

COSIN
IST-2001-33555

COevolution and Self-organization
In dynamical Networks

Customization and usability study of general purpose software tools for visualization of large networks

Deliverable Number:	D14
Delivery Date:	March 2004
REVISED VERSION ON	SEPTEMBER 2005
Classification:	public
Contact Authors:	Marco Gaertler and Dorothea Wagner
Document Version:	final
Contract Start Date:	01-03-2002
Duration:	42 months
Project Coordinator:	Guido Caldarelli
Partners:	INFM <i>Italy</i> Università "La Sapienza" <i>Italy</i> Universitat de Barcelona <i>Spain</i> Université de Lausanne <i>Switzerland</i> Ecole Normal Supérieure de Paris <i>France</i> Universität Karlsruhe <i>Germany</i>

Projects funded by the
European Commission under the
Information Society Technologies
Programme (1998-2002)



Abstract

The ability of drawing very large networks as e.g. large computer networks is of great significance in visualizing the evolution of stochastic models for evolving networks. One focuses on designing and implementing new algorithms and innovative software systems that display a large graph at different abstraction levels. For example, there is an increasing need of systems that show maps of the Web and support the user during her navigation, of systems that display and monitor the traffic on the Internet, and of systems that draw portions of the Internet as a graph. Until now, the vast majority of graph drawing algorithms that have been deeply studied and experimentally tested in the literature, like for instance for database schemes, can efficiently handle graphs of only hundreds of vertices. We aim at devising general algorithmic techniques for drawing large graphs and at experimenting their usage in new visualization systems, thus contributing to devising the technology transfer from the algorithmic research on graph drawing to its application in networks visualization. As part of this goal, we developed analysis-enhancing layouts and created in a cooperation with **CR8** Université de Paris Sud a novel technique that preserves the readability of abstract visualizations while showing all elements. On a different level we also report the activity that has been done in collaboration with Université de Paris Sud and **C01** Istituto Nazionale Fisica per la Materia in Rome. In the first case through k-core decomposition we can significantly reduce the complexity of the graph, still preserving some information. In the latter case the reduction is made by selecting the vertices to preserve according to their betweenness value.

General Techniques for Visualizing Large Graphs

In order to design a new tool that can handle larger networks, several issues have to be clarified first. Among many are the suitable representation of networks and precise descriptions for the connection between the analysis and the layouts. For this deliverable, the problem of finding a concise description is already tackled.

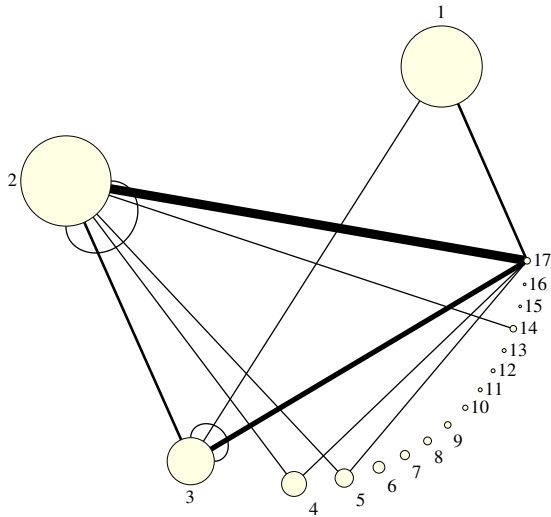


Figure 1: Reduced view of the Autonomous System graph. Nodes represent groups having similar connectivity and sizes are proportional to the number of elements in the group. Edges show the connections between groups and their thickness is proportional to the number of connections.

Several visualization techniques have been established that support drawing large graphs, like the collapse of subgraphs or groups, masking irrelevant parts or multi-views. However, these procedures are useless as long as they are not related to proper analysis methods. Network layouts and visual representation in general are usually created to support a penetrative analysis. On the other hand, images that are generated without taking into account fundamental properties of the network can readily lead to false conclusions or agendas. As a minor example the jellyfish representation of the Autonomous System graph can be listed. The model stated that the network consists of a heavy center with several also dense adjacent layers and some tree-like tentacles. Although this is true after all, in the original model one of the inner layers contained many nodes of very low degree.

As already pointed out, drawing large graphs is a difficult task and many already-existing tools restrict themselves to networks of small or medium size. We started to collect, evaluate and customize general algorithmic techniques for the handling of large graphs. In a first part, various transformations are presented. Most of them can be described as either *selecting interesting parts and masking the remaining* or as *abstracting the whole network to emphasize on the interaction of groups*. This ideas are illustrated by

Figures 1 and 2. Figure 1 shows a typical instance of abstraction. It gives a reduced view of the Internet at the Autonomous System level. Figure 2 gives an example of a hierarchical network drawn in mixed mode. It shows single actors as wells as groups or clusters. Note that some of the techniques developed involve clustering or grouping. As a basis for such methods, efficient clustering techniques are required. Therefore, we developed a new efficient clustering algorithm and evaluated its quality in an extensive experimental study. This work is presented in [7] and is explained in more detail as part of deliverable D06.

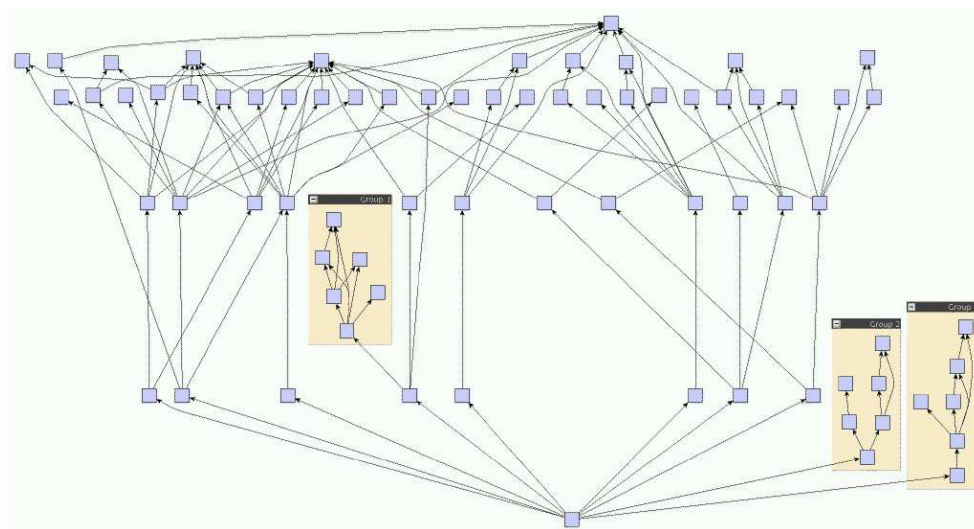


Figure 2: A hierarchical network drawn in mixed mode (single actors as wells as groups).

Besides reduction techniques, methods that target at a layout of the whole network are studied. In this case, we pay special interest to the time and space complexity of the algorithms. Last but not least, we also address non-standard ways of drawing large graphs. The following sections describe new visualization methods that were developed for large graphs from different fields of application. These methods are designed as special purpose algorithms. However, they can be viewed as models for more general tasks.

Visualizing Nested Decompositions

One form of clustering and grouping are nested decompositions, i.e., sequences of subsets, where every subset contains all following ones. They are a commonly used tool to model hierarchical structures and express structural importance or core-periphery compositions. Unfortunately, in most real networks, this kind of information is only implicitly given. Since manually classifying the elements of these instances is not possible, one needs to find proper graph-theoretic decompositions that approximate or synthesize the original information.

Such an instance is the physical Internet at the Autonomous System level. ASes can be roughly classified into: back-bone, national, regional, and local providers as well as customers. In [17], we verified that this structure is well related to an already known

and efficiently computable decomposition called k -cores. Based on this decomposition, we established in [4] one of the first non-trivial and readable drawings for the AS graph that showed all nodes and edges (Figure 4). Beside the visually pleasing representation, it also

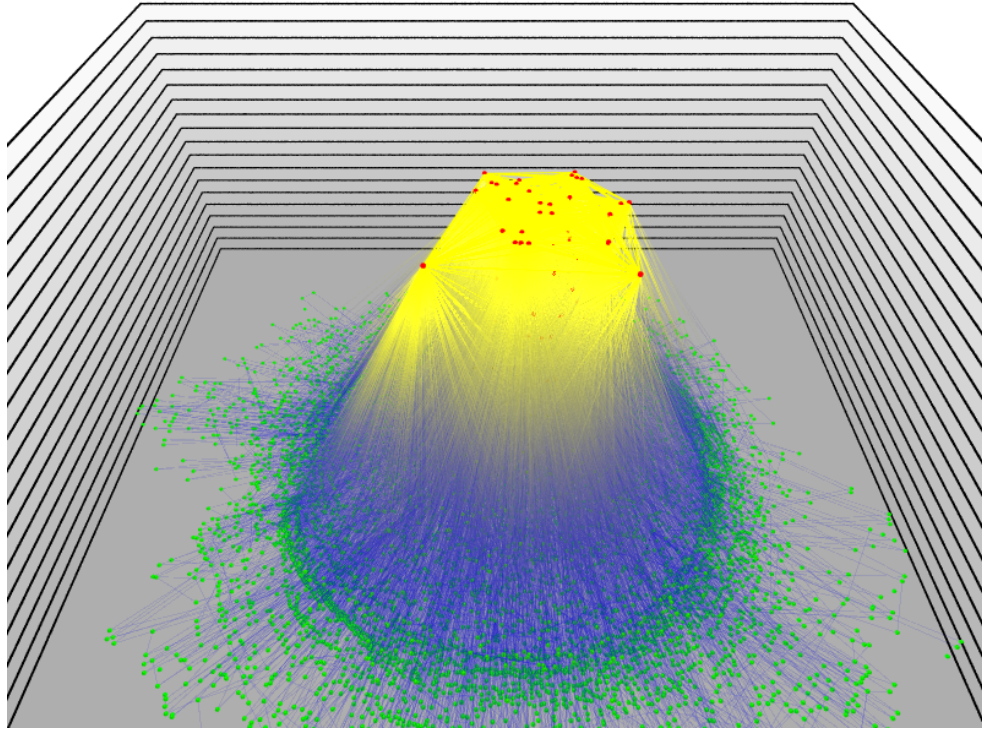


Figure 3: Hierarchical visualization of the Autonomous System network in 2.5D.

has further analytic relevance. For example, we studied the correlation between the AS graph and overlay networks, where we compared the induced topology of real and generated peer-to-peer networks. A first and simplified analysis showed that both networks are independent of the underlying AS network, while a visualization of the induced topologies revealed that both are structurally different. Thus, the effectiveness of our visualizations is demonstrated on the one hand by readable drawings and on the other hand by supporting further analyses. As ongoing research, we currently improve the technique to handle general nested decompositions and to speed-up the computation.

Pseudo-Abstract Visualizations

Visualizing large networks usually brings along a trade-off between, presented elements and information, achieved quality and computational effort. As mentioned above, one way to reduce the input complexity is abstraction. However, the accompanied loss of details makes the resulting layout often incompatible for further analyses. A compromise between complexity reduction and visually represented details are pseudo-abstract visualizations, i.e., visualizations that utilize a layout for an abstracted network as a blueprint. The

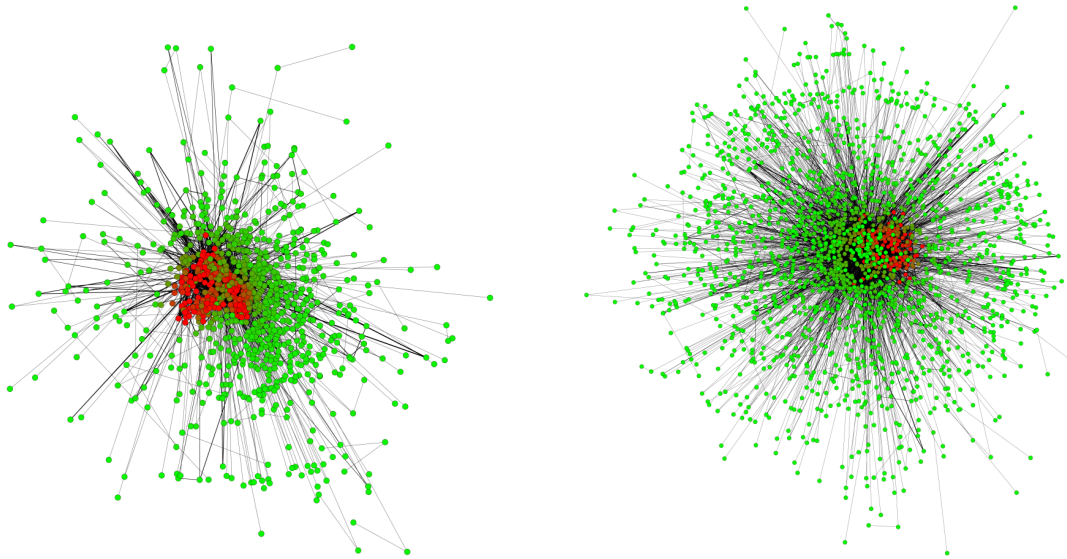


Figure 4: Comparing a real and a generated peer-to-peer network.

underlying shape of the network is usually fixed by the abstracted network and reflects certain characteristic properties. In a second phase, the actual elements of the network are placed. In this way, the readability of abstract visualizations is preserved while showing all elements of the network.

First results of such techniques were developed in a cooperation with Université de Paris Sud and are presented in [1, 2]. An example is given in Figure 5. Both use nested decompositions in the abstraction phase, however they differ in the choice of the underlying shape. In one case, concentric rings with different radii reflect selected properties of the nested decomposition, in the other, case angular segments are used. While the concentric rings have plenty degrees of freedom for placing the nodes, the angular segments provide more space for edges. Currently, the graphical representation of nodes reflects again se-

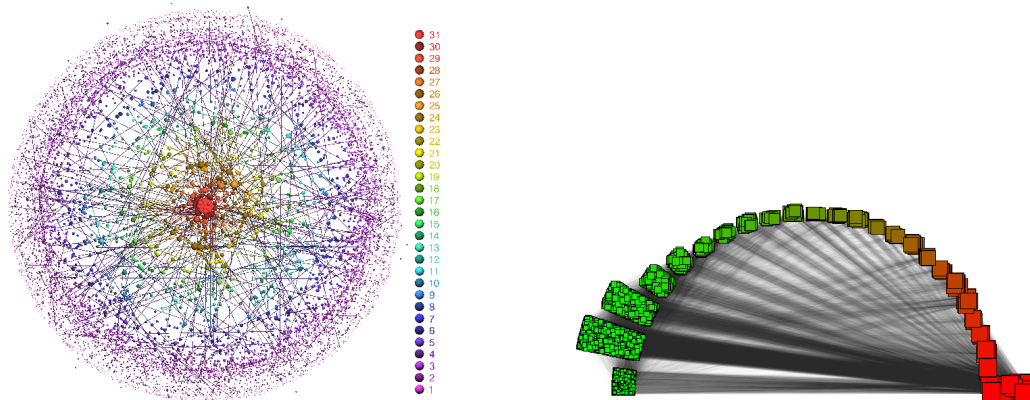


Figure 5: Pseudo-abstract visualizations of the AS network.

lected properties of the nested decomposition. In order to supply an analysis, it also could be defined by external parameters. Further, the two instances employ various methods to maintain readability. In the first work, a reduction of edge set via sampling improves the quality, while in the second work, graphical ‘tricks’ such as transparency are used to emphasize the important structures and to conceal irrelevant information. In both cases, the placement of the actual elements is very simplistic, yet the results are already satisfying and can be efficiently computed. Further improvement could reveal additional details or emphasize user-defined properties. Summarizing, such visualizations offer a user-scalable trade-off between quality, e.g., readability and recognition of included details, and the required computation time.

On the same line different decompositions are possible. One that could have some interest for the social network is based on the properties of the betweenness[9]. The *betweenness* b of a vertex i gives the probability that the site i is in the path from vertex j and vertex k . If the number of shortest pathways, between a pair of vertices (j, k) is $C(i, j)$, we denote with $C_i(j, k)$ the number of the shortest pathways running through i . The fraction $g_i(j, k) = C_i(j, k)/C(j, k)$ may be interpreted as the amount of the role played by the vertex i in social relation between two persons j and k . Betweenness of i is defined as the sum of $g_i(j, k)$ over all the connected pairs.

$$g_k = \sum_{i \neq j} g_k(i, j) = \sum_{i \neq j} \frac{C_k(i, j)}{C(i, j)}. \quad (1)$$

It has recently been shown[10] that the Probability distribution $P(b)$ for the betweenness b follows a power law,

$$P_B(g) \sim g^{-\eta}, \quad (2)$$

where the exponent η can take only the two values $\eta = 2$ and $\eta \approx 2.2 \pm 0.1$. Also the behaviour of $P(k)$ the probability distribution of the degree is very similar

$$P(k) \sim k^{-\gamma}, \quad (3)$$

where the exponent γ is all of the times in the interval $2 < \gamma < 3$.

Since this robustness in the critical exponents is therefore very tempting to re-express the parameters that define the system in terms of a simpler set by keeping unchanged the physical properties of the graph. This procedure is inspired to the Renormalization Group (RG) for critical phenomena and consist in the following two steps

- **Decimation** Select the site with lowest betweenness and remove it and the incident edges from the graph.
- **Rescaling** Compute again the value of the betweenness in the graph.

by following these two procedure all the statistical properties of the graph remain unchanged. We explicitly tested the form of the Degree distribution, the Clustering distribution, the Correlation $\langle K_{nn}(k) \rangle$ between degrees of neighbours vertices, and of course betweenness distribution. An example of the reduction is shown in Figure 6

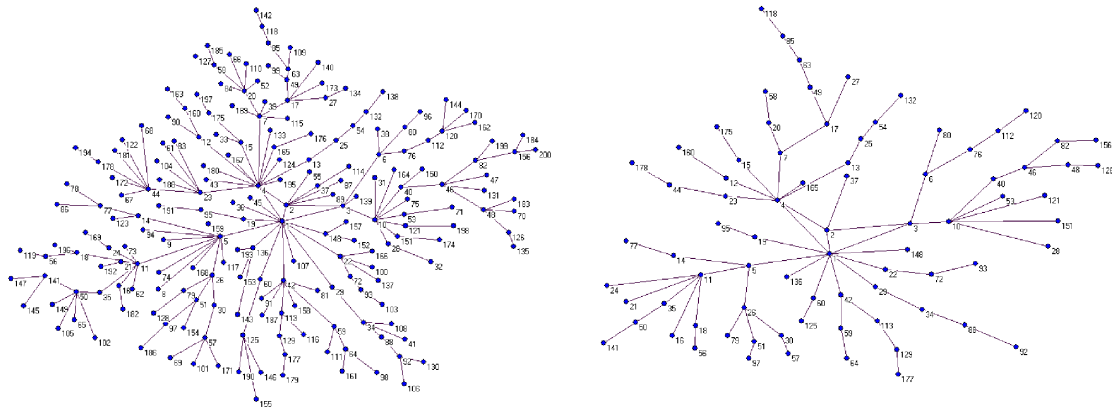


Figure 6: One step of decimation procedure for an Barabási-Albert network. The original graph is above. As scaling procedure we compute the new value of the betweenness on the graph below.

Visual Ranking in Large Graphs

Methods for ranking World Wide Web resources according to their position in the link structure of the Web are receiving considerable attention, because they provide the first effective means for search engines to cope with the explosive growth and diversification of the Web. Closely related methods have been used in other disciplines for quite some time.

In [11] we propose a visualization method that supports the simultaneous exploration of a link structure and a ranking of its nodes by showing the result of the ranking algorithm in one dimension and using graph drawing techniques in the remaining one or two dimensions to show the underlying structure. We suggest to use a simple spectral layout algorithm, because it does not add to the complexity of an implementation already used for ranking, but nevertheless produces meaningful layouts.

The effectiveness of our visualizations is demonstrated with example applications, in which they provide valuable insight into the link structure and the ranking mechanism alike. We consider them useful for the analysis of query results, maintenance of search engines, and evaluation of Web graph models.

The directed graph induced by the hyperlink structure of the Web has been recognized as a rich source of information. Understanding and exploiting this structure has a proven potential to help dealing with the explosive growth and diversification of the Web. Probably the most widely recognized example of this kind is the PageRank index employed by the Google search engine [15].

PageRank is but one of many models and algorithms to rank Web resources according to their position in a hyperlink structure. We propose a method to complement rankings with a meaningful visualization of the graph they are computed on.

While graph visualization is an active area of research as well, its integration with quantitative network analyses is only beginning to receive attention. It is, however, rather difficult to understand the determinants of, say, a particular ranking if its results do not

influence the way in which the structure is visualized. Standard rankings are based on spectral methods and iterative computation, but the same methods can also be used for graph layout. In the present application they are particularly well-suited, because densely connected subgraphs are clustered. On the Web, such subgraphs correspond to related resources and graphical clustering is therefore highly desirable. By using the axis separation principle and spectral layout techniques, a uniform approach to visual ranking of link structures is achieved.

Figure 7 shows spectral layouts of graphs generated according to the *small-world model* [24]. For this model, we initially generate a cyclic sequence of vertices and let a vertex link to a fixed number of predecessors and successors. Then, each edge is rewired with some small probability by choosing a new destination uniformly at random.

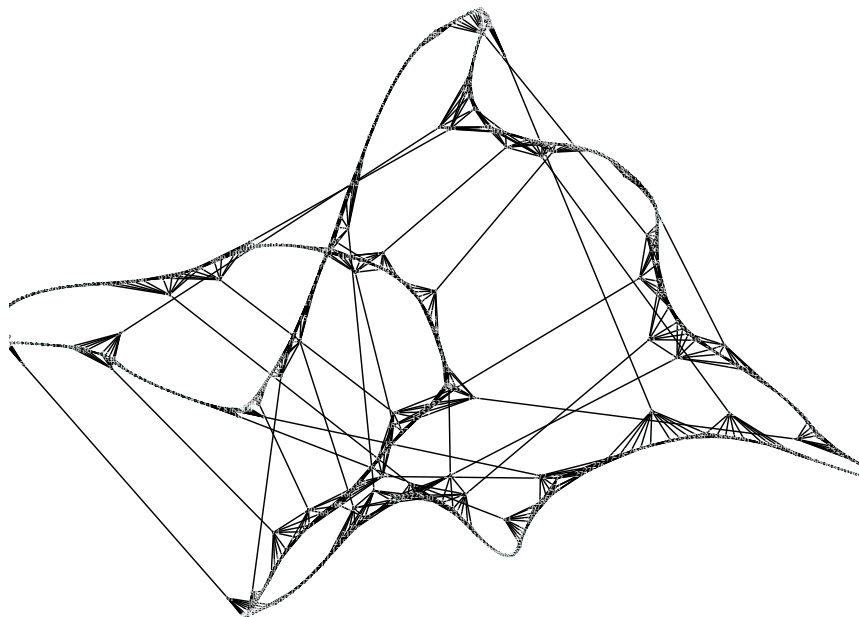


Figure 7: Spectral layout of a "small world"-graph

Figure 8 shows an example for visualizing rank in search engine query results. The data for this example was compiled in a way similar to the HITS algorithm [20]. We asked the AltaVista search engine for pages containing the word "java" and used the first 200 URLs it returned as the root set. It was then expanded by asking AltaVista for pages containing links to resources in the root set (backward extension), and adding resources linked to by pages in the root set (forward extension). The graph was completed by adding edges for all links between pages in the resulting set of vertices. The computations were carried out on the only large component of this graph from which some poorly connected vertices were removed to prevent extreme clustering. The graph has more than 5000 vertices and 15000 edges.

In Figure 8, this graph is shown with vertices positioned vertically according to the Fiedler vector, and horizontally according to one of two prominence indices. Links from

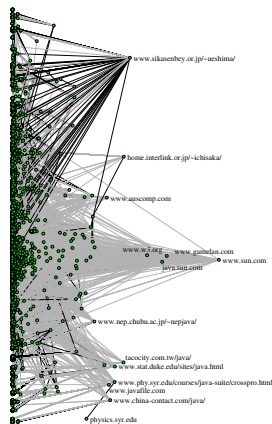


Figure 8: Authority and PageRank visualization of “java” query result

more to less prominent resources are colored black.

The most prominent resources under the PageRank measure match our expectations, but there are some surprising recommendations as well. It is clearly visible that some of these serve distinct user groups, like the japanese directory in the upper right. Note that, without zooming into the image, we may not conclude that vertically close vertices are closely connected. However, it is safe to assume that vertically separated vertices are relatively distant in the structure. This feature can serve to distinguish query results which contain a keyword that is used in different contexts (see the “jaguar”-query example in [20]).

Figure 8 also shows that the top authorities are surprisingly distinguished from the rest of the graph, and quite different from our expectations. Most of them are located at **Stars.com**, a large repository for developers (“Web Developer’s Virtual Library”). Since they are well connected among each other, it is by virtue of our layout approach that their vertical position is similar, and thus this phenomenon could be detected by visual exploration.

Visualizing Related Metabolic Pathways

Metabolic pathways are subnetworks of the complete network of metabolic reactions. They differ across organisms because different species may for example have developed different ways to synthesize a specific substance. We are interested in visualizing several related metabolic pathways in such a way that the inherent differences can be explored by trained biologists in order to understand the evolutionary relationships among species.

The structural characteristics of metabolic pathways make them particularly amenable to layered graph drawing methods. We therefore propose a method for visualizing a set of related metabolic pathways using $2\frac{1}{2}$ D graph drawing [12]. Interdependent, two-dimensional layouts of each pathway are stacked on top of each other so that biologists get a full picture of subtle and significant differences among the pathways. Layouts are deter-

mined by a global layout of the union of all pathway-representing graphs using a variant of the proven Sugiyama approach for layered graph drawing that allows edges to cross if they appear in different graphs.

To realize such a design, two graph drawing issues need to be addressed. We have to determine a suitable ordering to reduce the variation between consecutive pathways and we have to deal with dependencies introduced by the many substances and reactions present in more than one pathway. An interesting consequence is a new type of crossing number in which the weight of a crossing may be different for each pair of edges.

In [12], the utility of our approach is demonstrated on pathways extracted from the KEGG database [19] using the WilmaScope 3D graph visualization system [16]. The data consists of parts of the glycolysis and fructose/mannose metabolism pathways in seven organisms that show significant differences.

For network-based phylogenetic trees, we developed a visualization approach based on the idea of triangulation [13]. Phylogenetic analysis is an attempt to uncover the evolutionary relationships between organisms. It is an important tool in understanding evolutionary processes and in measuring genetic variations between species. Applications include the design of new drugs and reconstruction of the history of infectious diseases. The result of phylogenetic analysis is a phylogenetic tree representing hypothetical ancestral relationships among a set of entities. Established methods for phylogenetic analysis are based on morphological attributes or nucleotide and protein sequences.

Recently, new methods using metabolic pathway data have been introduced, and phylogenetic trees based on such data are becoming increasingly important. These trees have a complex structure as each node of the tree is a network rather than a sequence as in simple phylogenetic trees. This development complicates visualization of phylogenetic trees as sequences can be seen as one-dimensional information whereas networks cannot. Therefore, network-based phylogenetic trees require more elaborate visualization methods that provide a general tree overview as well as detailed visualization of the networks represented by nodes in the tree.

A naïve approach would be to draw the metabolic pathways inside the leaf nodes of their phylogenetic tree. Such diagrams, however, fail to convey the essential information in the data because viewers are not able to easily compare the similarities and differences of pathways.

We propose a visualization approach based on the idea of employing multiple complementary tools to study an object that is not easily understood when applying a single method. In this approach different visualizations are shown for different aspects and a diagram of the phylogenetic tree is used as the main selection panel to determine the data shown in the other views.

An early version of the system featuring only the $2\frac{1}{2}$ D related pathway view and the operational pathway view was recently evaluated in a user study with biologists. The ideas presented in [13] are designed to address issues raised in this study. The study involved a “cognitive walk-through” methodology in an interview type setting. That is, the domain experts were asked to use the system to explore a set of metabolic pathways, asked to think aloud as they proceeded with the exploration, and their feedback recorded.

In general the feedback regarding the use of the $2\frac{1}{2}$ D stacked view for comparing related pathways was very positive. However, most users felt that the number of pathways which could be successfully visualized simultaneously in this way was limited. Perhaps to around six to ten pathways at a time. A method for interactively selecting which of the available pathways are included in the stack is therefore essential. Thus, the phylogenetic tree view not only provides this facility, but does so in a way that is meaningful and intuitive for the biologists.

The same idea is also applicable to phylogenetic trees built from other complex structures such as protein-protein interaction networks or signal transduction pathways, and to multi-level structures in general. Future work will investigate utilizing these other data sources and application areas in our system, as well as modifications to address scalability issues.

Usability of existing packages

Current software tools for analyzing and visualizing networks have three major design issues: interactivity, range of analysis and layout methods. Unfortunately, interactivity is a large bottleneck with respect to the size of the networks. Several tools like Pajek [3] deal with this restriction by omitting a graph editor; networks can only be imported. Doing so they can easily analyze networks with 10,000 actors or more. However, most included layout algorithms are only suitable for graphs up to medium sizes.

As a first attempt to this deliverable, we collected informations and links of existing visualization software. In a case study, the advantages and disadvantages of these packages are illustrated. The results are made public at the COSIN web page. See <http://i11www.ira.uka.de/cosin/tools>. We also refer to [21].

Customization of general purpose software tools for visualization of large networks

Software Package **Visone**

We describe first **Visone**, a tool that facilitates the visual exploration of social networks [6, 14]. Social network analysis is a methodological approach in the social sciences using graph-theoretic concepts to describe, understand and explain social structure. The **Visone** software is an attempt to integrate analysis and visualization of social networks and is intended to be used in research and teaching. See also deliverable D6.

In contrast to more conventional mathematical software in the social sciences that aim at providing a comprehensive suite of analytical options, our emphasis is on complementing every option we provide with tailored means of graphical interaction. We attempt to make complicated types of analysis and data handling transparent, intuitive, and more readily accessible [23]. User feedback indicates that many who usually regard data exploration and analysis complicated and unnerving enjoy the playful nature of visual interaction.

Consequently, much of the tool is about graph drawing methods specifically adapted to facilitate visual data exploration. The origins of **Visone** lie in an interdisciplinary cooperation with researchers from political science which resulted in innovative uses of graph drawing methods for social network visualization, and prototypical implementations thereof. With the growing demand for access to these methods, we started implementing an integrated tool for public use. It should be stressed, however, that **Visone** remains a research platform and testbed for innovative methods, and is not intended to become a standard tool with all due consequences such as extensive user-support and product marketing. Essentially all components are in development and therefore subject to change. In a nutshell, **Visone** is a

- tool for interactive analysis and visualization of networks, in which
- originality is preferred over comprehensiveness, and that
- caters especially to social scientists.

The main application area of **Visone** is a methodological approach in the social sciences: *Social Network Analysis* uses graph-theoretic concepts to describe, understand and explain, sometimes even predict or design, social structure. The objects of interest are emergent patterns of relationships and their interplay with entity attributes.

Visualized information must neither be misleading nor hard to read. Hence there are two obvious criteria for the quality of social network visualizations:

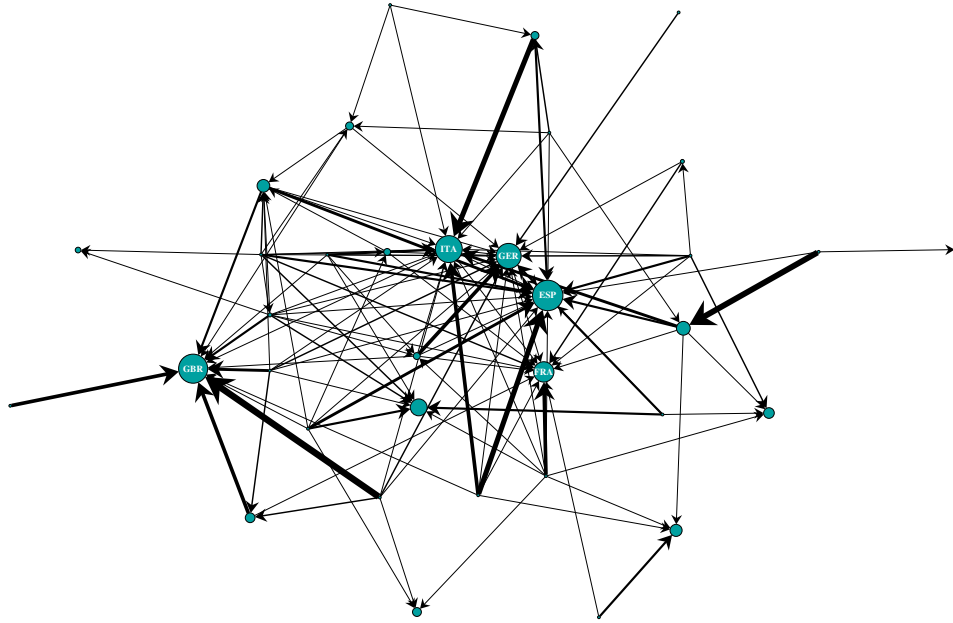
1. Is the information manifest in the network represented accurately?
2. Is this information conveyed efficiently?

With these criteria in mind, the following three aspects should be carefully thought through when creating network visualizations [8]:

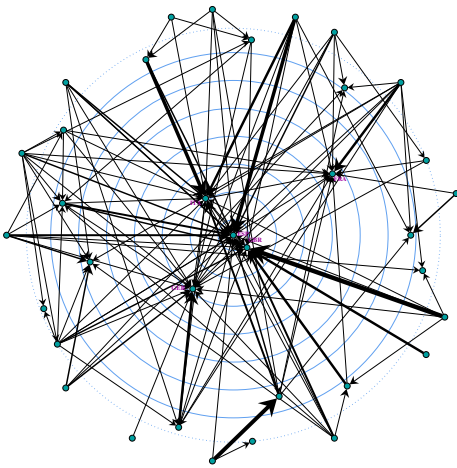
- the *substantive aspect* the viewer is interested in,
- the *design* (i.e. the mapping of data to graphical variables), and
- the *algorithm* employed to realize the design (artifacts, efficiency, etc.).

In addition to algorithms that try to produce what is often termed an “aesthetic” drawing of a graph (and thus are oblivious to the first aspect) we developed the following two types of visualization specifically for the vertex index analyses currently available in **Visone**.

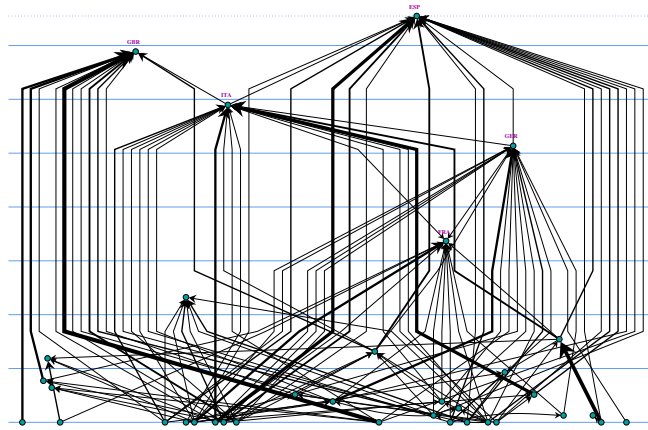
Depending on the context, actors of high structural importance are interpreted as being *central* or as having *high status*. With this substantive aspect in mind, we designed visualizations that represent vertex indices by constraining vertex positions to fixed distances from the center or from the bottom of the drawing, in either case depending linearly on the vertex index. See Figure 9 for illustration and note that relative scores are difficult to determine from the straightforward representation based on vertex size.



(a) vertex index represented by vertex size



(b) interpreted as centrality



(c) interpreted as status

Figure 9: Different means of visualizing a vertex index: most prestigious football leagues based on which ones the participants of the 1998 World Cup Final played in (network data courtesy of Lothar Krempel). Thickness of edges indicates number of players in foreign league. Like graph paper, background lines support determination and comparison of scores.

The information can thus be represented accurately, and it is up to the (constrained) graph layout algorithm to optimize readability. To avoid user dissatisfaction with suboptimal drawings, we strive to find at least locally optimal layouts that are not obvious for users to improve.

The **Visone** software is implemented in C++ using LEDA, the *Library of Efficient Data Types and Algorithms* [22]. While the user interface is a customized version of LEDA's GraphWin class, all graph generation, analysis, and layout algorithms have been implemented from scratch.

Starting with version 1.1, the main data format used in **Visone** is the XML sublanguage GraphML (Graph Markup Language) [6]. GraphML support is implemented in a LEDA extension package which is made available for public use. Besides GraphML, import and export in a number of simple formats and some formats customary in social network analysis and graph drawing are supported. To communicate results, visualizations can be exported in Scalable Vector Graphics (SVG) or PostScript format. Many conversion tools exist for both. The SVG export routine has been adopted into the core LEDA package. There is neither a macro language nor an interface for third-party extensions, but limited support of command-line options for batch-mode operations is planned in the future.

The **Visone** software is provided as a standalone executable for systems running Linux, Solaris, or Windows, and is free for academic purposes.

Technically, the user interface inherits from LEDA's graph editor class GraphWin, though several internal modifications were necessary to address the needs of researchers and students in the social sciences. With application-specific terminology it comprises the usual drawing canvas with pull-down and context (pop-up) menus. Therefore, network data can be imported from a file but also input or edited graphically.

Recently, a new version of **Visone** has been implemented where instead of LEDA, the yFiles tool [25] is used. Thus robustness and flexibility of **Visone** has been increased.

References

- [1] I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. k-core decomposition: A tool for the visualization of large scale networks. arXiv:cs.NI/0504107.
- [2] I. Alvarez-Hamelin, M. Gaertler, R. Görke, and D. Wagner. k-core decomposition: A tool for the visualization of large scale networks. arXiv:cs.NI/0504107.
- [3] V. Batagelj and A. Mrvar. Pajek – Analysis and Visualization of Large Networks In: P. Mutzel and M. Jünger (eds.) *Special issue on Graph Drawing Software*, Mathematics and Visualization, Springer, 2003.
- [4] M. Baur, U. Brandes, M. Gaertler, and D. Wagner. Drawing the as graph in 2.5 dimensions. In *Proceedings of the 12th International Symposium on Graph Drawing (GD'04)*, volume 3383 of *Lecture Notes in Computer Science*, pages 43–48, 2005.

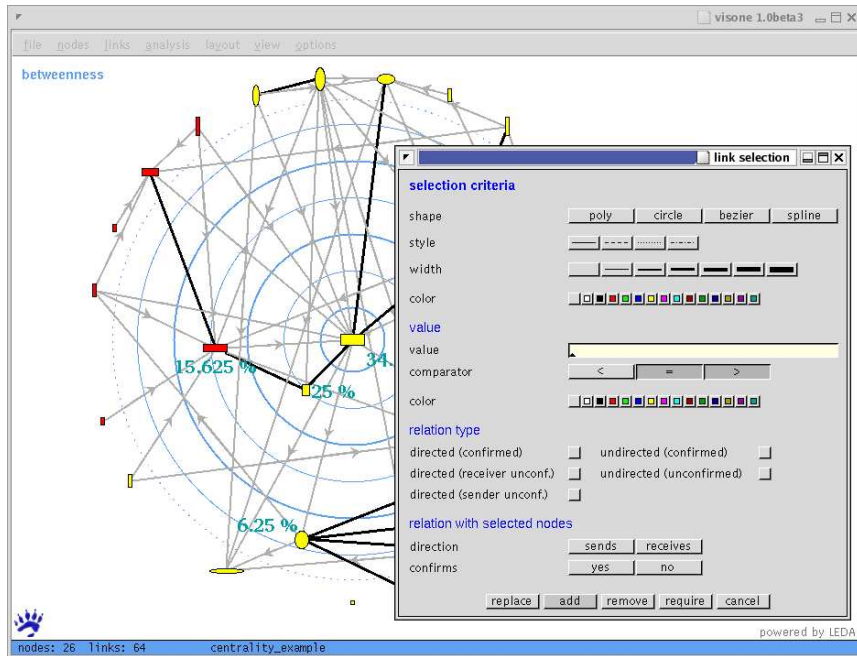


Figure 10: visone user interface with convenient selection options.

- [5] U. Brandes, J. Raab, and D. Wagner. Exploratory network visualization: Simultaneous display of actor status and connections. *Journal of Social Structure*, 2(4), 2001.
- [6] U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, and M. S. Marshall. GraphML progress report: Structural layer proposal. In: P. Mutzel, M. Jünger, and S. Leipert (eds.) *Proceedings 9th International Symposium on Graph Drawing (GD '01)*, LNCS 2265:501–512, Springer 2002. For up-to-date information see <http://graphml.graphdrawing.org/>.
- [7] U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering algorithms. In *Proceedings of the 11th Annual European Symposium on Algorithms (ESA'03)*, LNCS 2832, pp. 568–579, Springer-Verlag, 2003.
- [8] U. Brandes, P. Kenis, J. Raab, V. Schneider, and D. Wagner. Explorations into the visualization of policy networks. *Journal of Theoretical Politics*, 11(1):75–106, 1999.
- [9] G. Caldarelli, F. Rao and M. Gaertler. Renormalization Scheme for Graphs. in preparation 2005
- [10] K.-I. Goh, B. Kahng and D. Kim. Universal Behaviour of Load Distribution in Scale-free Networks. *Physical Review Letters*, 87, 278701(1-4), 2001.
- [11] U. Brandes and S. Cornelsen. Visual Ranking *Journal of Graph Algorithms and Applications* 7(2) pp. 181–201, 2003.

- [12] U. Brandes, T. Dwyer and F. Schreiber. Visualizing Related Metabolic Pathways in Two and a Half Dimensions. *Proc. 11th Intl. Symp. Graph Drawing (GD'03)*, LNCS 2912, pp. 111-122, Springer, 2004.
- [13] U. Brandes, T. Dwyer and F. Schreiber. Visual Triangulation of Network-Based Phylogenetic Trees. To appear in *Proc. 6th Joint Eurographics - IEEE TCVG Symp. Visualization (VisSym '04)*.
- [14] U. Brandes and D. Wagner. visone Analysis and visualization of social networks. In: P. Mutzel and M. Jünger (eds.) *Special issue on Graph Drawing Software*, Mathematics and Visualization, Springer, 2003.
- [15] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107-117, 1998.
- [16] T. Dwyer and P. Eckersley. The WilmaScope 3D graph drawing system. In P. Mutzel and M. Jünger, editors, *Graph Drawing Software*, Mathematics and Visualization, pages 55-75. Springer, 2003.
- [17] M. Gaertler and M. Patrignani. Dynamic analysis of the autonomous system graph. To appear in *Proceedings of IPS 2004, International Workshop on Inter-domain Performance and Simulation*.
- [18] C. Gkantsidi, M. Mihail, and E. Zegura. Spectral analysis of internet topologies. In *IEEE Infocom 2003*, 2003.
- [19] M. Kanehisa and S. Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acid Research*, 28(1):27-30, 2000.
- [20] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the Association for Computing Machinery*, 46(5):604-632, September 1999.
- [21] P. Mutzel and M. Jünger (eds.). *Special issue on Graph Drawing Software*, Mathematics and Visualization, Springer, 2003.
- [22] K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, 1999.
- [23] D. Wagner. Analysis and Visualization of Social Networks, *Proceedings of the 2nd International Workshop on Experimental and Efficient Algorithms (WEA'03)*, LNCS 2647, pp. 261-266, Springer, 2003.
- [24] D. J. Watts and S. H. Strogatz. Collective dynamics of “small-world” networks. *Nature*, 393:440-442, 1998.
- [25] R. Wiese, M. Eiglsperger and M. Kaufmann. yFiles – Visualization and automatic layout of graphs. In: P. Mutzel and M. Jünger (eds.) *Special issue on Graph Drawing Software*, Mathematics and Visualization, Springer, 2003.