

The Future of Software: Visualization+Computation Tools

Ed H. Chi

UIR-R-2000-12

The Future of Software: Visualization+Computation Tools



Ed H. Chi, Research Scientist
User Interface Research Group
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94301
echi@parc.xerox.com

Intro: The complexity of software have continuously risen since the invention of the computer, and while Moore's law predicts the growth in processor speed, it fails to take into account of our ability in managing complex software processes. Our utilization of the increase in processor speed is very much dependent on our ability to manage this complexity.

History: Xerox PARC has been in the forefront of software development for many years, providing industry leadership. Starting with the early software development environments in the Xerox Alto and the Star, the first GUI machines, Mesa, Smalltalk, object-oriented programming, to today's efforts in Aspect-oriented Programming, Information Visualization, and Ubiquitous-Computing in Physical User Interface and information appliances.

Every little bit helps: There has been a number of movements in recent years that helps in the management of software complexity, from the red-hot Open Source Software movement, a common set of API in Java development environments, to Application Service Providers' model of sharing software infrastructure. We can safely say that with the momentum behind these efforts, we will see these techniques to reside for the significant future in everyday software developer's toolbox, because each and every one of these capabilities helps in small but significant ways in combating the bloating of complexity.

Visual Tools will be important: Visualization tools will be increasingly important in combating the complexity in software. This is a clear case where information visualization can significantly increase the communication bandwidth between the computer and the user. Our ability to quickly assimilate information can be vastly improved by visualization techniques. However, for this strategy to succeed, there must be development of intuitive data visualization techniques that users can easily be trained to use. These visual tools will enable software developers to understand complexity, users to quickly monitor large amount of data. The figure below shows an example of visualizing a whole months worth of web server log data along with the changes in the website's structure and content. While it isn't intuitive, it enables users of this system to quickly assimilate large amounts of data in short

period of time. The saving in time is the holy grail in understanding of information.

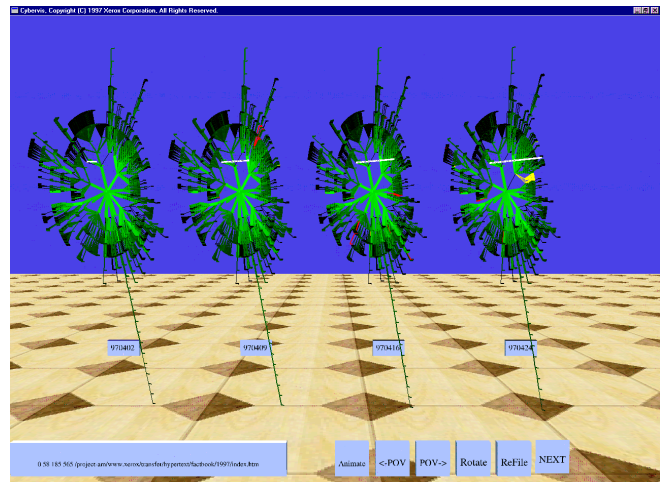


Figure: Our system in which the trends of document accesses are visualized as slices of Disk Trees, with each slice representing a week worth of access log.

Visual Tools+Computation: One of the lessons learned in information visualization is that good visualizations are coupled with good analysis algorithms. We can get the most power out of visualization if we use a sophisticated analysis computation that distills the data further from the raw data. For example, we can process Web server log data files to generate user paths through Web sites. This enables us to extract significant surfing patterns from the usage logs. However, the result from the computation still generates a large amount of data that cannot be analyzed easily without visualization techniques. This is where visualization tools can help in managing this complexity.

Conclusion: The marriage of visualization tools with sophisticated analysis computation will significantly help the usability of software. It can also help in the process of reducing the complexity of software development. This approach is another contribution to the arsenal of tools we can employ to combat complexity in the software of the future.