

# SoftViz

## Laboratory for Software Visualization

Worcester Polytechnic Institute

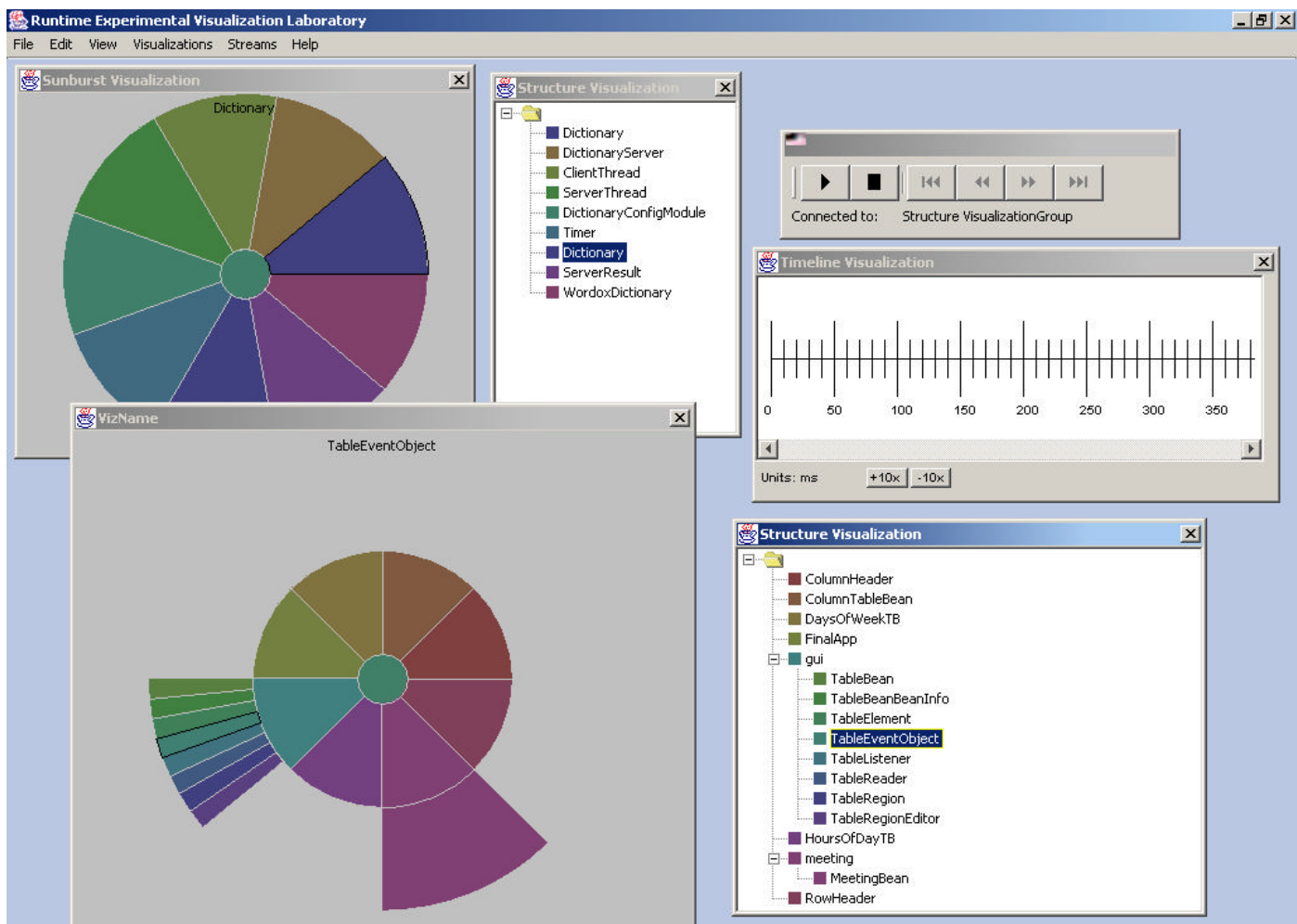
Computer Science Department

Worcester, MA 01609

George T. Heineman ([heineman@cs.wpi.edu](mailto:heineman@cs.wpi.edu))

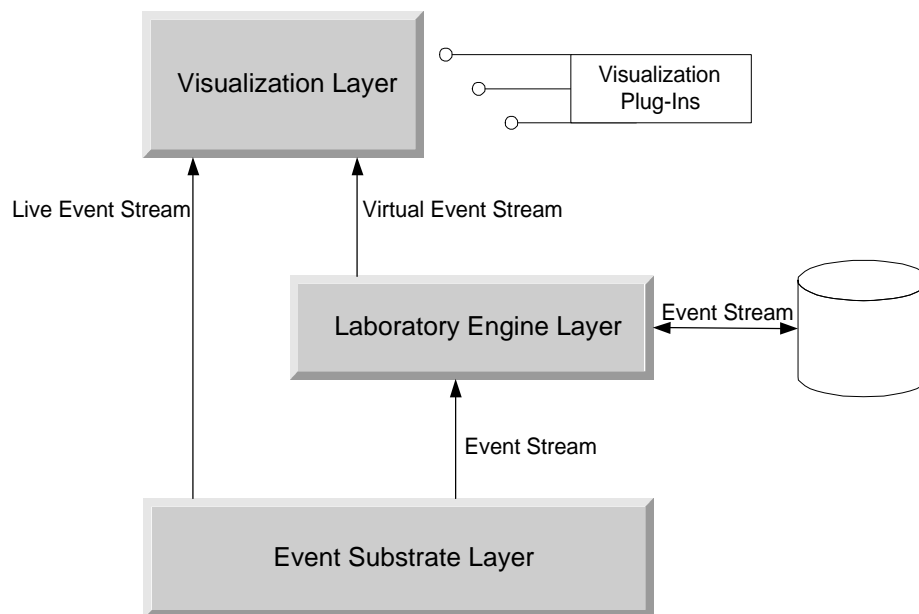
### Overview

Modern software engineers working on large software systems are faced with the daunting task of understanding enough about the structure and behavior of the system to correctly modify the system. Software systems involving thousands of source files and millions of lines of code are becoming commonplace. Armed only with debugger stack traces and a plain-text searching tool such as `grep`, a newly hired developer may need several frustrating and unproductive months to begin to comprehend how the system functions as a whole. In addition, system documentation is almost always a poor reflection of the actual system implementation, since it is not always properly updated when changes occur. What is needed is a tool that provides a user-configurable map of the system that tells the user “You are here.”



## Technical Details

*SoftViz* provides an interactive environment combining visualizations of both the static structure and the dynamic behavior of these complex systems, providing as much or as little information as the user needs. *SoftViz* is an open, layered system composed of an Event Substrate Layer (using DASADA Siena technology), a Laboratory Engine Layer (that manages and stores event traces) and a Visualization Layer (that provides an extensible interface for designing visualizations as plug-ins).



*SoftViz* helps DASADA gauge developers more rapidly construct specific gauges that capture anomalous behavior. Once a target system is instrumented using our companion EXTRACT/AIDE technology, *SoftViz* enables engineers to visualize the active regions of a software application as it executes. These visualizations can be transformed into gauges for locating or troubleshooting large, complex systems.

*SoftViz* is designed for:

- Scalability – target systems may range from simple programs to complex systems with millions of lines of code.
- Portability – by using Java, the environment can run on any platform and allows for multi-language software systems.
- Flexibility – users can perform a wide variety of tasks, including debugging and optimization.
- Extensibility – the environment allows for integration with existing software tools and for the addition of new user-defined visualizations.
- Openness – the environment can work alongside source control systems and integrated development environments.

## Software Specifications

**Platform:** Java 2 (JDK 1.3). Tested on Solaris and Microsoft Windows platforms

**URL:** <http://www.cs.wpi.edu/~heineman/dasada>

**Availability:** *SoftViz* tool now available