

Poster Gallery Report

Sue Whitesides

School of Computer Science
McGill University
3480 University St.
Montreal, Quebec H3A 2A7
Canada

GraphPack

M. S. Krishnamoorthy,^a F. Oxaal,^b U. Dogrusoz,^c D. Pape,^d A. Robayo,^e R. Koyanagi,^f Y. Hsu,^g D. Hollinger,^h A. Hashmiⁱ

^a moorthy@cs.rpi.edu, Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy NY 12180 USA

^b oxaalf@cs.rpi.edu, Step Tools Inc., Rensselaer Technology Park, North Greenbush NY USA

^c dogrusoz@cs.rpi.edu, Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy NY 12180 USA

^d Dept. of Computer Science, University of Illinois at Chicago, Chicago IL USA

^e AT&T Laboratories, Consulting Corners NJ USA

^f Wall Street Consulting Co., 200 Wall Street, NY NY USA

^g hsuyf@cs.rpi.edu, Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy NY 12180 USA

^h hollind@cs.rpi.edu, Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy NY 12180 USA

ⁱ hashmia@cs.rpi.edu, Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy NY 12180 USA

Abstract: We describe different methods for three dimensional embedding of graphs, as well as a two dimensional layout method, namely barycentric embedding. In addition, we present a program for automatic recognition of graphs. Finally, we describe servers for graph drawing routines that can be called from C or C++ programs and Mosaic. We also describe some of the capabilities of GraphPack, the system that we developed here at Rensselaer. We show several drawings of graphs and digraphs using GraphPack.

The Interactive Graph-Visualization System *daVinci*

Michael Fröhlich,^a Mattias Werner^b

^a mfr@Informatik.Uni-Bremen.DE, University of Bremen — FB 3, Institute for Formal Methods in Software-Engineering, PO-Box 330440, D-28334 Bremen GERMANY

^b mawe@Informatik.Uni-Bremen.DE, University of Bremen - FB 3, Institute for Formal Methods in Software-Engineering, PO-Box 330440, D-28334 Bremen GERMANY

Abstract: The graph-visualization system *daVinci* is an interactive tool that can be used by arbitrary application programs as a user interface for graph data structures. Besides a novel automatic layout algorithm for graphs, *daVinci* offers many interactive facilities such as fine-tuning of a layout, abstractions and scaling operations. A bi-directional application interface is used for tool communication with a program that controls the graph structure. *daVinci* is already in use in many different academical and commercial software projects.

Layout with Constraints and Animation

Xiaobo Wang,^a Bin Hu,^b Kazuo Sugihara^c

^a xiw@coby.ics.hawaii.edu, Information and Computer Science Dept., Univ. of Hawaii at Manoa, 2565 The Mall, #200, Honolulu HI 96822 USA

^b hubin@coby.ics.hawaii.edu, Information and Computer Science Dept., Univ. of Hawaii at Manoa, 2565 The Mall, #200, Honolulu HI 96822 USA

^c sugihara@coby.ics.hawaii.edu, Information and Computer Science Dept., Univ. of Hawaii at Manoa, 2565 The Mall, Honolulu HI 96822 USA

Abstract: LYCA is a graph tool which uses a spring embedder algorithm and Sugiyama's algorithm to draw undirected/directed graphs. LYCA's constraint manager detects the distortion caused by resolving constraints in the layout process and attempts to reduce the distortion by interaction between the constraint manager and the layout algorithms. The interaction is also used to bias the layout process to generate different layouts, depending on preferential constraints or clustering constraints. LYCA animates the layout process. The user can pause, play back, continue the layout process or set the animation speed and pause condition. The user can also edit the intermediate configuration as layout instructions. This aims at integrating the user's modification with the automatic layout algorithm to improve the layout quality and allows the user to control the layout by direct manipulation.

A Better Heuristic for Orthogonal Graph Drawing

Therese Biedl,^a Goos Kant^b

^a therese@rutcor.rutgers.edu, RUTCOR, Rutgers University, P.O. Box 5062, New Brunswick NJ 08903 USA

^b goos@cs.ruu.nl, Dept. of Computer Science, Utrecht University, Padualaan 14, 3584 CH Utrecht THE NETHERLANDS

Abstract: An orthogonal drawing of a graph is an embedding in the plane such that all edges are drawn as sequences of horizontal and vertical segments. We present a linear time and space algorithm to draw any n -vertex connected graph orthogonally on a grid of size $n \times n$ with at most $2n + 2$ bends. Each edge is bent at most twice. In particular for non-planar and non-biconnected planar graphs, this is a big improvement. The algorithm is very simple, easy to implement, and it handles both planar and non-planar graphs at the same time.

A Visual Programming Environment Based on HICHART Program Diagrams

Yoshihiro Adachi,^a Youzou Miyadera,^b Kimio Sugita,^c Kensie Tsuchida,^d Hiroshi Unno,^e Takeo Yaku^f

^a adachi@krc.toyo.ac.jp, Toyo University, Dept. Inform. & Compt., Toyo Univ., Kuji-
rai, Kawagoe, Saitama 350 JAPAN

^b miyadera@j.dendai.ac.jp, Tokyo Denki University, Dept. Inform. Sci., Tokyo Denki
Univ., Hatayaka, Saitama 350-03 JAPAN

^c sugita@cc.u-tokai.ac.jp, Tokai University, Dept. Math., Tokai Univ., Hiratsuka,
Kanagawa 259-12 JAPAN

^d kensei@krc.toyo.ac.jp, Toyo University, Dept. Inform. & Compt., Toyo Univ., Kujirai,
Kawagoe, Saitama 350 JAPAN

^e unno@jaist.ac.jp, Japan Advanced Institute of Sci. & Techno., Tastuguchi, Ishikawa
923-12 JAPAN

^f yaku@tansei.cc.u-tokyo.ac.jp, Dept. Appl. Math., Nihon Univ., Setagaya, Tokyo 156
JAPAN

Abstract: We have implemented a software development environment called ETA-AIDE3 which based on a graph drawing theory. In this environment, the tree-structured diagram "Hichart" and the structured specification document "Hiform" are treated. ETA-AIDE3 is used for building an advanced programming education and software engineering environment.

Do You Recognize This Graph?

Shimon Even^a

^a even@cs.Technion.AC.IL, Computer Science Department, Technion, Israel Inst. of
Tech., Haifa 32000, Israel

Abstract: A drawing of a (nice and symmetric, very well known) graph.

ICD-EDIT: A Server for 2&3/4-d Interactive Connection Diagram Graphics with Prolog Clients

David Dodson,^a Hugh Reeves,^b Rob Scott^c

^a ded@cs.city.ac.uk, City University, London, Computer Science, City University,
Northampton Square, London EC1V 0HB UK

^b lhr@cs.city.ac.uk, City University, London, Computer Science, City University,
Northampton Square, London EC1V 0HB UK

^c rbs@cs.city.ac.uk, City University, London, Computer Science, City University,
Northampton Square, London EC1V 0HB UK

Abstract: ICD-EDIT is a server for graphical interaction with 3d node-and-link diagrams. It provides a flexible medium for diagrammatic interaction with client applications and supports diagram editing with or without a client process. To aid responsive diagram manipulation without 3d hardware it uses a limited form of 3-dimensionality, 2&3/4-d, in which nodes always face the user and stay upright. Depth cueing is aided by optional “rocking” motion which cycles the display between three stored images. ICD-EDIT’s rationale, functionality, X/Motif implementation and connection to a Prolog client are illustrated.

TRIVIAL: Refocusing in Cooperative Diagrams with Ductile Space

David Dodson^a

^a dcd@cs.city.ac.uk, City University, London, Computer Science, City University, Northampton Square, London EC1V 0HB UK

Abstract: We are building diagrammatic interfaces which are intended to cooperate with users in the maintenance of connection, containment and alignment relations in the layout of 3d diagrams subjected to multi-focal browsing operations. Our method is to design systems of local force interactions which arbitrate between separation, adjacency and alignment preferences in a mechanical and - to a useful extent - an intuitively predictable way. The diagram undergoes simulated Newtonian metamorphosis, flowing into new configurations rather than being navigated-in. This approach is conceived as exploiting the situated kinetic intuition of users to empower them to make beneficial topology changes to diagrams undergoing gradual change in content. A “Ductile Space” metaphor is used to provide initial layout adjustment when a node of the diagram is required to wax, wane, disappear or reappear. This 3d analogue to rubber-sheet layout is briefly analysed and illustrated.

n-Dimensional RSPs, Right Multilayered Diagrams and Prolog

David Dodson^a

^a dcd@cs.city.ac.uk, City University, London, Computer Science, City University, Northampton Square, London EC1V 0HB UK

Abstract: Building diagrammatic interfaces which maintain their spatial arrangement relations of containment and alignment in 3d merits a data type in which these primitives can be combined. Right Simple Place (RSP), for want of a better term, is a data type for an n -dimensional place, $0 \leq n \leq 3$, the boundary of which is defined by independently assigning to each dimension either a fixed coordinate value or a finite or infinite range of coordinate values. This leads to considerable computational efficiency in a rich world of operations on RSPs. Uses of RSPs in modelling multilayer diagrams and the implementation of relevant operations on RSPs in Prolog are illustrated.

On Drawing Cubic Graphs

Liu Yanpei,^a Paola Marchioro,^b Rossella Petreschi^c

^a Institute of Applied Mathematics, Academia Sinica, Beijing 100080 CHINA

^b Dept. of Mathematics, University of Rome “La Sapienza”, P.le A. Moro, 5, 00185 Rome ITALY

^c petreschi@dsi.uniroma1.it, Dept. of Computer Science, University of Rome “La Sapienza”, Via Salaria 113 00198 Rome ITALY

Abstract: We present a new algorithm to draw on a grid a cubic planar graph, with the optimization of the number of bends. The peculiarity of the algorithm is that it does not require a planar embedding as input, but actually tests planarity at the same time it constructs a drawing of the graph. Furthermore, it is shown a new complete proof of the fact that any planar cubic graph, except the tetrahedron, is at most single-bend embeddable. Finally, from the algorithm is derived the fact that the total number of bends in any planar cubic graph with n vertices is less than or equal to $0.5n + 1$.

GraphEd

Michael Himsolt,^a Timo Hickl,^b Andreas Stuebinger^c

^a himsolt@fmi.uni-passau.de, Universitaet Passau, Innstrasse 33, 94032 Passau GERMANY

^b hickl@fmi.uni-passau.de, Universitaet Passau, Innstrasse 33, 94032 Passau GERMANY

^c stuebing@fmi.uni-passau.de, Universitaet Passau, Innstrasse 33, 94032 Passau GERMANY

Abstract: GraphEd is an extensible editor for graphs and graph grammars. It comes with several application modules. Many of them are graph theoretical or graph layout algorithms. Our poster gives an impression of how GraphEd looks on screen, and presents highlights from our scientific work.

A Software Environment for Algorithmic Research

James Abello,^a Antonio Miranda,^b Craig Smith^c

^a abello@cs.tamu.edu, Computer Science Dept., Texas A&M University, College Station TX 77843 USA

^b Computer Science Dept., Texas A&M University, College Station TX 77843 USA

^c Computer Science Dept., Texas A&M University, College Station TX 77843 USA

Abstract: We have designed and implemented the main elements of a software environment for creating and interacting with visual expositions of algorithmic concepts. The environment consists of manipulation tools, an algorithm animation system, an interpreted programming language and a scripting facility.

The manipulation tools allow one to construct and manipulate algorithmic objects interactively. A set of primitive operations is provided from which more powerful operations may be formed. Through the animation facilities one may develop applications which visually demonstrate the dynamics of algorithms allowing the user to specify run-time inputs. The scope of the scripting facility, ranges tremendously from taking research notes to preparing lecture presentations, recording algorithms test runs, giving homework problems and so on. Two prototypes called X-AGE and MIRAGE have been designed and implemented. Patterned by the client-server paradigm, X-AGE consists of a server that isolates its clients from the details of window management, user interaction and graph animation. MIRAGE is an interpreted C-like language with built in primitives to ease the task of algorithm animation. Experience suggests that this environment contributes to the understanding of algorithmic concepts and promotes research – the experience of implementing X-AGE alone has furnished several unsolved graph theoretical problems.

3D Graph Drawing

Qing Wen Feng,^a Patrick Garvan,^b Diet Ostry^c

^a qwfeng@cs.newcastle.edu.au, Dept. of Computer Science, The University of Newcastle, University Drive, Callaghan 2308 NSW AUSTRALIA

^b pgarvan@cs.newcastle.edu.au, Dept. of Computer Science, The University of Newcastle, University Drive, Callaghan 2308 NSW AUSTRALIA

^c dostry@cs.newcastle.edu.au, Dept. of Computer Science, The University of Newcastle, University Drive, Callaghan 2308 NSW AUSTRALIA

Abstract: Drawing graphs in 3D rather than 2D has the potential to represent information in a more understandable way. We are investigating several aspects of 3D graph drawing, including drawing planar graphs as strictly convex polyhedra (P. Garvan), 3D drawing methods for hierarchical graphs (D. Ostry) and drawing clustered graphs (Q.W. Feng).

TOSCANA — A Graphical Tool for Analyzing and Exploring Data

Rudolf Wille,^a Frank Vogt,^b Wolfgang Kollwe,^c Martin Skorsky^d

^a wille@mathematik.th-darmstadt.de, Technische Hochschule Darmstadt, Fachbereich Mathematik, Schlossgartenstrasse 7, D-64287 Darmstadt GERMANY

^b vogt@mathematik.th-darmstadt.de, Technische Hochschule Darmstadt, Fachbereich Mathematik, Schlossgartenstrasse 7, D-64287 Darmstadt GERMANY

^c kollwe@mathematik.th-darmstadt.de, Technische Hochschule Darmstadt, Fachbereich Mathematik, Schlossgartenstrasse 7, D-64287 Darmstadt GERMANY

^d skorsky@mathematik.th-darmstadt.de, Technische Hochschule Darmstadt, Fachbereich Mathematik, Schlossgartenstrasse 7, D-64287 Darmstadt GERMANY

Abstract: TOSCANA is a computer program which allows an online interaction with larger (relational) data bases to analyse and explore data conceptually. It uses labelled line diagrams of concept lattices to communicate knowledge coded in given data. The basic problem to create online presentations of concept lattices is solved by composing prepared diagrams to nested line diagrams. A larger number of applications in different areas have already shown that TOSCANA is a useful tool for many purposes.

Applications: classification of documents, analyzing situations, retrieval of information, decision support, analysis of data in medicine, sociology and psychology, optimization of production process, presentation of thesauri

Keywords: diagrams of hierarchical structures (lattices), relational databases, communication of knowledge

VEGA — Doing Discrete Mathematics with Computers

Tomaž Pisanski,^a Bor Plestenjak,^b Jože Marinčec^c

^a Tomaz.Pisanski@uni-lj.si, University of Ljubljana, IMFM/ Dept. of Theoretical Computer Science, Jadranska 19, SLO 61111 Ljubljana SLOVENIJA

^b Tomaz.Pisanski@uni-lj.si, University of Ljubljana, IMFM/ Dept. of Theoretical Computer Science, Jadranska 19, SLO 61111 Ljubljana SLOVENIJA

^c Tomaz.Pisanski@uni-lj.si, University of Ljubljana, IMFM/ Dept. of Mathematics, Jadranska 19, SLO 61111 Ljubljana SLOVENIJA

Abstract: The scope of The VEGA system will be presented. Vega is neither a graph drawing package nor a graph editing system. It is an integrated environment for doing discrete mathematics. It enables the user to manipulate with ease relatively sophisticated discrete mathematical objects like graphs, groups, Cayley graphs, polyhedra, and maps on surfaces. Some of its Postscript graphic output will be displayed.

Impressions from the Gem Algorithm

Arne Frick,^a Andreas Ludwig,^b Heiko Mehlau,^c

^a frick@informatik.uni-karlsruhe.de, Universitaet Karlsruhe, Postfach 6980, Vincenz-Priessnitz-Str. 3, D-76128 Karlsruhe GERMANY

^b Universitaet Karlsruhe, Postfach 6980, Vincenz-Priessnitz-Str. 3, D-76128 Karlsruhe GERMANY

^c Universitaet Karlsruhe, Postfach 6980, Vincenz-Priessnitz-Str. 3, D-76128 Karlsruhe GERMANY

Abstract: This poster shows many visual examples from actual runs of the Gem algorithm for the layout of undirected graphs [Frick, Ludwig, Mehlau, paper and demo, this conference]. It includes snapshots of runs on a single graph at different times as well as final results for other types of graphs. We show evidence that in many cases, the Gem algorithm matches or improves the results published

for other heuristic approaches [Kamada/Kawai, Reingold/Fruchterman, Davidson/Harel] either in speed or in quality or both. We also show examples for the inherent limits of methods optimizing for the same set of aesthetics criteria as Gem does.

Encoding Presentation Emphasis Algorithms for Graphs

Emanuel G. Noik^a

^a noik@db.toronto.edu, Computer Systems Research Institute, 6 King's College Road,
Toronto, Ontario M4S 1A1 CANADA,

Abstract: While graphs have been demonstrated to be effective at visualizing one or more relations on a set of elements, drawings of large graphs can be very difficult to understand. As such, numerous presentation emphasis techniques for visualizing graphs such as fisheye views have been proposed and extensively studied. A recent survey paper [Noik, Graphics Interface 1994, Banff, Alberta] described an abstract space of techniques and identified a number of common shortcomings. In the GD '94 paper we introduce a high-level language that addresses several of these limitations. The poster will present a number of examples that could not be included in the paper due to page limit constraints.

Symmetric Graph Drawing

Joseph Manning^a Mikhail Atallah,^b

^a manning@cs.vassar.edu, Dept. of Computer Science, Vassar College, Poughkeepsie
NY 12601 USA

^b mja@cs.purdue.edu, Dept. of Computer Sciences, Purdue University, West Lafayette
IN 47907 USA

Abstract: This poster illustrates the results of our research into using *symmetry* to produce good drawings of abstract graphs. The display of axial and/or rotational symmetry appears to be a powerful and useful criterion in drawing graphs, and can produce some very appealing results. Its importance may stem from the fact that, given a symmetric drawing, a conceptual understanding of the entire graph can be built up from that of a smaller subgraph, replicated a number of times. However, we have proved that the basic problem of determining if a general abstract graph possesses any such symmetry is \mathcal{NP} -complete, so our work has accordingly focused on detecting symmetry in planar graphs. We have succeeded in finding optimal linear-time algorithms for detecting and displaying both axial and rotational symmetries in *trees*, *outerplanar graphs*, and *embedded planar graphs*. This poster presents a short outline of the algorithms, along with several symmetric graph drawings which they generated.

(Research supported by the National Science Foundation under Grant CCR-9202807.)

Interactive Graph Based Visual Languages

Jeffrey D. McWhirter^a

^a jeffm@cs.colorado.edu, Dept. of Computer Science, Campus Box 430, University of Colorado, Boulder CO 80309 USA

Abstract: Graphs and their pictorial representations serve to facilitate human-computer communication in a wide range of areas. In the context of interactive graph based computer applications, the role of the pictorial representation of the graph and the mechanisms that support interactions with the graph are crucial to the user's ability to create, manipulate and understand the underlying graph based information structure. In this poster we present Escalante, a development environment that supports the rapid construction of interactive, graph based applications. Escalante provides a realization of general domain model, a rich structured graphics library for defining the representation of the graph and a flexible architecture that supports multiple graph representations.

Drawing Graphs by Example Efficiently: Trees and Planar Acyclic Graphs

Isabel F. Cruz,^a Ashim Garg^b

^a isabel@cs.tufts.edu, Dept. of Electrical Engineering and Computer Science, Tufts University, Medford MA 02155 USA

^b ag@cs.brown.edu, Brown University, Dept. of Computer Science, Providence RI 02912-1910 USA

Abstract: Traditionally the algorithmic and declarative approaches to graph drawing use textual specifications. However, it is common practice to explain the layout of a graph using pictures that describe the drawing of the different components of a graph and the constraints that specify their relative positions. A facility for writing visual programs for drawings graphs allows the programmers to transform directly their mental picture of the desired drawing into a visual program. This greatly eases the task of programming and produces (visual) code that is easy to understand and maintain. As a first step towards providing such a facility, in our GD '94 paper we show how to write visual programs for constructing planar drawings of trees, series-parallel graphs, and acyclic digraphs using a visual language. We show that these programs construct the drawings in linear time.