Pictures of Applications Using Garnet

Edited by
Brad A. Myers

October 1993

Abstract

This is a collection of pictures of real applications written by Garnet users. You can print just this section of the manual on a color printer to see some dramatic color images.

This research was sponsored by the Avionics Lab, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, Arpa Order No. 7597. Additional support for Garnet has been provided by NEC, Apple, Adobe, and General Electric.

1. Some Garnet Applications

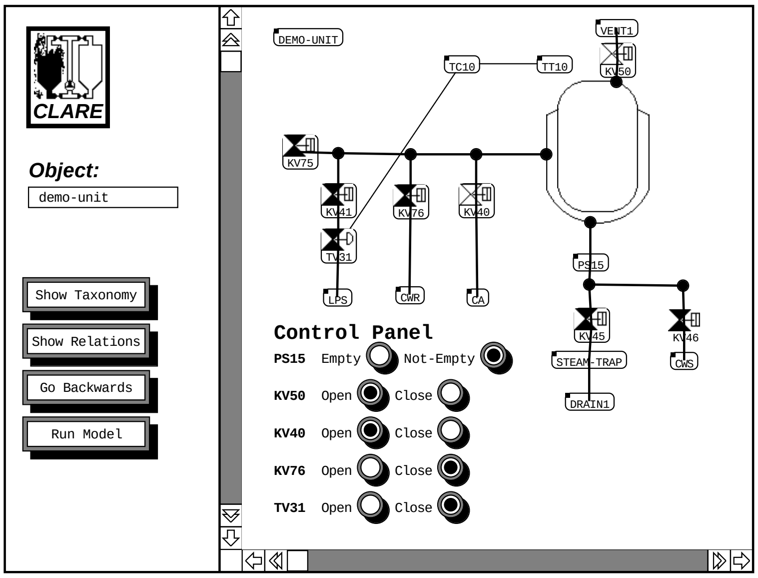
This section presents some examples of applications written using Garnet. As of September 29, 1993, we know of over 50 applications written using Garnet, listed below. If you are using Garnet and you are not in the list, please send mail to `garnet@cs.cmu.edu`. The figures on the following pages show some of the applications. These were generated using the Garnet utility `Opal:make-ps-file`. Many of these figures are in color, so if you have a color printer or postscript previewer, you might try looking at this file in color.

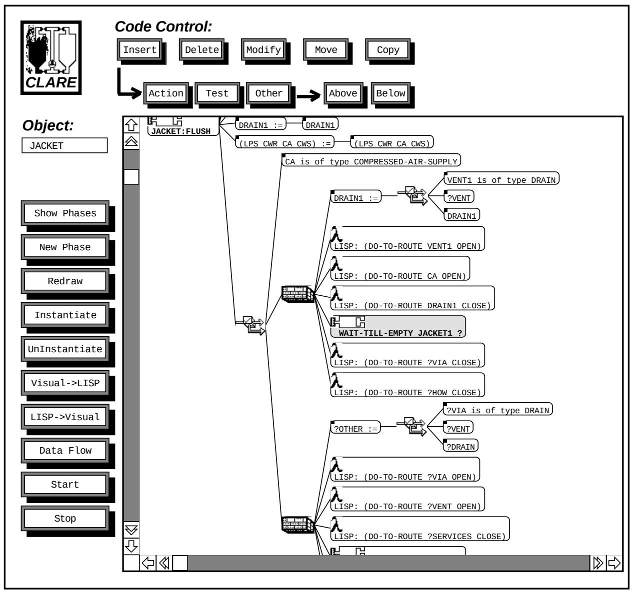
Companies

1. Corporation For Open Systems
   Automated Protocol Analysis/Reference Tool
   Frank J. Wroblewski
2. Design Research Institute (Cornell & Xerox)
   (various)
   Jim Davis
3. Deutches Forschungszentrum fuer Kuenstliche Intelligenz GmbH
   COSMA
   Stephen P. Spackman
4. GE Research and Development Center
   Metallurgical Expert Systems for Manufacturing
   K. J. Meltsner
5. GE Research and Development Center
   SKETCHER
   K. J. Meltsner
6. Hughes AI Center
   NLP Project
   Seth Goldman or Charles Dolan
7. Institute of Knowledge Engineering, Madrid, Spain
   KIISS: Knowledge Interface Interactive Surgery System
   Roberto Moriyon
8. Lawrence Livermore National Lab
   PLANET (Pump Layout ANd Evaluation Tool)
   Tom Canales
9. Microelectronics & Computer Tech. Corp. (MCC)
   Scan: Intelligent Text Retrieval
   Elaine Rich
10. MITRE Corporation
    AIMI (An Intelligent Multimedia Interface)
    John D. Burger
11. National Science Center Foundation
    Learning Logic
    Lawrence Freil
12. PTT Research, Groningen, The Netherlands
    R&D Employee Information Retrieval Service
    Colin Tattersall
13. Rank Xerox EuroPARC
    (various CSCW-related projects)
    Paul Dourish
14. StatSci
    GRAPHICAL-BELIEF
    Russell Almond
15. Transarc Corporation
    Lisp GUI for Encina
    Mark Sherman
16. U S WEST Advanced Technologies
    KBNL natural language system
    Randall Sparks
17. USC/ISI
    Humanoid
    Pedro Szekely
18. USC/ISI
    SHELTER knowledge-based development environ.
    Pedro Szekely

Universities

19. Carnegie Mellon University, CS
    Miro
    J. D. Tygar and J. M. Wing
20. Carnegie Mellon University, CS
    Learning Calendar System
    Conrad Poelman
21. Carnegie Mellon University, CS
    Interactive Fiction Editor
    Merrick Furst
22. Carnegie Mellon University, CS
    Redstone
    Jeff Schlimmer
23. Carnegie Mellon University, CS
    PURSUIT
    Francesmary Modugno
24. Carnegie Mellon University, ERDC
    LOOS
    Ulrich Flemming or Robert Coyne
25. Carnegie Mellon University, Math
    Educational Theorem Proving System
    Peter Andrews
26. Carnegie Mellon University, Psychology
    Soar Graphics Interface
    Frank Ritter
27. Carnegie Mellon University, Robotics
    MICRO-BOSS
    Norman Sadeh
28. Carnegie Mellon University, Robotics
    SAGE Data Visualization
    Steve Roth
29. Georgia Institute of Technology
    KritikTutor (interactive learning environment)
    Andres Gomez de Silva Garza, Nathalie Grue
30. MIT, Dept. of Brain and Cognitive Sciences
    SURF-HIPPO Neuron Simulator
    Lyle J. Borg-Graham
31. MIT, LCS, Computation Structures Group
    Debugging tools for the Id Language
    Steve Glim and R. Paul Johnson
32. New York University, Courant Institute
    COMLEX (Common Lexicon) dictionary project
    R. Grishman, C. Macleod, S. Wolff, and K. Rajan
33. Oregon State University
    Forms/3 (visual language)
    Margaret Burnett
34. Oregon State University, Dept. of CS
    CarGo (Game of Go)
    Peter Dudey
35. State University of New York at Buffalo, CS
    Air Battle Simulation
    Henry Hexmoor
36. State University of New York at Buffalo, CS
    SNePS Graphical UI
    John S. Lewocz
37. Tulane University, CS
    Natural Language Processing
    Robert Goldman
38. Tulane University, CS
    THESEUS
    Raymond Lang
39. Universidad de las Americas-Puebla (Mexico)
    Visual Language
    Miguel Sanchez
40. University College London
    The Cognitive Browser
    Gordon Joly
41. University of Calif. at Santa Barbara, CS & ECE
    Graphical Tools for the Development of Concurrent Systems
    L. Dillon, G. Kutty, P. Melliar-Smith, L. Moser, Y. Ramakrishna
42. University of Chicago, CS
    Shopper (planning and visual navigation)
    Daniel Fu
43. University of Chicago, CS
    SEAL (situation-driven execution and constructivist learning)
    Charles Earl
44. University of Chicago, AI Lab
    Roentgen (case-based aid for radiation therapy)
    Jeff Berger
45. University of Leeds
    Graphical Multi-User Domain Designer
    Roderick J. Williams
46. University of Leeds
    CLARE
    Nikos Drakos
47. University of Leeds
    ADVISOR
    Andrew J. Cole
48. University of Leeds
    PORSCHE
    Colin Tattersall
49. University of Michigan, EE&CS
    Science works: Nuclear Engineering
    K. Abotel, E. Solloway, C. Quintaina, W. Martin
50. University of Queensland, Australia
    Conceptual Modeling CASE tool
    Anthony Berglas
51. University of Queensland, Australia
    Interoperating Network Servers
    Hung Wing
52. University of Saskatchewan
    DISCUS (DIstributed Computing at the U of S)
    Beth Protsko (User Interface only)
53. University of Southern California
    Dynamic Aggregation in Qualitative Simulation
    Nicolas Rouquette
54. University of Southern California
    Revenge: Knowledge-based S/W for (Re)Design-for-Assembly
    Gerard Kim, Dr. George Bekey,
55. University of Washington, CS
    Multi-Garnet
    Michael Sannella
56. University of Washington, CS
    Electronic Encyclopedia Exploratorium
    Mike Salisbury

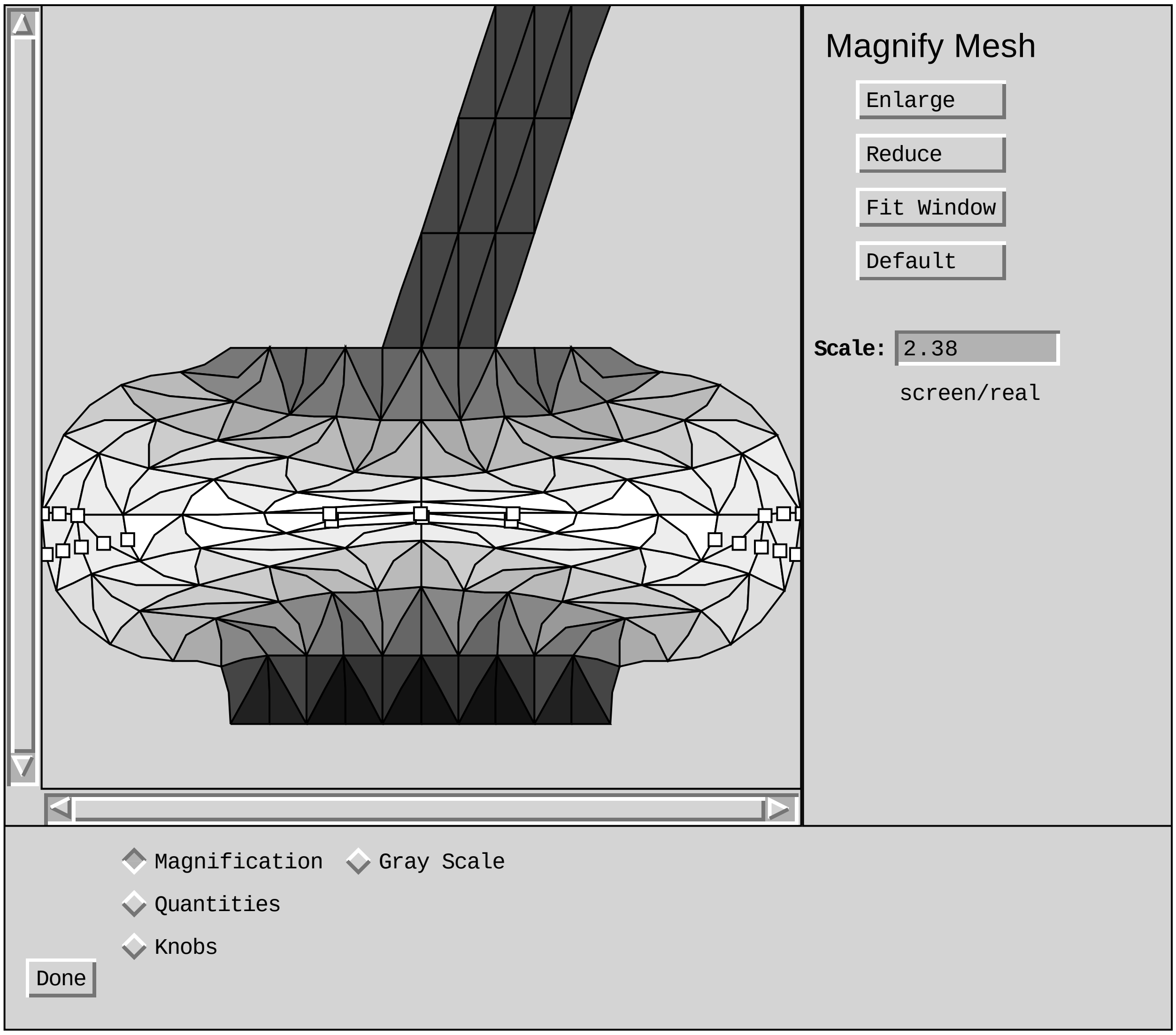




Nikos Drakos
Computer Based Learning Unit, University of Leeds, UK.
CLARE
nikos@cbl.leeds.ac.uk

A collaborative project on an environment for the specification, testing, maintenance and automatic generation of application software. The context is batch process control in Chemical Engineering although it is envisaged that the applicability of the environment will be more general. A ‘domain expert’ will be able to specify knowledge about plant subsystems, plant configurations, and the allowable generic operations and constraints on each plant subsystem. An ‘application engineer’ will then use the system to ‘glue’ together predefined operations in order to make specific products. The system will then generate process control code for particular target hardware. Garnet is being used to capture and visualise plant and process information through schematics, process diagrams, interactive simulations and simple animations.

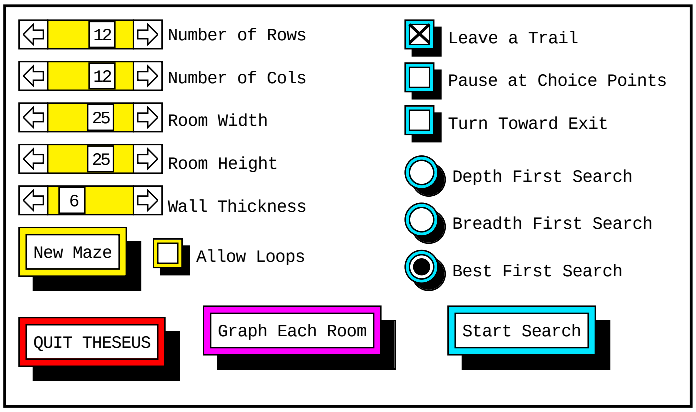
Nikos Drakos, ‘‘Object Orientation and Visual Programming’’, in Mamdouh Ibrahim, editor, *OOPSLA ’92 Workshop on Object-Oriented Programming Languages: The Next Generation*, Vancouver, B.C. Canada, October 18 1992. Extended Abstract. pp. 85-93.

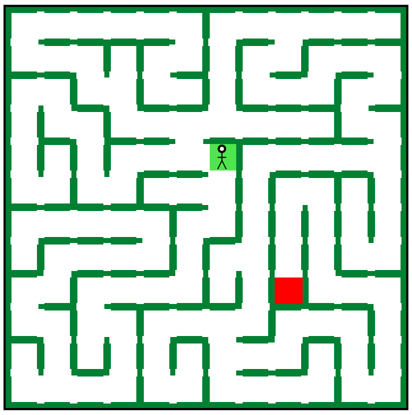


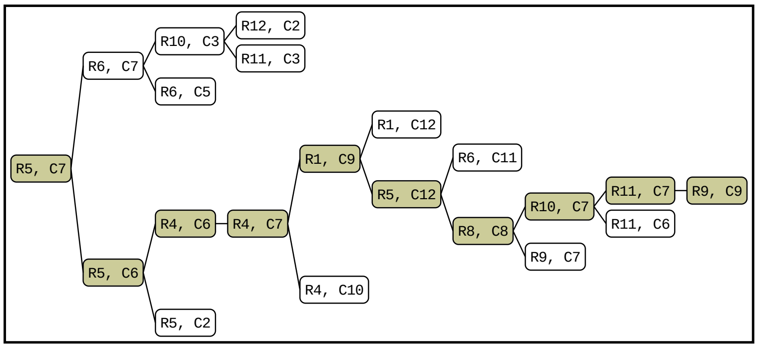
Kenneth Meltsner
General Electric Company, Corporate Research and Development
Metallurgical Expert System
meltsner@crd.ge.com

A mesh created using a virtual aggregate for the polygons and another virtual aggregate for the square knobs. For the polygons, the virtual aggregate is passed a prototype for a polygon, and an array containing the list of points and the color for each polygon. The virtual aggregate then pretends to allocate an object for each element of the array, but actually just draws the prototype object repeatedly.

Kenneth J. Meltsner. ‘‘A Metallurgical Expert System for Interpreting FEA,’’ *Journal of Metals*, Oct, 1991, vol. 43, no. 10, pp. 15-17.



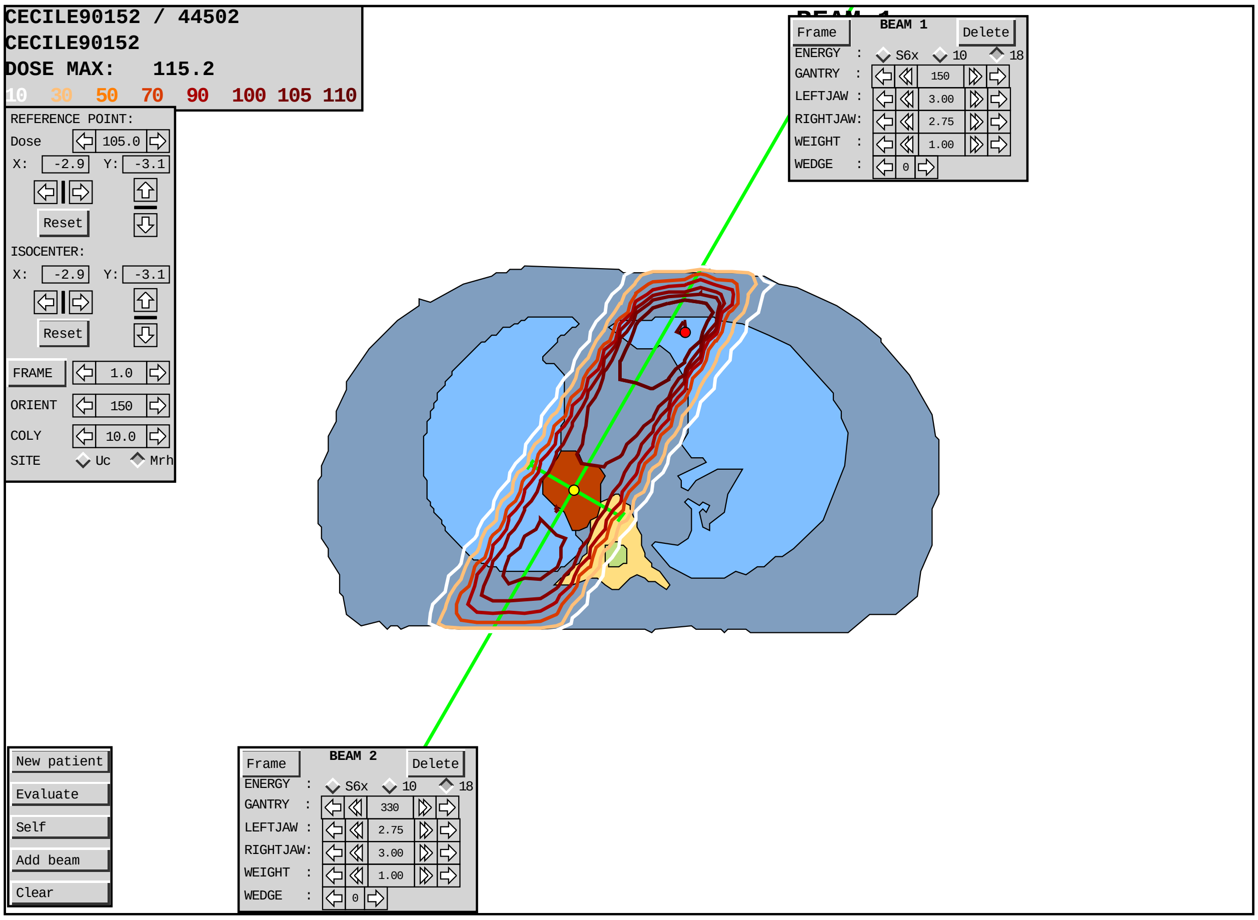




Raymond Lang
Tulane University, Computer Science Department
THESEUS
`lang@rex.cs.tulane.edu`

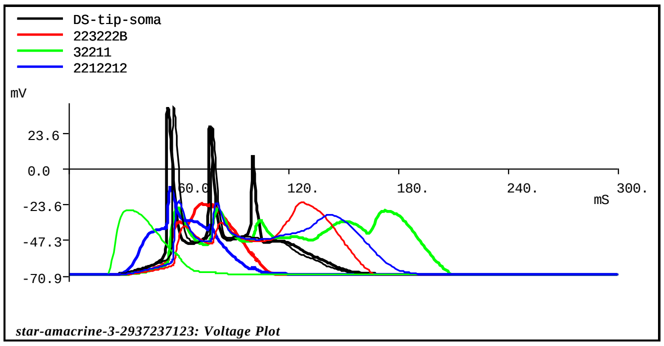
These are images of windows from the THESEUS application used by the Tulane University Computer Science Department on guided tours of the department given to visiting high school seniors and other interested parties. THESEUS is intended to be used as part of a presentation on what the study of computer science entails. It does this by showing graphically the progress and results of common search methods applied to the problem of finding the exit of a randomly created maze. THESEUS was developed in CMU Common Lisp version 16d and the Garnet X-Windows toolkit version 2.01.

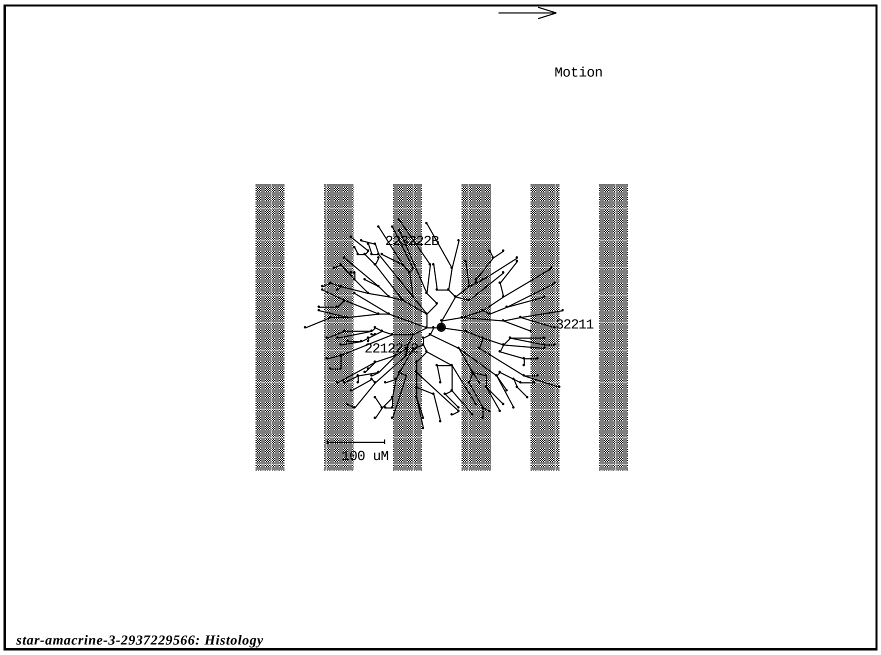
R. Raymond Lang, *THESEUS: Using Maze Search to Introduce Computer Science*. Technical Report. Computer Science Department, Tulane University. 1992.



Jeff Berger
University of Chicago, Artificial Intelligence Laboratory
Roentgen
berger@cs.uchicago.edu

Roentgen is a case-based aid to radiation therapy planning. It relies on an archive of past therapy cases to suggest plans for new therapy patients. Roentgen supports therapy planning by: 1) retrieving the case which best matches the geometry and treatment constraints of the new patient; 2) tailoring the plan to the specific details of the patient; 3) evaluating the results of applying the plan; 4) repairing the plan to avoid any discovered faults in treatment results. This final plan is the system's suggestion to the human planner. Roentgen breaks new ground in solving problems in a domain dominated by spatial reasoning and the satisficing of constraints.

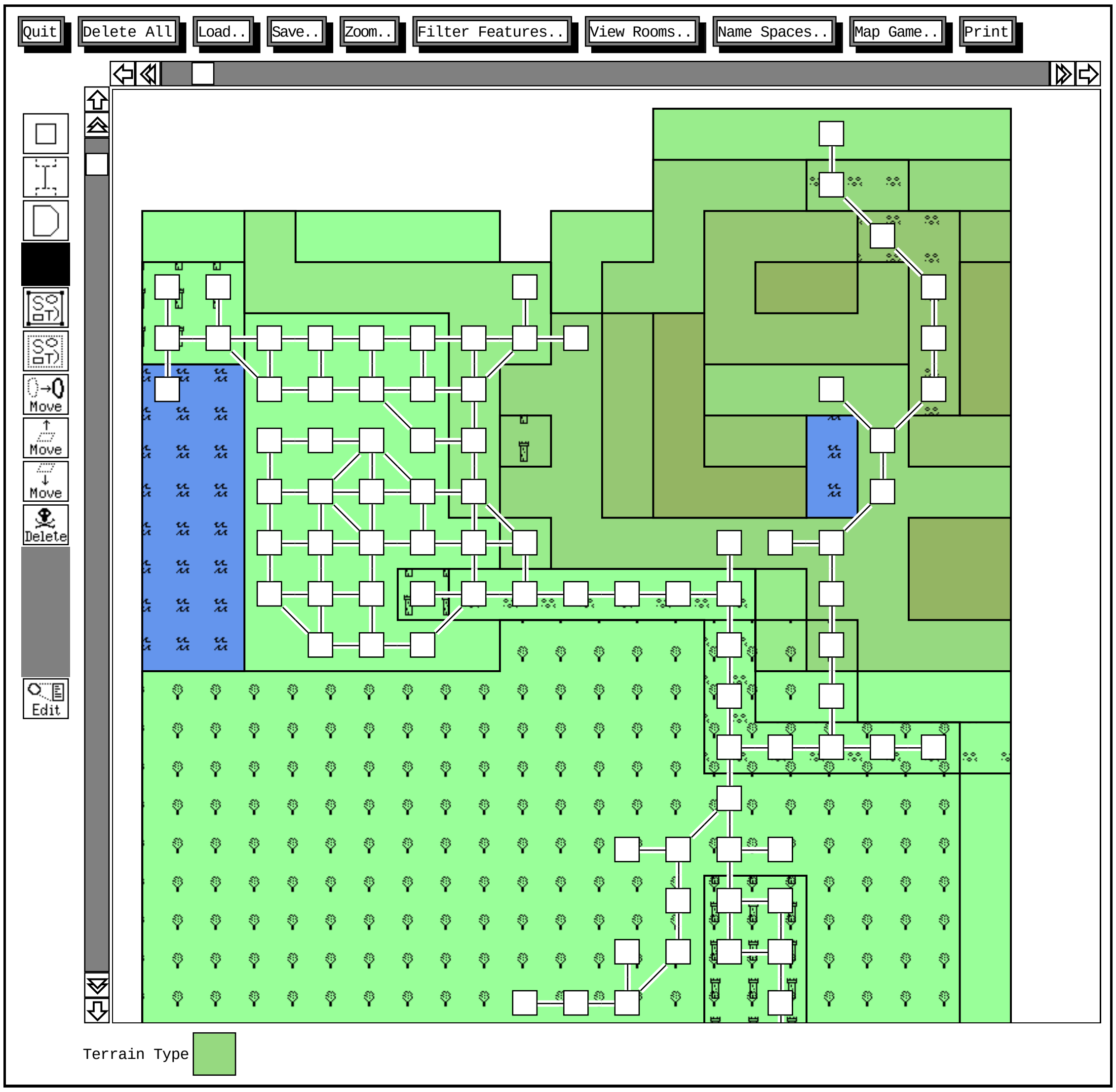




Lyle J. Borg-Graham
MIT Dept. of Brain and Cognitive Sciences
Surf-Hippo
`lyle@ai.mit.edu`

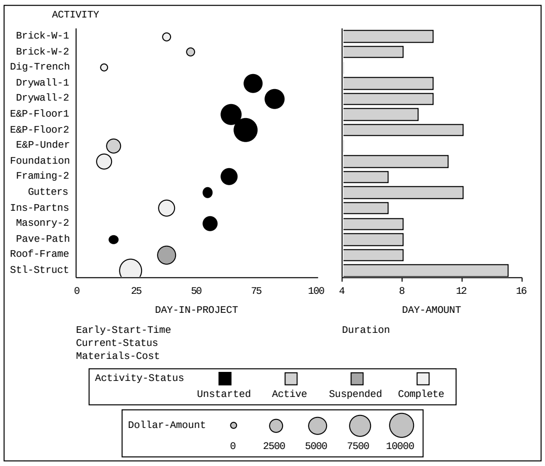
The SURF-HIPPO Neuron Simulator is a circuit simulation package for investigating morphometrically and biophysically detailed models of single neurons and small networks of neurons. SURF-HIPPO allows ready construction of multiple cells from various file formats, which can describe complicated dendritic trees in 3-space with distributed non-linearities and synaptic contacts between cells. Cell geometries may also be traced from the histology directly on the screen, using the mouse. An extensive user interface is provided, including menus, 3D graphics of dendritic trees, and data plotting. Data files may also be saved for analysis with external tools. A research version of SURF-HIPPO (available by anonymous ftp from ftp.ai.mit.edu [pub/surf-hippo]) is written in LISP, and is configured to run using the public domain CMU Common Lisp and Garnet packages. Our version is compiled for SPARC workstations, and should be easily ported to other UNIX machines running X. LISP is a useful simulator language because it has the benefits of a powerful interpreted script language, but it may also be compiled. Thus it is convenient to integrate custom code into SURF-HIPPO. The simulator may also be used with a minimum of programming expertise, if desired.

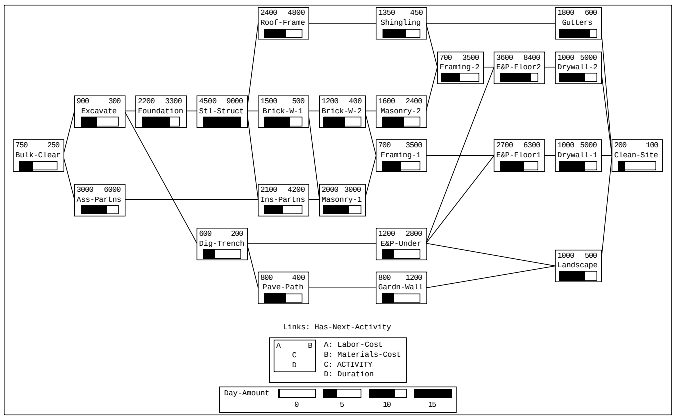
Borg-Graham, L. and Grzywacz, N. M. ‘‘A Model of the Direction Selectivity Circuit in Retina: Transformations by Neurons Singly and in Concert,’’ in *Single Neuron Computation*, edited by T. McKenna, J. Davis, and S. F. Zornetzer. Academic Press, 1992.

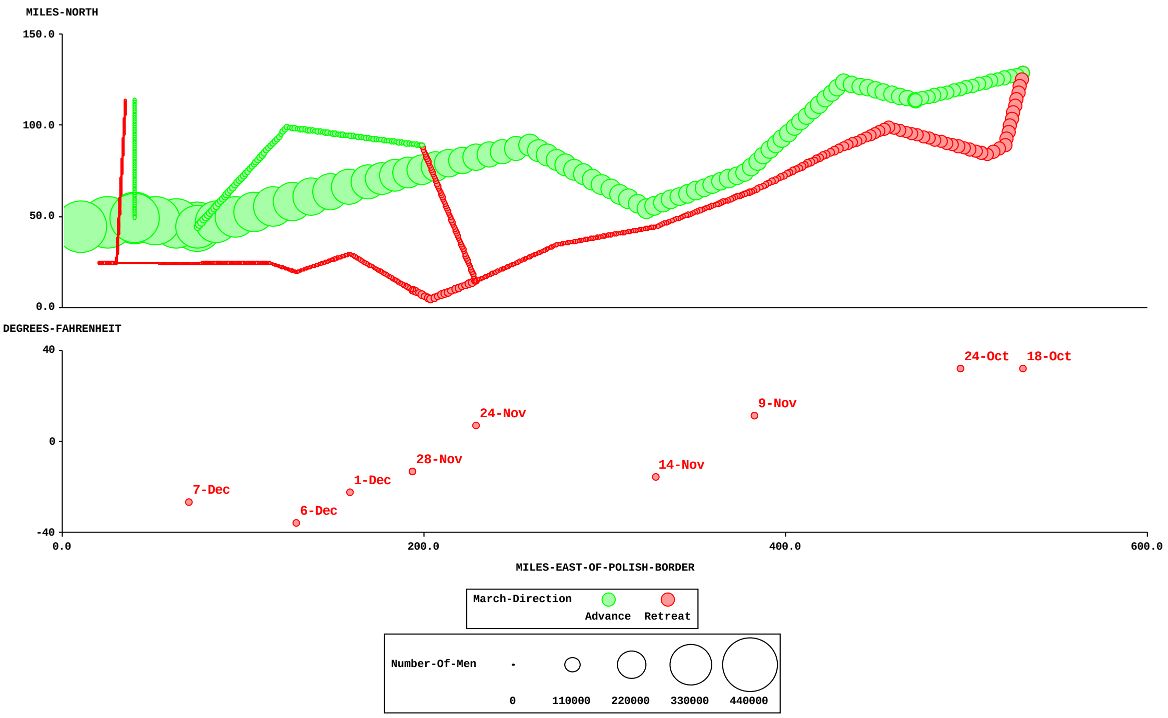


Roderick J. Williams
The University of Leeds, Leeds, LS2 9JT, UK.
GMD (Graphical Mud (Multi-User Domain) Designer)
rodw@cbl.leeds.ac.uk

This application is aimed at supporting the creation of text-based multi-user domains. Current techniques use text-based tools to create these environments, but these tools have very little computer support so complexity and consistency are sacrificed. Our new application supports the graphical creation of MUD areas and enforces topological constraints together with hierarchical grouping of features. The graphical tool can be used in a number of modes which allow the information to be filtered, zoomed and viewed in 2.5 D. Areas created can be printed and additionally they can be saved as native code that can be executed.



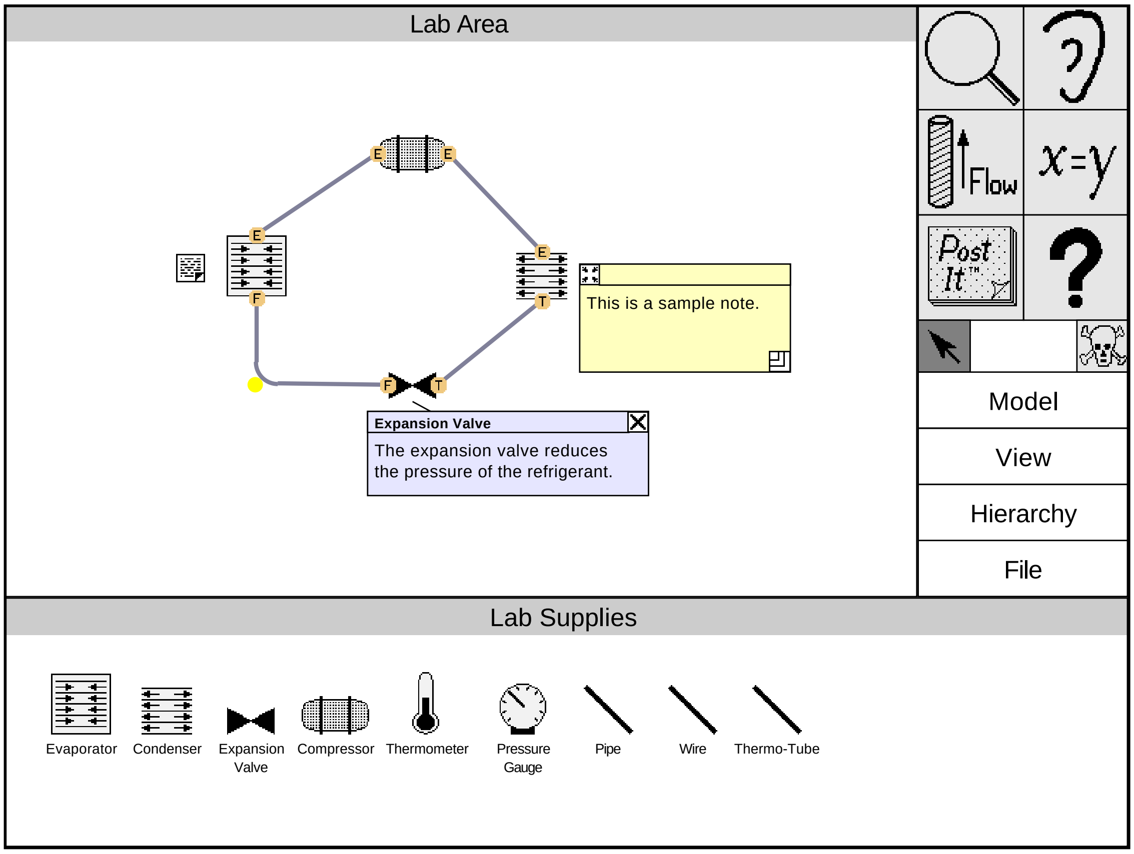




Steven F. Roth
Carnegie Mellon University, Robotics Institute
SAGE
`roth@isl1.ri.cmu.edu`

The SAGE project is developing systems which automate the process of designing presentations of information. An automatic presentation system is an intelligent interface component which receives information from a user or application program and designs a combination of graphics and text that effectively conveys it. It's purpose is to assume as much responsibility for designing displays as required by a user, from layout and color decisions to broader decisions about the types of charts, tables and networks that can be composed within a display. The SAGE project is developing an interactive data exploration environment which contains automatic display design capabilities integrated with data navigation, manipulation and modification tools. These tools are being used to explore large amounts of diverse data from marketing, logistical, real estate, census and other databases.

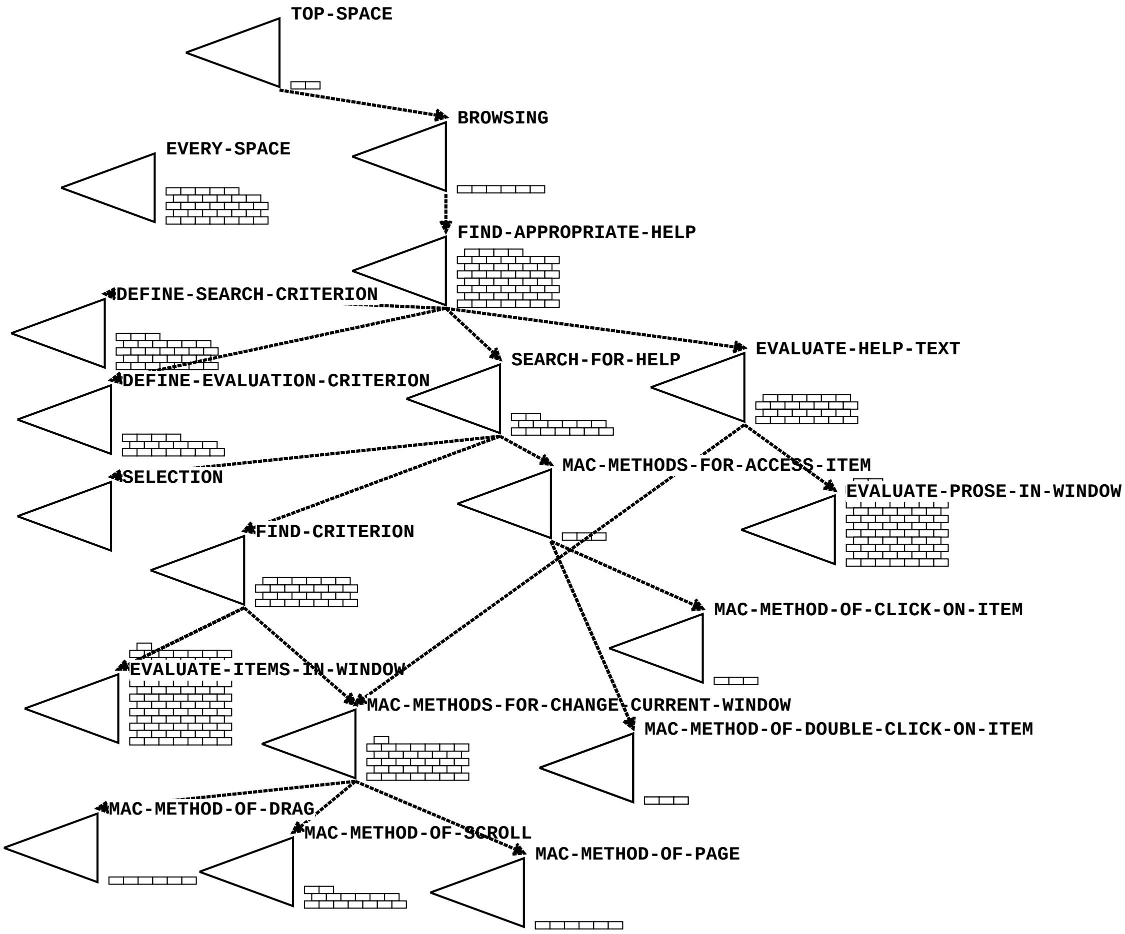
Roth, S.F. & Mattis, J.A. "Data Characterization for Intelligent Graphics Presentation", In *CHI'90: Proceedings of the ACM/SIGCHI Conference on Computer Human Interaction*, Seattle, April, 1990. pages 193-200.



Mike Salisbury
University of Washington, Department of Computer Science
Electronic Encyclopedia Exploratorium
`salisbur@cs.washington.edu`

The Electronic Encyclopedia Exploratorium is an electronic how-things-work book. It allows the user to learn about devices by experimenting with the components of those devices in a lab simulation setting. A causal model simulator lies beneath the user interface which simulates the current device and can provide causal explanations of the results of that simulation. Other high-level tools are planned for future enhancement.

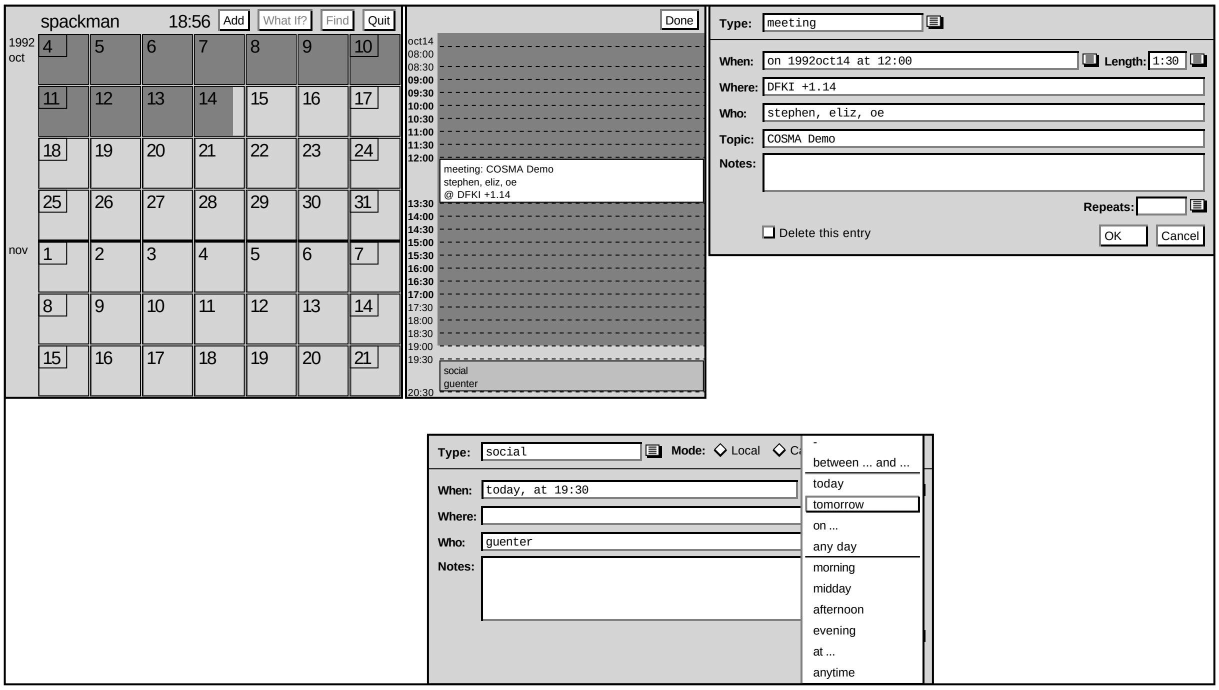
F. G. Amador, D. Berman, A. Borning, T. DeRose, A. Finkelstein, D. Neville, Norge, D. Notkin, D. Salesin, M. Salisbury, J. Sherman, Y. Sun, D. S. Weld, and G. Winkenbach. *Electronic "How Things Work" Articles: A Preliminary Report.* University of Washington, Department of Computer Science and Engineering Technical Report 92-04-08. June, 1992.



Frank E. Ritter
Department of Psychology, U. of Nottingham
The Developmental Soar Interface
Ritter@psyc.nott.ac.uk

The Developmental Soar Interface provides a graphical and textual interface to observe and modify models (programs) for Soar, an AI programming language that also realizes a unified theory of cognition. Garnet is used to graphically represent Soar's goal stack and internal state, and to help users modify and observe structures in Soar.

Ritter, F. E. (1993) *TBPA: A methodology and software environment for testing process models' sequential predictions with protocols*, PhD thesis, Department of Psychology, Carnegie-Mellon University. Reprinted as techreport CMU-CS-93-101, Carnegie-Mellon University.



Stephen P. Spackman
Projekt DISCO
Deutches Forschungszentrum fuer Kuenstliche Intelligenz GmbH
COSMA, the CoOperative Scheduling Management Agent
`spackman@dfki.uni-sb.de or stephen@acm.org`

The calendar window shows the dark bar of the past sweeping, one pixel each half hour of the day and night, across a horizontal line [not visible in this Postscript image] summarising by its width and height the user's working hours and appointments, tentative and firm. The marginal time tags can be dragged up and down, and it will eventually be possible to type over the top of them to jump to a given time. The datebook window presents an expanded view of time as an infinite tape from which appointment forms can be popped up by pointing or by sweeping out free areas. Most importantly, when arrangements involve several people the system communicates with its peers and with meeting participants by reading and writing email in German; the displays are updated in real time.

The fields of the appointment form are semi-structured: they can be filled in with the help of menus - such as that visible on the lower window - that drop down from the small icons on the right; numeric, date and time values within them can be incremented and decremented directly with mouse buttons; and experienced users can type structured values straight in. Unconstrained German text (the graphic interface will soon be English/French/German trilingual, but the natural language parser and generator speak only German) can also be entered. It is routed to the natural language system for analysis; planned improvements to the pragmatics module will allow you to give up on the structured form completely and type informal questions and instructions into the notes field, as you might for a human secretary who had stepped out of the room.

The work underlying this picture was supported by a research grant, FKZ ITW 9002 0, from the German Bundesministerium fuer Forschung und Technologie to the DFKI project DISCO.

Elizabeth A. Hinkelman and Stephen P. Spackman, ‘‘Abductive Speech Act Recognition, Corporate Agents and the COSMA System,’’ in *Abduction, Beliefs and Context: Proceedings of the second ESPRIT PLUS workshop in computational pragmatics*. W. J. Black and G. Sabah and T. J. Wachtel, eds. Academic Press, 1992.

Table of Contents