



---

## Contents

- Introduction
  - An Example
  - Software
  - Information
- 

## Introduction

**Swan** is a data structure visualization system developed as part of the NSF Educational Infrastructure project. Using **Swan**, a C or C++ program can be annotated to provide views of the data structures used in the program. The **Swan** Annotation Interface Library is designed primarily for ease of use, so that instructors and students can annotate existing programs with relatively little effort, much as a programmer might place print statements in a program to get information to help in debugging. **Swan** is designed to support visualization of graphs, including arrays, lists, trees and general graphs.

Our goal is to provide the viewer with a better understanding of the data structures in a program. One innovative feature of **Swan** is that it allows visualization to be a two-way communication process between

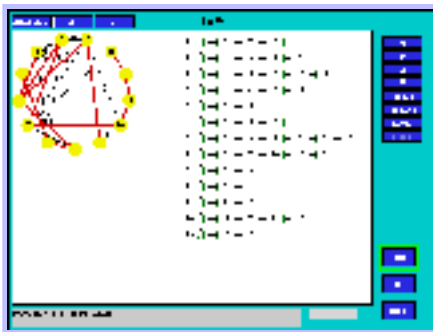
the view and the program. In particular, the algorithm animator can provide viewers with the ability to modify data structures as they proceed through a visualization. In this way, students can experiment with a data structure or algorithm to gain further understanding of its operation.

Several graph layout algorithms are implemented in **Swan**, including special layouts for arrays, lists and trees. Multiple algorithms are available for automatic layout of general graphs. Automatic layout allows the annotator to concentrate on logical structures of views without worrying about their graphical display. **Swan** annotation functions also allow the annotator to give precise specifications for graph layout should this be desired, for example when developing instructional courseware. Thus, the annotator has a range of options in terms of how much effort to put into graph layout.

**Swan** is capable of serving many roles: as a presentation medium for instruction in data structure and algorithm courses, as a graphical debugging aid for students in programming courses, and as a platform for experimenting with various graph layout algorithms.

---

## An Example



Views of MST

The above image shows a **Swan** window generated by an annotated *minimum cost spanning tree* algorithm. There are two views for the input graph. The view on the left shows the logical topology of the graph. This illustrates one of our standard general graph layout

algorithms, with nodes evenly distributed on a circle. The view on the right is an adjacency list representation of the graph, a visualization of the physical implementation used by the annotated program.

After these views are initialized, the viewer can click buttons **STEP** or **RUN** to go through the algorithm. Whenever a new edge is added into the minimum spanning tree, its color changes from blue to red, and it becomes a thick line.

The viewer can modify the topology of the graph by selecting graph editing functions from the **Edit** menu. Nodes and edges can be inserted or deleted from the graph. The views are updated to reflect these changes.

---

## Software

The source code of **Swan** Annotation Interface Library and examples are available for the following systems:

- DECstation with X-Windows

The on-line **Swan User's Manual** is also available.

---

## Information

If you need help or for more information about **Swan** , contact:

Dr. Cliff Shaffer  
Department of Computer Science  
Virginia Tech  
Blacksburg, VA 24061  
(703) 231-4354  
geosim@cs.vt.edu

Please report any problems or bugs, so that we may try to address them in future versions of **Swan**.

---

## Virginia Tech Information

Information about other Internet services at Virginia Tech is available



here.

---

Last updated: April 17, 1995

*Jun Yang* < [yangj@csgrad.cs.vt.edu](mailto:yangj@csgrad.cs.vt.edu) >