



Code Maps: A Scalable Visualisation Technique for Large Codebases

Nathan Hawes
June 2013



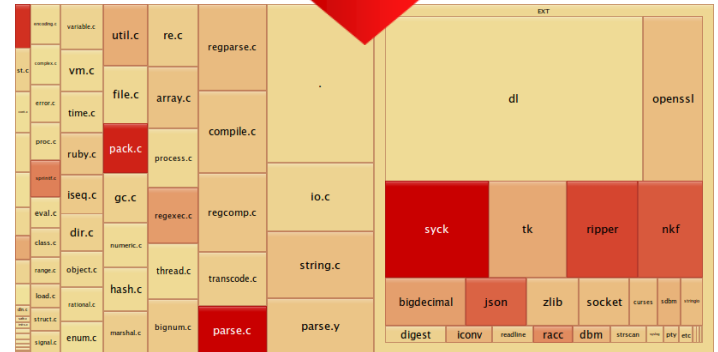
Today's Codebases

- Large, 10^6 to 10^7 lines of code
- Written by hundreds of developers at once
- Often over several decades
- May have little or out-of-date documentation
- Hard to understand, modify and maintain

Software Visualisation

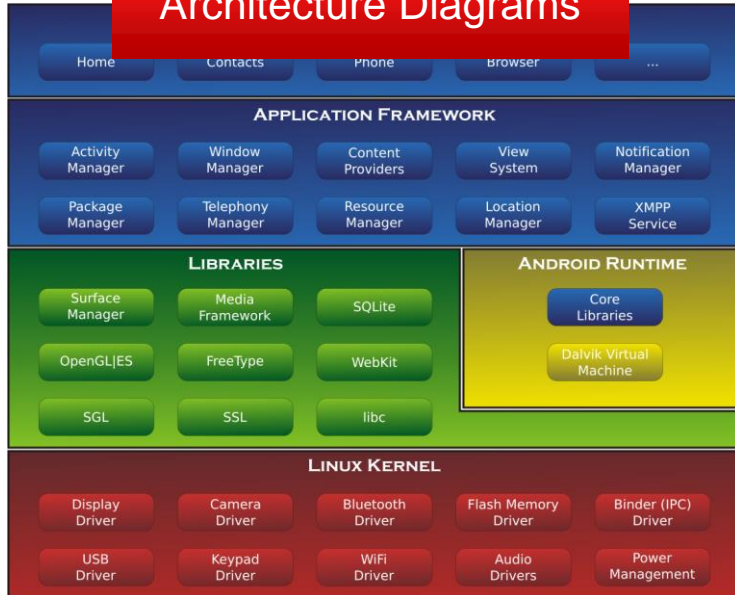
- Visualise program information
 - Bugs, metrics, dependencies, coverage data, etc.
- Denser than text
- We're great at pattern recognition
- Allows for visual data mining

```
Source => /home/parfait/vm_method.c
Function => rb_mod_public_method
LineCount => 2
Length => 9
Vocabulary => 3
Volume => 9
Level => 0
Effort => 0
Cyclomatic => 1
Operators => 3
Nesting => 1
DefUseChains => 0
Source => /home/parfait/vm_method.c
Function => rb_mod_public_method_defined
LineCount => 7
Length => 24
```



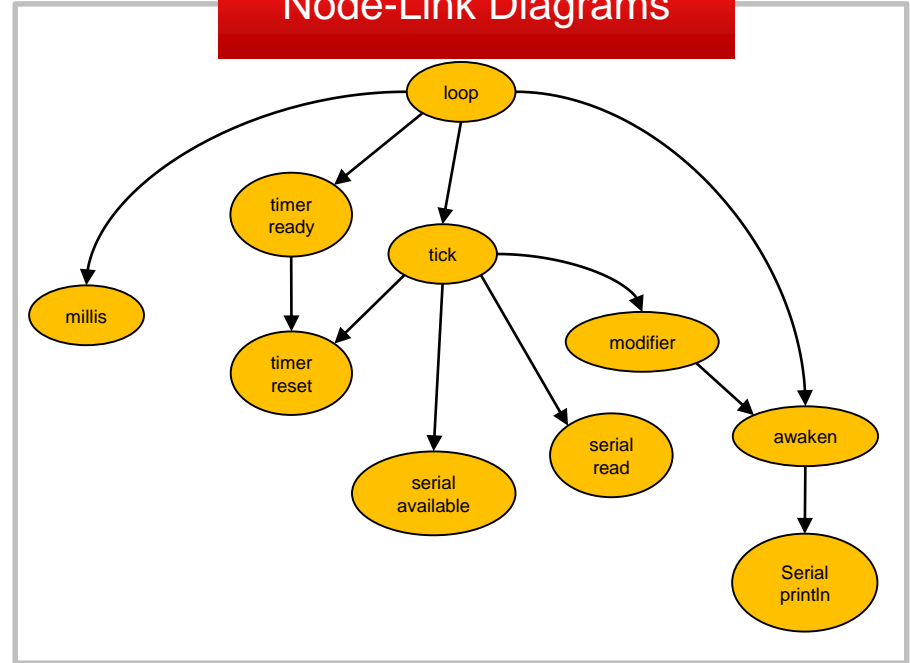
Widely Used Visualisations

Architecture Diagrams

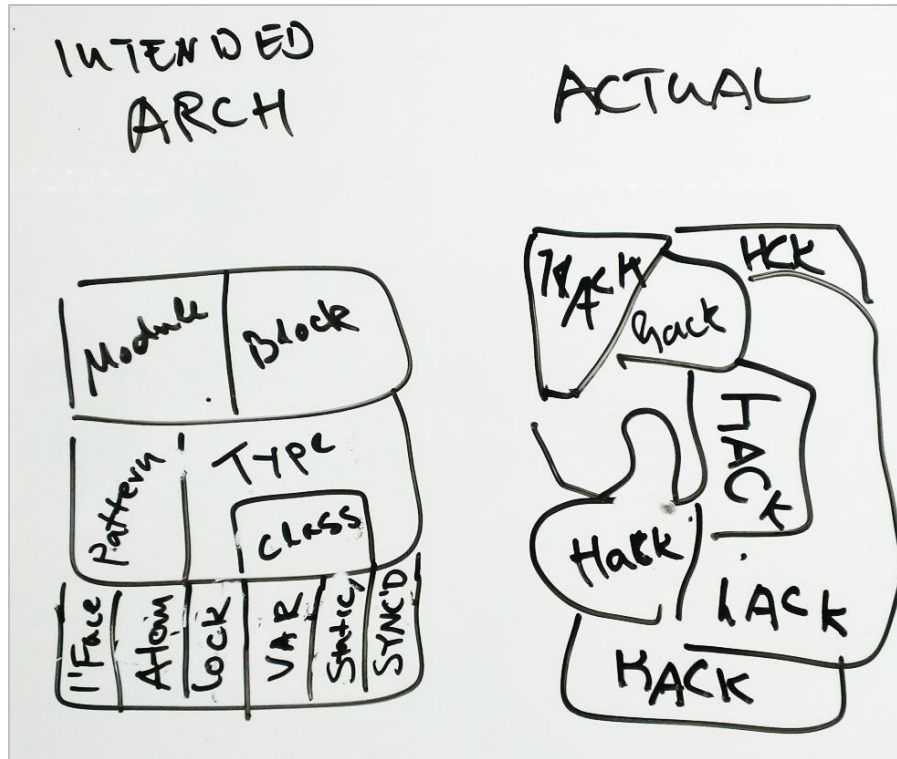


By Smieh (This file was derived from: Sample.svg) [CC-BY-SA-3.0 (<http://creativecommons.org/licenses/by-sa/3.0>)], via Wikimedia Commons

Node-Link Diagrams

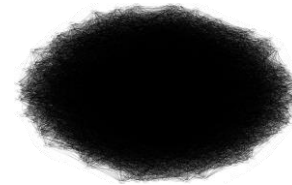
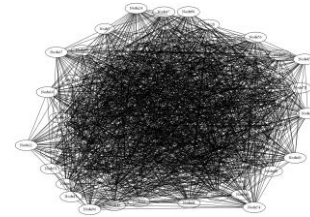
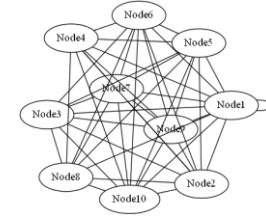
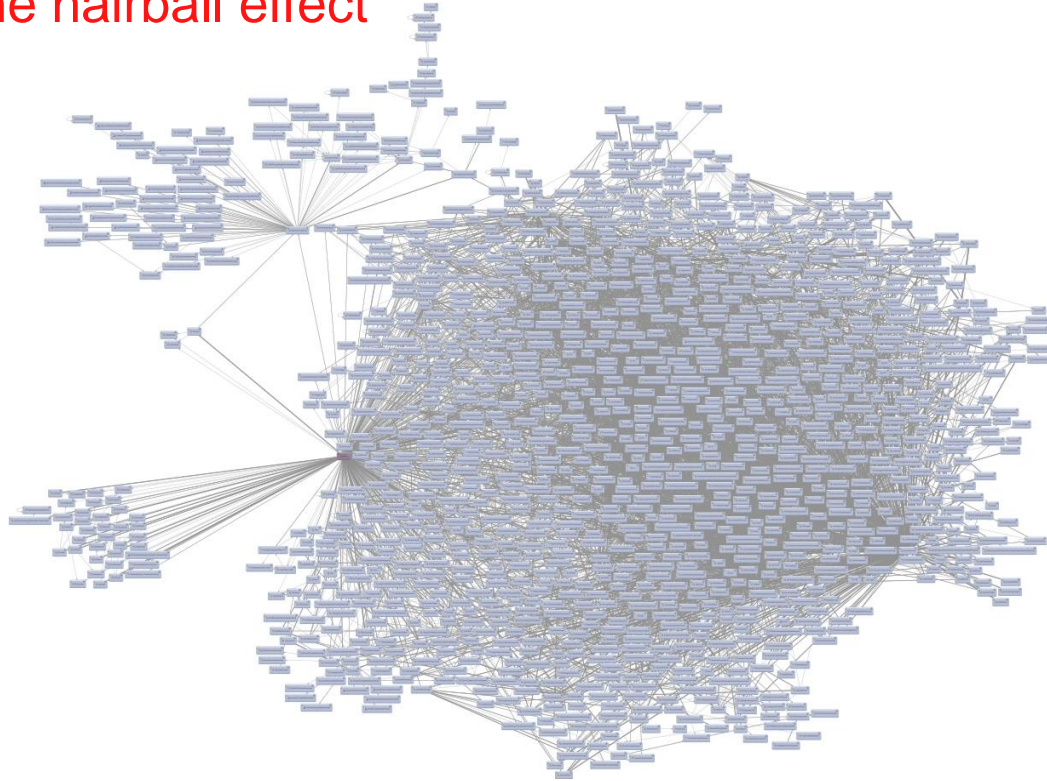


Issues with Larger Codebases



Issues with Larger Codebases

The hairball effect



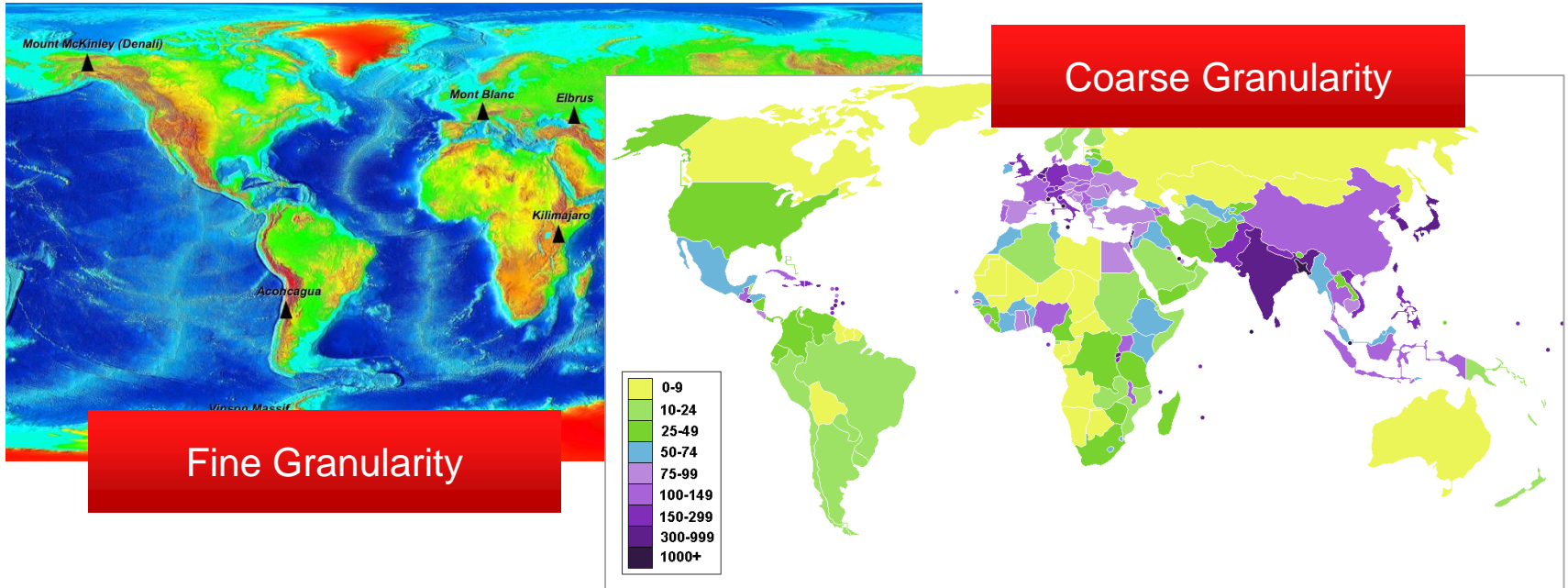
Code Maps

A world map metaphor

- Continent/country/state/city → module/package/class/method
- Easy to stay oriented
 - Distinctive shapes and positions serve as landmarks
 - Can develop a sense of direction over time (stable)
 - Intuitive pan/zoom interaction model
- Information implicit in geography
- Can overlay a variety of information

Overlay Metrics

E.g. Cyclomatic complexity



Overlay Flows

E.g. Path to a bug or assert

```
2622. them */
2623.     mxr = (kccrn)kccgmX(&ctxb, KCCDETB5);          /* max
2624.     used record */
2625.     ttsns = (ktsn *)ksmals(mxr * sizeof(ktsn),
2626.     "kpdbcucr:tttsns");
2627.     /* pointer allocated with kghstack_alloc(...)
2628.
2629.     /* look at records in the controlfile until we find the
2630.     one we want */
2631.     for (tsi = 1; tsi <= mxr; tsi++)
2632.     /* <unknown> leaks when tsi > mxr
2633.     {
2634.     kccts ts;
2635.     kcffo_offline_dbf(tafns, (kcvcpg *)0, FALSE, TRUE,
2636.     FALSE, TRUE);
2637.     for(tsi = 0; tsi < ttsncnt; tsi++)
2638.     /* and tsi >= ttsncnt
2639.     {
2640.     ktsn tsn = ttsns[ttsncnt];
2641.     /* Clear out kpdblast entry for this pdb */
2642.     CLRSTRUCT(kpdblast[pdbid]);
2643.     /* Release afn entries in cdb_file$ */
2644.     /* Get rid of undo */
2645.     ktuapun((kcbds *)NULL, (kcocv *)NULL, ucxt, flag, TRUE);
2646.     MEMORY LEAK
2647.     Memory leak of pointer '<unknown>' allocated with kghstack_alloc(...)
```

Overlay Markers

E.g. Search results

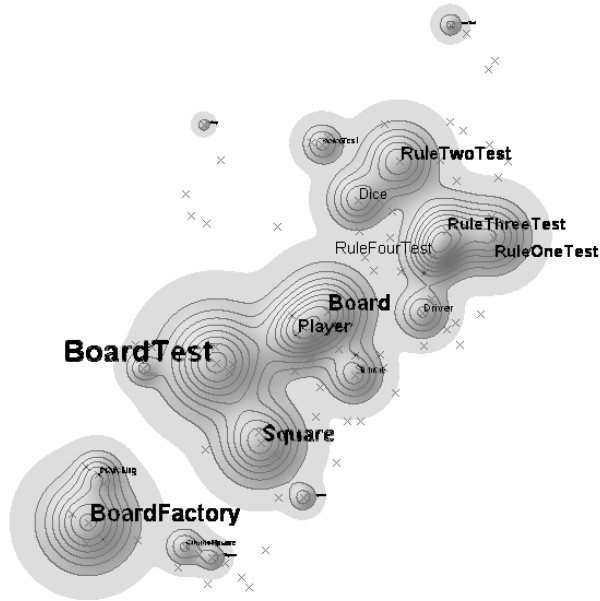
The screenshot displays the Google Maps interface for a search of 'oracle'. On the left, a list of results is shown, each with a red location pin icon and a letter label (A-E). The results include:

- A ORACLE, представительство** - nabrezhnaya Presnenskaya, d.10, Moscow, Russia (+7 495 641-14-00)
- B Oracle Deutschland GmbH** - Thurnithstraße 2, 30519 Hannover, Germany (+49 511 95787 ext. 0)
- C Oracle Polska Sp. z o.o. Oprogramowanie do zarządzania przedsiębiorstwem** - Przyokopowa 31, 01-208 Warszawa, Poland (+48 22 690 87 00)
- D Oracle România S.R.L.** - Calea Floreasca 169A, București, Romania (+40 21 367 8700)
- E Oracle** - Lexington Tower Prestige St, St. John's Woods No. 18, 2nd Cross Road, Chikka Audugodi, S.G. Palya, Sadduguntepalya, Bangalore, India (+91 98453 560 029)

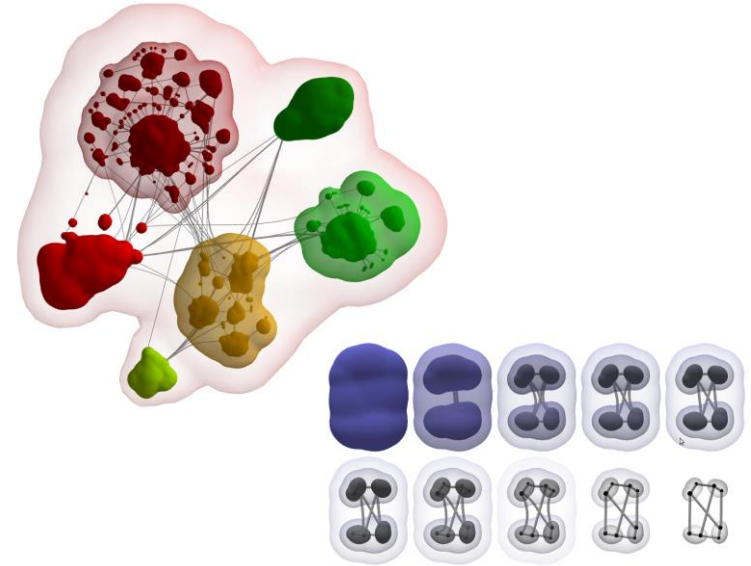
On the right, two map views are shown. The left map is a world map with red location pins overlaid across various continents, corresponding to the search results. The right map is a zoomed-in view of a region in Queensland, Australia, showing several red location pins labeled A through H, with 'A' and 'E' corresponding to the results in the list.

But How?

Influences



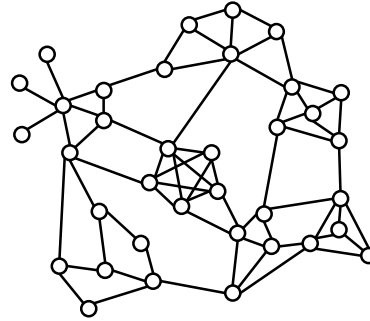
Kuhn, A.; Erni, D.; Loretan, P. & Nierstrasz, O. **Software Cartography: thematic software visualization with consistent layout** J. Softw. Maint. Evol., John Wiley & Sons, Inc., 2010, 22, 191-210



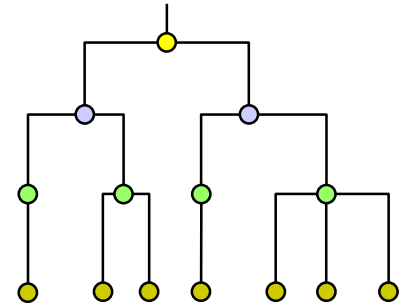
Balzer, M. & Deussen, O. **Level-of-detail visualization of clustered graph layouts** Visualization, 2007. APVIS '07. 2007 6th International Asia-Pacific Symposium on, 2007, 133 -140

Input

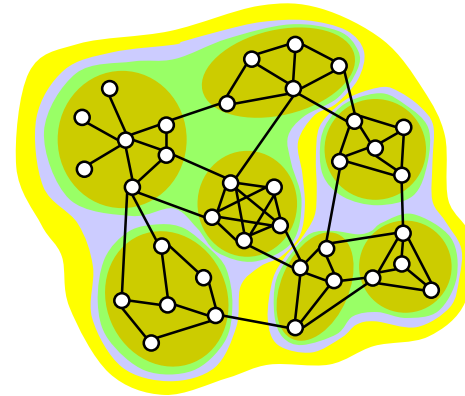
- Dependency graph
 - Any weighted graph
 - Default: callgraph + global variable reads/writes
- Abstraction hierarchy
 - Any tree (e.g. directory/package)
 - Default: derived from dependency graph using clustering algorithm



Dependency Graph



Abstraction Hierarchy

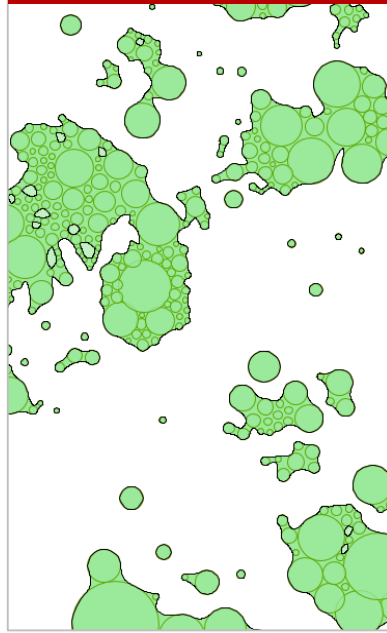


Map Generation

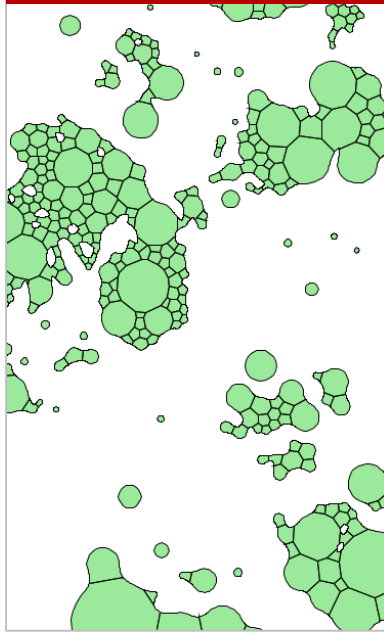
Graph Layout



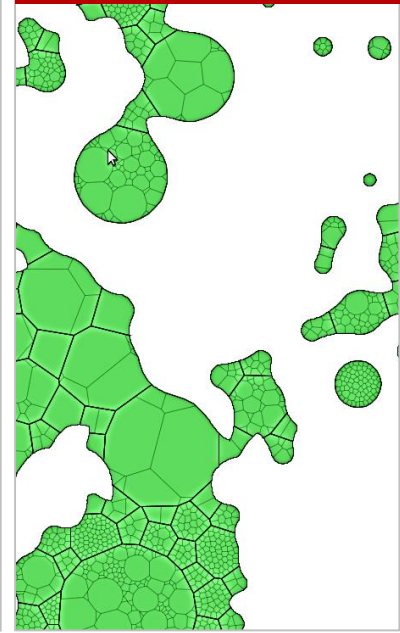
Implicit Surface Generation



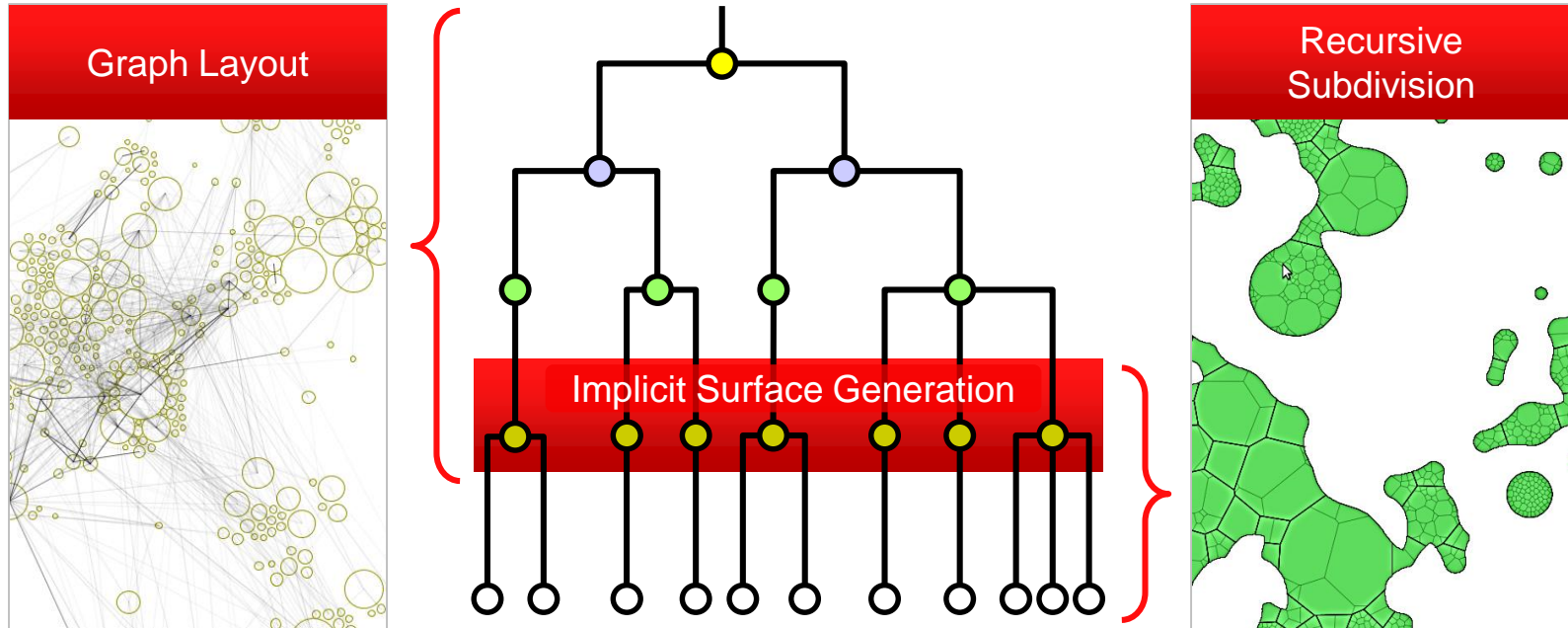
Surface Subdivision



Recursive Subdivision



Map Generation



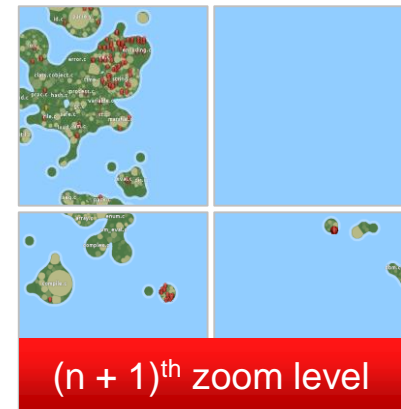
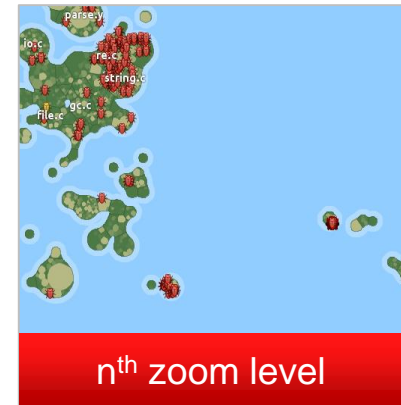
Noack, A. & Lewerentz, C. **A space of layout styles for hierarchical graph models of software systems** Proceedings of the 2005 ACM symposium on Software visualization, ACM, 2005, 155-164

Nocaj, A. & Brandes, U. **Computing Voronoi Treemaps: Faster, Simpler, and Resolution-independent** Computer Graphics Forum, Blackwell Publishing Ltd, 2012, 31, 855-864

Output

Pre-rendered tiles

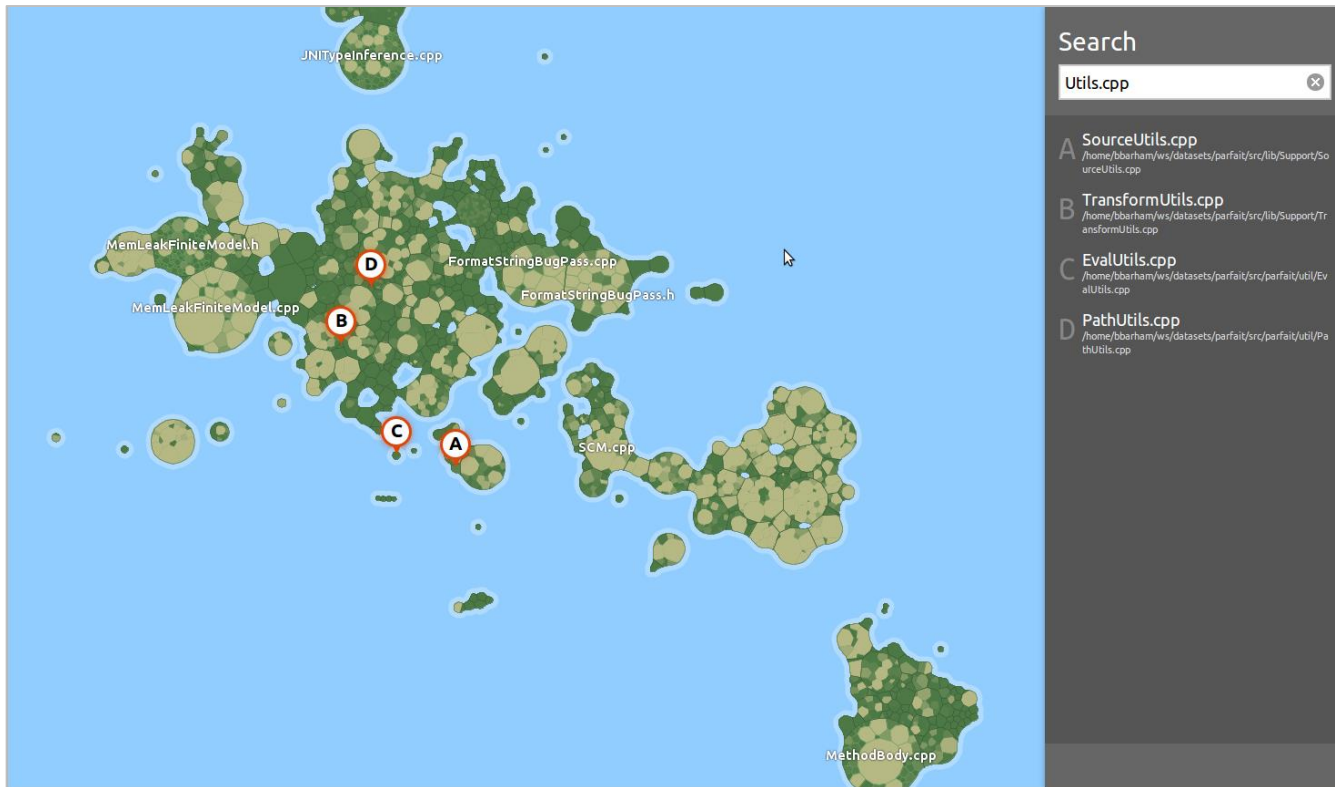
- Position labels
 - Active zoom range
 - Avoid overlaps
- Overlay program information
 - Colour, markers, edges
- Render tiles
 - **Exponential** over number of zoom levels



Performance

	Ruby (C Only)	Solaris ON
Lines of Code	0.5M	10M
Source Files	370	12 976
Functions	9K	210K
Layout	1.18s	602.92s
Surface Generation	0.09s	1.03s
Surface Subdivision	11.5s	121.95s
Rendering (6 levels)	97.25s	1410.85s
Total (without/with render)	15.84s / 113.10s	772.37s / 2183.22s

Demonstration



Our Next Steps

- Ongoing evaluation
- Improve fit to metaphor
 - Crossing water more difficult than land
- Ensure stability
 - Code evolution as continental drift



Further Research

- 3D height (mountains) to display another metric
- Finding clustering algorithms that work well for software
- Adaptive precision floating-point
- Graph layout that more directly optimises for adjacency

ORACLE®