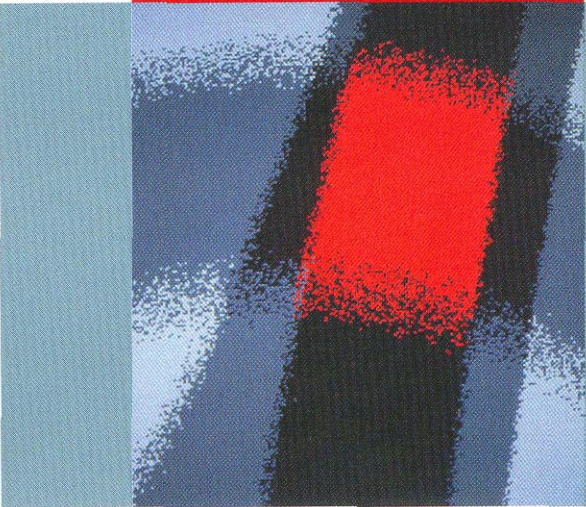




Connectivity

8.1 Connections

There is a class of phenomena whose underlying feature is that of connection, examples of which can be found in a wide range of fields. Harry Beck's static map of the London Underground railway system of 1931 (Figure 8.1) focused on connections since it was the problem of traveling underground from one station to another, *with little* concern for geography, that was the motivator (Garland, 1994); nevertheless, some element of spatial relevance was still essential. Earlier maps had also shown connectedness (or topology) but had allowed familiar geography to be the overriding consideration (Figure 8.2). Descriptions of wars to be found in historical or archaeological texts (Arnold, 1997) are also essentially describing connections, but additionally contain useful information such as who attacked who: here, a simple graphic presentation allows useful visualization (Figure 8.3). In many examples the basic connectivity information is enhanced by parameterization. Thus, SO-grams ('significant other' grams) employed by sociologists to describe close personal relationships (Davenport and Buckner, 1998; Davenport *et al.*, 1998) have a topology as their basis (Figure 8.4) but become far richer from parameterization with regard to the nature of interaction ('we speak on the phone every day', 'we exchange gifts at Christmas', 'we meet every Friday in the pub'). The organization chart of a company illustrates connectedness of a special kind in which personnel exist at clearly defined levels within a hierarchy. Similarly, visualization of the function call graph (that is, connectedness) associated with a software system can lead to the proposal of effective compilation strategies (Eick and Wills, 1993).

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



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

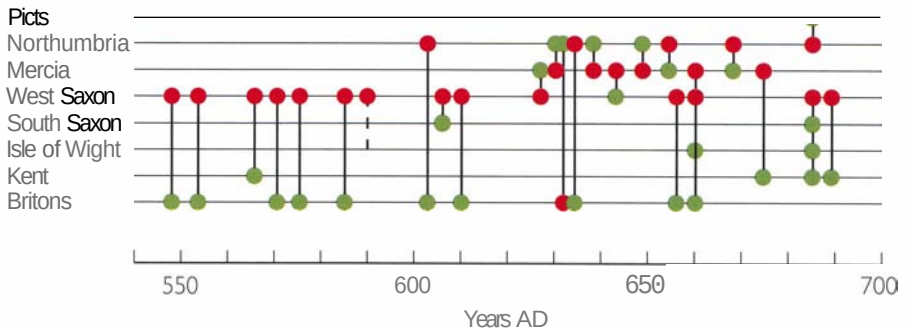
Source: London Underground Map designed by H.F. Stingemore (1927) © London Transport. Reproduced by kind permission of London Transport



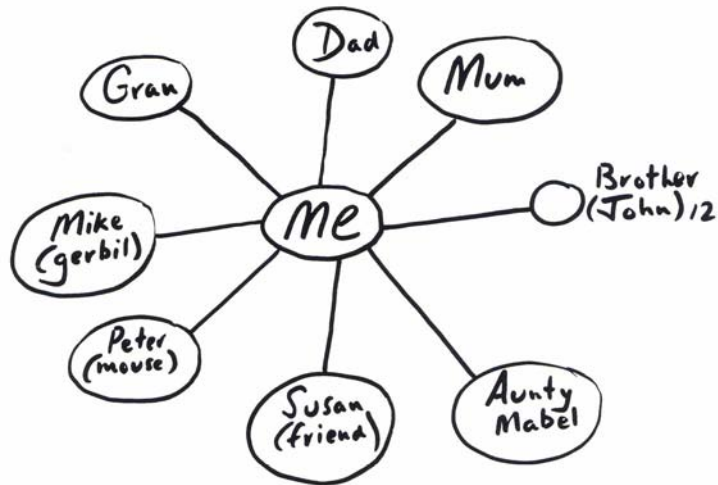
FIGURE 8.3

The incidence of warfare in early Anglo-Saxon England (AD 550–700). The red dots indicate the recorded aggressors

Source: After Arnold (1997)

**FIGURE 8.**

A hand-sketched significant other- (SO-) gram



Perhaps the principal current interest in the visualization of connectedness arises from the proliferation of communication networks such as telephone networks, email networks and the Internet. What characterizes many of these applications is their sheer size, a feature which renders the visualization of their structure and behaviour a non-trivial task.

8.1.1 Graph theory

A useful formalism for the study of connectivity is that of graphs and graph theory (Bondy and Murthy, 1976), the relevant terminology of which is briefly reviewed here. Connectedness is most conveniently represented by a graph composed of nodes and connecting links. An example from Chapter 2 (Figure 8.5) represented telephone conversations between a number of people: here, nodes represent people and links the occurrence of conversations. The example shows a disjoint graph reflecting the absence of telephone conversations between different clusters of individuals. Such a representation can easily contain information regarding directedness (leading to a directed graph). Parameters are often associated with links. In a telephone network, for example, each link can be characterized by its capacity, by the current traffic density, and by the

ratio of incoming to outgoing traffic. Similarly, nodes can have parameters associated with them, such as aggregated incoming and outgoing calls. The graphs associated with modern communication networks tend to be very large and extremely complex, so that one problem posed for the successful visualization of a network and its operation is the fact that its graph can rarely be presented in two dimensions without crossovers and without a danger of occlusion.

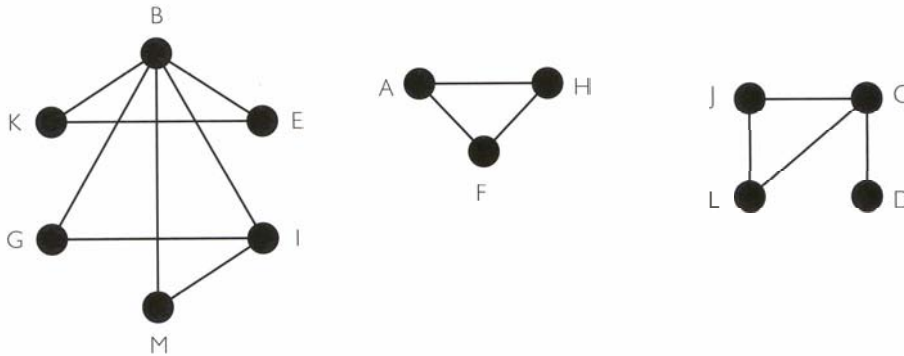


FIGURE 8.5

A network
representing
conversations

Overall, the treatment of connectedness falls into two classes. The first, which often presents the most severe problems, is concerned with general networks in which any node can be connected to any other node. The second class contains those examples in which the underlying graph is a tree; in other words, there are no loops in the graph. This class is far easier to handle, but is not without challenging problems.

8.2 General networks

8.2.1 The US telephone network

Our first example of a large general network will be the **AT&T** Long Distance network which is used by **AT&T** to carry long-distance calls within the United States. Over 110 nodes (called 'switches') are involved and they are nearly completely connected by over 12,000 links. A considerable volume of data is collected, typically every five minutes, concerning the performance of this network. Some data is associated with nodes and some with links. Node data includes aggregated incoming and outgoing traffic; link data includes traffic flow, capacity and overload.

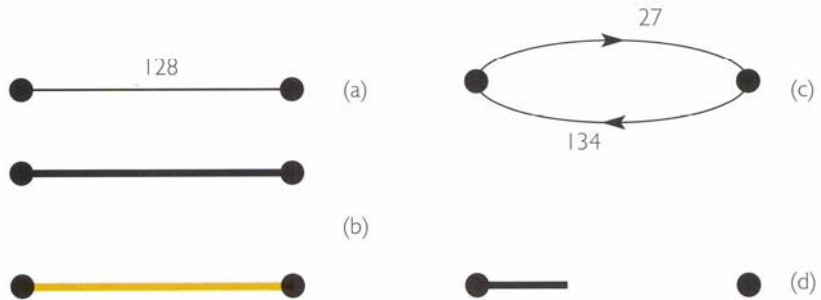
The insight being sought by the operator of the network will determine the appropriate visualization **technique(s)** to be used. Such insight might be in response to any of a wide range of questions, so we select as our example (Becker et al., 1995) the day of the San Francisco earthquake, October 17, 1989. During this event, relevant questions would include the following:

1. Where are the overloads?
2. Which links are carrying the most traffic?

3. Was there network damage?
4. Is there any underutilized capacity?
5. Are calls into the affected area completing or are they being blocked elsewhere in the network?
6. Is the overload increasing or decreasing?

It is obviously necessary to represent the connections – the links – in some way. An obvious solution is to use line segments between the nodes and characterize the traffic flow quantitatively (Figure 8.6(a)). Alternatively the traffic flow can be encoded by thickness or colour (Figure 8.6(b)). Any directional property can be indicated either by using two links (Figure 8.6(c)) or by dividing the link into two segments (Figure 8.6(d)). If one of the data values is zero, the division of Figure 8.6(d) has the advantage of being able to reduce clutter, though there may then be some doubt concerning the location of one of the two terminal nodes.

FIGURE 8.6
Representation
of links and
their
parameters



The representation of links is illustrated in the **linkmap** of Figure 8.7 which shows the overload into and out of the **Oakland** node during the earthquake. The divided representation of Figure 8.6(d) is employed to encode the overload by direction. However, if the same technique is used to investigate overload over the entire network (Figure 8.8), a disadvantage becomes apparent: there are many long lines that obscure most of the country. It is also difficult to see where a divided link terminates. These disadvantages can be ameliorated to some extent by drawing important links last (so that they appear on top) and by using line thickness as an encoding mechanism.

Figure 8.9 is another example of the presentation of connectivity-related data. For a local telephone network it shows, by the (colour coded) size of the dots, talk time within a four-hour period and, by line thickness, the talk time between individuals.

The impression should not be gained that it is a simple matter to choose the parameters which associate node and link quantities with the encoding mechanisms. As **Becker et al. (1995)** remark:

... it takes talent, and sometimes luck, to select the proper parameter values. ... The analyst manipulates the display parameters dynamically while watching the display change: good parameter[choice] is achieved when the display shows meaningful information about the data.

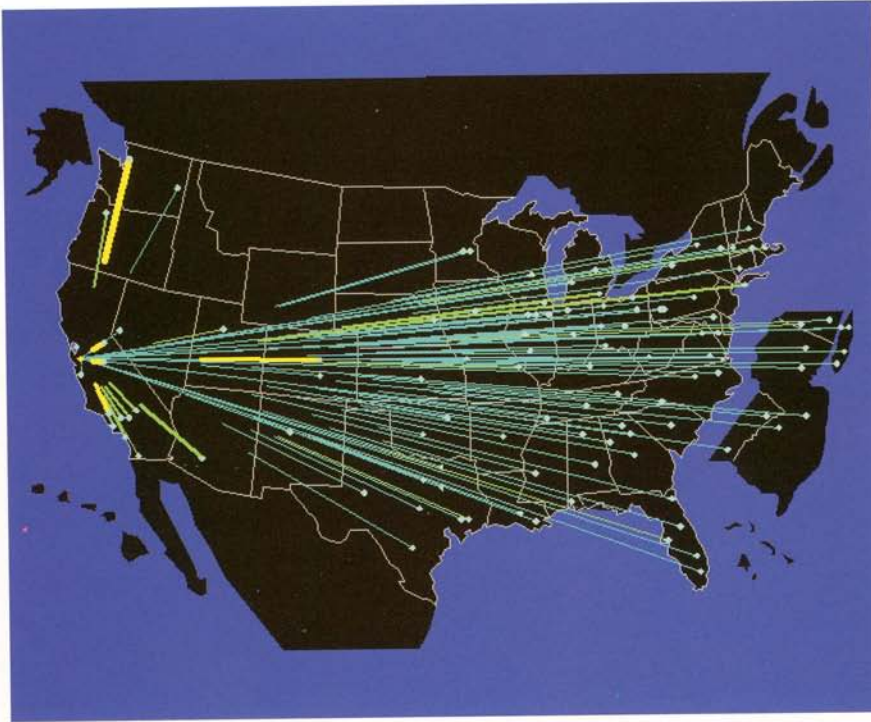


FIGURE 8.7
A linkmap showing the overload into and out of the Oakland node during the earthquake of October 17, 1989

Source: © 1995
IEEE

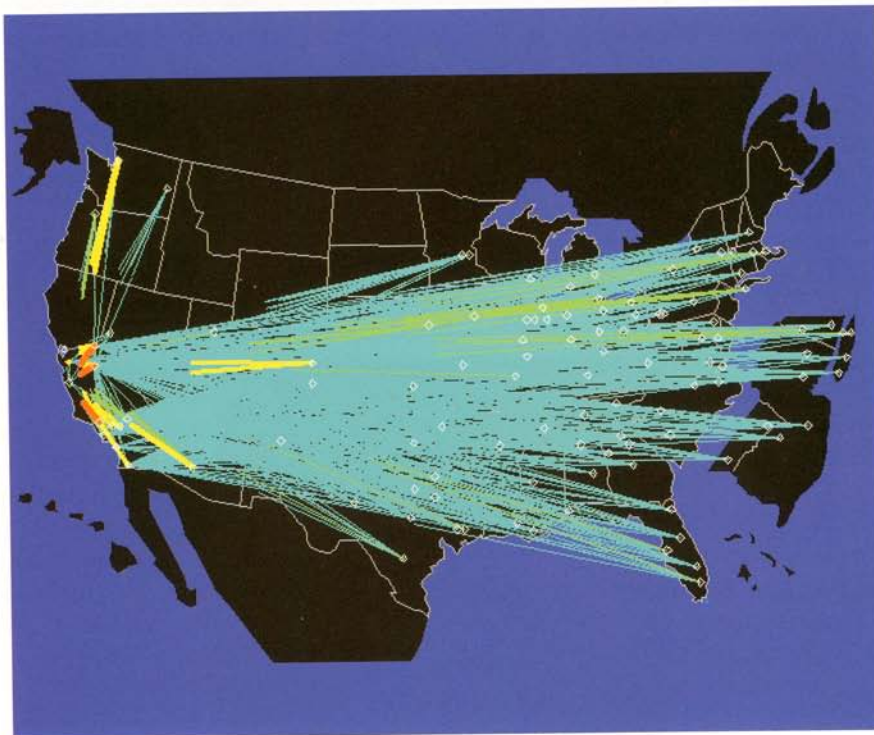
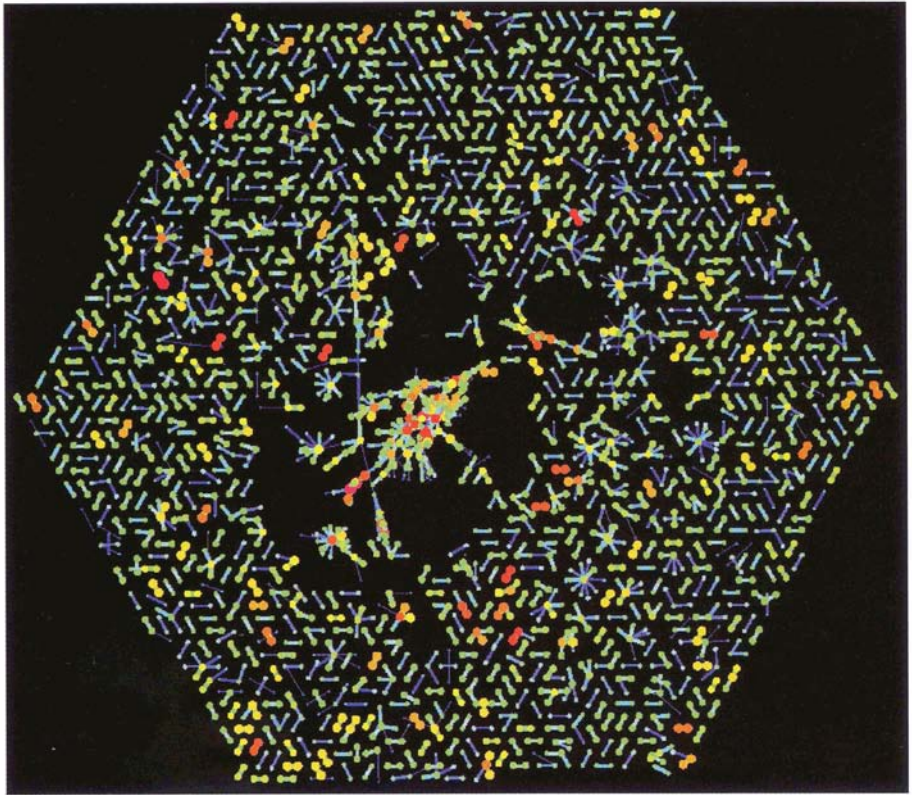


FIGURE 8.8
Illustrating the danger of occlusion

Source: © 1992
IEEE

FIGURE 8.9
Communication
within a local
telephone
network

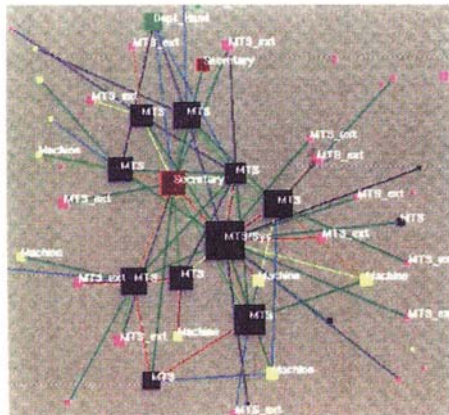


8.2.2 Electronic mail

Many visualization issues are common to telephone networks and the use of electronic mail. At a first glance the representation of **email** usage within a company department, as shown in Figure 8.10, has much in common with previous figures showing telephone traffic. There are, however, significant differences. With **email**, geographic location is not so relevant: while San Francisco has remained in its current and well-known position for a long time, the location of

FIGURE 8.10
Representation
of **email** usage
within a
department

Source: © 1993
IEEE



Mr Smith's office is not regarded as of great importance: instead, what is more important is for his node to be so placed within a display as to convey significant information about his email usage. It is also useful to classify different functions associated with the nodes: thus, red represents a clerical worker, whereas technical staff are blue and managers are green. However, just as with the telephone network presentation, node size indicates the aggregated volume of traffic, and links are encoded, in this case using the 'heat scale'⁷ – blue for few, through green and then yellow to red for many – to represent volume of email transmission.

The display of Figure 8.10 was generated by the HierNet tool (Eick and Wills, 1993) which offers many functions. An important one is that of placement, since proximity in the display can convey a great deal of understanding: for example the close collaboration between a small group of workers and the lack of interaction between others. Such placement is achieved by 'node placement algorithms' (Kruskal and Wish, 1979; Eick and Wills, 1993). In Figure 8.10 placement was determined according to the weighted links between nodes: not only did this allow certain interpretations regarding interpersonal collaboration but, as a bonus, tended to reduce clutter in the display. The HierNet tool also provided the ability to investigate statistics in the general manner discussed in Chapters 4 and 5, and also offered animation to investigate, for example, the change in personal collaborations over time.

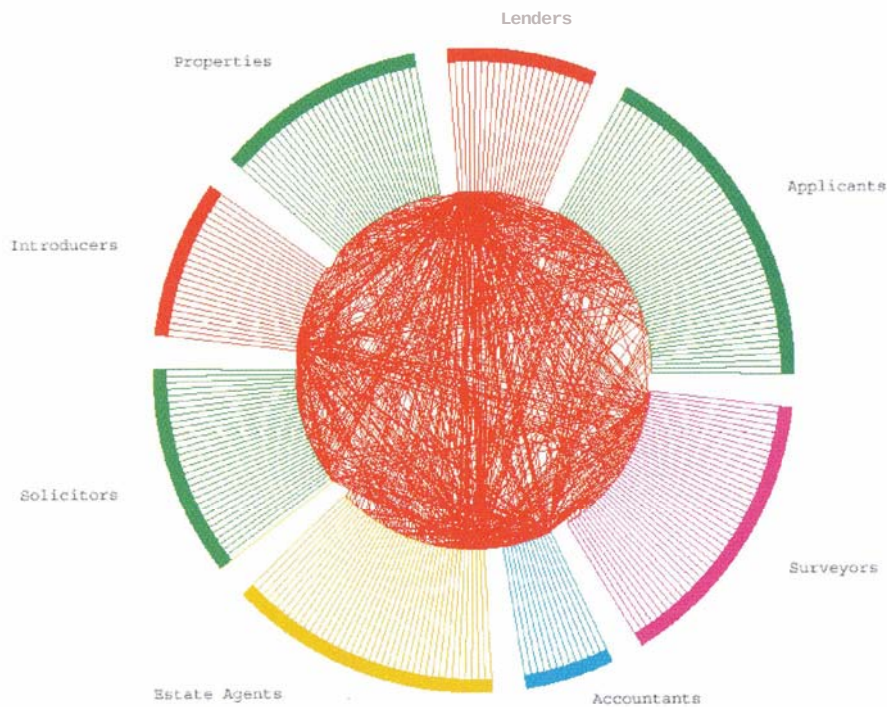
8.2.3 The Netmap visualization tool

Graphs can also usefully represent the connections between people and things and services. For example, if I decide to buy the house at 12 Park Street by borrowing money from the Dodgy Bank, and am professionally assisted in this activity by John Moneymaker, there is a connection, or relationship, between two people, a thing (the house) and an institution (the bank). How graphs representing such connections can provide a useful visualization tool can be illustrated by a particular, and quite remarkable, application of the Netmap tool¹ (Westphal and Blaxton, 1998; Davidson, 1993) to the activity of fraud detection.

The Serious Fraud Office (SFO) of the United Kingdom recently spent eight person years examining 48 file drawers of data in order to identify the perpetrator of a suspected building society fraud. An alleged perpetrator was identified, sent for trial and judged guilty. The same task was then set to a single investigator provided with the Netmap visualization tool which, in turn, had access to the same data. The same alleged perpetrator was identified within four weeks! What is more, the Netmap software made it possible, at the same time, to additionally identify the person who was behind the fraud.

The basic Netmap display has the form shown in Figure 8.11. Within an annulus, groups of radial segments are associated with similar entities; in the fraud example these include money lenders ('lenders'), house buyers ('applicants') (Figure 8.12), houses ('properties') and house sellers ('estate agents'). The central area is reserved for straight lines which represent connectivity. In our

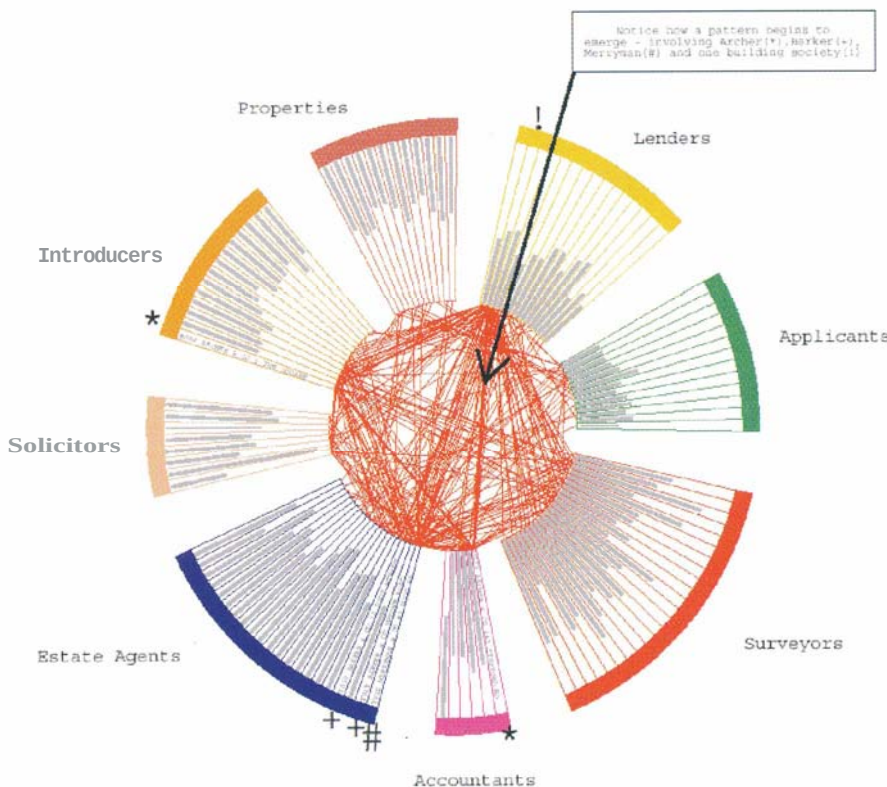
¹ Netmap is a registered trademark of ALTA Europe Ltd.

**FIGURE 8.13**

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

Source:

<http://www.altaeurope.com>

**FIGURE 8.14**

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

Source:

<http://www.altaeurope.com>

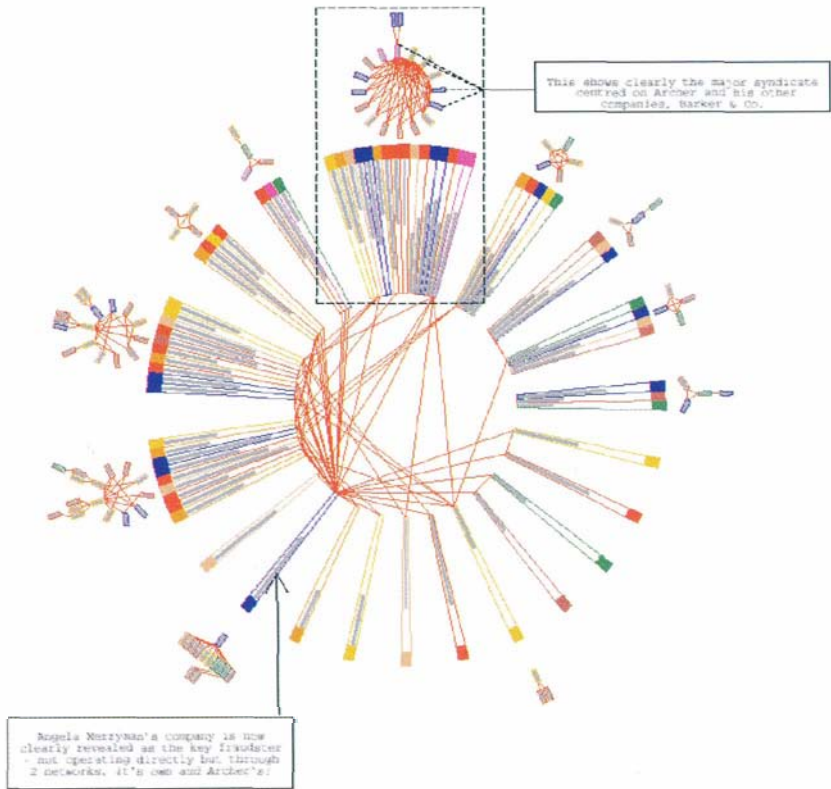
(indicated by !).² Based on the selected threshold, strongly interconnected items can now be brought together (Figure 8.15) by a clustering algorithm which groups items having more than 50 per cent of their linkages with each other, resulting in circles, outside the main annulus, which allow the investigator to focus on relevant data. The cluster at the top of Figure 8.15 clearly shows the major syndicate responsible for the fraud, and Merryman's company is revealed as the key fraudster – not operating directly but through two networks, its own and Archer's.

FIGURE 8.15

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

Source:

<http://w.aftaeurope.com>



Fraud investigation is just one example of the application of **Netmap**. By way of contrast its application to the investigation of social interaction within a community (Davenport and **Buckner**, 1998; Davenport *et al.*, 1998; Connolly, 1998) has been explored. **Netmap's** displays are not restricted to the 'wagonwheel' form shown in Figures 8.11 to 8.15. For example, the 'step-link' facility supports the ability to group nodes according to the way they are connected (Figure 8.16). Other connectivity-based tools are discussed by Westphal and Blaxton (1998), one of which – Daisy – has something in common with **Netmap**. Figure 8.17, for example, shows how attributes related to the nodes can usefully be encoded by bar charts within a Daisy display.

² All names and institutions are fictitious, for obvious reasons.

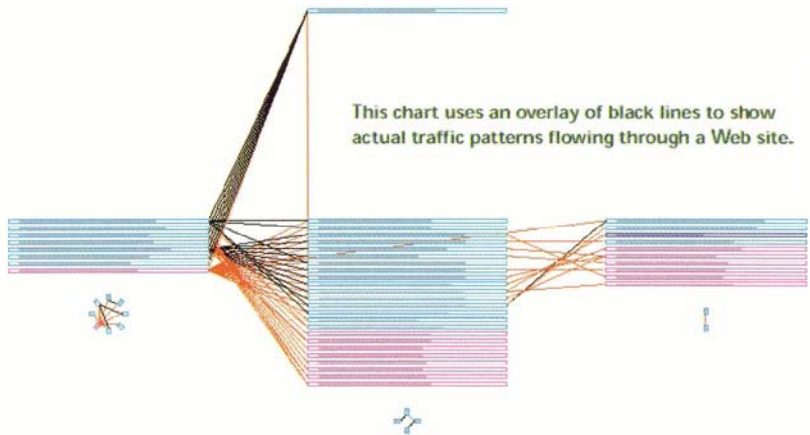


FIGURE 8.16
Illustration of the 'step-link' facility within Netmap

Source:
<http://www.altaeurope.com>

Daisy Example 11

Printed directly by Daisy on a Hewlett-Packard Colour LaserJet printer.

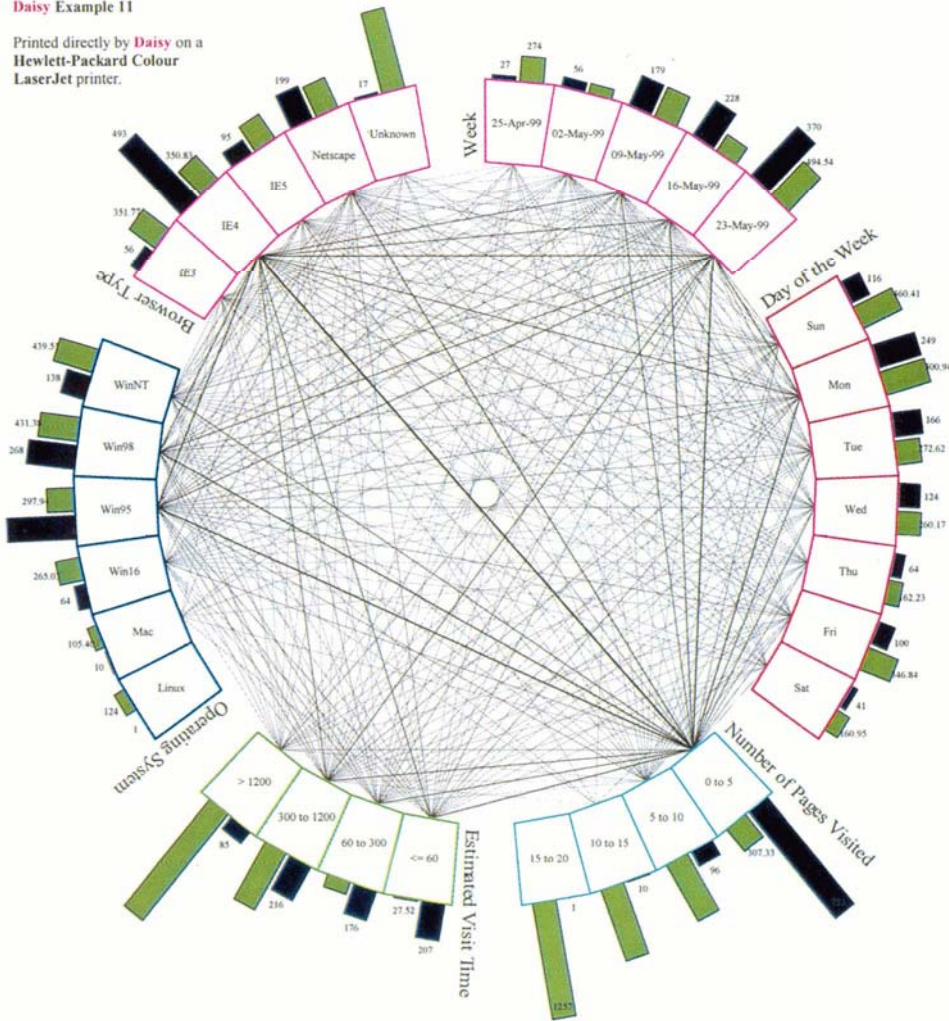


FIGURE 8.17
Bar charts associated with nodes in the Daisy visualization tool

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

8.2.4 World Wide Web

The World Wide Web is of enormous size and complexity (Bray, 1996), but is often held up as a prime example of anarchy: its description as the 'biggest car boot sale in the world' reflects something of its character. How big is it? Fifty million Web addresses and rising. How many Web sites? Over 100,000 and again rising rapidly. What about the content? A typical page size is around one to ten Kbytes, and around half contain at least one image reference. How 'connected' is it? Many pages contain at least one URL link: a large majority of sites are pointed to by from one to ten other sites.

From these statistics it is clear that, if the connections implicit in the Web were to be represented by a graph, it would be an enormous graph. So large, in fact, that there is little point in trying to represent or visualize it in its entirety. We must therefore consider the underlying motivation for Web visualization and, thereby, discover what visualization techniques can be of value (Hendley *et al.*, 1995).

Broadly speaking there are two classes of people who might wish to visualize aspects of the Web. One group has much in common with the telecommunication network specialists who **employ** displays such as those of Figures 8.7 to 8.9 in order to manage and understand global aspects of telephone traffic better. The group would, for example, include **website** administrators concerned with a site's effectiveness: they would ask questions such as 'Are people entering my site in the way I had imagined?', 'How are people getting to my site?' and 'What is the average path length to a particular node?'. Displays to support appropriate visualization would have much in common with those employed by telecommunication network specialists.

Web users

The other group for whom Web visualization is crucial are Web users, a truly enormous population. As suggested by Chapter 6, the basic reason for movement from one page to another can vary. The motivation may, for example, be to browse with a view to forming a cognitive collage (Figure 8.18) or, and perhaps concurrently, to search for either a well- or poorly-defined target (Figure 8.19). We do, however, know something of their Web traversal habits (Tauscher and Greenberg, 1997). For example, for a wide variety of reasons, the probability that the next page visited has previously been seen is about 58 per cent. This strongly suggests that, to ease a user's task, a display of their past trajectory (called a trace) would be useful, especially if a single direct interaction could



FIGURE 8.18
Browsing to form a cognitive collage

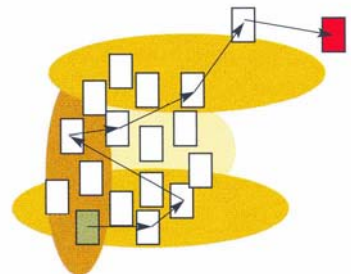


FIGURE 8.19
Search for a target

transfer them back to a previously visited page. Such *selective retreat* (selective revisitation), together with a display of trace (that is, previous nodes visited) was, in fact, found by Field and Apperley (1990) to be a valuable feature of conventional hierarchical menu search: these features were considered, on the basis of experimental evidence, to provide the user with an improved internal model that eased later use of the same menu structure.

Mendelzon (1996) has pointed out the value of a presentation of trace, especially if that presentation offers direct selection to effect transition to a page visited earlier. In fact, a graphical representation (Figure 8.20) of a past trajectory can be enriched considerably by using colour, position, size and arrow direction – to name only a few techniques – to encode such parameters as node dwell-time, **bookmarking**, revisitation and sequence. Such a facility has other advantages. As well as recognizing that a display of trace with selective retreat reduces a user's cognitive and physical burdens, Tauscher and Greenberg (1997) point out that 'they decrease resource usage by supplanting search engines for finding old pages, and by eliminating navigation through intermediate pages en route to the destination'.

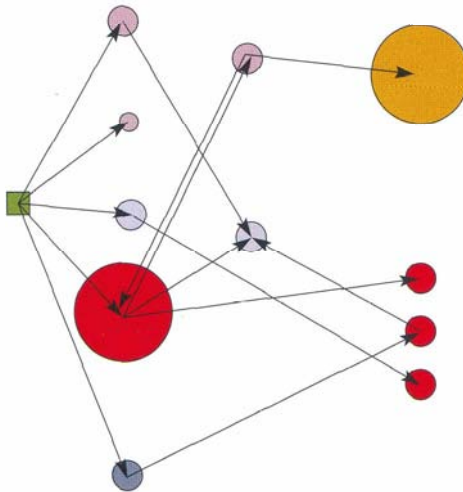


FIGURE 8.20

A display of trace, with retreat to an earlier page achieved by direct selection

A display of trace is also expected to be of value following movement, towards a target, that has been automated by a search query (Figure 8.21). Here, the user is teleported a long (semantic) distance and then requires a localized search to find a target which may only be a few links away: local trace would be helpful. Such 'query initiated browsing' (Furnas 1997) is an emerging paradigm on the Web, and will require appropriate visualization techniques.

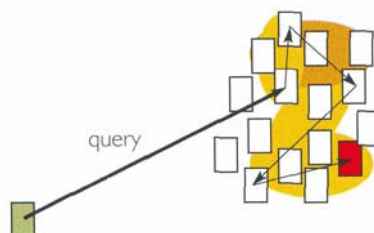


FIGURE 8.21

Finding an item of information: query initiated browsing

'Where can I go now?'

'Where am I?' and 'Where have I been?' are not the only important questions asked by the Web user. Especially if they have an intended destination or problem in mind, they also ask 'Where can I (usefully) go from here?'. If the computer is given no hint as to the user's needs (that is, if no query facility is used), it can only provide some indication of which destinations can be reached from the current page. Such an indication can be useful either as context to enhance the user's internal model of their current location or because a move to one of the destinations might bring the user closer to a desired page. Such a 'look ahead' facility was in fact investigated in the context of textual hierarchical menus by Snowberry et al. (1985) and found to be instrumental in success in arriving quickly at a required leaf node, suggesting that a look-ahead facility may also be found valuable by Web users.

A number of visualization tools are available to provide 'look ahead' information. A sketch illustrating one possibility is shown in Figure 8.22. Here the concept of a 'bubble' (Boardman, 1995, 2000) is used to represent a page: a bubble can be 'blown' to conceal all inner detail, or 'burst' to display link destinations. Selection of a link destination could automatically blow the current bubble to conceal its contents and take the user to the selected page. Retreat to an earlier page will burst that bubble and disclose its own link destinations. In this way a display of trace and the facilities of selective retreat and look-ahead are combined in a single visualization tool. There are many opportunities for encoding links and bubbles, and rendering them sensitive to interaction, in a display such as Figure 8.22.

Where should I go?

Once aware of possible next steps, the question as to which a user should take to satisfy some intention must be answered, and hopefully supported by visualization. In current practice, the Web user makes a subjective assessment based on a – perhaps cursory – examination of the available links, and makes a decision about which page to visit next. An alternative approach, though a challenging one for the visualization tool designer, is to provide, within what is directly visible, some clue or 'scent' (Pirulli, 1997) indicating in more or less detail what is available by selecting a particular page; in other words, some sensitivity measure. Such provision is difficult within an essentially anarchic information structure such as the Web. On the other hand, within a well administered and managed structure such as an intranet it might be possible, through the use of coding techniques such as those discussed in Chapter 4, to provide such scent. In conventional menu systems the administrator attempts to provide scent through the appropriate choice of category labels. In the browsing of a very large text collection (Pirulli, 1997) scent was provided automatically by an analysis of available texts. An interesting variant of the concept of scent is that of 'footprints' (Wexelblat and Maes, 1999) descriptive of popular paths explored by others, and which could easily be represented in a display such as that of Figure 8.22.

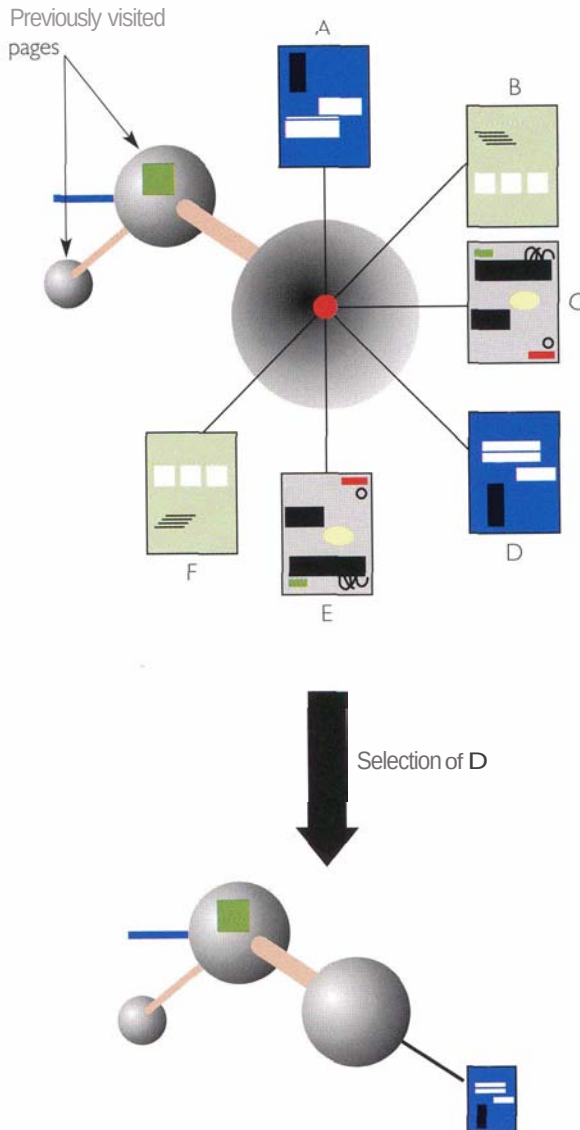


FIGURE 8.22
The use of 'bubbles' to form a representation of trace and to permit both retreat and the display of detail

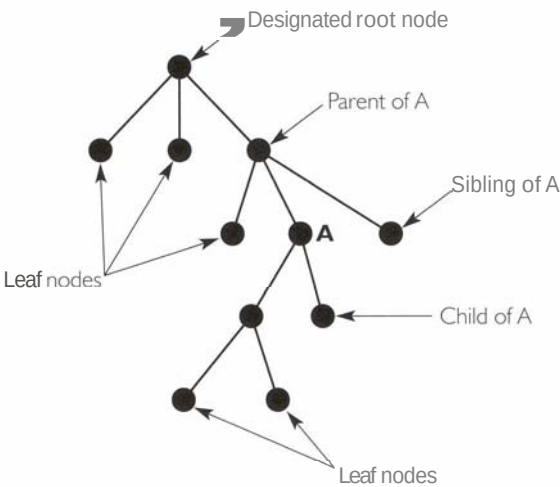
8.3 Tree structures

Many relations are characterized by connectedness of a special sort, that which is represented by a graph which is a tree. Examples have already been given in Section 8.1. One is a company reporting structure, with the CEO at the 'root' and many others reporting to a single person and being reported to, in turn, by subordinates. In such an example the designation of a 'root node' is not fundamental to the graph but is determined by what the graph represents: any node in a tree can be so designated. Another example is a software package in which each subroutine is called only by one superordinate function. In the case where a subroutine may be called by more than one function, one or more loops will be

introduced into the graph; nevertheless, it may still be possible to employ visualization techniques originally devised for trees (Furnas and Zachs, 1994). Nodes at the (conventional) extremities of the tree are often referred to as 'target nodes' in view of the fact that the function of non-target nodes is frequently to guide the user through a menu to a single source of information. Within a tree we frequently refer to parent, sibling and child nodes (Figure 8.23).

FIGURE 8.23

The terminology associated with trees



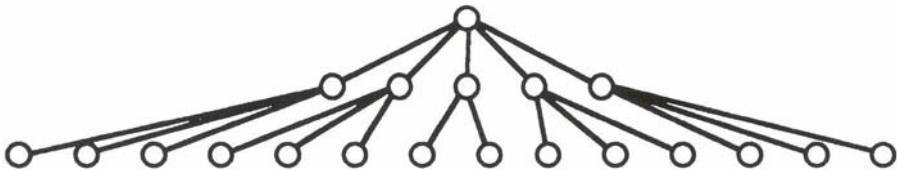
A major problem with trees is that their conventional representation (Figure 8.24) occupies a great deal of space, usually to the extent that for more than three levels and more than 50 nodes at the third level, the tree is too large for effective presentation on a display screen. The search for a better presentation of a tree, and ways in which interaction can be supportive, has resulted in some effectiveschemes, three of which are now presented.

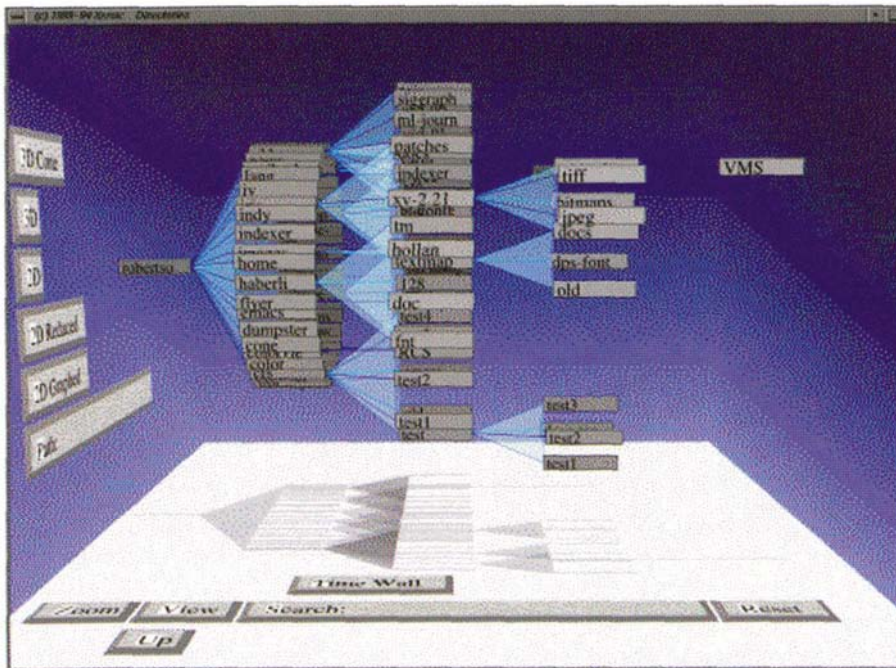
8.3.1 Cone-Trees

Imagine a 'flat' tree, as shown in two-dimensional space in Figure 8.24, to be remodeled, in three-dimensional space, in such a way that all nodes subordinate to a given node are arranged in a circle which, together with the superordinate node, forms a 'cone' as shown in Figure 8.25. A two-dimensional view of that rearrangement, as shown in Figure 8.25, is now more compact and,

FIGURE 8.24

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



**FIGURE 8.25**

A Cone-Tree representation of hierarchical data

Source: Xerox Corporation

notwithstanding substantial occlusion, easier to handle. Such a view is called a Cone-Tree (Robertson et al., 1991). The user of a cone tree may, for example, wish to see the reporting path of an employee within an organization represented by the cone tree, in which case entry of the employee's name by some means will bring about any cone rotations needed to position that employee (as well as his reporting path) to the foreground. Following this action, the user might wish to locate another employee to see how they are related to the previous one, in which case entry of the new name would again cause rearrangement of the cones to bring the new name to the fore. Recalling the relevant remarks in Chapter 6, however, we realize that rotation of the cones should not occur instantaneously, and for good reason: a smooth rotation, supported by 'shadows' of the cones on the base plane, supports the maintenance of the user's internal model of the hierarchy. In the words of Stu Card, these techniques 'constitute animation [which] allows the perceptual system to track rotations. The perceptual phenomena of object constancy enables the user to track substructure relations without thinking about it'. A horizontal orientation of the cone tree, called the cam tree (Robertson et al., 1991) may be more convenient for the layout of node names.

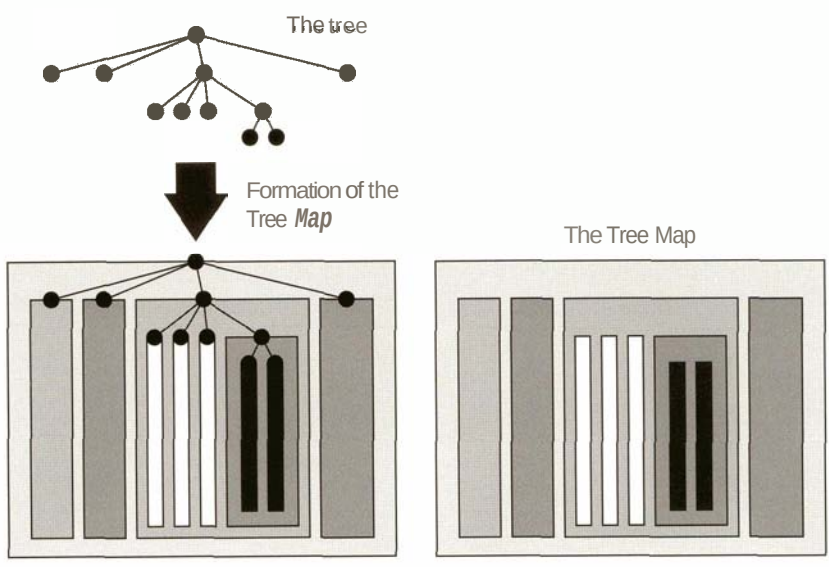
8.3.2 Tree maps

An alternative representation of a tree is the Tree Map (Johnson and Shneiderman, 1991; Shneiderman, 1992). Its derivation from the conventional representation (Figure 8.24) is straightforward and is illustrated in Figure 8.26. The procedure is simple: starting with the designated root node one draws a rectangle; usually, in order to make good use of screen space, it will be almost the same size as the display. Within that rectangle are smaller rectangles, one for

FIGURE 8.26

The derivation of a Tree Map from a tree

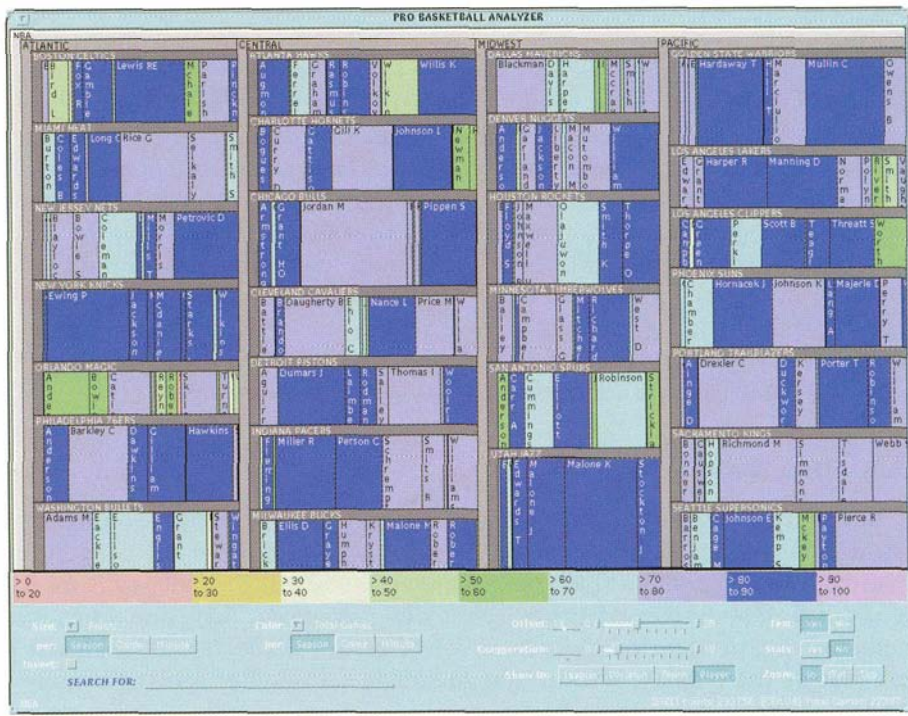
Source: Ben Shneiderman



each subordinate node of the node just considered. This construction is repeated until all nodes are accounted for. There is no constraint (except the resolution of the display) on the depth of the tree, and no requirement that all target nodes are at the same level or that the fan-out of each node is the same. Figure 8.27 shows a tree-map representation of baseball data. Reportedly (Johnson, 1993), fifteen minutes of training can lead to effective use of a Tree Map.

FIGURE 8.27

A Tree Map representation of baseball data



Once derived, individual rectangles can be encoded by colour to denote some attribute: the age of a file, for example, or its composition. If sufficient space is available textual annotation can be added, though the long thin rectangles naturally arising from the process illustrated in Figure 8.26 militate against this. A solution to this problem is available, however (Figure 8.28) and involves alternate horizontal and vertical divisions ('slice and dice') of a rectangle to represent subordinate nodes. In this way a more manageable aspect ratio is associated with most nodes, so that labeling is more easily achieved. An example taken from an author's collection of reports is shown in Figure 8.29.

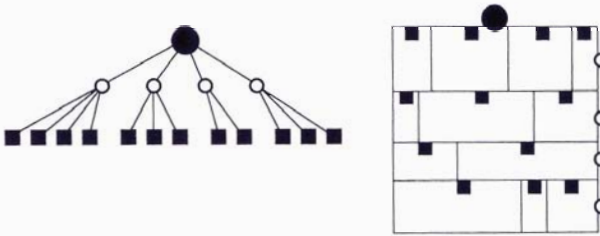


FIGURE 8.28
A modified
Tree Map
construction
with alternating
horizontal and
vertical divisions

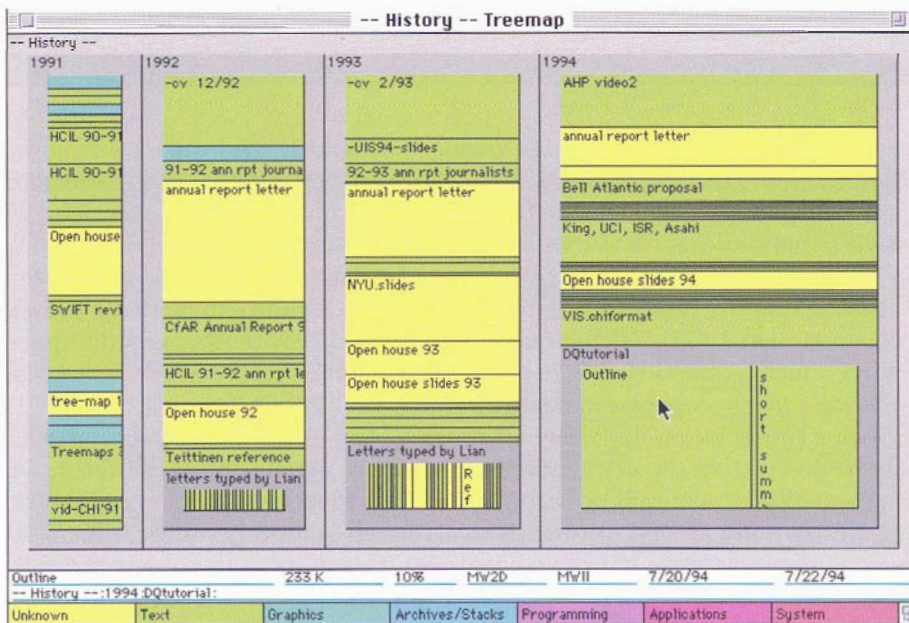


FIGURE 8.29
Tree Map
display of an
author's
collection of
reports
Source: Ben
Shneiderman

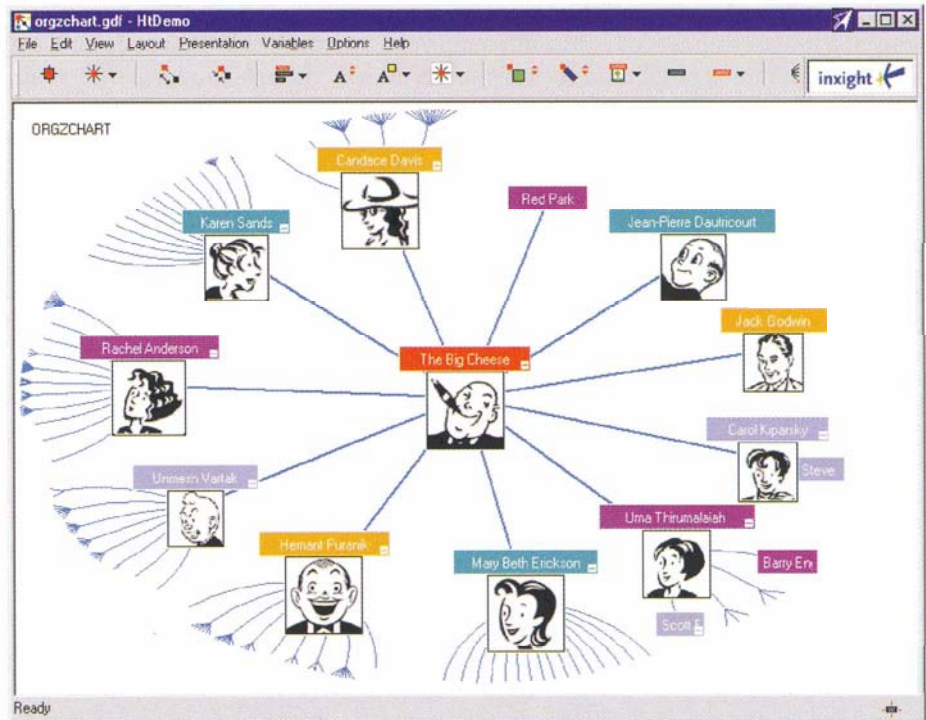
8.3.3 Hyperbolic browser

An ingenious technique whereby an entire tree can be kept within the confines of a circular area on a conventional display screen was derived by Lamping and Rao (1994). Without going into the sophisticated mathematical detail involved (Lamping *et al.*, 1995; Lamping and Rao, 1996), the method is based on a hyperbolic geometric transformation with the result that the tree has an appearance of the form shown in Figure 8.30. The designated root node is initially in the

FIGURE 8.30

Appearance of the hyperbolic browser

Source: *Inxight Software, Inc.*



center, with its subordinate nodes arranged around it, and their own subordinate nodes arranged similarly, and so on. As one proceeds outwards from the **center** of the display the distance between a node and its subordinates decreases in such a manner that, as a result of the hyperbolic transformation, the entire tree 'fits' within the circular area, although drawing of the tree stops below one pixel resolution. In view of its mathematical basis the term 'hyperbolic browser' has been assigned to the technique. It can be said that the hyperbolic browser employs a form of distortion (Chapter 7) to keep the display within bounds and to provide a focus + context view of the data.

Such a presentation has many advantages. Any node which is of interest can easily be moved into the central area so that more detail can be examined. Such movement can be achieved by dragging the node to its new position or by identifying which node should automatically, though smoothly, be brought to the **center**. With a new node positioned centrally (Figure 8.31) the rest of the tree is appropriately repositioned (as if the node of interest were now considered the root node). An attendant advantage of the hyperbolic display is that a node's context can be viewed in all directions within the tree: its parent, siblings and children are all in close proximity. The mathematical parameters defining the display can be chosen to provide, for the more centrally located nodes, essentially non-overlapping display regions (Figure 8.32) within which details of that node can be displayed or, alternatively, a 'long text' format so that up to 25 characters can be used for each node (Figure 8.33). Indeed, as a node is repositioned, the displayed detail can be adjusted automatically in a 'zoom and bloom' effect, otherwise known as semantic zoom.

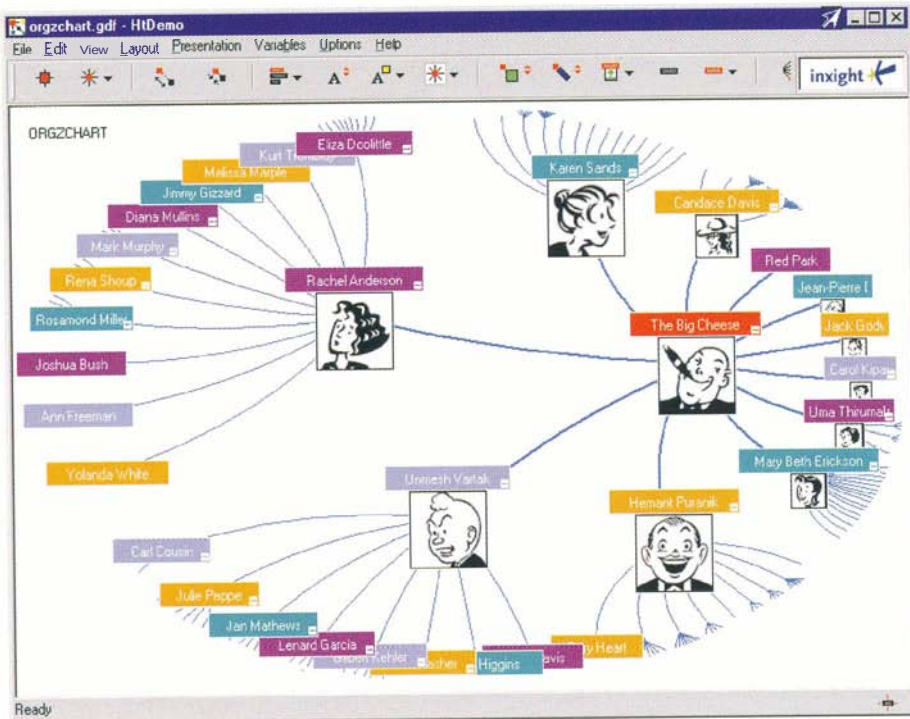


FIGURE 8.3 1

A new node
has been
dragged
towards the
center

Source: *Inxight Software, Inc*

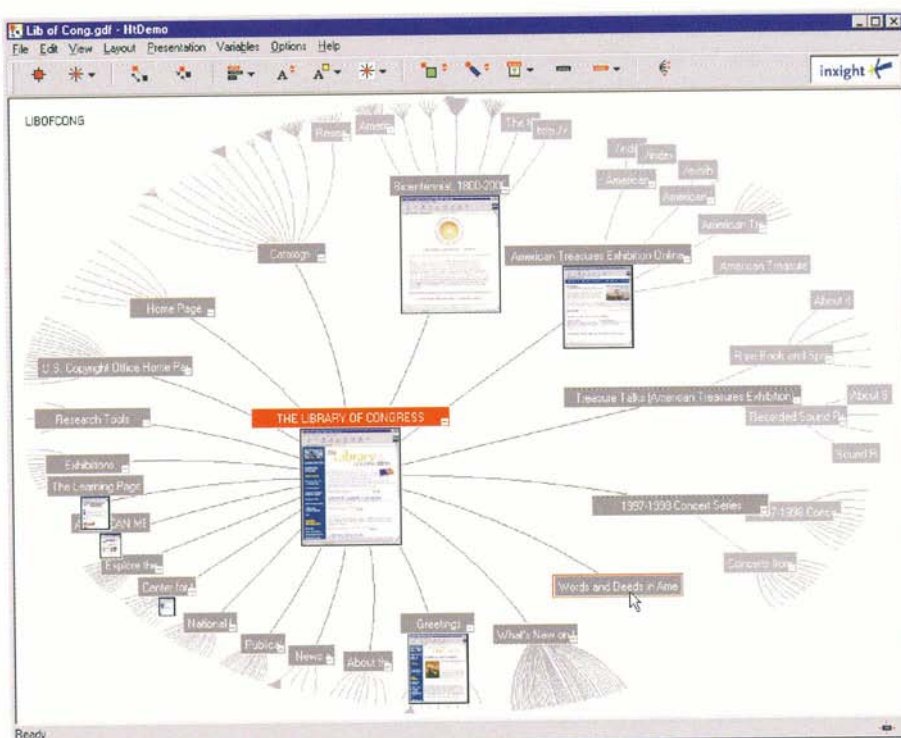


FIGURE 8.32

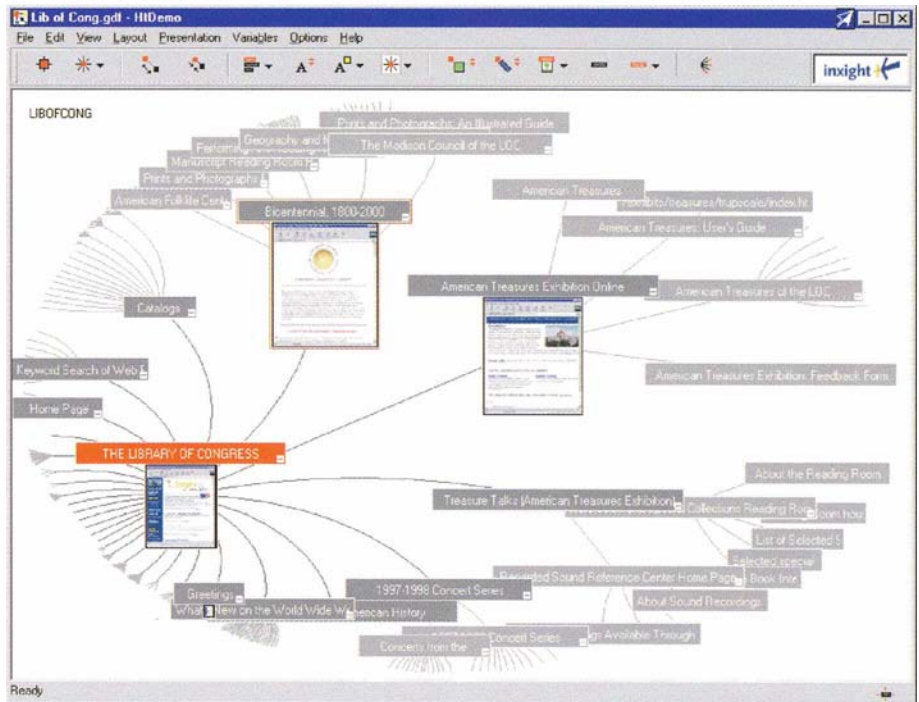
Effective use of display space in the hyperbolic browser

Source: *Inxight Software, Inc*

FIGURE 8.33

The 'long text' format of node labeling

Source: *Inxight Software, Inc.*



Various extensions and enhancements to the hyperbolic browser are possible. One is a modified transformation that will **allow a 'dual focus' presentation**, as illustrated by the rubber sheet model in Chapter 7. Another is the use of color-coded nodes to provide 'landmarks' that enable a user to move around the tree (in other words, successively move nodes to the **center**) with some confidence. Its inventors point out that the hyperbolic transformation is equally applicable to a general graph as well as to a tree, although the 'crossing links' problem would still remain. Many other techniques described in earlier chapters can be introduced into the hyperbolic browser, and will typically depend upon the application. Potential applications are many and varied. For example, the Web visualization tool discussed earlier might place the currently visited page at the root of a hyperbolic browser, with representations of available destinations one link away and their own destinations two links away, and so on.

While the attractive features of the hyperbolic browser can be appreciated to some extent from a written and illustrated description such as the one provided above, a full appreciation of its effectiveness is undoubtedly gained from its interactive use.