Views of Storage Allocation in Icon

Ralph E. Griswold

Department of Computer Science, The University of Arizona

Introduction

The first visualization tool written for Icon was MemMon [1], which provides a relatively literal view of how storage is allocated in Icon. MemMon shows memory regions to scale and objects are shown according to their size and in the order in which they are allocated. Different colors are used to distinguish different types of data.

The visualization tool described here, allocview, takes a different approach. It provides a variety of ways for viewing storage allocation that are more metaphorical than literal.

Like MemMon, allocview uses different colors to represent different types. In some views, an allocated object is shown by an area of color that is proportional to its size. In other views, all allocated objects are the same size. Most views do not attempt to show the relative location in memory where allocation occurs. In some views, location is random. In the jargon of the trade, location does not encode information in these views.

The metaphors of the different views are indicated by their names:

blinker	a blinking light
curtain	a beaded curtain
flowers	flowers with petals
haystack	randomly placed straws
mosaic	a mosaic of tiles
nova	an exploding star
phoenix	buildings rising from the ashes
pinwheel	revolving blades
pulsar	a pulsating star
rays	random rays
splatter	random dots on a canvas
tapestry	woven threads
vortex	concentric rings
web	a random cobweb
weed	a randomly growing weed

See the examples at the end of this report.

Running the Visualization Tool

When allocview is run, one of the views listed above can be selected. Multiple views can be obtained using Eve[2].

allocview is run as follows:

allocview [*options*] *program* [*arguments*]

where *program* is the program whose allocation is to be monitored. Standard input and output for *program* can be specified in the usual fashion.

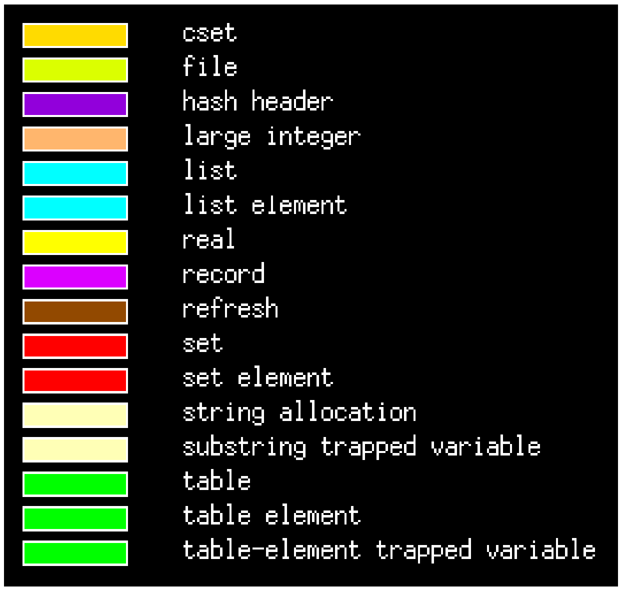
The options available for allocview are:

–v	view, default "pinwheel"
–H	height of view in pixels, default 300
–W	width of view in pixels, default 300
–x	number of columns (where applicable), default 30
–y	number of rows (where applicable), default 30
–s	number of sectors (where applicable), default 240

Colors

In addition to the eight data types for which storage is allocated during program execution*, there are several internal types that also are allocated. For example, a list consists of a list block and one or more list-element blocks. In allocview, the same color is used for such related types.

*Co-expression allocation is not shown in allocview.

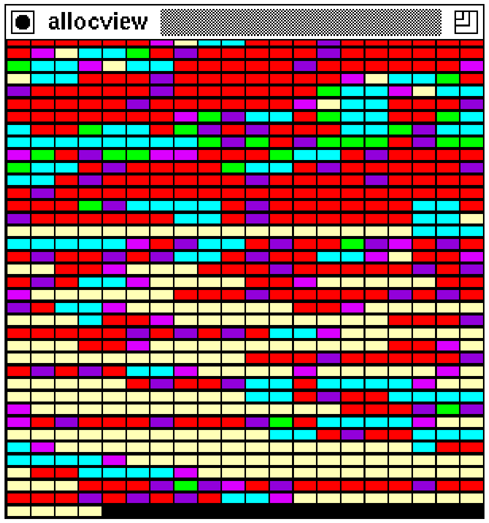


🟨	cset
🟨	file
🟪	hash header
🟧	large integer
🟦	list
🟦	list element
🟨	real
🟪	record
🟫	refresh
🟥	set
🟥	set element
🟨	string allocation
🟨	substring trapped variable
🟩	table
🟩	table element
🟩	table-element trapped variable

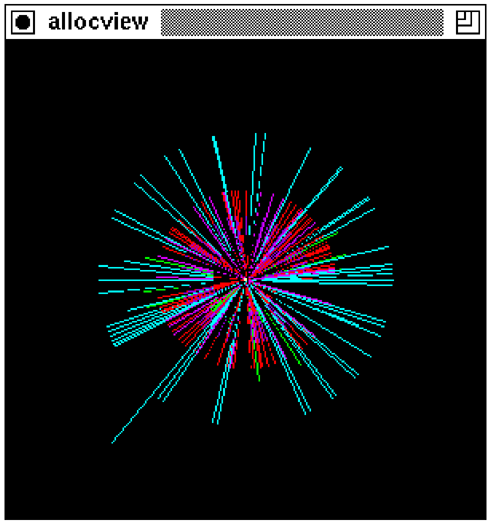
References

1. *The Visualization of Dynamic Memory Management in the Icon Programming Language*, Ralph E. Griswold and Gregg M. Townsend, Technical Report TR 89-30, Department of Computer Science, The University of Arizona, 1989.

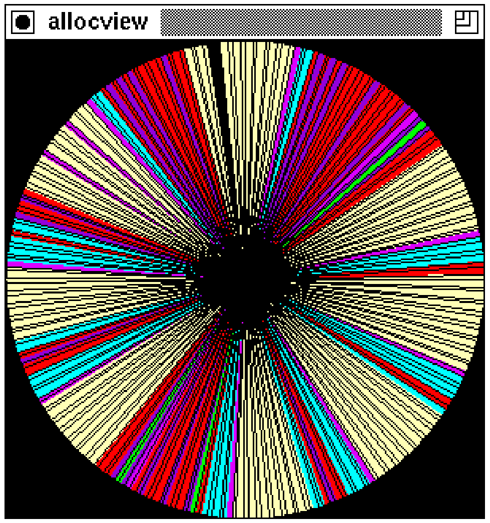
2. *Eve: An Icon Monitor Coordinator*, Clinton L. Jeffery, Icon Project Document IPD179, Department of Computer Science, The University of Arizona, 1992.



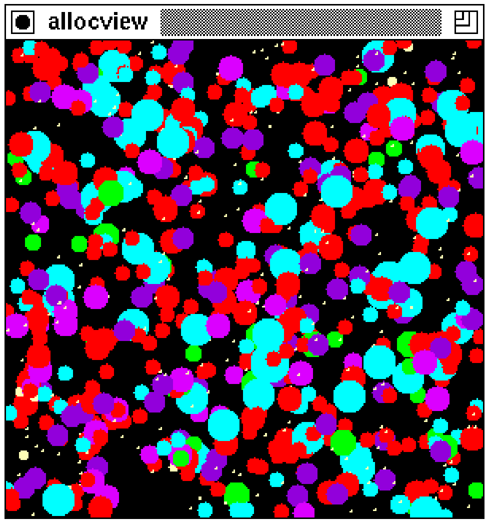
mosaic



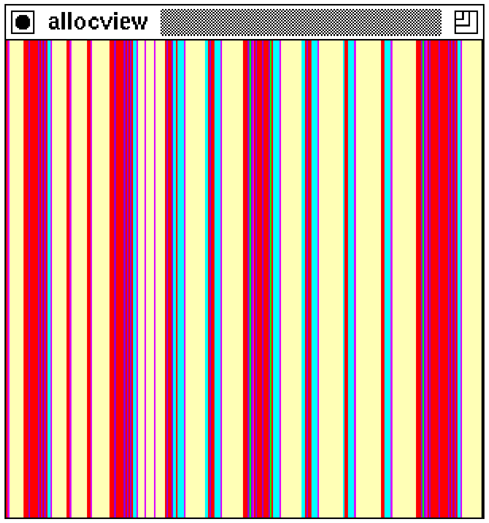
nova



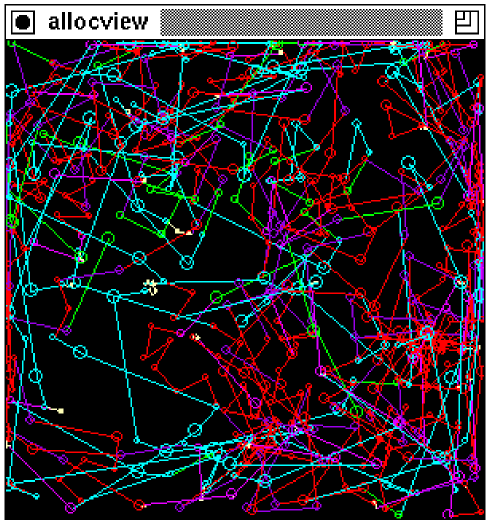
pinwheel



splatter



tapestry



web

Views of Storage Allocation in Icon