Master’s Thesis

Dipl.-Ing. Markus Weninger
Institute for System Software

P +43 732 2468 4361
F +43 732 2468 4345
markus.weninger@jku.at

Secretary:
Birgit Kranzl
Ext. 4341
birgit.kranzl@jku.at

Using Tree Visualizations to Investigate Heap Evolution Over Time

Student:	Lukas Makor, BSc.
Advisor:	Dipl.-Ing. Markus Weninger
Start date:	01.10.2019

Linz, September 9, 2019

AntTracks comprises a modified Java VM based on the Hotspot VM, the *AntTracks VM*, and an offline post-processing analysis tool, the *AntTracks Analyzer*.

The VM's aim is to enable tracking of an application's entire memory lifecycle by writing information about certain events to a trace file. These events include object allocations, object movements by the garbage collector, pointers between the objects and so on.
Such an event trace can then be analyzed in the AntTracks Analyzer. Based on the information parsed from the trace file the tool is able to reconstruct the heap for any garbage collection point.

A reconstructed heap in AntTracks contains various information about every object that has been live at the given point in time: Address, type, allocation site, the address of all referenced objects, and so on. To analyze a heap state, the objects can be arbitrarily grouped using AntTracks’s classification system. For example, all objects could be first grouped by their types, and then grouped by their allocation sites. Such a grouping results in a hierarchical *classification tree*. AntTracks uses this classification system throughout its various analyses and typically visualizes the resulting classification trees in a TreeTableView control.

The goal of this thesis is to integrate a new visualization system into AntTracks that can visualize classification trees using graphical means instead of using a TreeTableView. Since a vast amount of techniques exists to visualize trees, the first part of this work is to study related work. A good source on tree visualizations that have been presented in scientific publications can be found on treevis.net [1]. The student should become familiar with the most common tree visualization techniques such as treemaps, sunburts, icicle plots and variations thereof.
During this study, suitable approaches to visualize heap memory, more specifically AntTracks’s classification trees, should be selected and discussed with the thesis supervisor. A set of at least three different visualization approaches will be defined that will be implemented over the course of this thesis.
In a first step, a prototype of the selected visualization approaches should be implemented to visualize a single heap state. This prototype should then be extended to support basic interaction, for example, selecting a certain node of the tree as new tree root in the visualization. Interactions on one of the visualizations

should be propagated to the other ones, e.g. drilling down on one visualization should also drill down on the other visualizations.

After the visualizations of a single heap state have been successfully implemented, the next step is to extend the visualizations to support the visualization of memory evolution. The memory evolution of an application can be interpreted as a sequence of classification trees, one classification tree per garbage collection point. The user should be able to navigate through time, going from one garbage collection to the next one. The visualizations should be updated accordingly to reflect the heap state of the currently selected garbage collection.

The AntTracks Analyzer is developed as a standalone JavaFX application that also offers to display information through a web interface. Thus, the project should be implemented in HTML and JavaScript, using well-known JavaScript frameworks such as vue.js [2] and d3.js [3, 4].

The thesis should contain a qualitative evaluation in which the visualizations are applied on real-world applications. The evaluation should show how memory anomalies such as memory leaks are reflected in the visualizations.

The final version of the written thesis must be submitted not later than 01.10.2020.

References:

[1] H. Schulz, "Treevis.net: A Tree Visualization Reference," in IEEE Computer Graphics and Applications, vol. 31, no. 6, pp. 11-15, Nov.-Dec. 2011. doi: 10.1109/MCG.2011.103

[2] https://vuejs.org/

[3] https://d3js.org/

[4] https://www.d3indepth.com/layouts/