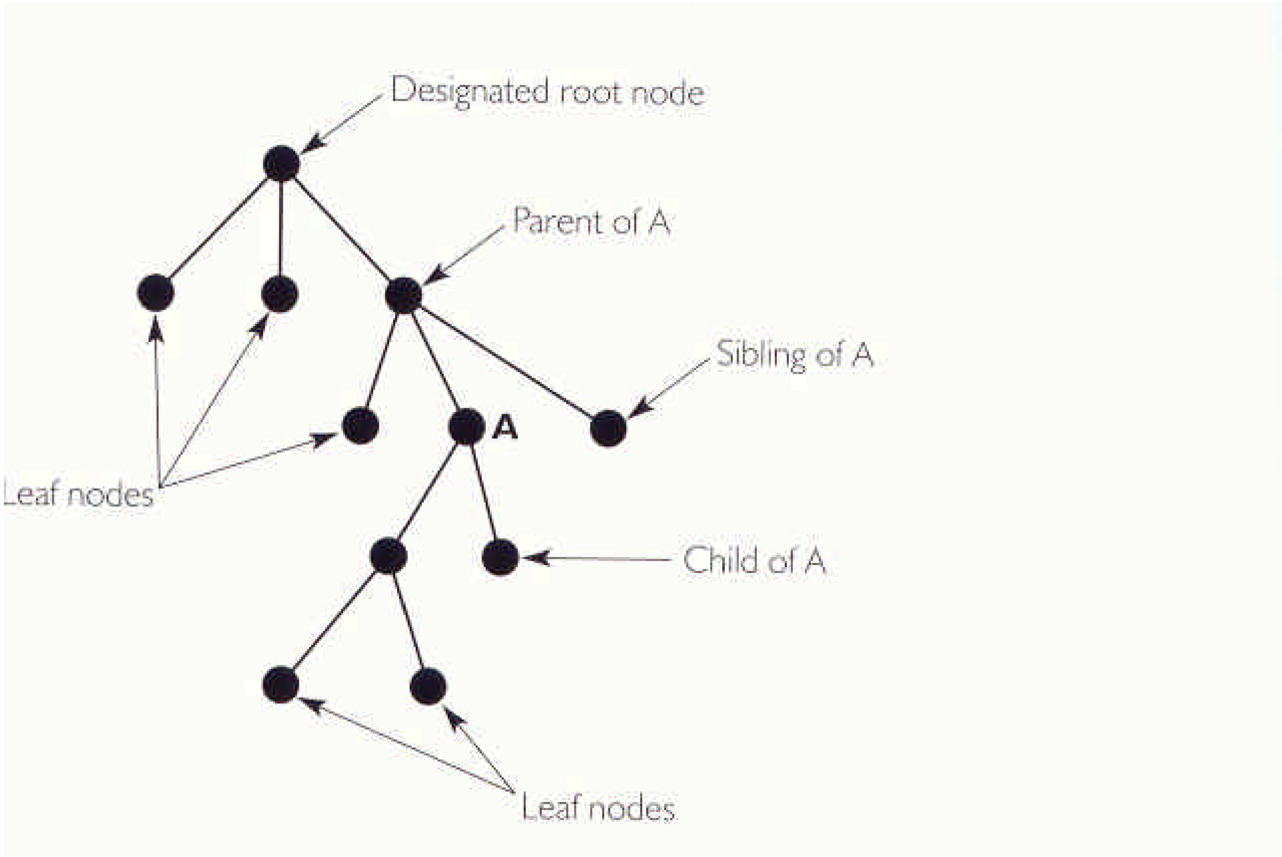
Trees

- We can encode hierarchical data (trees) using
  - Connection or
  - Containment (enclosure)



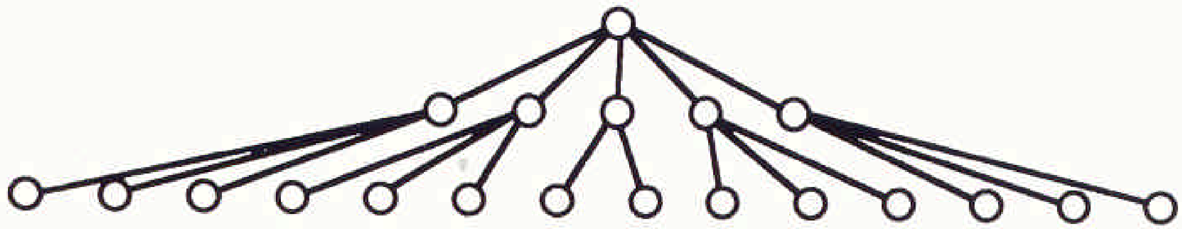
- Use different approaches to show different kinds of information:
  - Node link better for trees that have an uneven shape, enclosure (such as Treemaps) preferred if there is a quantitative variable you want to encode using size
  - But it really depends on the questions you are trying to answer or the concepts you are trying to communicate…

Terminology of a tree



Broad trees

FIGURE 8.24
The representation of a tree quickly becomes difficult to handle within a conventional display

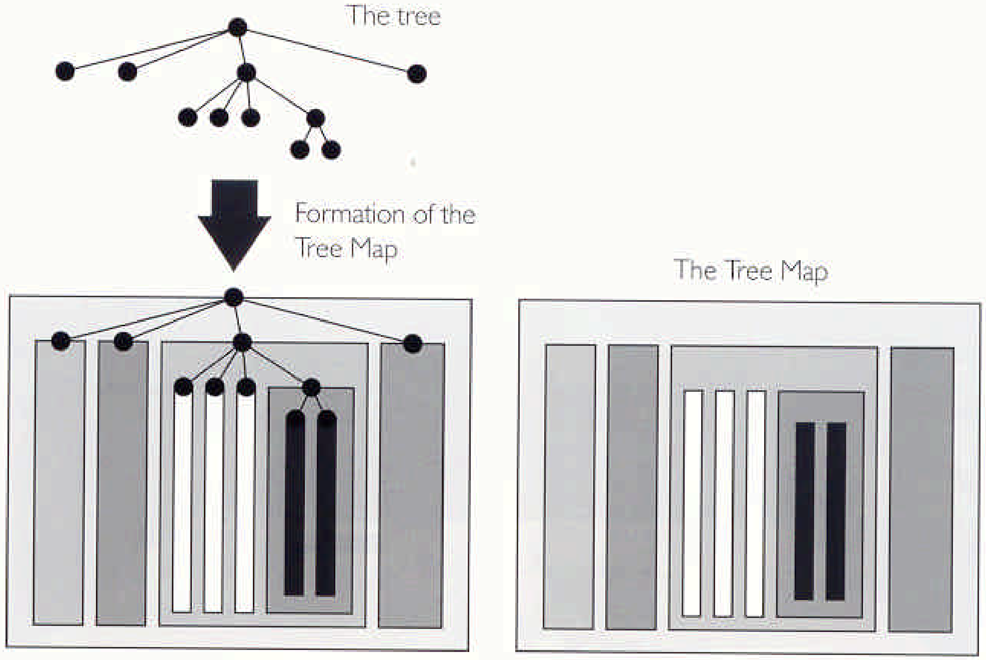


Treemaps

- Issues:
  - Nesting vs. non-nesting, when to nest?
  - May be hard to use if large
  - Layout issues
- Advantages
  - Interactive
  - Customizable (for example colour, depth)
  - Shows both structure and content
  - Shows the “gestalt” nature of the data

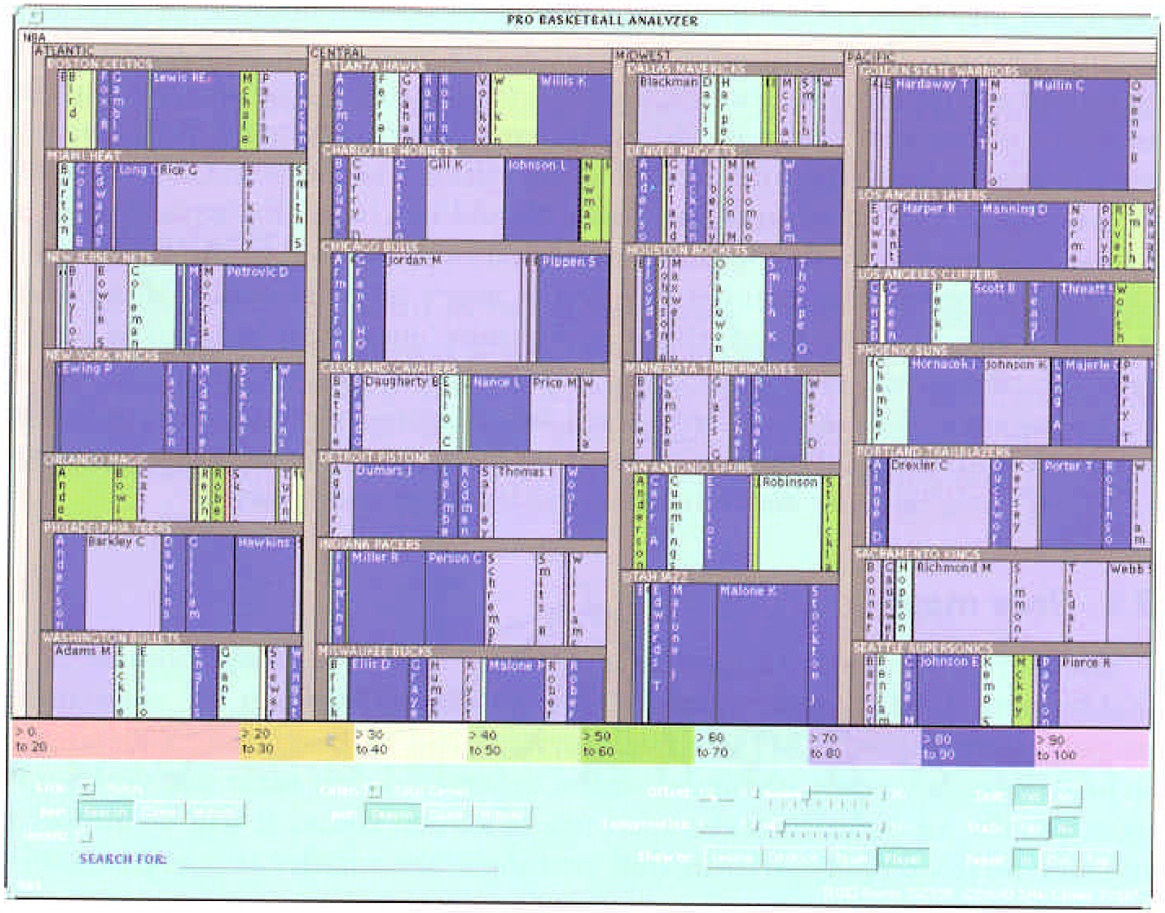
Treemap idea

FIGURE 8.26
The derivation of a Tree Map from a tree
Source: *Ben Shneiderman*



Treemap (baseball data)

FIGURE 8.27
A Tree Map representation of baseball data



Treemap idea

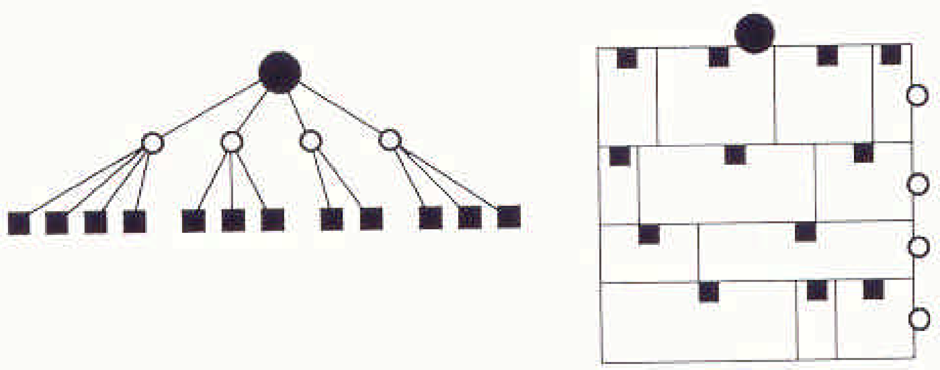


FIGURE 8.28
A modified Tree Map construction with alternating horizontal and vertical divisions

Treemap example

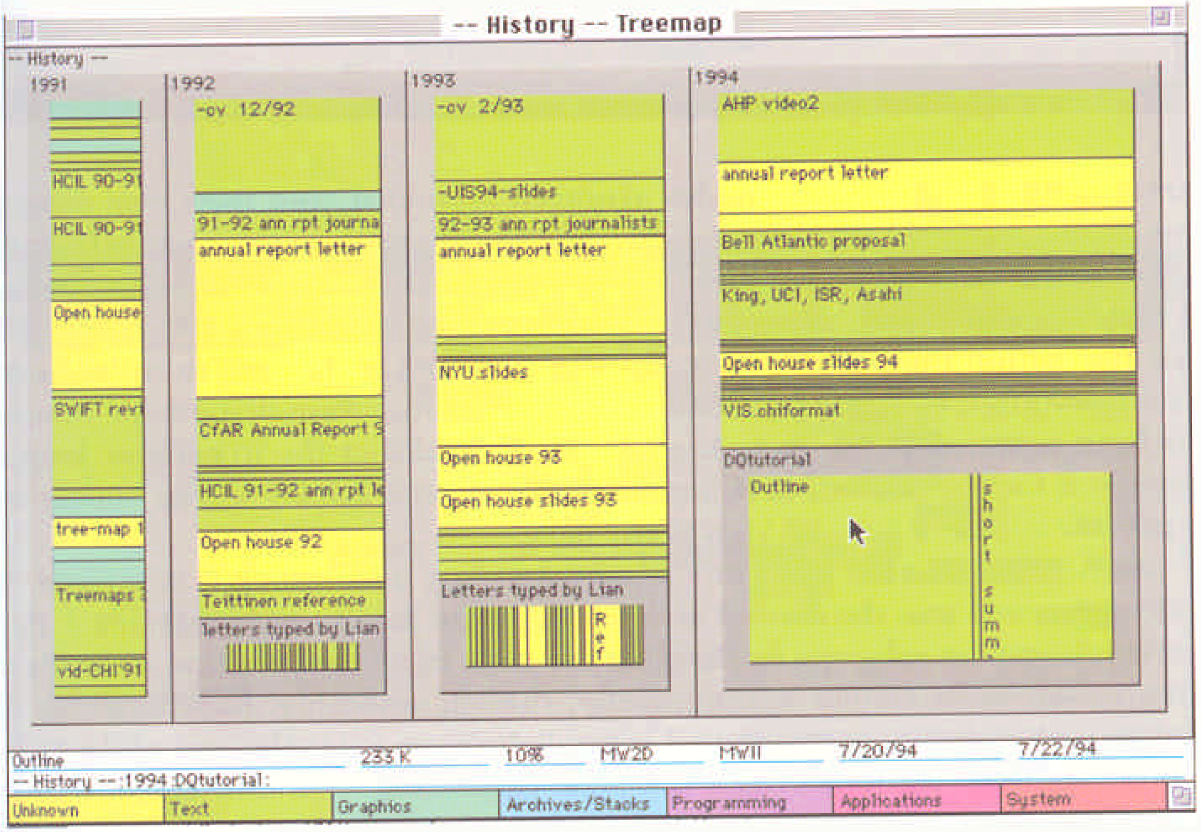
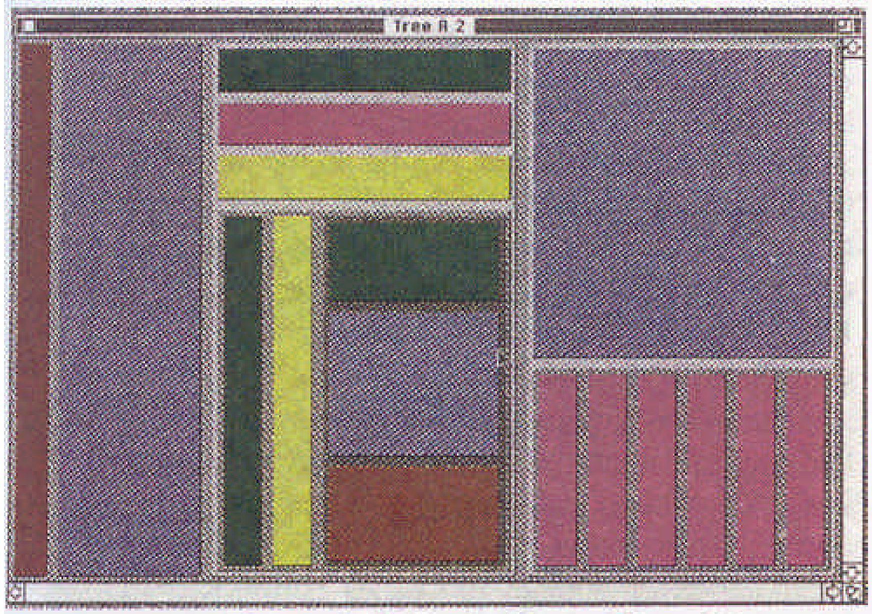
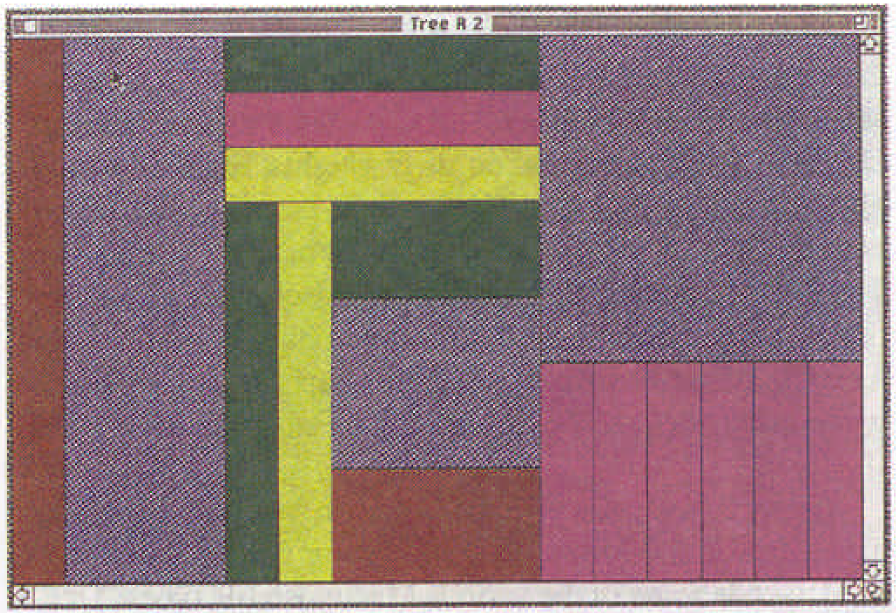


FIGURE 8.29
Tree Map display of an author's collection of reports
Source: *Ben Shneiderman*

Treemaps (nesting vs. non-nesting)

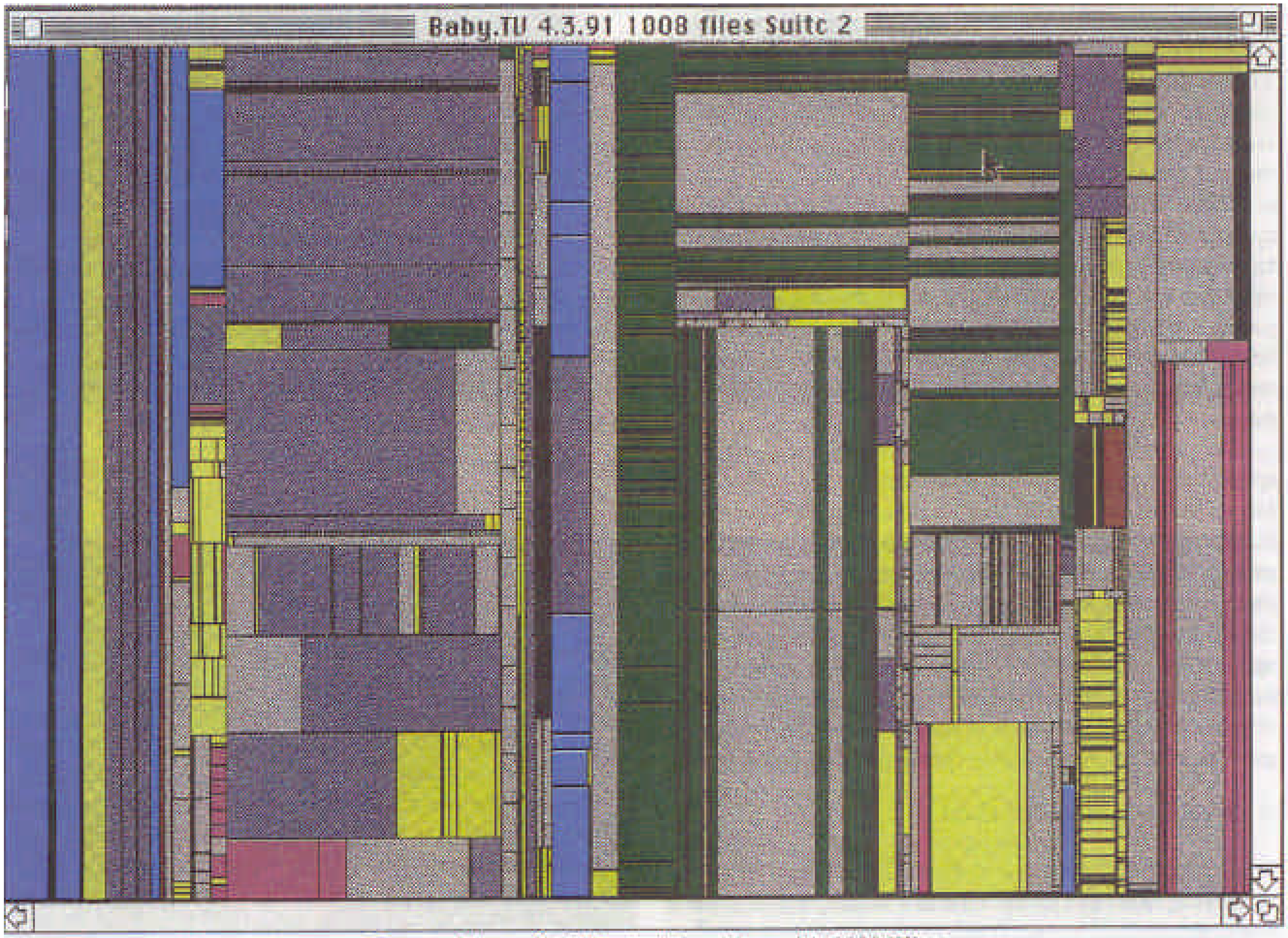


Johnson Figure 6 (Color). Nested Tree-Map



Johnson Figure 7 (Color). Non-nested Tree-Map

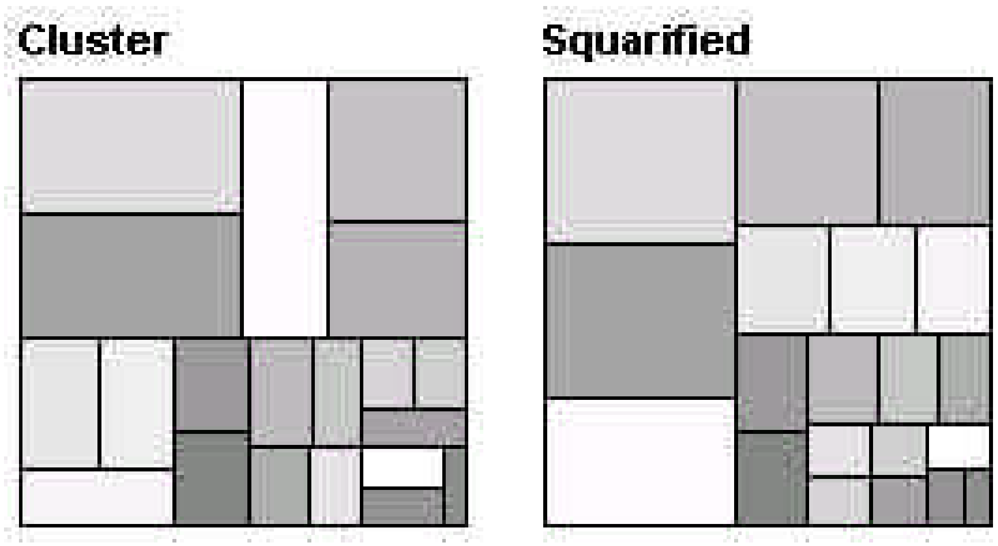
Treemap of 1000 files



Johnson Figure 8 (Color). Tree-Map with 1000 Files

Other approaches

- Original algorithm – preserves order, stable with respect to small changes, but an result in areas with a high aspect ratio
- Other approaches (not stable, order not preserved):

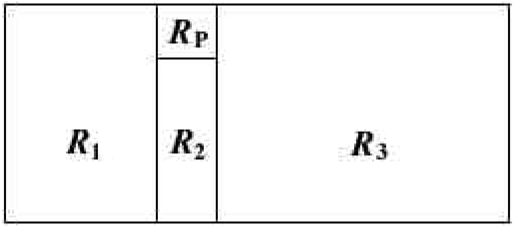


Ordered Treemaps

- See ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/2001-06html/2001-06.htm
- Observation – it is possible to layout items that are adjacent in a list adjacent in a treemap (so not strictly linear ordering)
- Idea – place the largest item first

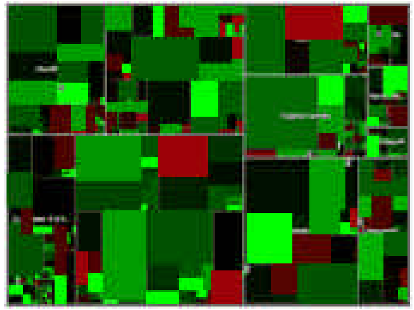
Algorithm

1. Let *P*, the pivot, be the item with the largest area in the list of items.
2. If the width of *R* is greater than or equal to the height, divide R into four rectangles, R_1, R_P, R_2, and R_3
3. Place *P* in the rectangle R_P, exact dimensions of it determined in Step 4.
4. Divide the items in the list, other than *P*, into three lists, L_1, L_2, and L_3, to be laid out in R_1, R_2, and R_3. L_1 and L_3 all may be empty lists. (Note that the contents of these three lists completely determine the placement of the rectangles in Figure 3.) Let L_1 consist of all items whose index is less than P in the ordering. Let L_2 and L_3 be such that all items in L_2 have an index less than those in L_3, and the aspect ratio of *P* is as close to 1 as possible.
5. Recursively lay out L_1, L_2, and L_3 (if any are non-empty) in R_1, R_2, and R_3 according to this algorithm.

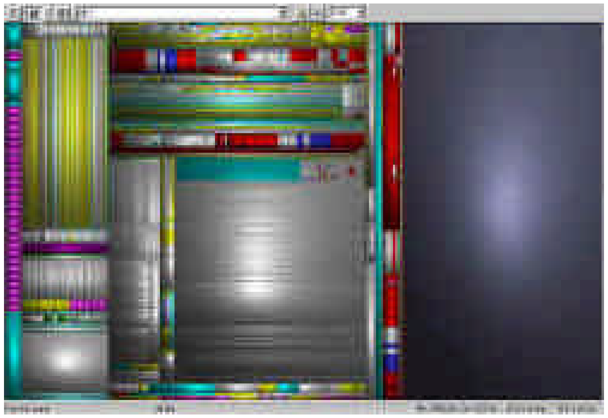


Other Treemap examples

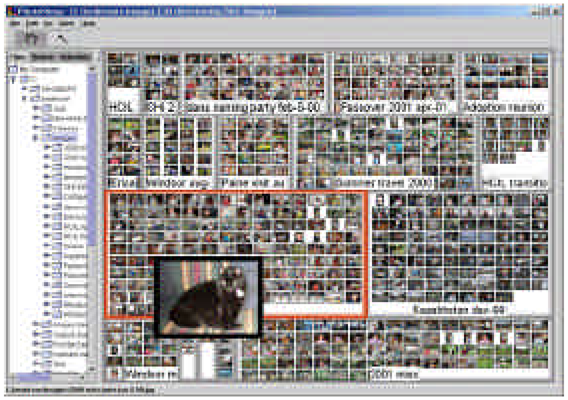
www.smartmoney.com



SequoiaView



PhotoMesa



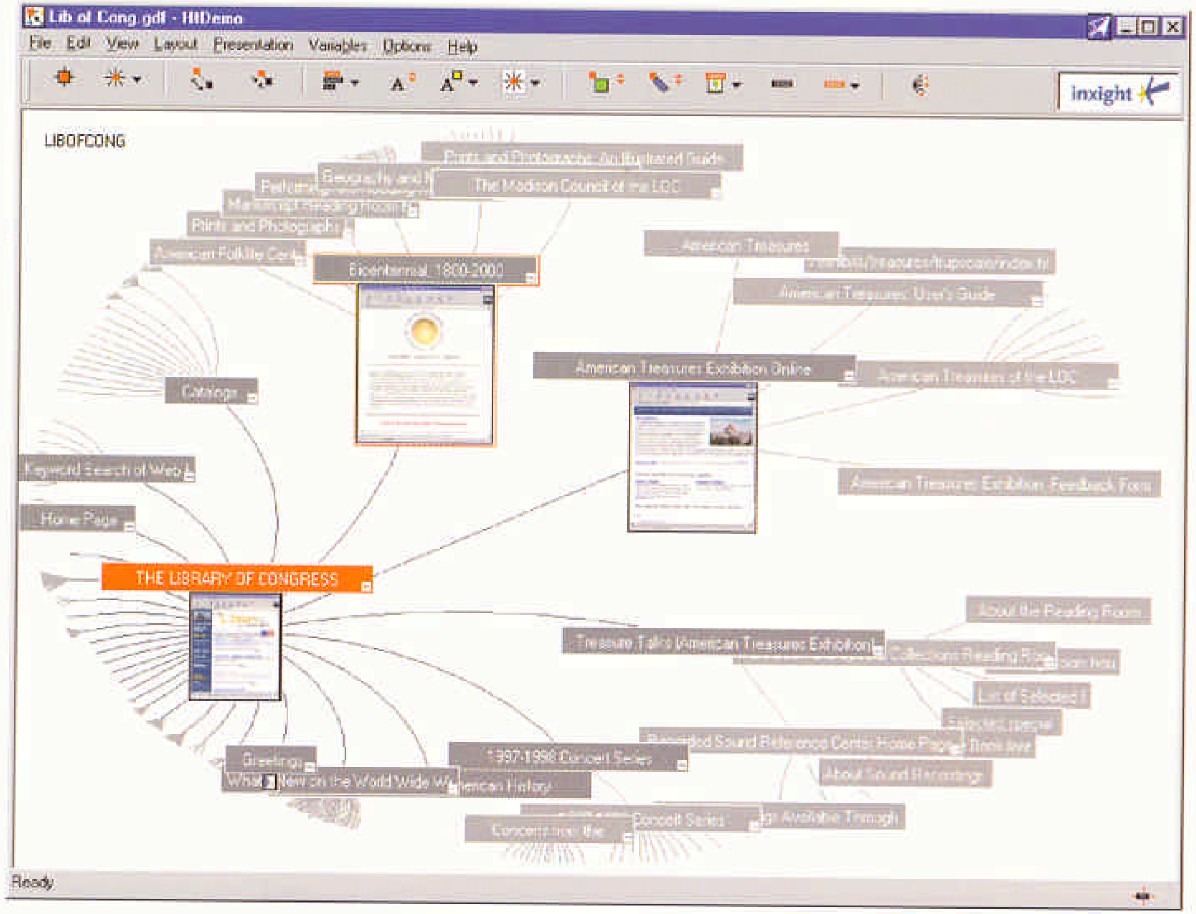
More on Treemaps

- History -- http://www.cs.umd.edu/hcil/treemaps/
- Algorithm variations
  - SliceAndDice - Ordered, very bad aspect ratios, stable
  - BinaryTree - Partially ordered, not very good aspect ratios, stable
  - Ordered - Partially ordered, medium aspect ratios, medium stability
  - Squarified - Unordered, best aspect ratios, medium stability
  - Strip - Ordered, medium aspect ratios, medium stability
  - http://www.cs.umd.edu/hcil/treemaps/java_algorithms/LayoutApplet.html -- compare them, open source available

Hyperbolic Trees

FIGURE 8.33
The 'long text' format of node labeling

Source: *Inxight Software, Inc.*



Cone Trees

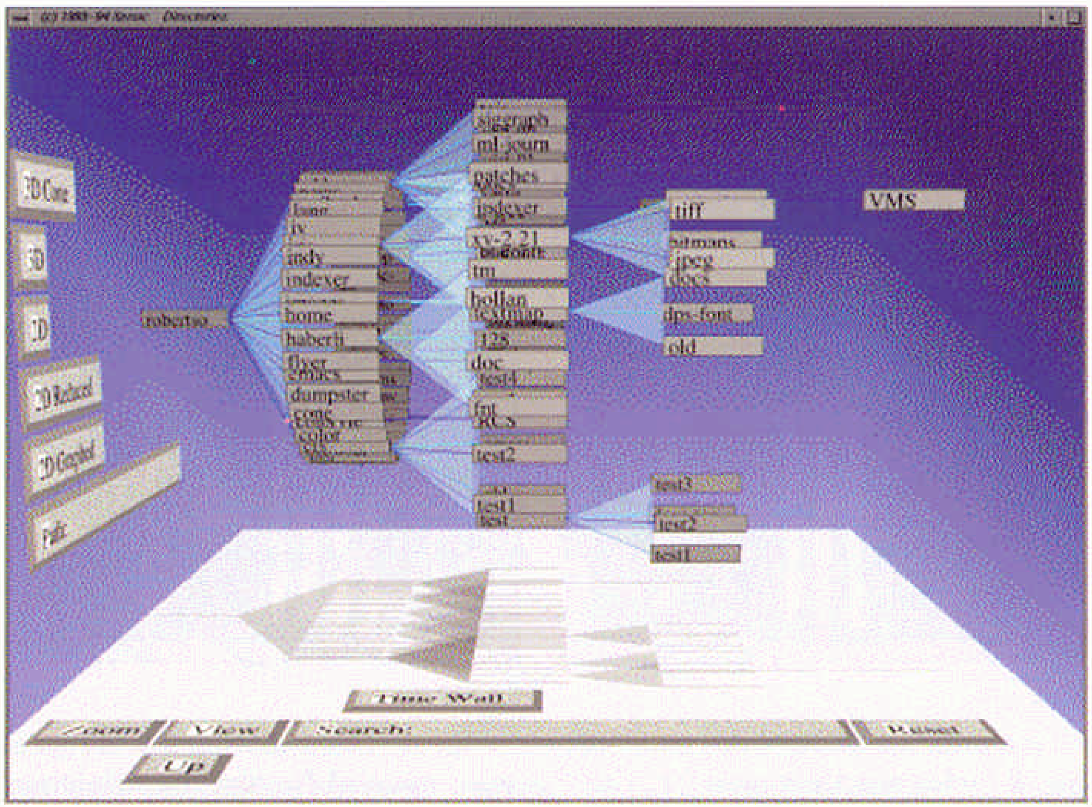
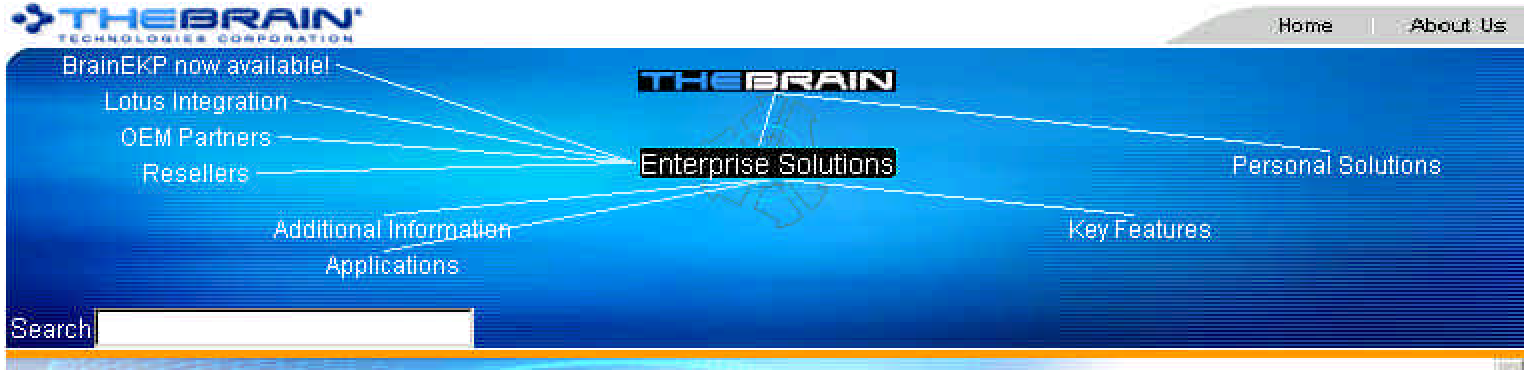


FIGURE 8.7
A Cone-Tree representation of hierarchical data

Source: *Xerox Corporation*

The Brain



http://www.thebrain.com/BrainEKPtour/ekptour.htm

Networks

- Network structures used for many things:
  - WWW, telephone networks, personal communications…
- Network have cycles (consequently not suitable for containment layouts)
- Often very large, with lots of links
- Problems:
  - Positioning nodes
  - Managing links
  - Scalability
  - Interactivity

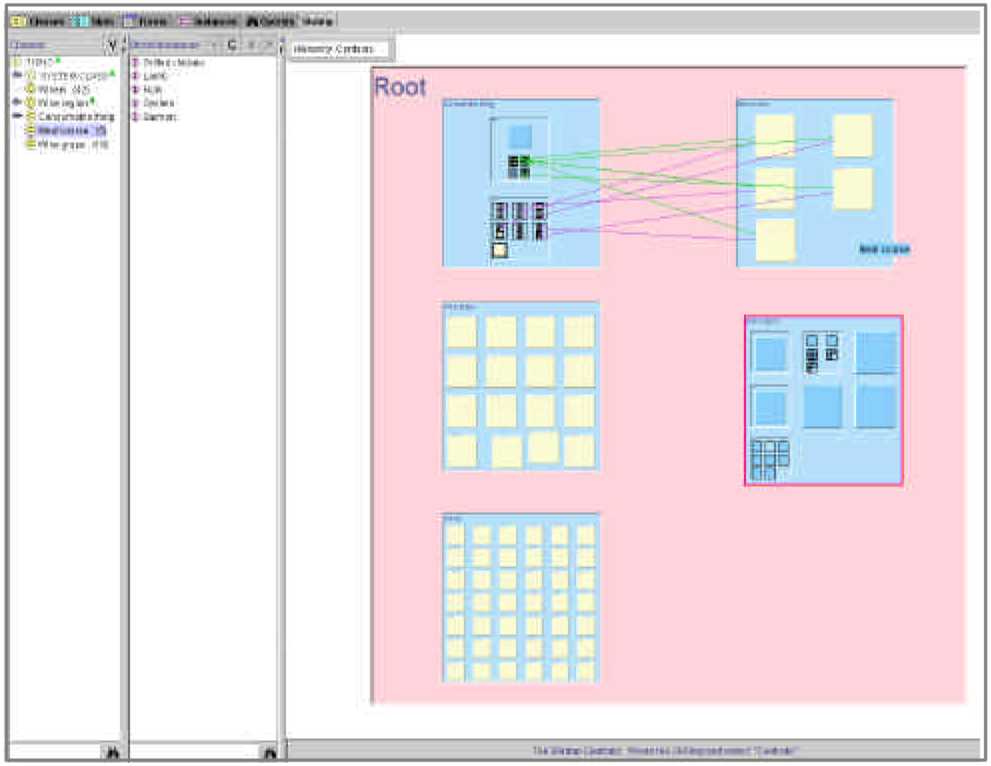
SHriMP Views

- **S**imple **H**ierarchical **M**ulti-**P**erspective Views
- A prototype environment for integrating various visualization techniques
- Improves use of limited screen area
- Integrates text browsing using hypertext (HTML objects) embedded in a graphical view

- Supports navigation and exploration of diverse perspectives of the information space
- Domain independent

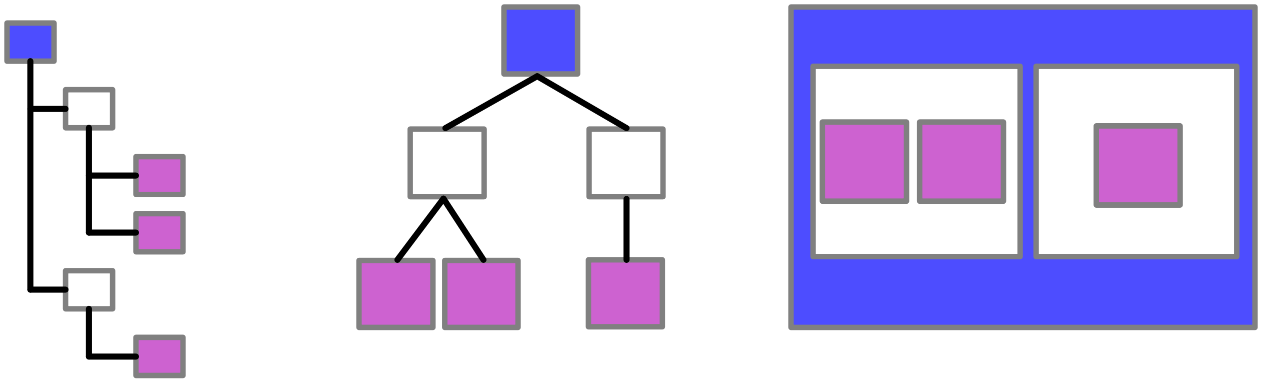
Jambalaya

- Protégé + SHriMP, using the Java Bean plug-in architecture supported by both tools
- The integration enabled alternative visualization and navigation mechanisms to be used in Protégé



Using Jambalaya to model knowledge

- Directed graph consisting of nodes and arcs
- Nodes represent concepts (classes) and instances
- Arcs represent relations between concepts and instances
  - Hierarchy relations (is-a, instance-of)
  - Structural relations and properties
- Nested nodes can be used in place of arcs for any kind of relation



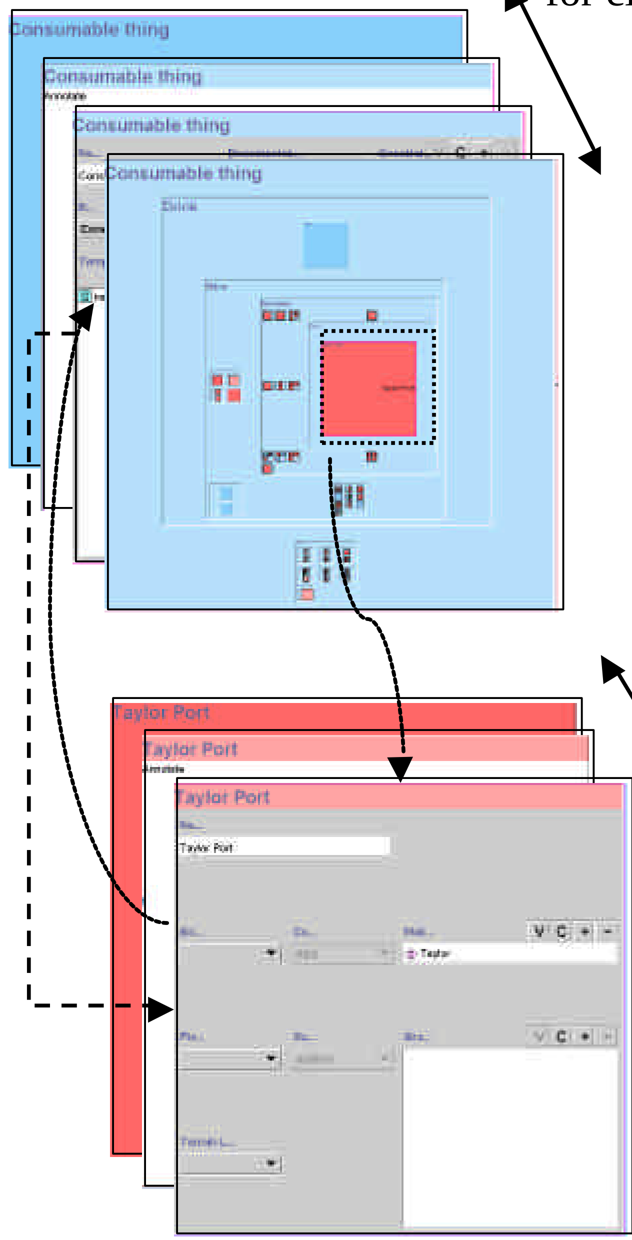
Nested Interchangeable Views in Jambalaya

Operations for switching views:

Zoom in/out
(default view shown on zoom in/out action is configurable)

Semantic zoom
(e.g. following a slot value to an instance)

Switching between interchangeable views using the hotbox



Interchangeable views for class nodes:

- Closed
- Annotation view
- Form view
- Children view (subclasses, instances of leaf classes)

Interchangeable views for instance nodes:

- Closed
- Annotation view
- Form view

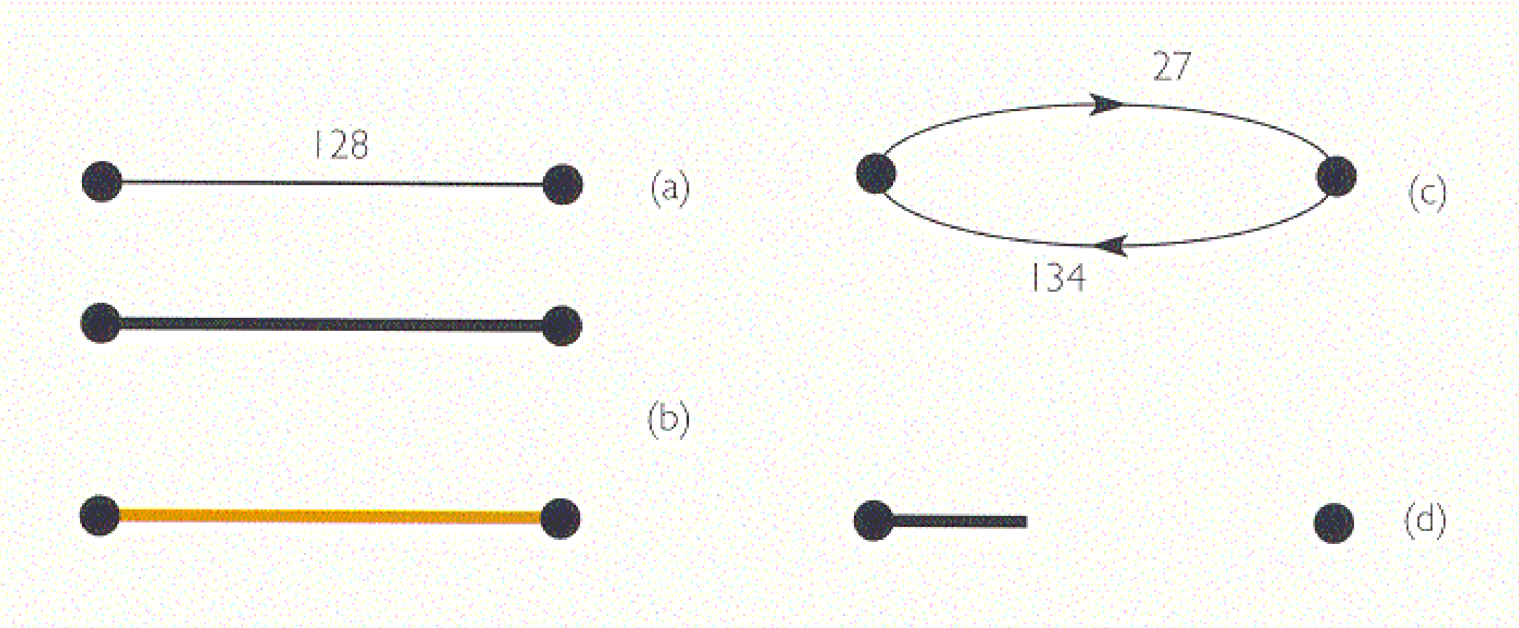
Navigation and Browsing

- 2 aspects of navigation
  - Recognizing location (orientation)
    - Current viewpoint
    - Show path to the current location
  - Controlling location
    - Relative movement
    - Absolute movement
    - Teleportation (bookmarks)
    - Hyperspace movement (using relationships)
    - Moving the space

Scalability – dealing with links

- Filtering
- Fisheye views (distortion)
- Abstractions (nodes and arcs)

Link representations (SeeNet)



SeeNet

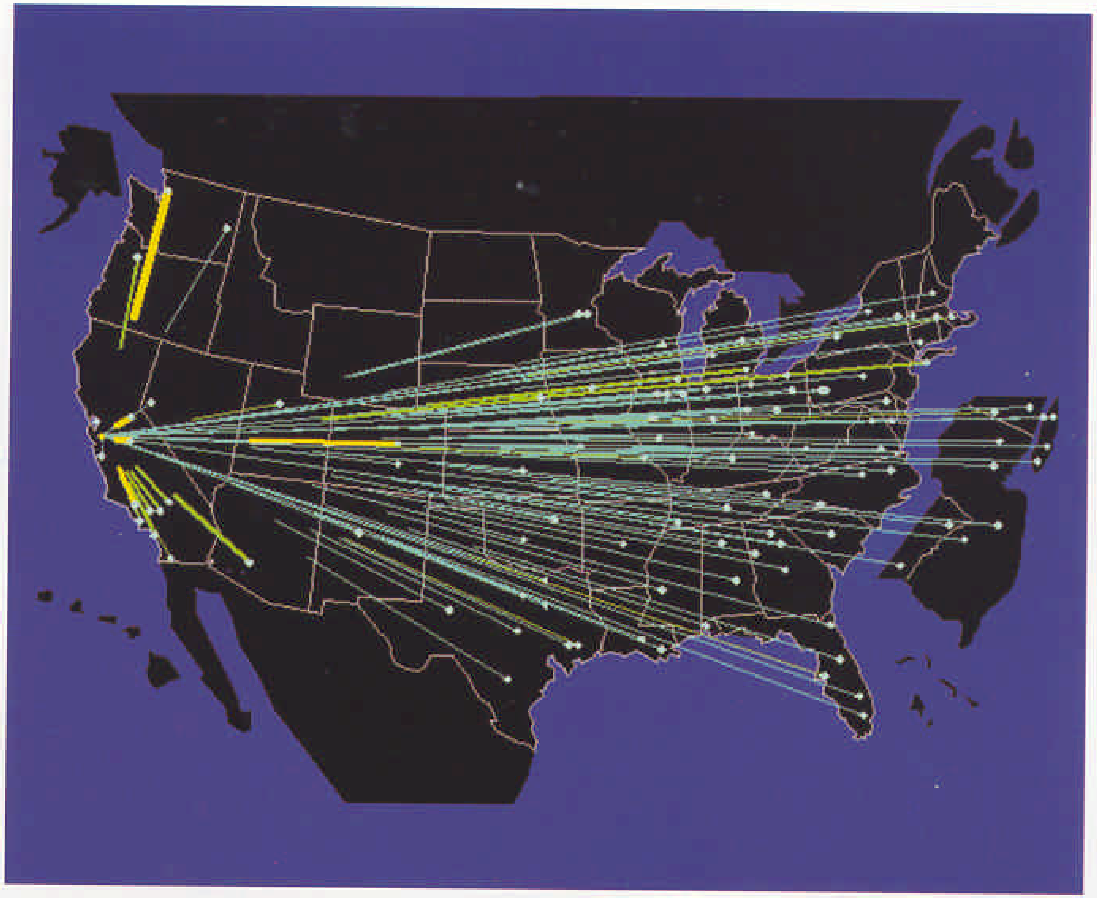


FIGURE 8.7
A linkmap showing the overload into and out of the Oakland node during the earthquake of October 17, 1989

Source: © *1995 IEEE*

SeeNet -- Linkmap

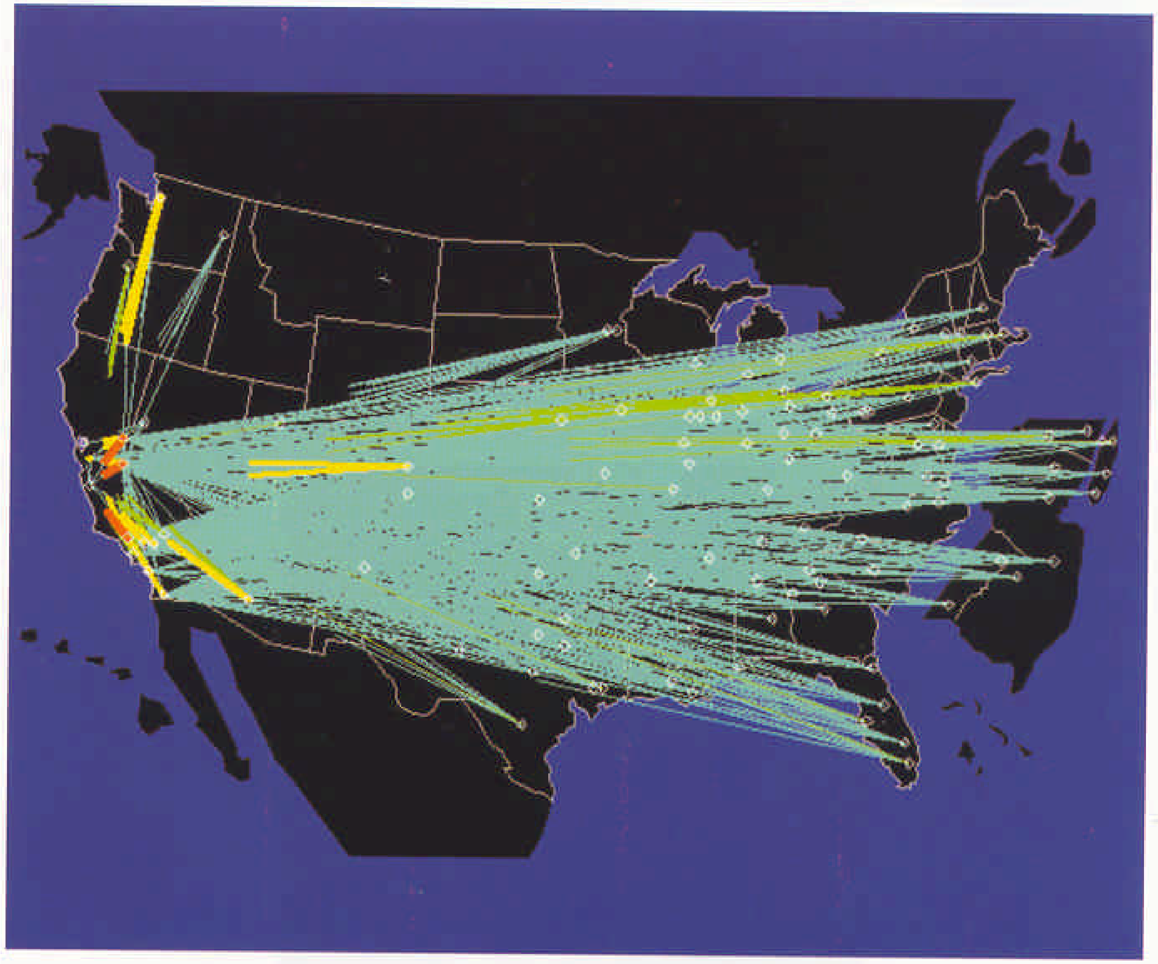


FIGURE 8.8
Illustrating the danger of occlusion
Source: © 1992 IEEE

SeeNet-- Nodemap

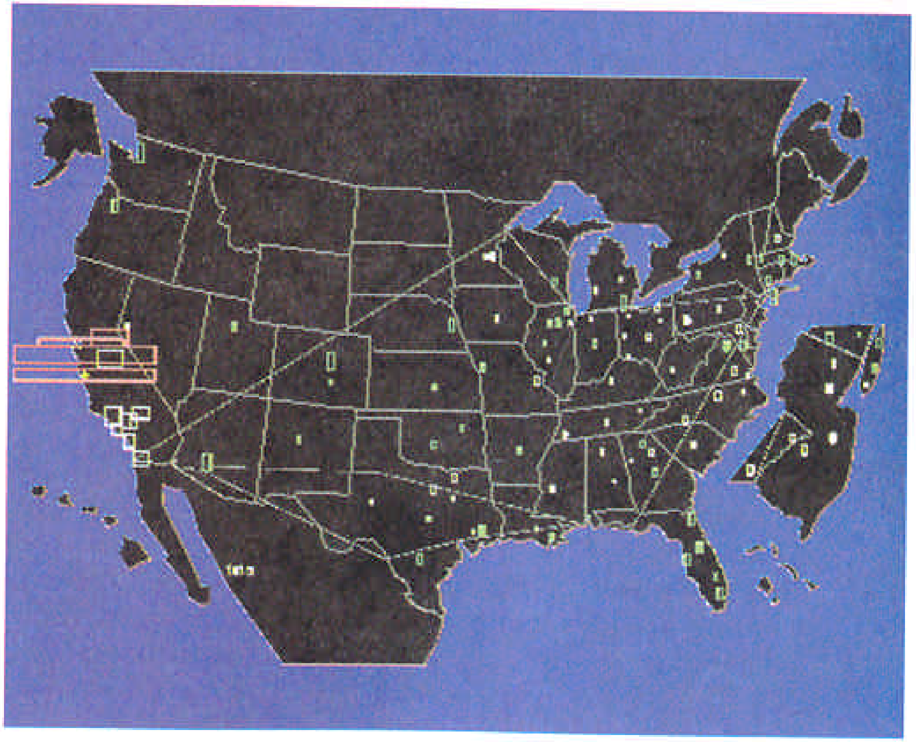
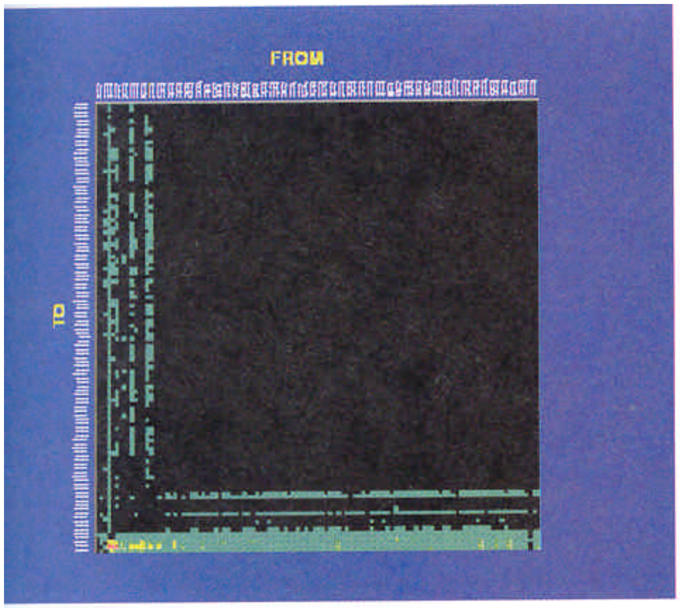


FIGURE 4

Overload Following Earthquake. Each rectangle shows the aggregate overload over all links; its horizontal dimension is proportional to the square root of the number of incoming calls in the preceding 5-minute period, and the vertical dimension encodes the outgoing calls.

SeeNet (variations in dealing with many links)



GURE 5

twork Overload As Matrix. The same overload as in Fig. 3 shown ing a matrix representation instead of a network map. The nodes shown along the rows and columns in approximate west-to-east der in matrix form, with columns corresponding to "from" nodes d rows corresponding to "to" nodes. At the intersection of each v and column there is a square whose color codes the link tistic. The colored squares on the left and bottom correspond to lines on Fig. 3. The nonsymmetry is due to the directed nature of traffic.

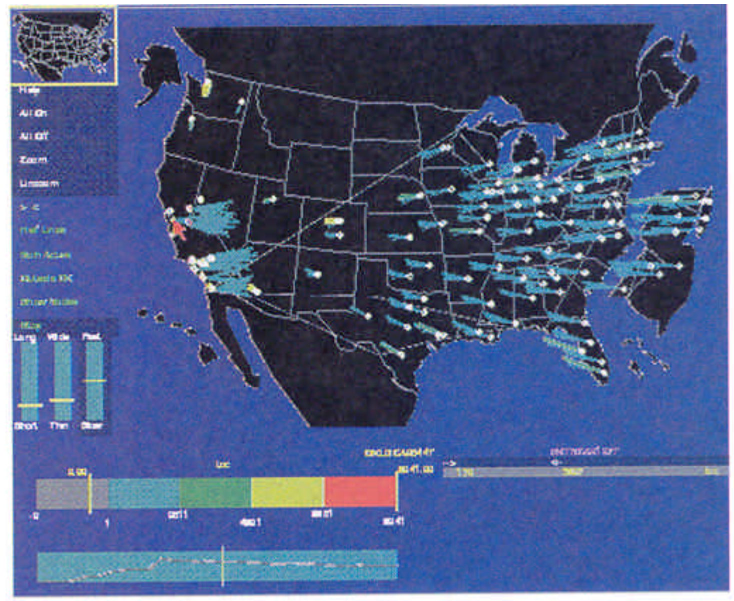
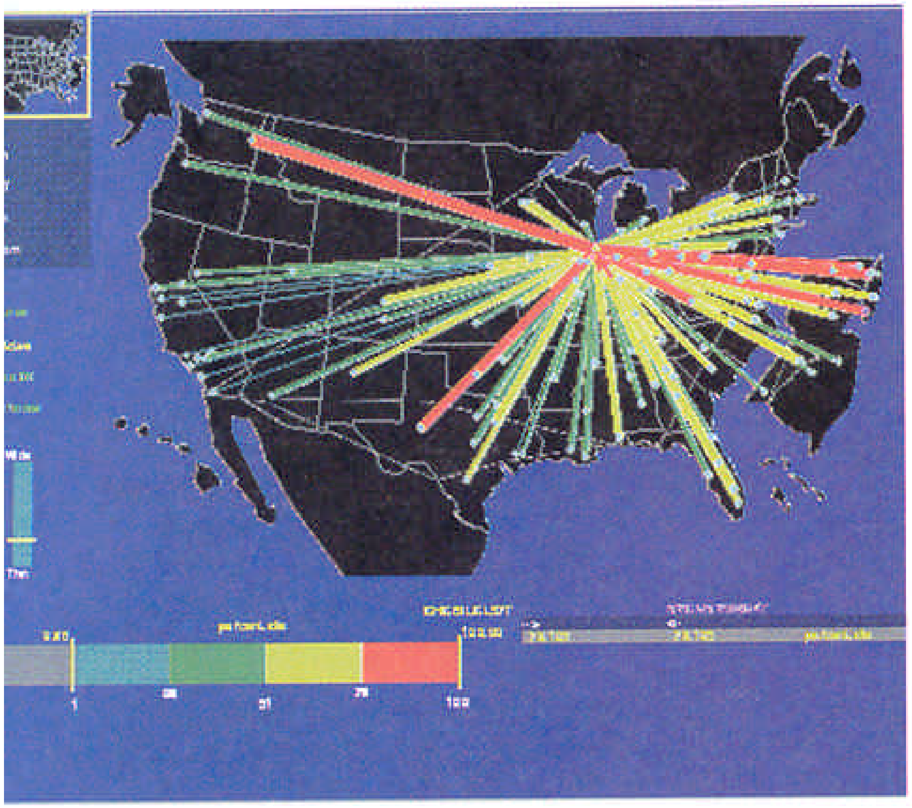


FIGURE 6

Line Shortening. The figure shows the same date as Fig. 3, except the half-lines are drawn only part way between the nodes that they connect.

SeeNet – looking at inverse of information



ƷURE 7

ı Network Capacity. The percentage of idle capacity on links into
l out of a Chicage node. By turning off all nodes and interactively
ıing on selected nodes, we can study a pervasive network
ːistic.

SeeNet – zooming in

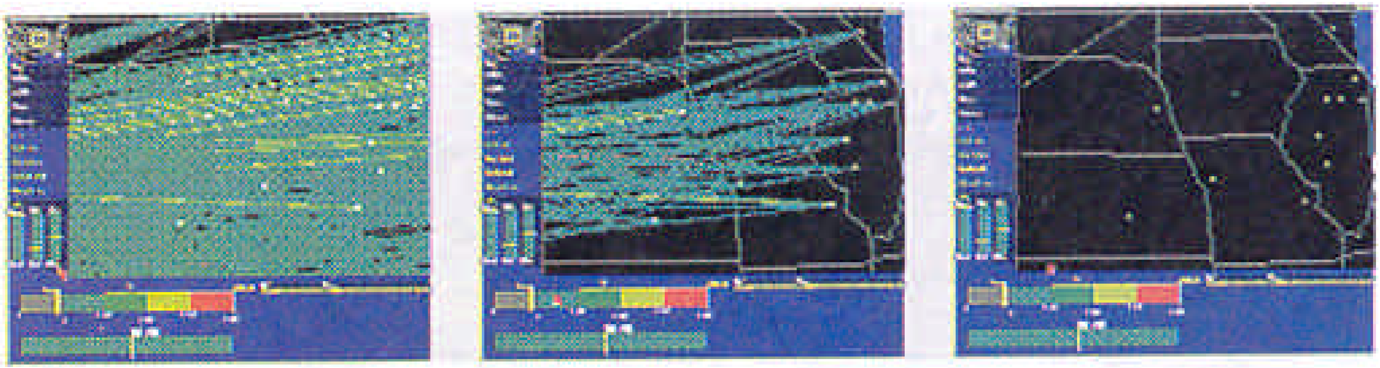
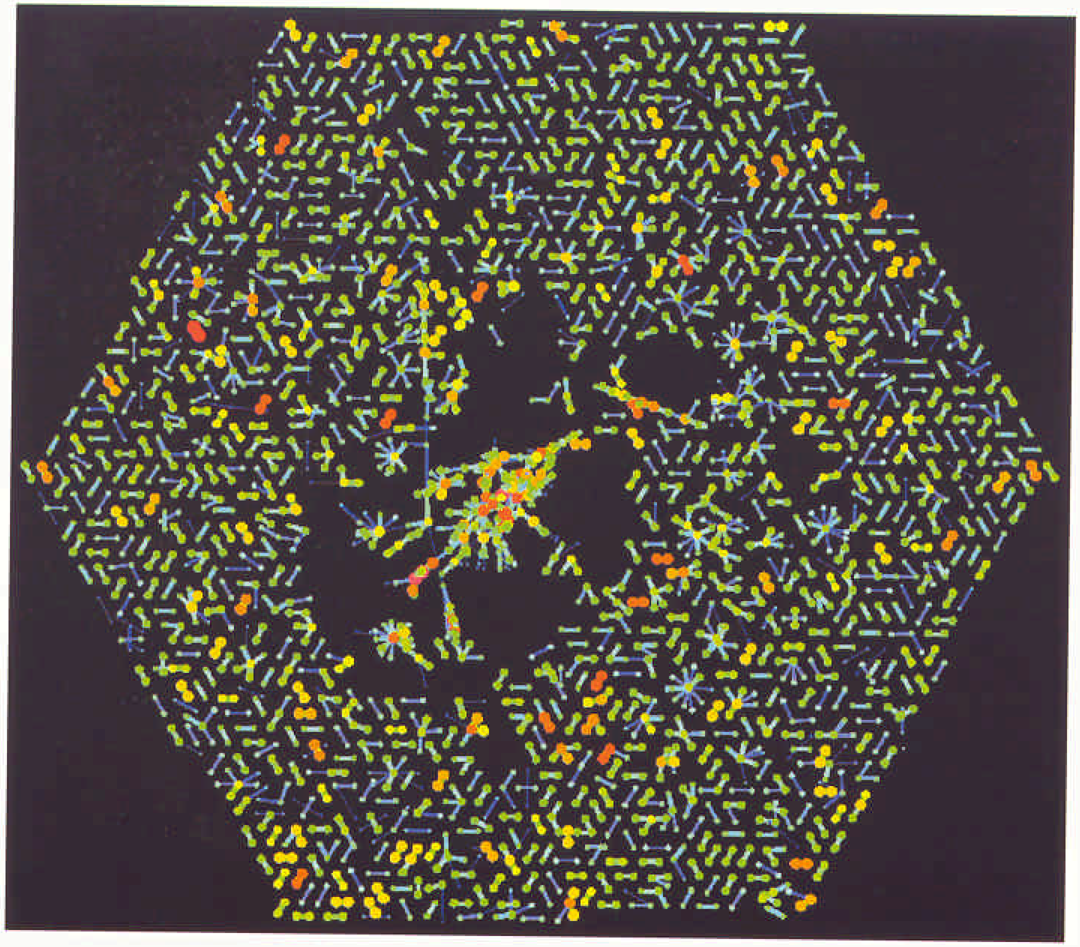


FIGURE 8

Interaction Between Links And Zooming. The zoomed area is in the interior of the network shown in Fig. 3. The left pane shows all lines, the middle pane shows all lines termination within the zoomed area, and the right pane shows all lines that both originate and terminate in the zoomed area.

Local telephone network

FIGURE 8.9
Communication within a local telephone network

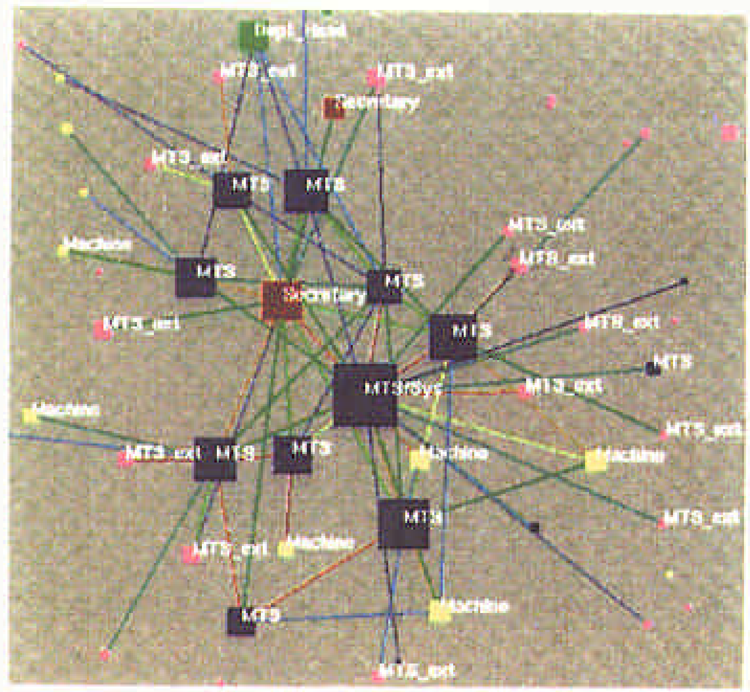


Electronic mail

FIGURE 8.10

Representation of email usage within a department

Source: © *1993 IEEE*



Netmap

FIGURE 8.11 The basic Netmap display, with groups of radial segments within an annulus, each segment associated with a particular person or institution or object

Source: *http://www.altaeurope.com*

Netmap

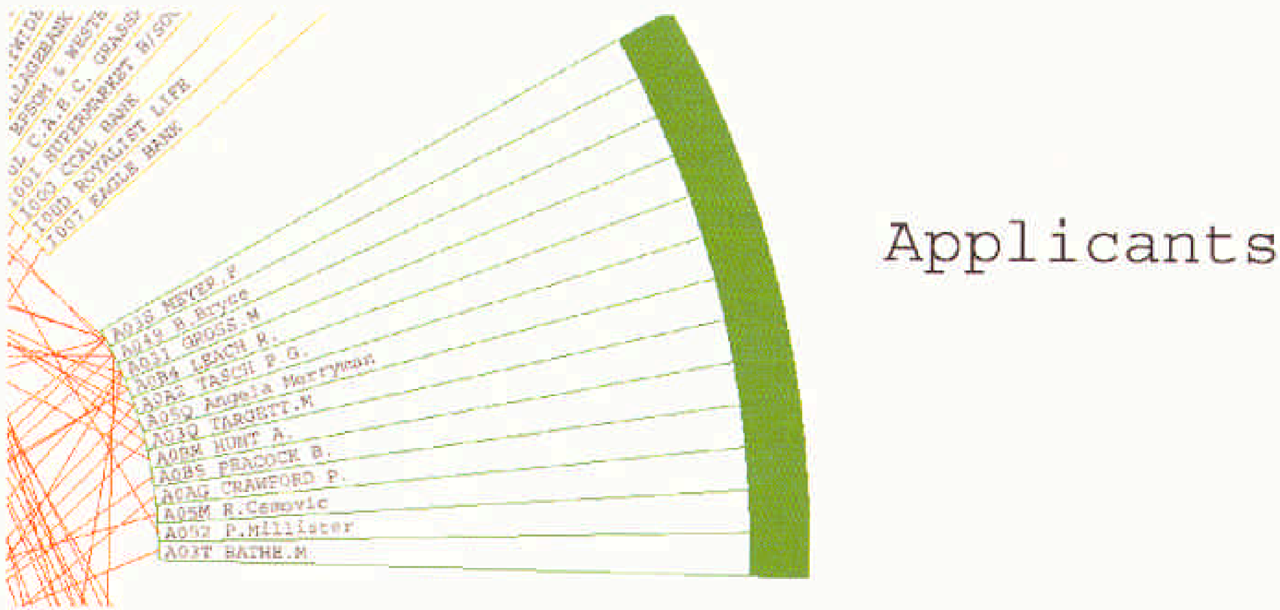


FIGURE 8.12 Detail of segments within a Netmap display

Source: *http://www.altaeurope.com*

Netmap

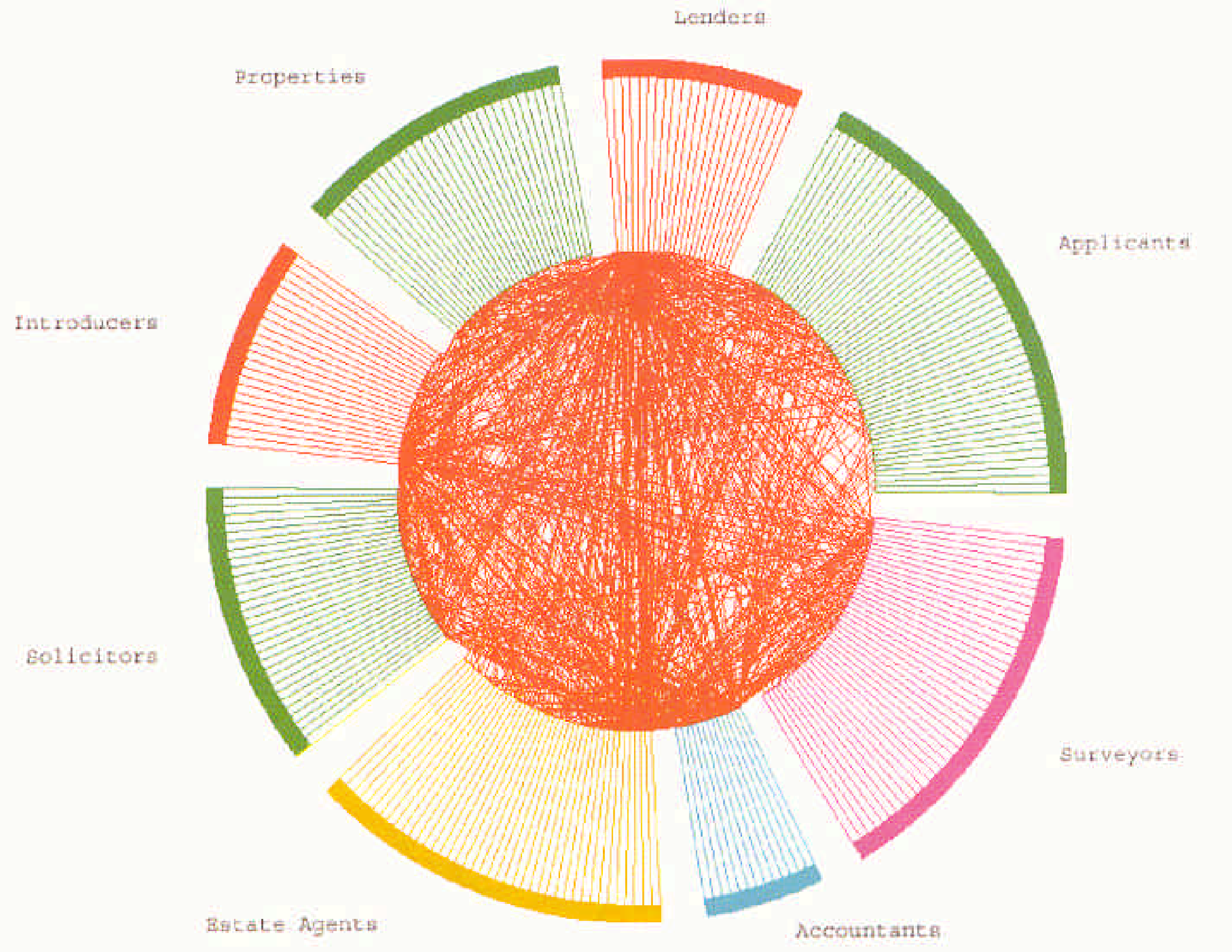


FIGURE 8.13
Lines in the interior of the Netmap display represent connections between items

Source: *http://www.altaeurope.com*

Netmap

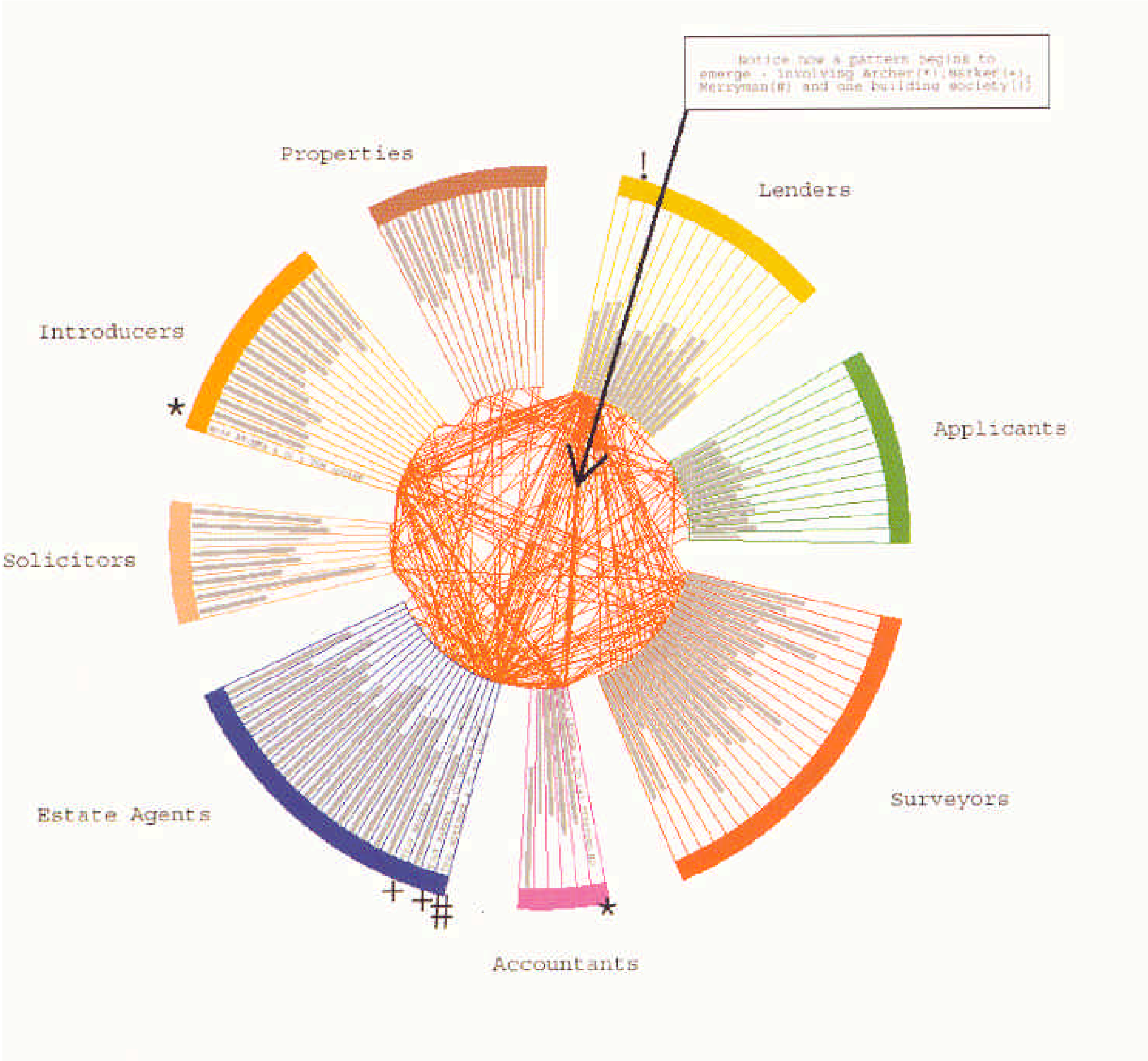


FIGURE 8.14
A threshold of three connections to a single item begins to show a pattern

Source: *http://www.altaeurope.com*

Netmap

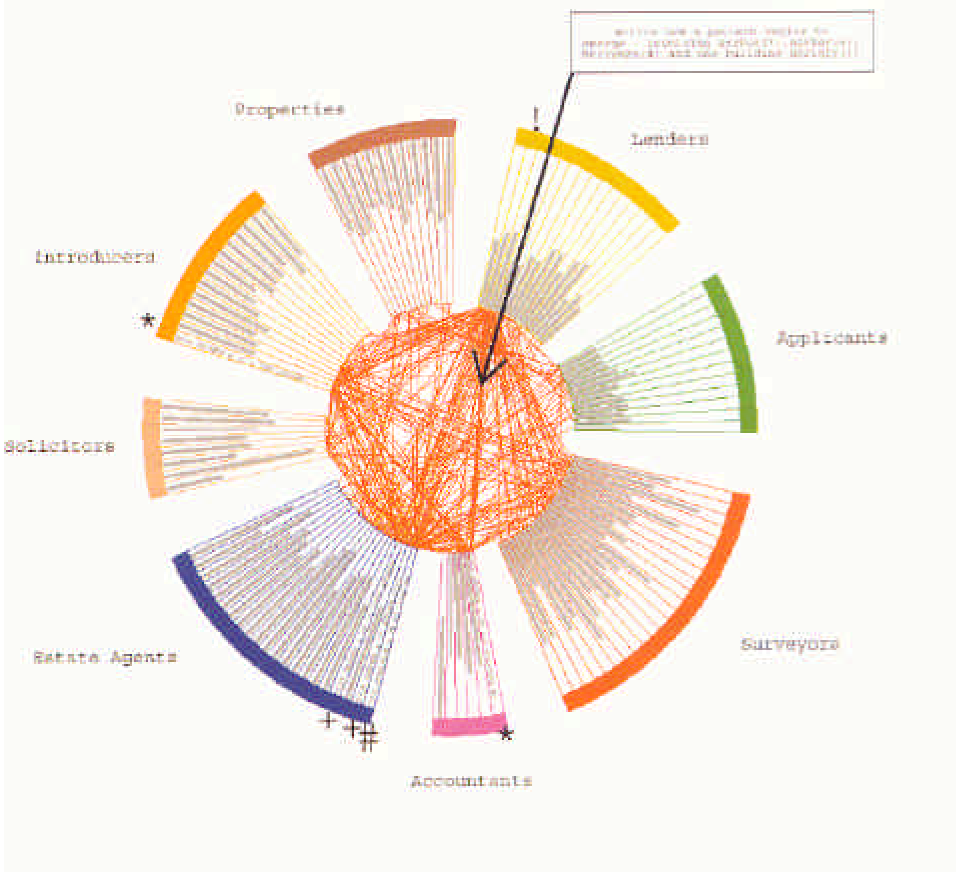


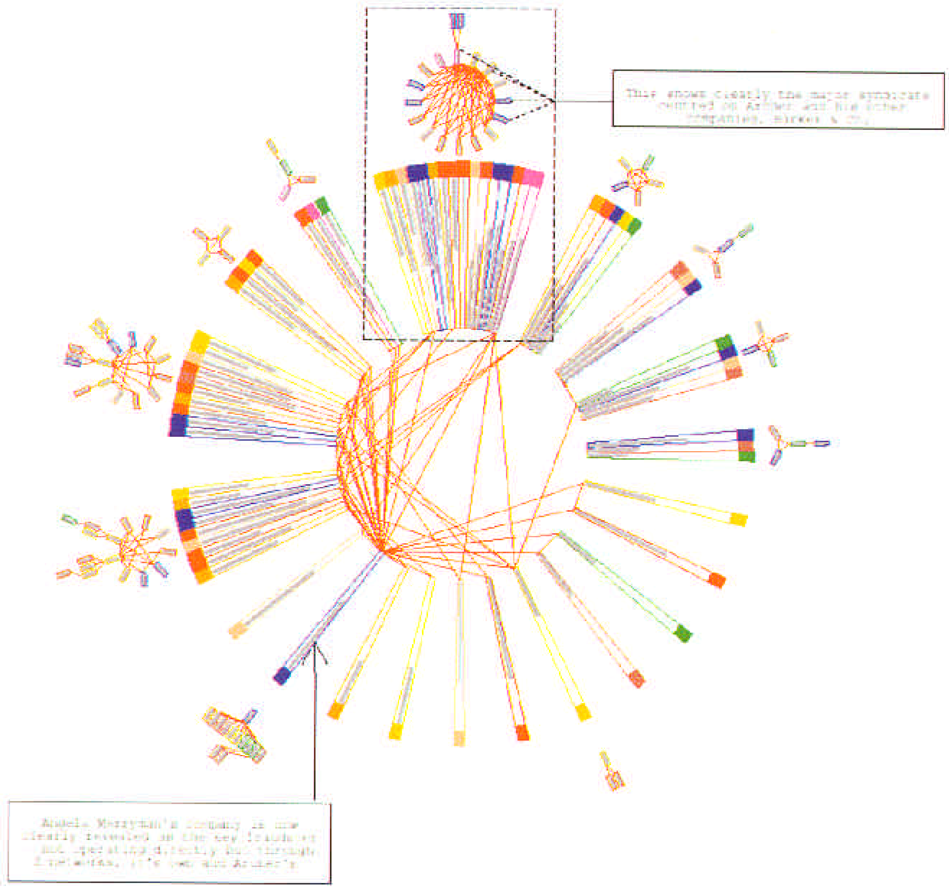
FIGURE 8.14 A threshold of three connections to a single item begins to show a pattern

Source: http://www.citdeurope.com

Netmap

FIGURE 8.15
The result of a clustering algorithm applied to items having 50% or more of their links with each other

Source: http://www.altaeurope.com



Daisy

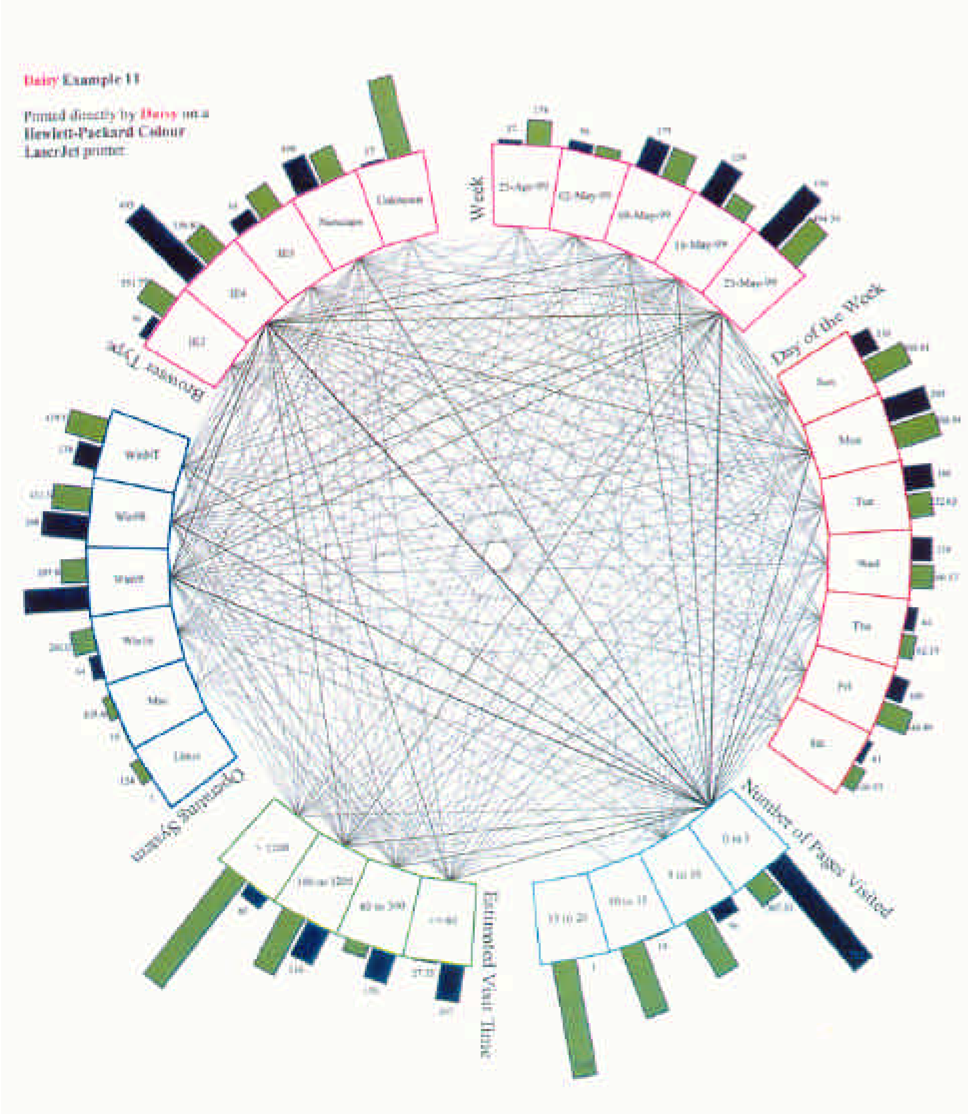


FIGURE 8.17 Bar charts associated with nodes in the Daisy visualization tool

Source: http://www.daisy.co.uk

Parameter Focusing

- Statistics
- Levels (thresholds)
- Geography/Topography
- Time
- Aggregation
- Size
- Color

Summary of Direct Manipulation Controls in SeeNet

- Identification
- Linkmap Parameter Controls
- Matrix Display Parameter Controls
- Nodemap Parameter Controls
- Animation
- Zooming and birds-eye views
- Conditioning (filtering)
- Sound
  - Node state changes
  - Slider values
  - Animation frame changes