Chapter 5
Networks and Hierarchies

All datasets considered in the previous chapters are organized in the form of simple tables that, following the process of visualization described in Section 2.1, have been converted into a visual representation. This visual representation, if well designed, with a proper elicitation of users' needs, can be adopted by a particular type of user to carry out specific tasks.

However, not all datasets we deal with in the real world have a linear structure: Just think of the city transport network, or the organization of the staff in a company. These cases involve data that, by their nature, have a very particular and important characteristic, that of *relation* (or *connection*) and/or of *enclosure* (or *containment*). In an urban transport network, the main elements, the bus stops or subway stations, are **connected** by means of the bus or subway lines. Within a company, each employee **belongs** to a specific unit or sector, under the direction of another company employee. Data structured through relationships (such as the urban transport network) are said to be organized in a *network* system, as the relationships between the elements can be thought as a network made up of connected elements. Data in which each element of the system (except for the top element) is a subordinate to a single other element are said to be organized *hierarchically*, by the hierarchy that is derived from placing the element that contains all the other elements at the top and descending step by step, with the contained elements immediately below the containing one.

The structural properties of these data types are crucial to understanding how the dataset is organized and are often the foremost aspects that must be made clear through visual representations. The properties of relation and enclosure can be represented very simply and intuitively by graphs and trees, illustrated in the following sections.

R. Mazza, *Introduction to Information Visualization*,
DOI: 10.1007/978-1-84800-219-7_5, © Springer-Verlag London Limited 2009

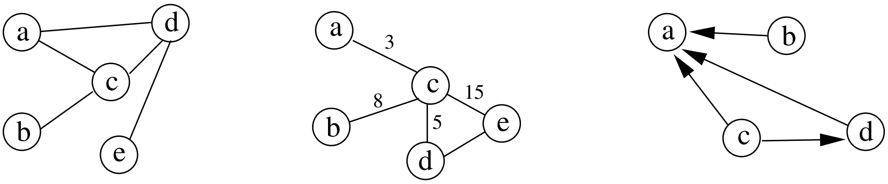


Fig. 5.1 Examples of graphs. On the left is a normal graph, in the center is a graph in which each edge is given a numerical value, and to the right is a directed graph.

5.1 Network Data

The data organized in a network structure can be naturally represented by *graphs*. Graphs are those visual representations in which the points, called *nodes* or *vertices*, represent instances of the data. Nodes are correlated by connections, called *edges*, which represent relationships between the instances. The edges of the graphs can also have direction (in this case, we speak of *directed graphs*) and values, which are called *weights* (in the case of numerical values) or *labels* (in the case of textual descriptions). Examples of graphs are represented in Fig. 5.1.

For a long time, graphs have been studied in mathematics and information technology (think, for example, of the *graph theory* or of the *finite-state automata*) and are naturally suited to representing entities where there is a network organization to represent. The following are issues to consider when representing structured data through a graph [8]:

1. *Positioning of nodes.* Information visualization graphs often represent abstract data types that don't have a natural spatial location. It is necessary to decide on which criteria to arrange the nodes in the space (that is, the *layout*). Some techniques, such as *multidimensional scaling* (MDS) [18], can convert a collection of multivariate data (of any dimension), mapping them into one, two, or three dimensions, from which the Cartesian coordinates are derived to position the nodes in the space. Some layout techniques will be illustrated later. Moreover, we can decide to map some attributes to the shape, color, and dimension of the nodes.
2. *Representation of the edges.* A relationship between two nodes has to be represented by an edge. These can have associated weights and can be direct or indirect. The weights can be indicated next to the edges or mapped to the color or to the width of the edge.
3. *Dimensioning.* Some datasets can have thousands or even millions of records, which can't be directly represented by graphs in a one-to-one relationship with nodes and edges. It is necessary to find solutions in such cases.
4. *Interaction with graphs.* Modern visual representation software allows the user to interact with the generated view, which is very helpful when we deal with complex graphs with a high number of nodes and edges. The user can manipulate the representation of a graph to move the nodes, zoom in on a part of the graph, and hide or show edges or even a part of the graph.

We will show some examples illustrating the use of graphs to represent data organized in a network structure in various contexts, and we'll see how these factors have been handled within the various applications.

5.1.1 Concept Maps and Mind Maps

Concept maps , proposed by Joseph Novak at the beginning of the 1970s, are diagrammatic representations showing the relationships among concepts of a complex and structured domain. They are built from a series of concepts (*semantic nodes*), which then proceed to their connections by means of labeled edges (*propositions*) that describe the type of connection. It is also possible to begin from a general concept, associating it, little by little, to more specific concepts, to have a sufficiently detailed description of the domain. It is possible to create maps for any type of subject: a website, a book, a service, a product, etc. The concept maps are best used in teaching, as they manage to outline the structured knowledge in a very synthetic and reasoned manner. Fig. 5.2 depicts a concept map describing the "website" concept. [1]

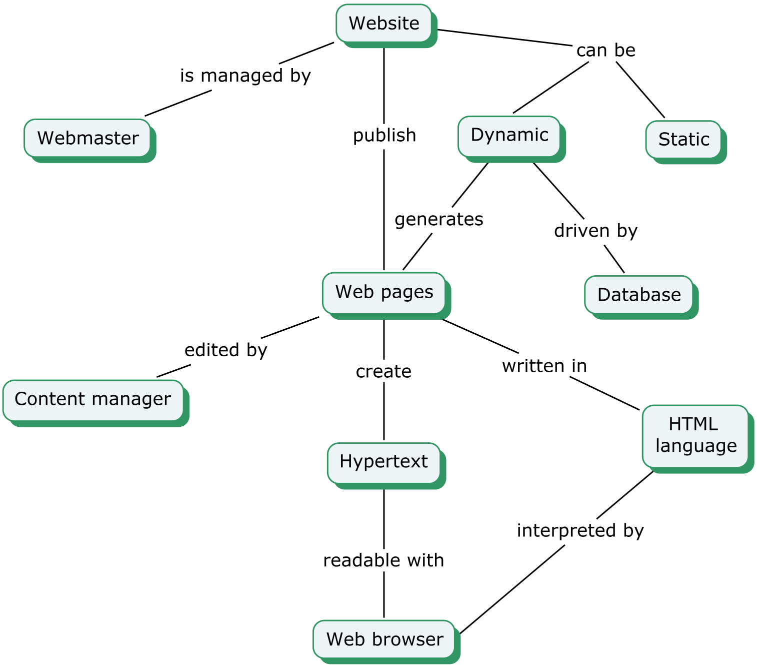


Fig. 5.2 Example of a concept map that describes concepts regarding a website.

[1] The map was created using the IHMC Cmap software tool; http://cmap.ihmc.us.

Mind maps are similar to concept maps. These are used to describe ideas, situations, projects, organizations, etc. by graphical associations. These find many applications in brainstorming sessions, in educational environments, or when organizing ideas. For example, when attending a lesson or presentation, an effective mnemonic and organizational technique consists of tracing the most important concepts in a graphical-textual manner and associating them to each other, graphically, through logic-associative relationships. Mind maps are usually organized by starting from a central concept, from which more correlated nodes are emitted, which will then be further specialized and divided. What distinguishes the mind map from the concept map is, therefore, the fact of having a single topic as a starting point (as opposed to concept maps, which can have many), besides having a radial structure, where the nodes develop from a base subject, according to a number of levels, whereas concept maps are based on connections between concepts. An example of a mind map, which represents the early ideas for the preparation of a university course, is illustrated in Fig. 5.3.[2] Recently, Tony Buzan, the English researcher who coined the term "mind map," tried to theorize this approach, designing a number of rules and best practice to use the mind map, such as the use of colors, images, symbols, and various character sizes. For a complete treatment, consult [7].

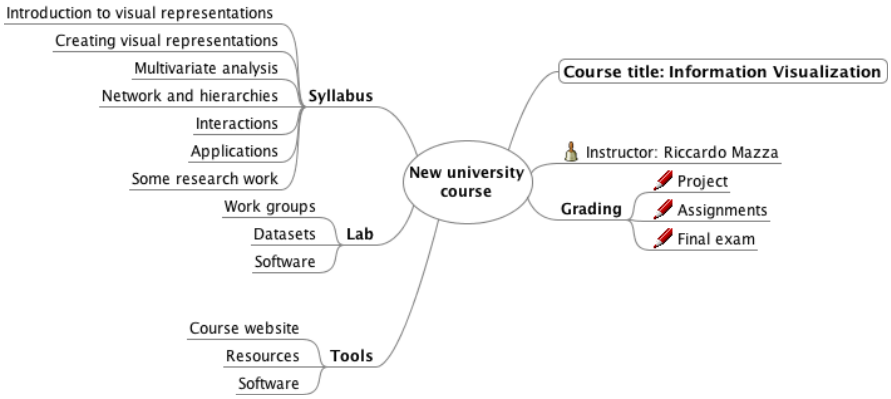


Fig. 5.3 Example of mind map for the planning of a university course.

5.1.2 Complex Network Data

Graphs are a very efficient form of representation for network data, but unfortunately they have the disadvantage of not being very scalable: By increasing the number of nodes, the graph becomes too complex and not very readable. The graph in Fig. 5.4 describes the network of social relationships among a number of individuals. Each

[2] The map was realized using FreeMind software; http://freemind.sourceforge.net.

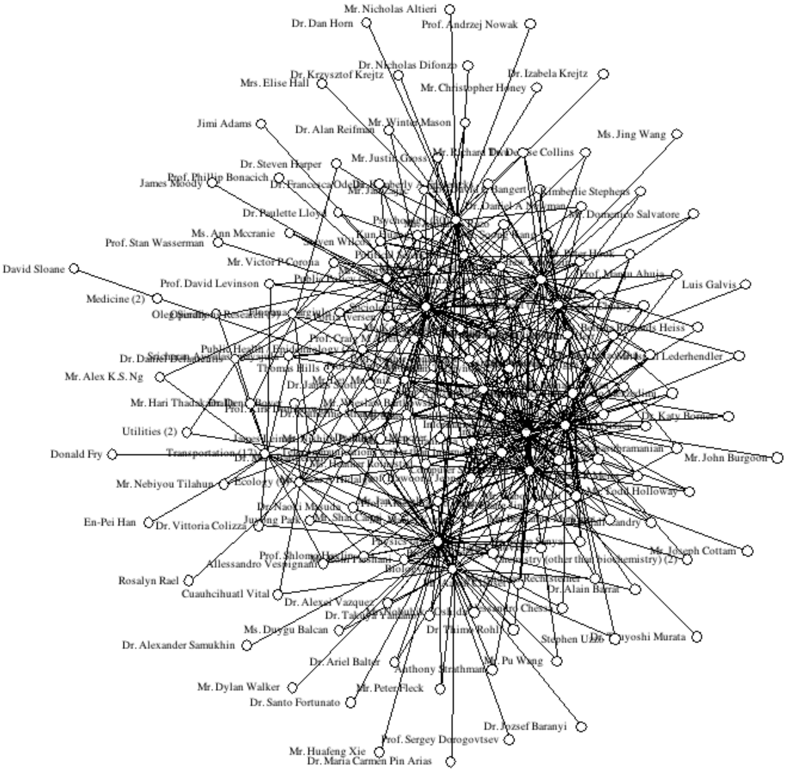


Fig. 5.4 An example of a complex graph.

node represents an individual, while the edges represent social relationships among these individuals, such as family, friends, colleagues, or acquaintances. Visual representations of this type, called *social networks*, are important to understand social relationships among various individuals, and to determine qualitative aspects such as leadership or informal structure. The problem of the representation in this figure is the density of nodes and edges concentrated in a small space, which makes it impossible to distinguish among the graphical elements placed in the center of the figure.

There is a great deal of ongoing research into the problem of complex graphs. Periodically, new methodologies are proposed to increase the number of nodes representable in a graph, or to improving their legibility. These problems can be tackled by adopting one or more of the following strategies:

- using new geometric arrangements (layout) for the graph design in order to improve the readability,

- approximating the structure with a reduced, but more readable, graph; for instance, by reducing the number of edges displayed on the graph or by hiding relationships of lesser interest after they are displayed,
- adding interactivity to the software that generates the visual representations to generate dynamic graphs, which can be manipulated and explored according to the user's needs.

Interactive techniques will be the subject of the following chapter. Here we will illustrate some examples of how geometric arrangements and approximations, which are required to produce more readable graphs, have been applied.

5.1.2.1 Optimizing Layout

The basic problem in representing a graph with a very large number of nodes is that these often have such a large number of crossing edges that it becomes impossible for the user to perceive the graph's general structure. The ideal would be to arrange the nodes in the space so as to minimize the number of crossing edges. The most common layout techniques that attempt to optimize the positioning of the nodes are called *spring-embedder* [17] and *force-directed* [22]. These techniques use algorithms that position the nodes of a graph in two- or three-dimensional space so that all the edges are of more or less equal length and there are as few crossing edges as possible. The resulting graph also has good aesthetic properties (uniform edge length, uniform edge distribution, and some symmetry). Figure 5.5 represents a graph, created by Jeffrey Heer's tool, *prefuse*,[3] that shows a social network making use of a *force-directed* algorithm.

5.1.2.2 Reducing Graph Complexity

When the number of nodes is very large (such as a data structure with tens of thousands of nodes), geometric techniques also show their limits, both for the reduced space of the screen and for the complexity of the layout algorithms, that would require longer computation time. The only possibility we have of visualizing a graph that could be of some use to the user is to create a "reduced" version of the graph, meaning approximating it with a scaled-down version that sacrifices the information somewhat, but simultaneously ends up being more readable and keeps the global structure of the data, allowing identification of the main patterns. To allow for a good final visual representation of the graph, it is necessary to attempt to reduce the number of objects represented and, at the same time, preserve the global structure of the graph (that is usually the main point of interest for analysis). A very simple technique, called *link reduction*, consists of visualizing only the edges having weights above a certain value, or that satisfy specific criteria. In this way, only the edges that could be of interest to the user are represented.

[3] http://www.prefuse.org/.

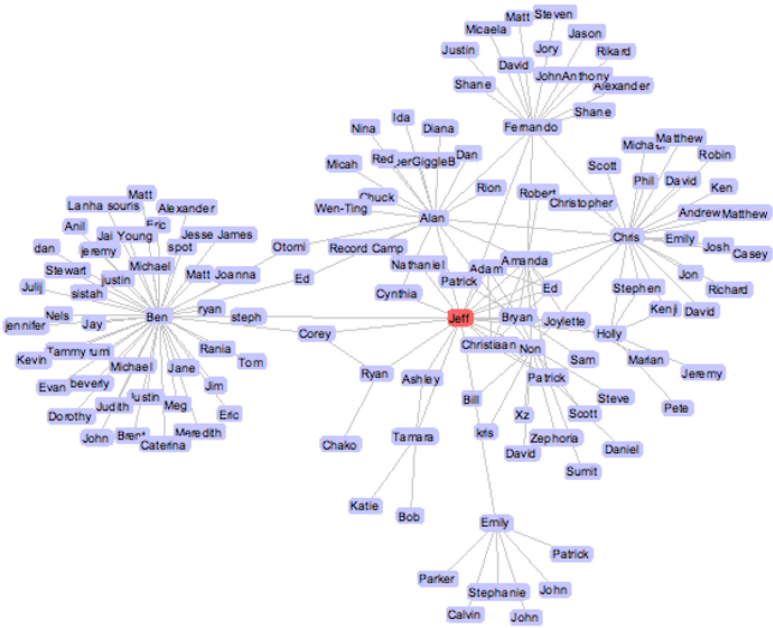


Fig. 5.5 An example of a graph that uses a force-directed algorithm to represent a social network. Image created with the prefuse tool and reproduced with the permission of Jeffrey Heer, University of California, Berkeley.

Other, more sophisticated techniques, such as the *minimum spanning trees (MST)* and *pathfinder network scaling (PFNET)* techniques, analyze the topological structure of the nodes and edges to eliminate the redundant edges and maintain the most significant links. Represented in Fig. 5.6 are a complete graph (left) and its reduced version (right), using the *pathfinder* technique.

To reduce the complexity of a graph, one can also intervene by attempting to diminish the number of nodes visualized. There are *clustering* techniques that can be applied to data to be represented by graphs. These techniques tend to visualize a group of "similar" nodes by combining them in one node (the *cluster*) to reduce the number of nodes and edges to be visualized. The degree of "similarity" between two nodes depends on the application type and the domain of the data visualized through the graph.

Some solutions have been studied specifically for generating and managing graphs with an extremely high number of nodes. These solutions use a combination of layout techniques, approximation, and interaction. One of these, NicheWorks [67], is able to treat up to a million nodes in a few minutes. It deals with a prototype developed at Bell Lab (now *Alcatel-Lucent*) in the mid-1990s, with the aim of exploring a visual approach to the study of telephone frauds. The algorithm was then generalized for generic problems, where there is a need to visualize graphs with a

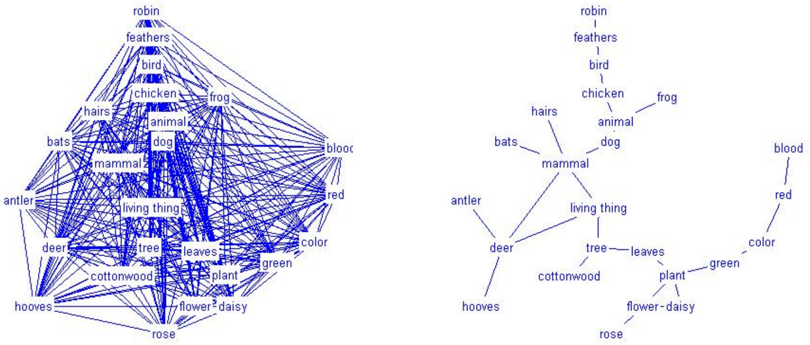


Fig. 5.6 Complete graph (left) and the reduced link version (right), achieved using the pathfinder technique. Graphs were generated with the KNOT analysis tool. Images courtesy of Interlink.

large number of nodes, and applied to other domains. NicheWorks uses radial positioning to optimize the positioning of the nodes in the space, in a way that avoids the crossing edges and sets a high number of nodes in the visible space. Furthermore, the weights associated with the edges are taken into consideration in the construction of the graph, so that the nodes are positioned at a distance inversely proportional to the weights of the edges that connect them (meaning the higher the values of the weights that connect two nodes, the closer the nodes are).

Figure 5.7 shows three types of layouts proposed by NicheWorks:

- *Circular layout.* The nodes are arranged circularly in the immediate periphery of a single circle.
- *Hexagonal grid.* The nodes are arranged at the grid points of a regular hexagonal grid.

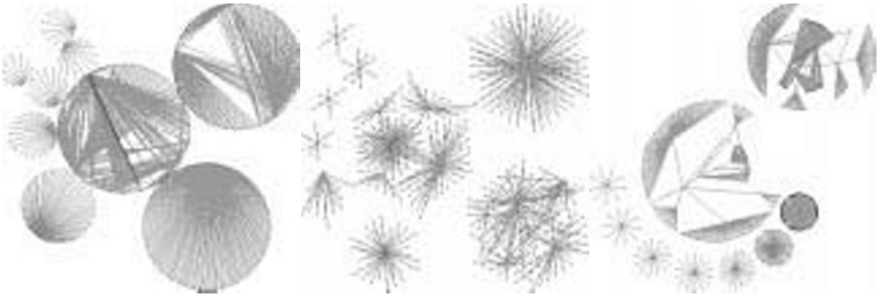


Fig. 5.7 Main NicheWorks layouts: circular (left), hexagonal (center), and tree (right). Image by [67]; Reprinted with permission from *The Journal of Computational and Graphical Statistics.* ©1999 by the American Statistical Association. All rights reserved.

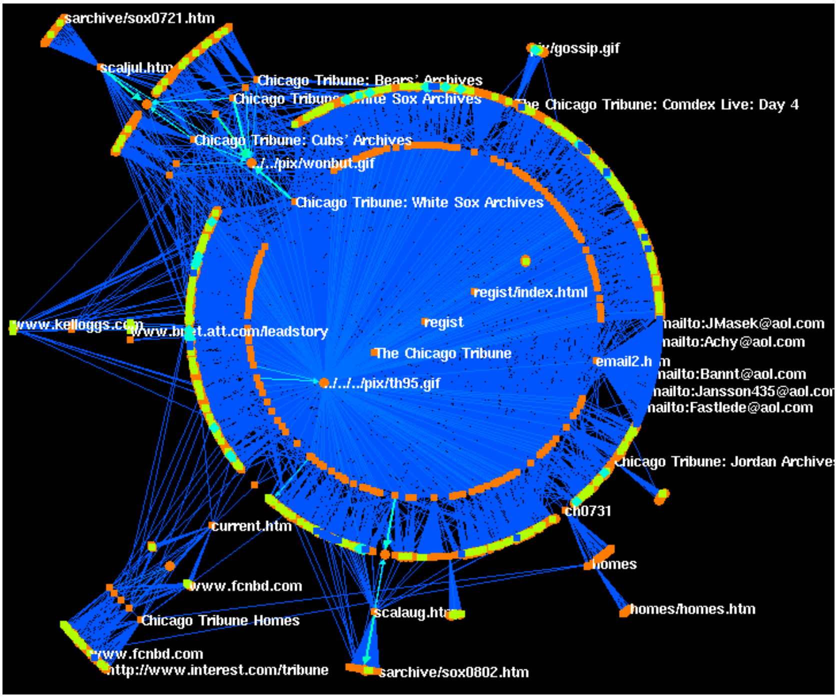


Fig. 5.8 NicheWorks shows the graph of the *Chicago Tribune's* website structure. Image reproduced with the permission of Graham Wills, SPSS Inc.

- *Tree layout.* The nodes are arranged with the root node at the center, around which each node is arranged on a circle. This type of layout is appropriate for hierarchical structures but can also be applied to network data.

NicheWorks has been used to study fraudulent international telephone calls [14] and to analyze complex website structures. Fig. 5.8 displays a graph of the structure of the pages and internal links, back in 1997, of the Chicago Tribune's website, one of Chicago's major daily newspapers. The goal of the representation is to understand how the website was structured and to see which design criteria were used in its production. The graph was represented using the tree layout (see Fig. 5.7). The shape and the color of the nodes have the following mapping:

- orange square: local pages,
- orange circle: local images,
- blue square: external pages,
- yellow square: program interface for managing forms or dynamic pages (CGI).

Some of the most important nodes are labelled. The graph shows the entrance pages of the site (positioned at the center of the figure) that link to a local page (internal circle). These pages contain a connection to further dynamic pages (external circle). Therefore, a structure of three levels (main page of the website, internal circle, and external circle) globally dominate. Only a limited number of pages have a different structure (top left of the image), which are mainly the pages dealing with sports subjects.

The advantage of approaches based on visual representations, as opposed to the automatic analysis techniques such as those that use *data mining* algorithms,[4] lies in their flexibility and the human visual system's ability to adapt. According to the authors of NicheWorks [14], people who attempt to defraud telephone companies adopt systems that elude the systematic checks that are carried out on international calls. The use of visualization systems can instead be adapted dynamically to allow recognition of possible changes that are then used by the perpetrators of these abuses.

5.1.3 Geographic Representations

Often graphs are used to describe the topology of networks in which a spatial or geographic component is present. In this case, this component is used to position the nodes in the space. For example, we are used to representing telecommunication networks with graphs, where we match a physical component to the nodes: a server, router, hub, etc., while the edges represent the connections (physical or virtual) among the various elements of the network. Frequently, the nodes have a wide-scale geographic positioning, which must be depicted visually. Figure 5.9 uses a layer technique to represent the geographic coverage that the backbone NSFNET T1 served during September 1991 [12]. The backbone and the connections with the various cities are visually differentiated by the raised layer of the geographic map, so as to create a clear visual separation between the network and the territory. A color scale that goes from purple (zero bytes) to white (100 million bytes) indicates the volume of incoming traffic measured at every point served by the backbone.

Other examples of a network topology on a world scale are represented in Fig. 5.10 by SeeNet [13], a result of the work directed by Stephen Eick at Bell Labs. SeeNet visualized the amount of Internet traffic throughout 50 nations, measured by the NSFNET backbone in 1993.

However, these types of representations are often appreciated more for their "aesthetic" value than for their effective use, so much so that they have been collected and published on a website, *The Atlas of Cyberspaces*, as well as appearing in a successful publication written by Martin Dodge and Rob Kitchin [32].

[4] The *data mining* algorithms seek to extract useful information from large quantities of data automatically or semiautomatically. They look for patterns in the data to hypothesize on causal type relationships between phenomena tied to the data.

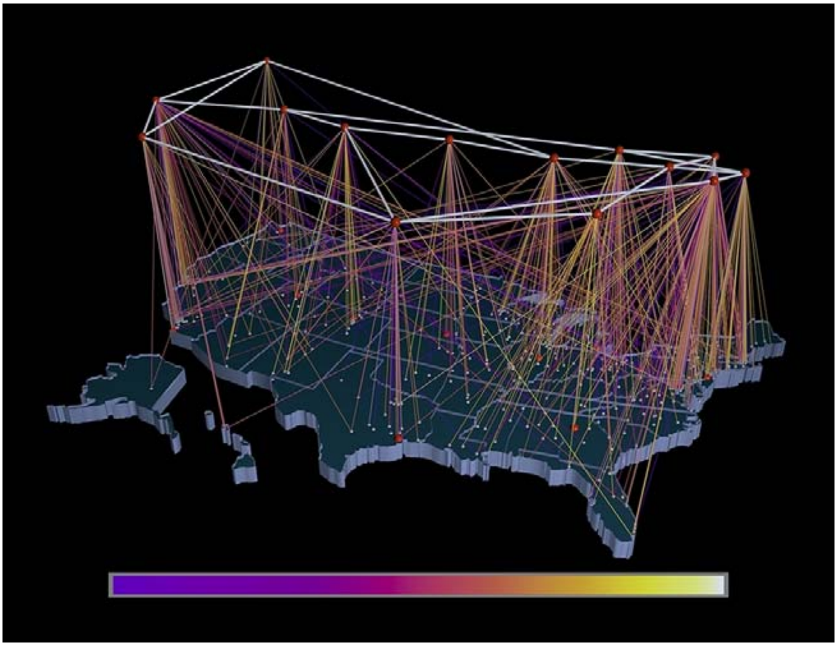


Fig. 5.9 An image that shows the topology of the NSFNET T1 backbone network in September 1991 in the United States. The situation today is certainly more complex than that represented in the figure. Created by D. Cox and R. Patterson/NCSA, UIUC. © 1994 The Board of Trustees of the University of Illinois.

5.1.4 Transport Networks

A very common example of information represented in graphical form is the transport networks, represented on a geographic map, in which the cities connected by the railway lines are highlighted. There is, however, a very particular type of transport representation: the maps of the underground train. Figure 5.11 shows a representation of a (unofficial) map of the Madrid Metro network. This type of map was conceived of in 1931 by Harry Beck, a London public transport employee, who, in his free time, designed the first draft of what is used today as the model of public transport networks all over the world. He proposed a new type of map inspired by the electrical circuit systems. His genial intuition was to understand that a traveler wants to know how to reach a destination when leaving from a specific station, and is not interested in the physical position of the stations. What matters is the topology of the network, or how the various stations are connected and which lines to take to reach a certain destination. In this way, Madrid Metro map represented in Fig. 5.11, is not a "map" in the true sense of the term, since the proportions of the physical distances between the stations are not respected. Rather, it deals with a graph (or a diagram) in which the lines, stations, and zones of the transport network are represented, and the correspondence with the physical position of the stations has been

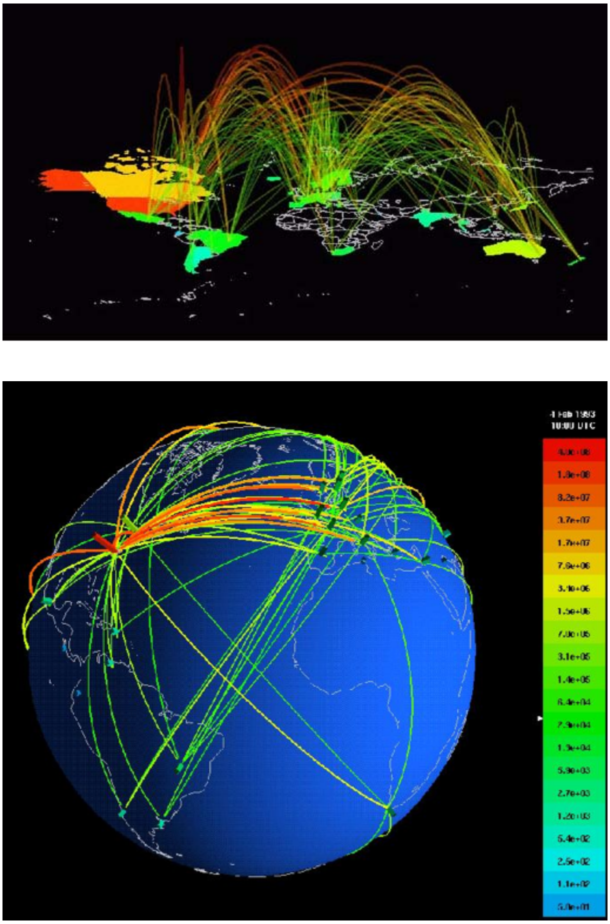


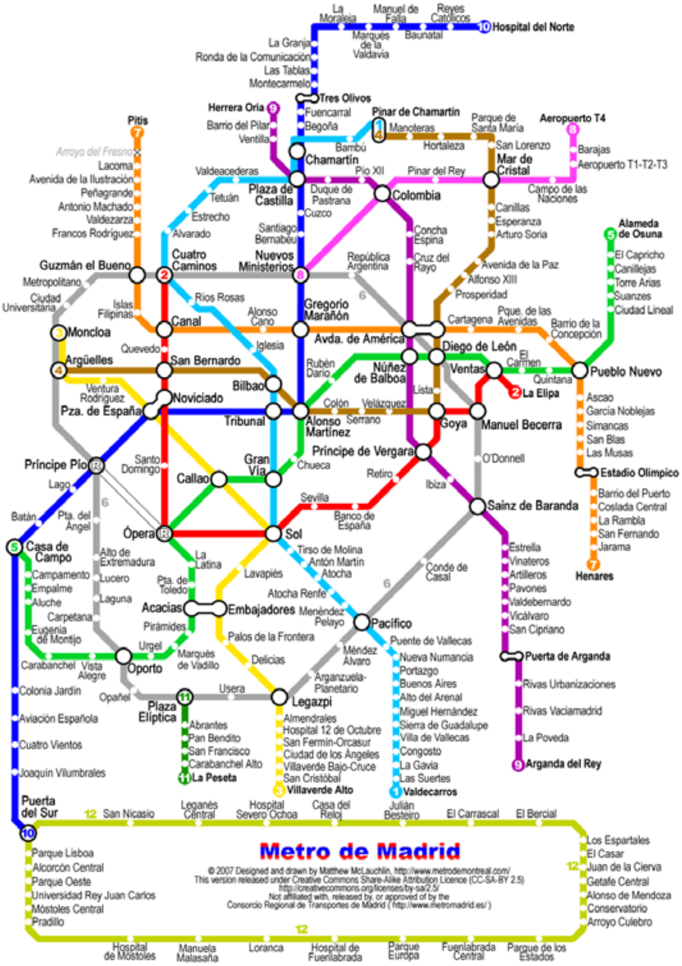
Fig. 5.10 SeeNet, 3D representation of the Internet traffic across the NSFNET backbone network in 1993. Images reproduced with the permission of Stephen Eick, SSS Research, Inc.

distorted, to concentrate the graph in the smallest space possible. It can therefore be called a topological, rather than geographical, map.

5.1.5 3D Graphs

Some modern tools for the construction of graphs are able to generate three-dimensional layouts. In Fig. 5.12, we report two examples generated by Tulip,[5] one of the best toolkits for generating graphs. The 3D layout graph can be turned and

[5] http://www.tulip-software.org.



© 2007 Designed and drawn by Matthew McLauchlin, http://www.metrodemontreal.com/
This version released under Creative Commons Share-Alike Attribution Licence (CC-SA-BY 2.5)
http://creativecommons.org/licenses/by-sa/2.5/
Not affiliated with, released by, or approved of by the
Consorcio Regional de Transportes de Madrid (http://www.metromadrid.es/)

Fig. 5.11 A map of the Madrid Metro system. Images licensed under Creative Commons Share-Alike.

moved, to allow the user to change the view and make visible any objects that could possibly be occluded.

5.2 Hierarchical Data

We speak of hierarchy when considering data characterized by containment properties. Examples of hierarchy are the organization of files and directories in the computer (the files are contained within the directories, which are in turn contained within other directories), the structure of books (organized in parts, chapters, sec-

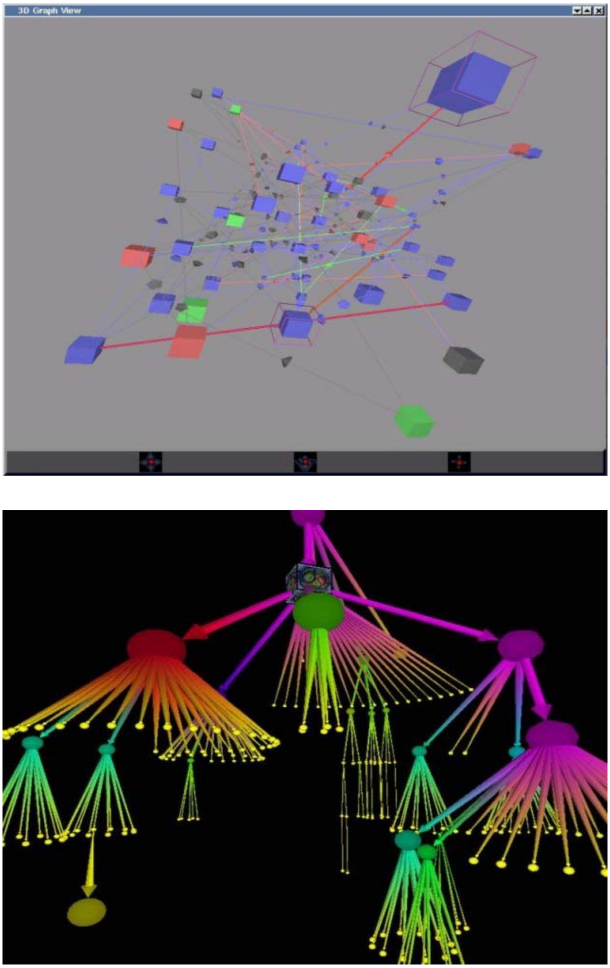


Fig. 5.12 Examples of graphs with three-dimensional layouts generated by Tulip software.

tions, etc.), company organization (president, director, advisors, supervisors, etc.), and the taxonomies used in biology. A hierarchy can be represented through a graph with a starting node called *root*. Each node has zero or more *child nodes*, which are usually represented below the ancestor, and the ancestor is called the *parent node*. A node has one parent, at most. Graphs of this type are defined as *trees*, precisely due to their similarity to actual trees, but in difference to the botanical trees, they are represented upside down, with the root at the top and the leaves at the bottom. An example of a tree is reproduced in figure 5.13, which represents a simplified version of the classification of the wind instruments as proposed in 1914 by Curt Sachs and Erich von Hornbostel.

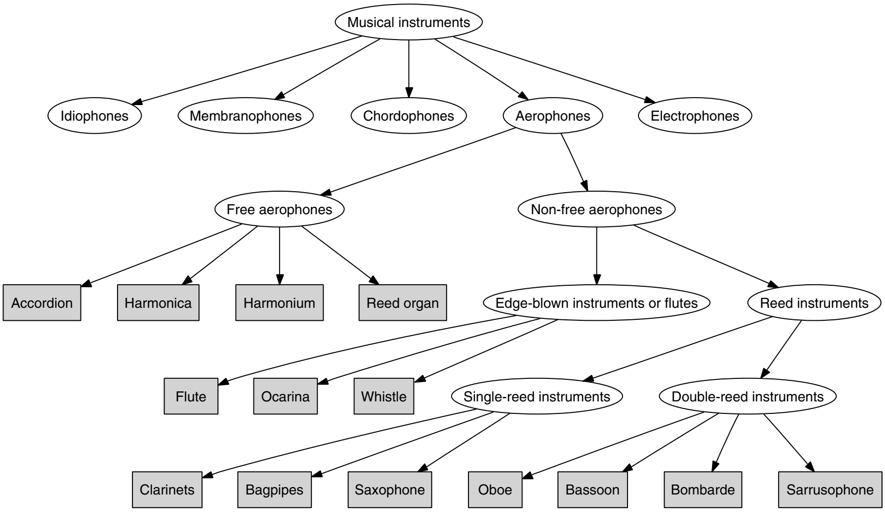
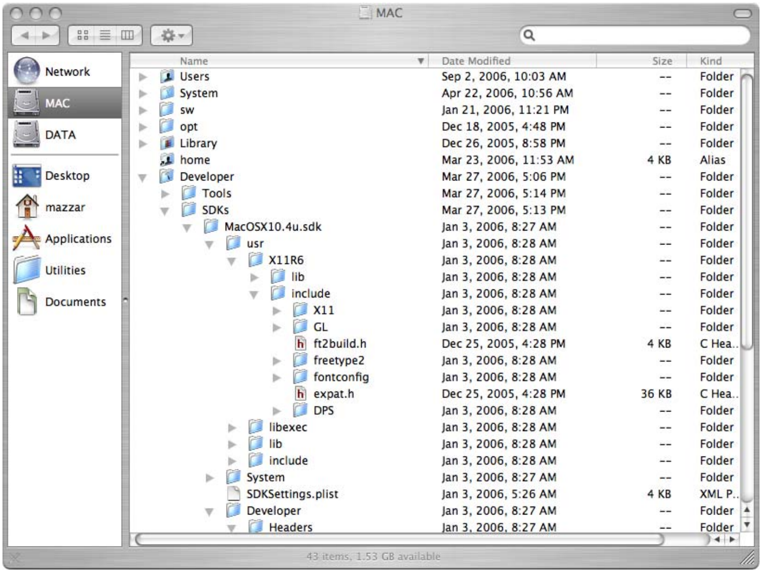


Fig. 5.13 A tree representing the classification of the wind instruments according to Curt Sachs and Erich von Hornbostel.

5.2.1 File System

For anyone who spends a great part of the day working with computers (such as the author of this text), the most familiar hierarchical structure is probably the *file system*, which is the mechanism with which the files are stored and organized on a mass storage device like a hard disk. In the file system, the files are organized hierarchically, starting from a *directory* (or *folder*) called root that contains files or other directories. The file system can be represented both textually (through the shell of the operating system) and graphically, through a file browser. In the visual representation used by all modern operating systems (e.g. Fig. 5.14 of the Apple Mac OS X systems), the folder metaphor, which contains documents (the files) and other folders, is generally used. However, this is a partial representation of the file system, as it may contain tens of thousands of files and directories. Particularly those who have to manage servers with large quantities of data shared among many people, the management of the file system is crucial. The system administrator must have tools that permit efficient monitoring of the file collections on the disk, to be able to identify situations in which system administrator intervention could be required.

As early as the beginning of the 1990s, the first graphical tools were proposed to integrate the command-line interface typical of UNIX-like systems with much more explicative and immediate visualizations. In 1991, Phil Dykstra developed *Xdu*, a system for graphical monitoring of the UNIX file system. Xdu visualizes the results of the UNIX command *du* (which returns some statistics on disk usage)



Name	Date Modified	Size	Kind
Users	Sep 2, 2006, 10:03 AM	--	Folder
System	Apr 22, 2006, 10:56 AM	--	Folder
sw	Jan 21, 2006, 11:21 PM	--	Folder
opt	Dec 18, 2005, 4:48 PM	--	Folder
Library	Dec 26, 2005, 8:58 PM	--	Folder
home	Mar 23, 2006, 11:53 AM	4 KB	Alias
Developer	Mar 27, 2006, 5:06 PM	--	Folder
Tools	Mar 27, 2006, 5:14 PM	--	Folder
SDKs	Mar 27, 2006, 5:13 PM	--	Folder
MacOSX10.4u.sdk	Jan 3, 2006, 8:27 AM	--	Folder
usr	Jan 3, 2006, 8:28 AM	--	Folder
X11R6	Jan 3, 2006, 8:28 AM	--	Folder
lib	Jan 3, 2006, 8:28 AM	--	Folder
include	Jan 3, 2006, 8:28 AM	--	Folder
X11	Jan 3, 2006, 8:28 AM	--	Folder
GL	Jan 3, 2006, 8:28 AM	--	Folder
ft2build.h	Dec 25, 2005, 4:28 PM	4 KB	C Hea..
freetype2	Jan 3, 2006, 8:28 AM	--	Folder
fontconfig	Jan 3, 2006, 8:28 AM	--	Folder
expat.h	Dec 25, 2005, 4:28 PM	36 KB	C Hea..
DPS	Jan 3, 2006, 8:28 AM	--	Folder
libexec	Jan 3, 2006, 8:28 AM	--	Folder
lib	Jan 3, 2006, 8:28 AM	--	Folder
include	Jan 3, 2006, 8:28 AM	--	Folder
System	Jan 3, 2006, 8:27 AM	--	Folder
SDKSettings.plist	Jan 3, 2006, 5:26 AM	4 KB	XML P..
Developer	Jan 3, 2006, 8:27 AM	--	Folder
Headers	Jan 3, 2006, 8:27 AM	--	Folder

43 items, 1.53 GB available

Fig. 5.14 Representation of the file system by the graphical interface Finder in the Mac OS X system.

in graphical form. Fig. 5.15 gives an example created by the *xdiskusage* tool,[6] a modern and improved version of Xdu. A representation with rectangles is used, which are placed into the screen from left to right. The positioning of the rectangles reflects the hierarchy of the file system. Each rectangle corresponds to a directory, and the dimension of the rectangle is mapped to the physical space (bytes) occupied by every directory on the disk.

Other tools have also attempted visualization in 3D dynamics. That was made possible by the availability of increasingly fast 3D graphical boards on personal computers and the availability of powerful graphical languages, like OpenGL. *Fsv*[7] is one of the tools that has been very successful in 3D graphics for navigating a file system. Fsv offers two types of three-dimensional representation: *MapV view*, which returns to the style of xdiskusage but with three-dimensional blocks, and *TreeV view* (see Fig. 5.16), which, instead represents the directory like a "platform," which supports three-dimensional blocks that represent the files contained in it. The captivating fsv visual representation should not deceive us: The system administrators and professionals who manage file systems with great volumes of data have always expressed perplexities about the effective usefulness of these types of visual representations.

[6] http://xdiskusage.sourceforge.net/.

[7] http://fsv.sourceforge.net/.

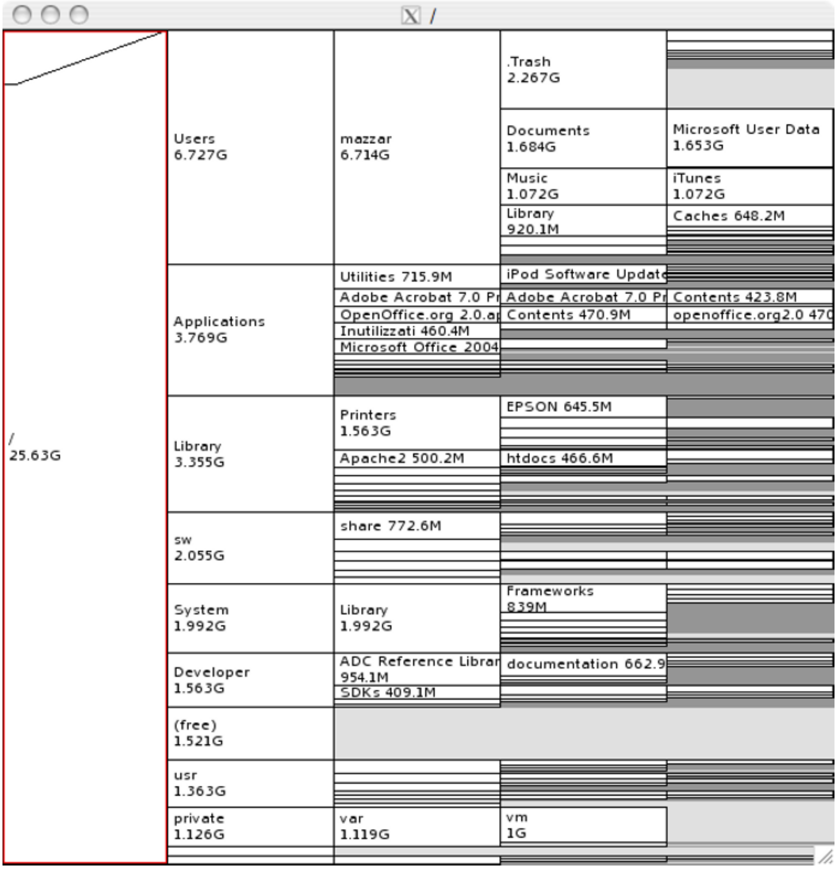


Fig. 5.15 Xdiskusage uses a representation with rectangles of the file system.

5.2.2 *Representing Evolutionary Data with Trees*

Philology is a science that studies texts, that may have been modified over time, to restore them to their original form. *Textual criticism* of a text consists of attempting to determine its original form through the study of its variations in the course of the process of traditional handwriting and/or print. In philology, trees are often used to represent the variations of a text over time. In fact, before the invention of mechanical printing, the texts were copied by hand (in ancient Egypt, for example, the scribes were specialized in the transcription of holy texts) and often, either by error due to an oversight or for explicit censors, modifications were introduced to the various transcriptions. The copied (and modified) texts could have been used as models to create other copies that may in turn have been subjected to other modifications, and so on. Philology, which attempts to rebuild the original form of a text, analyzes its various versions and pays close attention to the variations. These are based on

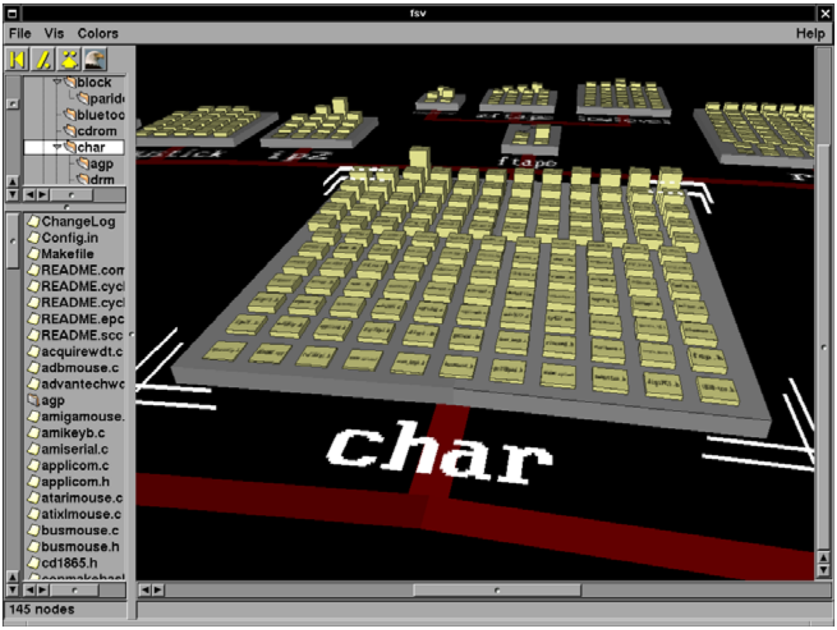


Fig. 5.16 Fsv in the TreeV version visualizes the organization of the file system by a metaphor of an interconnected platform (the directory) on which the blocks (the files) are laid. Image reproduced with the permission of Daniel Richard.

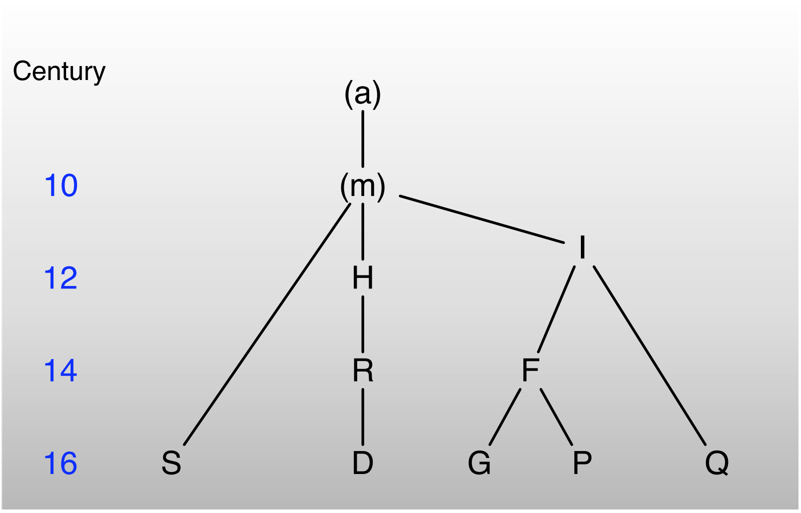


Fig. 5.17 Visual representation of a *stemma codicum* of a text.

the assumption that each copier tended to report the errors of the text that was being used as the model and then introduced some of his own. Another basic principle is that if two different versions of the text share a certain number of unusual errors, then it is very likely that they have a common model. Through this analysis, a tree is created, called *stemma codicum* in philology, that reports the “tradition of the text.” An example of stemma codicum treated by a concrete linguistic study, in which an essay on Hippocrates [47] was analyzed, is represented in Fig. 5.17, where the various “witnesses” (manuscripts) that it reports are indicated, taken from the first, original version of the text (a) called *archetype*.

5.2.3 Cone Tree

Cone tree [50] is a three-dimensional visualization technique for hierarchical data developed by Robertson, Mackinlay, and Card at the Xerox PARC laboratories in long ago 1991 (see fig. 5.18). It was developed to represent hierarchies with a large number of nodes. For this reason, a 3D representation, which extends the physical limits of the two-dimensional display, was chosen. The tree is built from the root node, with all the children nodes arranged at equal distances from the parent, to form a semitransparent cone. The process is then repeated for every node of the hierarchy, with the diameter of the base of each cone being reduced at every level, to ensure the arrangement of all nodes in the space. A particular innovation introduced by this application (which is reported here more for historical reasons than for its practicality) is the possibility of being able to rotate the cone to bring the nodes that may be occluded to the foreground, and the possibility of hiding a cone and all that is below. The particular arrangement of the nodes in the cone is worth mentioning, as it allows for the perception of the density of the nodes in each cone, thanks to the use of transparency and the shadow projected on the low level.

5.2.4 Botanical Tree

In 2001, researchers at the Eindhoven University of Technology in Netherlands made an interesting observation. The structure of trees is commonly used to represent data organized hierarchically, but this metaphor does present some limitations when we try to represent organizations of a certain complexity. However, when we observe a real botanical tree, we notice that, even if it has a large number of leaves and branches, we can always make out its various leaves, branches, and general structure. Then why not represent the hierarchical organization similarly to how trees were formed in nature?

Fig 5.19 gives an example of a botanical tree that represents the structure of a file system [33]. It uses a 3D representation to create trees that are reminiscent of botanical trees (in fact, a difference from the trees seen up to now is that the root

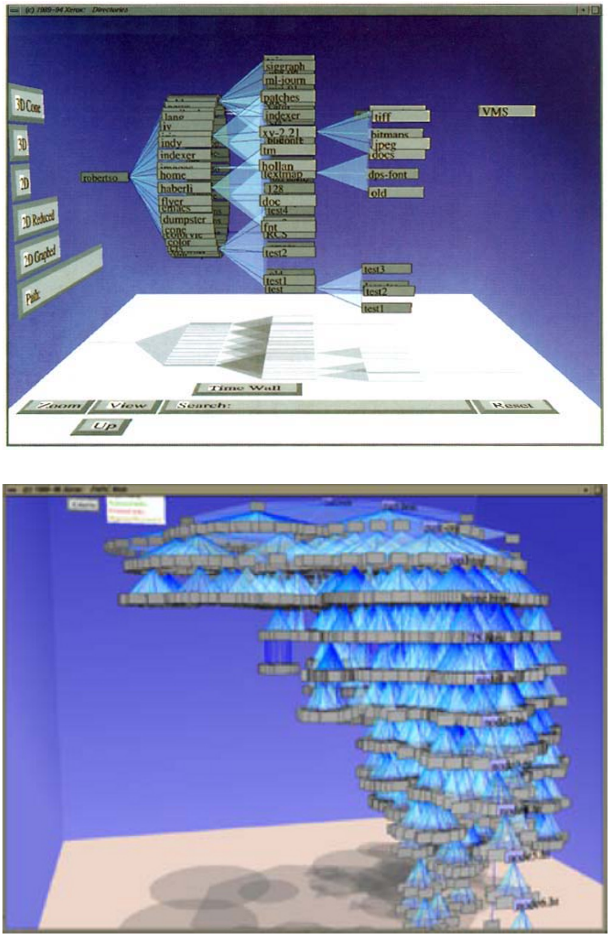


Fig. 5.18 Representation of hierarchical structure by a cone tree. Images reproduced with the permission of Stuart Card, George Robertson and Xerox PARC.

node is placed at the bottom, as opposed to at the top), and for this reason the fruits are visualized, as opposed to the simple leaves. In the example in Fig. 5.19, each fruit represents a collection of files, to avoid cluttering effects that can be generated by a high number of leaves. Each fruit has some colored “spots,” which correspond to a file; the area and the color of these spots are mapped to the dimension and file type, respectively.

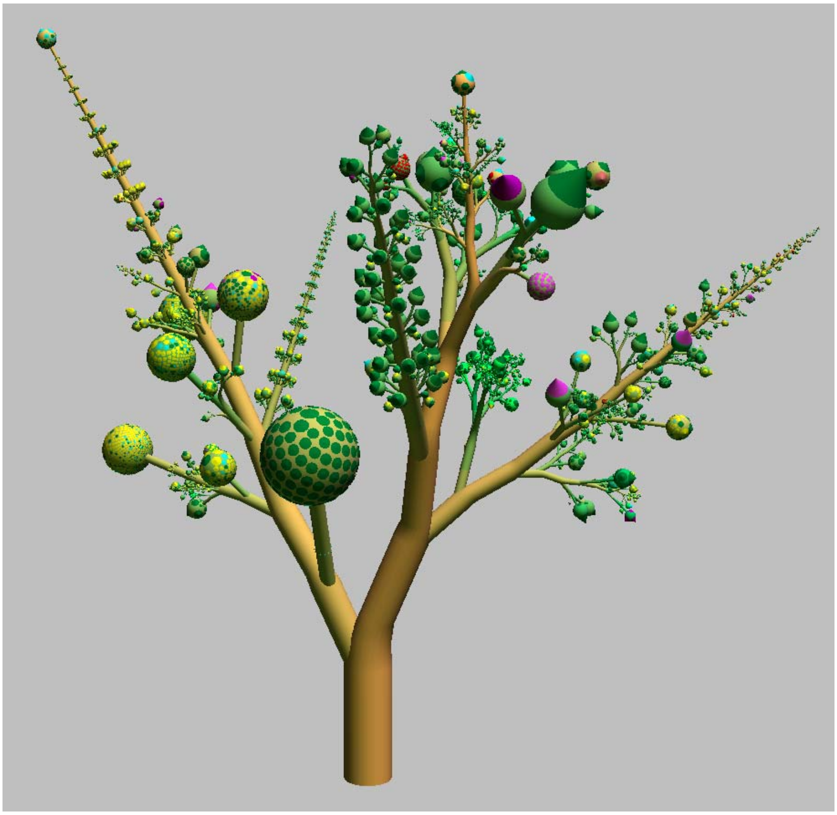


Fig. 5.19 Botanical tree. Image courtesy of Jack van Wijk, Eindhoven University of Technology.

5.2.5 Treemap

In 1990, Ben Shneiderman, one of the most active researchers in the human–computer interaction and information visualization, found himself facing the problem of a full hard disk on one of his servers at the University of Maryland. Shneiderman had to find a way to determine which were the biggest files that could be canceled and who of the 14 users of the server made the most use of the space on the disk. He was unsatisfied with the analysis tools available at the time (everything was more or less based on tree-type representations), he studied an alternative to be able to visualize the hierarchy of the files on the disk. It was then that he had the brilliant idea of using a space-filling technique called *treemap* [51]. Even today, the treemap is one of the most widely used visualization techniques for hierarchical data, used in dozens of applications.

The technique is called *space filling* because it uses all the available space, displaying hierarchical data using nested rectangles. The screen space is divided into rectangles starting from the root node, and then each rectangle is further divided for

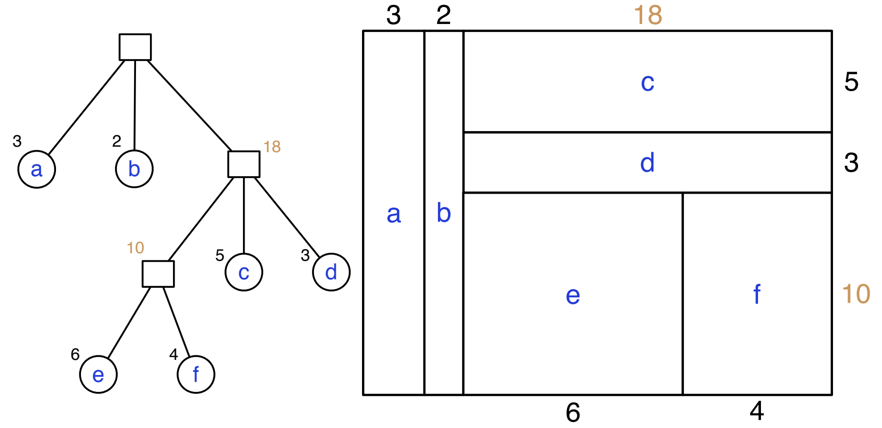


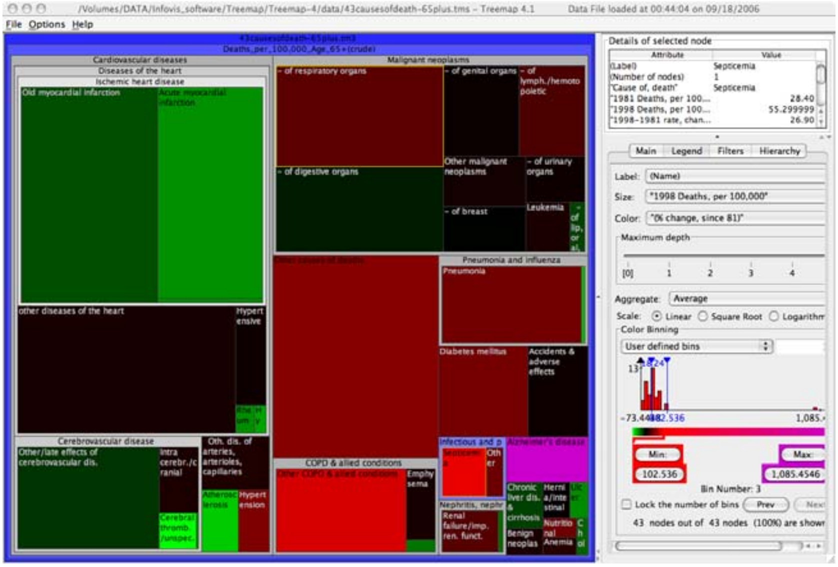
Fig. 5.20 On the left is an example of a tree representation, where a numerical value is associated with each leaf. The internal nodes report the sum of the values of the nodes below. On the right is the treemap representation of the tree.

each level of the hierarchy, until all components of the hierarchy are placed. We'll show how this technique works, in its original 1992 conception version, by applying it to an example.

A tree structure is represented on the left of figure 5.20. In this structure, the values are associated with the leaf nodes (we can think of it as a structure of directories, where a file corresponds to each leaf having a value that corresponds to the dimension of the file). The algorithm of construction of the treemap proceeds recursively, starting from the root node and considering the nodes derived from it. The example that we are observing has a root node from which three nodes derive: node *a* with a value of 3, node *b* with a value of 2, and an internal node with a total value (that is the sum of all the values below) of 18. The algorithm therefore partitions the available space into three rectangles (see Fig. 5.20, right): a rectangle for *a*, a rectangle for *b*, and another rectangle for the internal node assigned to the containment of the nodes below. The blocks are created by partitioning the area vertically in such a way that the proportions of the values are respected. Then the successive nodes continue, although the space is partitioned horizontally this time. The blocks are created in this way for nodes *c*, *d* and the additional internal node. The algorithm carries on analyzing all the levels of the tree, always partitioning the remaining space in an area proportional to the value, alternating horizontal and vertical positioning for every level of the tree.

Fig. 5.21 represents an example generated by version 4.1 of treemap software, developed at the University of Maryland.[8] The figure represents a visualization that shows the cases of mortality in 43 types of pathologies, organized according to the hierarchy defined by the *International Community Health Services* (ICHS), with data taken in the United States in 1998. The size of the rectangles indicates the per-

[8] The software can be downloaded from http://www.cs.umd.edu/hcil/treemap



/Volumes/DATA/Infovis_software/Treemap/Treemap-4/data/43causesofdeath-65plus.tms – Treemap 4.1 Data File loaded at 00:44:04 on 09/18/2006

Details of selected node

Attribute	Value
(Label)	Septicemia
(Number of nodes)	1
"Cause of, death"	Septicemia
"1981 Deaths, per 100...	28.40
"1998 Deaths, per 100...	55.299999
"1998–1981 rate, chan...	26.90

Label: (Name)

Size: "1998 Deaths, per 100,000"

Color: "(% change, since 81)"

Aggregate: Average

Scale: ● Linear ○ Square Root ○ Logarithm

Color Binning: User defined bins

Min: 102.536

Max: 1,085.4546

Bin Number: 3

□ Lock the number of bins

43 nodes out of 43 nodes (100%) are shown

Fig. 5.21 Treemap that represents cases of mortality in 43 types of pathologies. Image reproduced with the permission of Ben Shneiderman, University of Maryland, Human–Computer Interaction Lab.

centage of deaths per 100,000 inhabitants, while the color indicates the percentage of variation found from 1981 to 1998. As can be deduced from the figure, cardiovascular diseases are the main cause of death, but the green coloration indicates that the cases are decreasing (in particular, there was a 41% decrease from 1981 to 1998), while it is the contrary for other rarer diseases, such as Alzheimer's, for example, which, with its purple coloring, presents the maximum increase (which was 1,085%).

This technique is very efficient for representing hierarchical data, where the representation of the nodes through the dimension and color of blocks helps the user to immediately single out and compare nodes, individualize patterns, and identify exceptions.

Over the years, algorithms that generate this type of representation are constantly being improved, with a certain number of layouts available [4], and have produced innumerable applications that use this captivating technique to represent data of all types. We'll show two applications that have attained a good level of success: NewsMap, Map of the Market, and Sequoia View.

5.2.5.1 Newsmap

In 2004, Marcos Weskamp developed an innovative modality of presenting the news from Google News. The application is called Newsmap and is visible in real time by

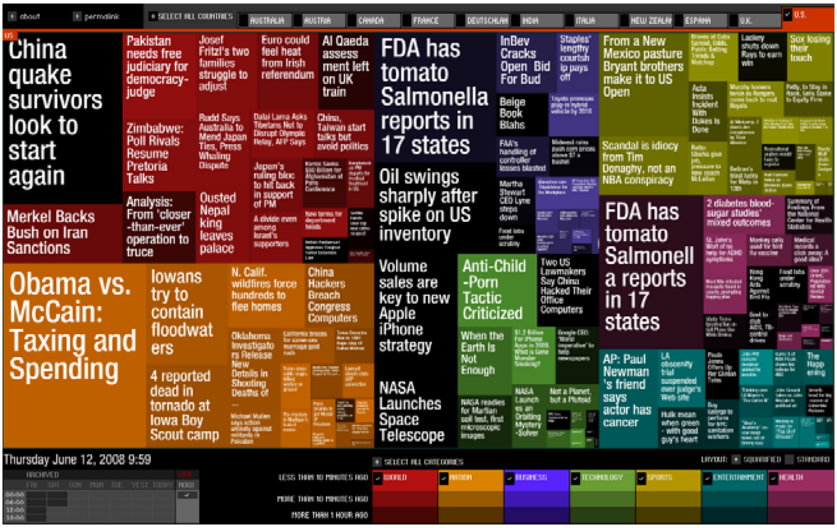

about
permalink
SELECT ALL COUNTRIES
AUSTRALIA
AUSTRIA
CANADA
FRANCE
DEUTSCHLAND
INDIA
ITALIA
NEW ZEALAND
ESPANA
U.K.
U.S.
US
China quake survivors look to start again
Merkel Backs Bush on Iran Sanctions
Pakistan needs free judiciary for democracy-judge
Zimbabwe: Poll Rivals Resume Pretoria Talks
Analysis: From 'closer-than-ever' operation to truce
Josef Fritzl's two families struggle to adjust
Rudd Says Australia to Mend Japan Ties, Press Whaling Dispute
Ousted Nepal king leaves palace
Euro could feel heat from Irish referendum
Dalai Lama Asks Tibetans Not to Disrupt Olympic Relay, AFP Says
Japan's ruling bloc to hit back in support of PM
A divide even among Israel's supporters
Al Qaeda assessment left on UK train
China, Taiwan start talks but avoid politics
Obama vs. McCain: Taxing and Spending
Iowans try to contain floodwaters
4 reported dead in tornado at Iowa Boy Scout camp
N. Calif. wildfires force hundreds to flee homes
China Hackers Breach Congress Computers
Oklahoma Investigators Release New Details in Shooting Deaths of ...
FDA has tomato Salmonella reports in 17 states
Oil swings sharply after spike on US inventory
Volume sales are key to new Apple iPhone strategy
NASA Launches Space Telescope
InBev Cracks Open Bid For Bud
Staples' lengthy courtship pays off
Beige Book Blahs
FAA's handling of controller losses blasted
Martha Stewart CEO Lyne steps down
Food labs under scrutiny
Anti-Child-Porn Tactic Criticized
Two US Lawmakers Say China Hacked Their Office Computers
When the Earth Is Not Enough
NASA readies for Martian soil test, first microscopic images
NASA Launches an Orbiting Mystery-Solver
Not a Planet, but a Plutoid
From a New Mexico pasture Bryant brothers make it to US Open
Scandal is idiocy from Tim Donaghy, not an NBA conspiracy
Lackey shuts down Rays to earn win
Sox losing their touch
Acta Insists Incident With Dukes Is Done
FDA has tomato Salmonella reports in 17 states
2 diabetes blood-sugar studies' mixed outcomes
AP: Paul Newman's friend says actor has cancer
LA obscenity trial suspended over judge's Web site
Hulk mean when green - with good guy's heart
The Happening
Thursday June 12, 2008 9:59
ARCHIVED
LIVE
FRI
SAT
SUN
MON
TUE
YEST
TODAY
NOW
00:00
06:00
12:00
18:00
SELECT ALL CATEGORIES
LESS THAN 10 MINUTES AGO
MORE THAN 10 MINUTES AGO
MORE THAN 1 HOUR AGO
WORLD
NATION
BUSINESS
TECHNOLOGY
SPORTS
ENTERTAINMENT
HEALTH
LAYOUT:
SQUARIFIED
STANDARD

Fig. 5.22 Newsmap uses a treemap algorithm to represent news from Google News U.S. on June 12, 2008. Image reproduced with the permission of Marcos Weskamp.

connecting to the website http://marumushi.com/apps/newsmap. Figure 5.22 shows the interface screen that visualizes the news coverage for Google News on June 12, 2008. The color defines the type of news (e.g., sport, entertainment, reports) and the size indicates how many articles deal with each story, from all of the sources considered by Google News. This application, thus, assigns some criteria of importance to the number of sources that report the news, assuming that the more important the news, the more it is present in the various journalistic sources. The intensity of color is also taken advantage of to understand how recent the information is: Recent news has a light coloring, while less recent news has a darker coloring.

5.2.5.2 Map of the Market

SmartMoney's Map of the Market (http://www.smartmoney.com/map-of-the-market) application was inspired by treemaps, to visualize the variations in stock exchange shares of over 500 stock markets in a single screen (Fig. 5.23). Map of the Market has attained notable success thanks to its very compact and elegant representation of a remarkable number of share titles, which provides a collective vision of progress for the entire shares market through a single map and, at the same time, allows visualization of the details of every share title. By using a colored rectangle visualization, organized hierarchically according to the 11 sectors (health care, finance, energy, technology, etc.), and grouped according to the industry type (e.g., the technological sector is further subdivided into software, hardware, Internet, telecommunications, semiconductors, and peripherals), the map automatically updates every 15

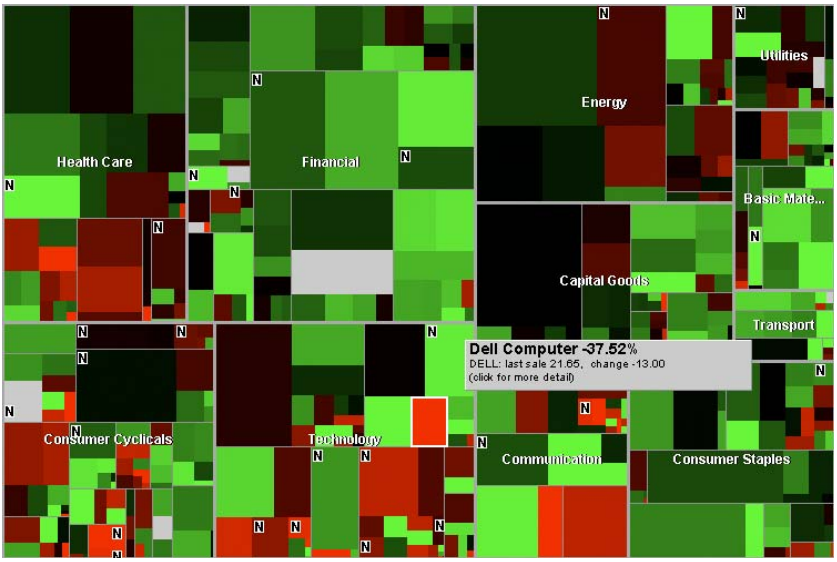


Fig. 5.23 Smartmoney.com Map of the Market. The map reports the variations during the past year on the national market on June 14, 2006. Image reproduced with the permission of ©SmartMoney.com.

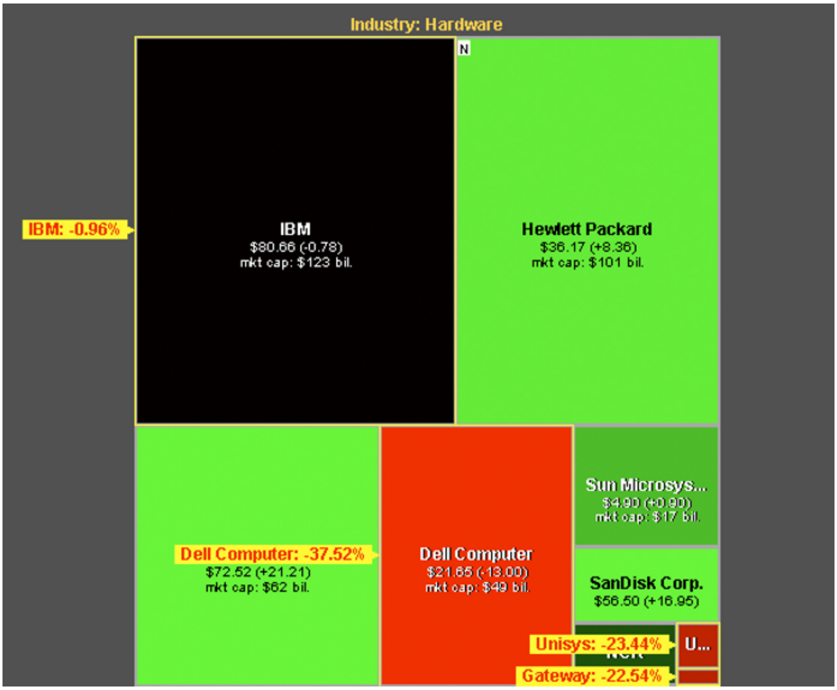


Fig. 5.24 Map of the Market shows the details for companies that deal with hardware. Image reproduced with the permission of ©SmartMoney.com.

minutes to allow the financial operator to check the entire stock market at a glance. Every colored rectangle of the map represents an individual company quoted on the stock market. The rectangle's size reflects the company's capitalization, which is the total market value of the shares issued by the company (in other words, the "big" companies are represented by rectangles of larger areas). The color reflects price performance in the period of time considered. In the example in Fig. 5.23, green means the stock price is up, red means it's down, dark colors indicate stationary situations, and the intensity of color reflects the importance of the variations. The user can get an idea of the entire stock market sector and compare different sectors and titles through the color of the blocks. The map is created by a Java applet, which the user can open with a browser connected to the website, and is very interactive. Using the appropriate controls, a user can see the details of every single title (for example, in Fig. 5.23, Dell Computer have been selected and present a strong negative variation over the last year), change the color scheme (particularly useful to color-blind people), change the time period of reference, or highlight the top five gainers or losers. Above all, the user can explore the map by zooming in on a particular sector or type of industry (the example in Fig. 5.24 represents the details of some companies that deal with computer hardware, indicating the shares with major losses).

5.2.5.3 SequoiaView

SequoiaView[9] and its clone applications, KDirStat for UNIX[10] and WinDirStat for Windows,[11] use treemaps to display the disk usage in terms of the dimension of files and folders. SequoiaView introduced a variant to treemaps: the *squarified cushion treemaps* [63] [6]. The screen is subdivided so that the rectangles resemble a square as closely as possible, to improve the readability of small files, which often leads to thin rectangles in the original treemap. Also, ridges are added to each rectangle. SequoiaView was developed by the Computer Science Department of the Technische Universiteit Eindhoven. A screenshot appears in Fig.5.25.

5.3 Conclusion

Several real-world datasets are organized other in the form of connections or containments, which can be naturally represented by graphs and trees. In these representations, items are encoded with nodes and relationships with edges. These types of representation are critical when there is a high number of nodes or edges: Crossing edges and overlapping nodes may make the visual representation unreadable.

[9] http://www.win.tue.nl/sequoiaview.

[10] http://kdirstat.sourceforge.net.

[11] http://windirstat.info.

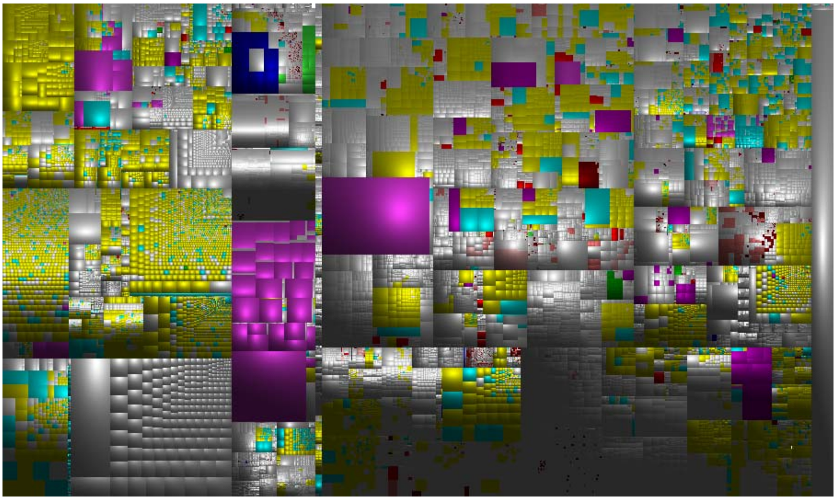


Fig. 5.25 Squarified cushion treemap in SequoiaView showing the whole content of the file system.

To reduce the complexity of a graph, one can intervene by attempting to diminish the number of nodes or edges visualized or using geometric node arrangements that improve readability or add interactivity. Trees are convenient way to represent hierarchical data. However, there are other ways to represent hierarchical data that allow to see their attributes and to identify specific patterns or properties of the hierarchy, such as the treemaps. In this chapter, we have shown some techniques that may help with the representation of complex graphs or trees.