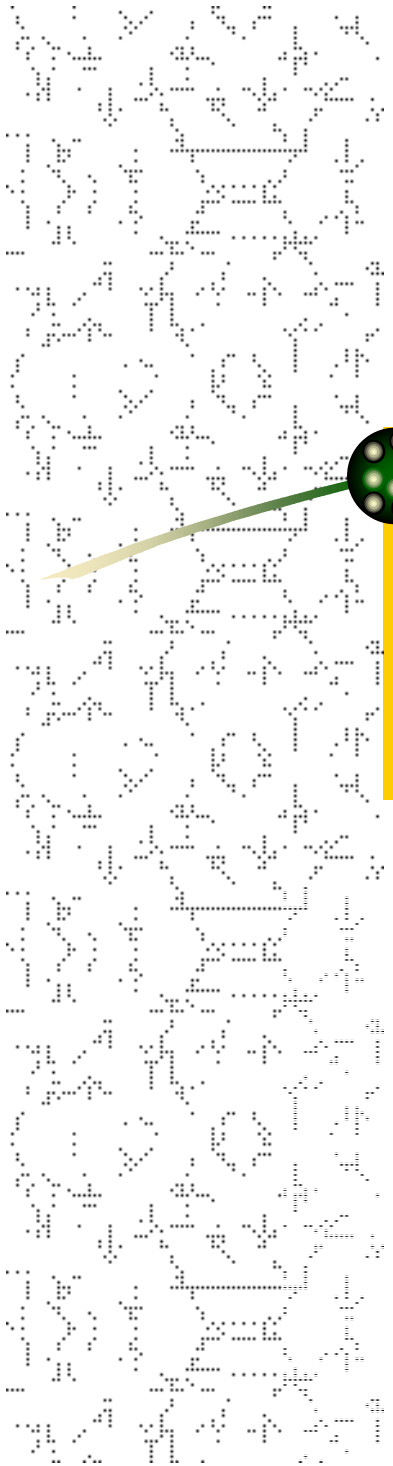


IVA: Visualizing Software Instability

Jennifer Bevan
University of California, Santa Cruz
Jbevan@soe.ucsc.edu

Problem: Software Decay

- Software decays as a result of incompatibilites between the operating environment and the implemented artifact.
  - Failure to meet requirements, specify accurate requirements, or anticipate changes in requirements.
- The existing software architecture can hinder the effectiveness of the maintenance process.
  - “golden handcuffs”, intransigent code.

Hypothesis and Proposal

- Hypothesis: an analysis of historical modification data can identify and classify problematic, high-maintenance software regions.
  - These regions can be described as “instabilities”.
  - Such knowledge can direct software redesign efforts.
- Proposal: IVA, a tool to visualize and analyze software instabilities.
  - The visualization can direct focused analysis.

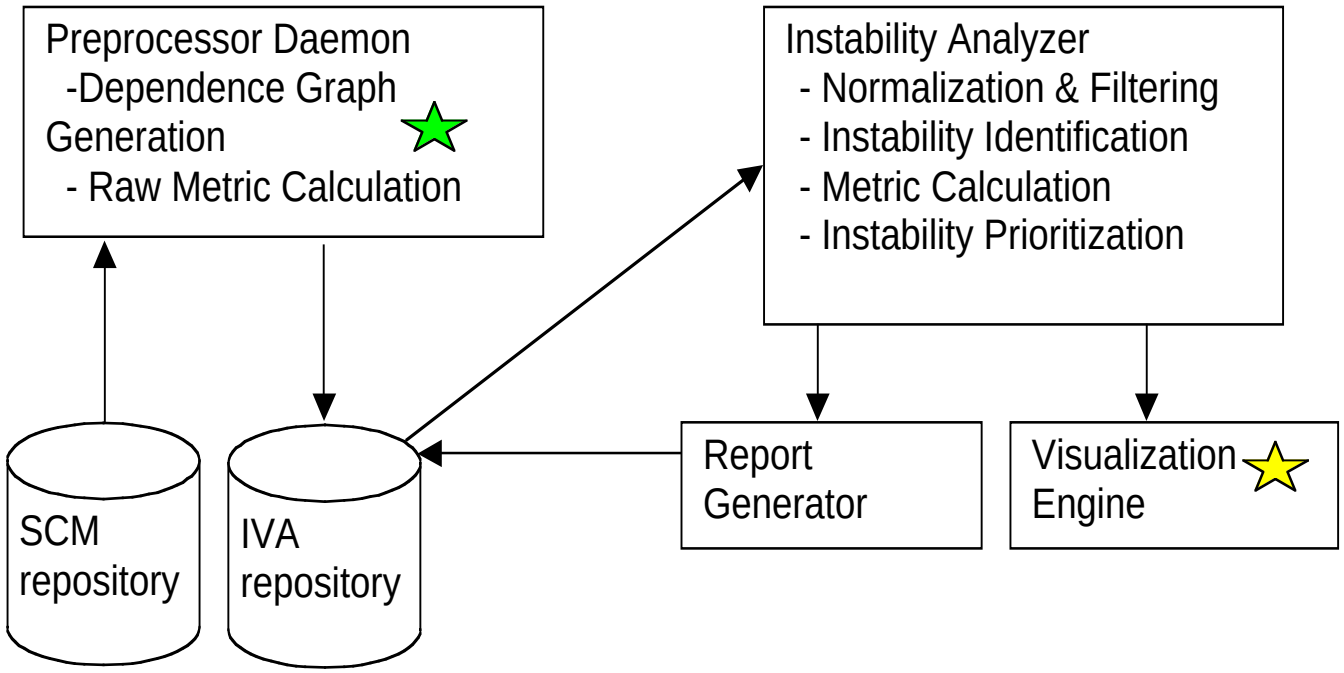
Related Research

- Static software analysis…
  - Uses dependence graphs of a single revision to generate code metrics (cohesion, coupling, complexity) or conduct change impact analyses.
- Software evolution either…
  - Analyzes software modification data to create process-level metrics and models of evolution.
  - Attempts to automatically evolve software.

IVA Is Different Because…

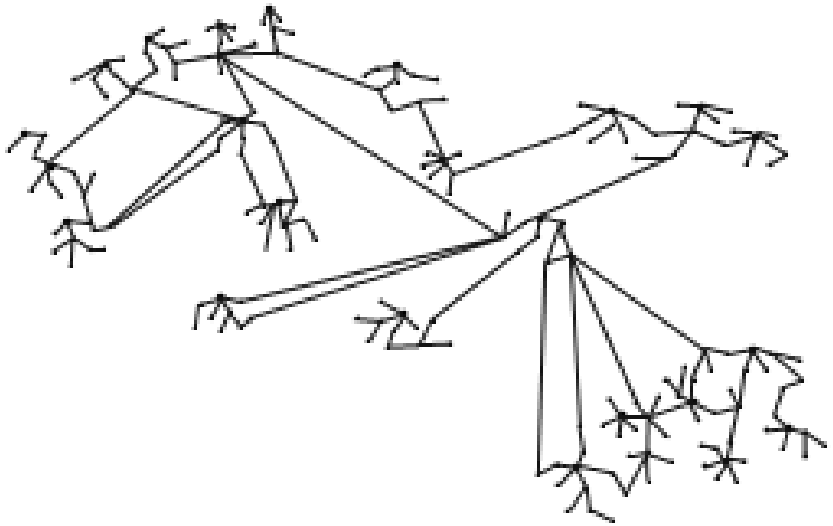
- IVA distinguishes between dependence-related changes and changes made during the same "commit".
- IVA does not require advanced change management data for basic functionality
  - Only requires when, where, what, but not why.
- User controls IVA filtering and aggregating of change data.
  - Different users are interested in different things.

IVA Architecture



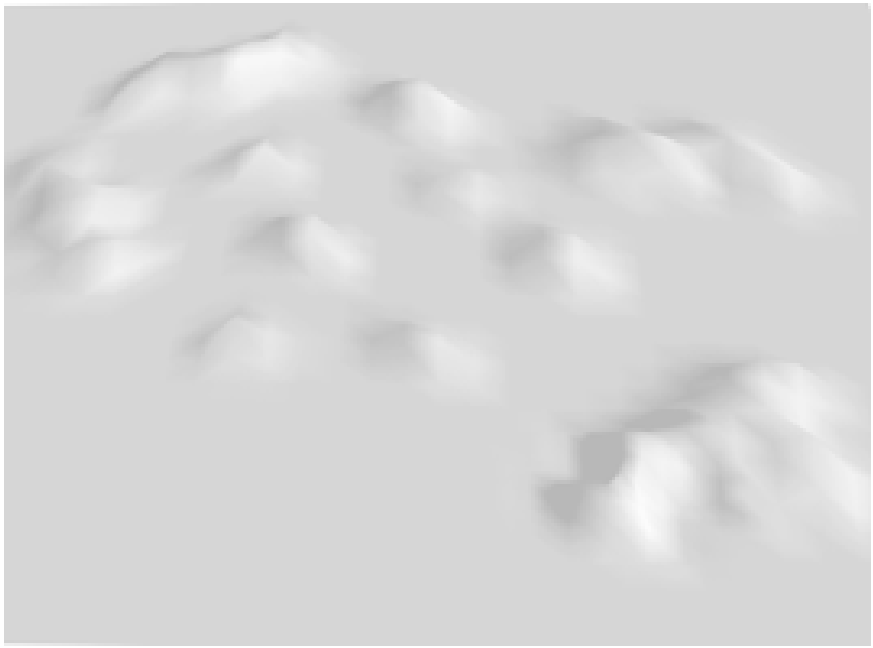
Instability Visualization (1 of 3)

- Dependence graph nodes positioned using heirarchical relationship.
- Causes spatial clustering of related nodes:
  - Package, class, method
  - Directory, file, function



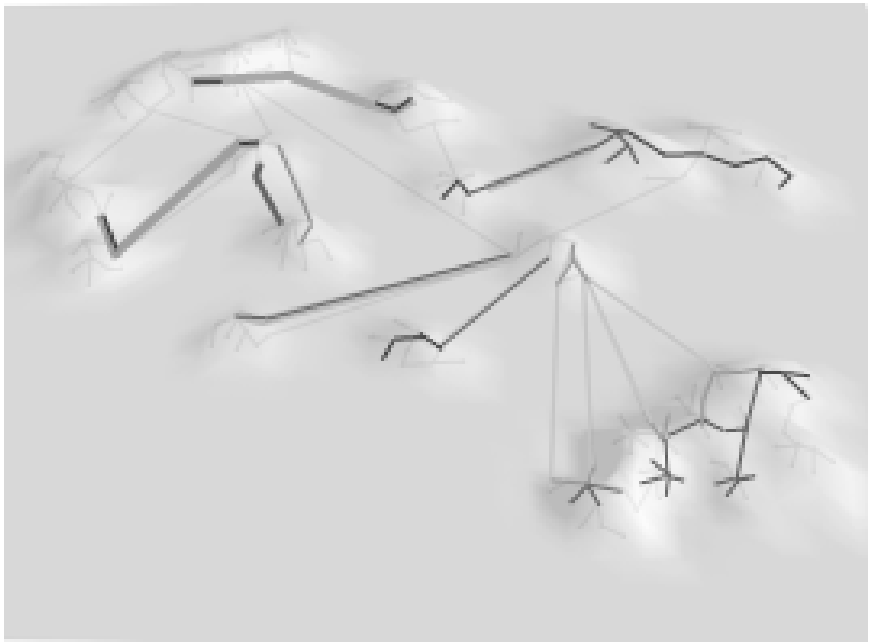
Instability Visualization (2 of 3)

- Surface map generated from dependence graph layout.
- Retains global context of data (code location, etc.)
- Hides edges, reduces clutter.



Instability Visualization (3 of 3)

- Classified instability regions are overlaid on the surface map.
- Instabilities follow edges of underlying dependence graph.
- Color and width denote user-controllable metrics; distance denotes span of coupling.



Use In Collaborative Development

- IVA can analyze and provide feedback on a given implementation of collaborative development.
  - Does task breakdown force contention?
    - Coloration based on number of different committers.
  - Does system architecture force contention?
    - High severity and number of different committers.
  - User can control visualization by directing color, line width, or aggregation algorithms.

Conclusion

- IVA will leverage the data stored in change control systems (CVS as a minimum) by identifying and classifying historical change patterns.
- A proof-of-concept IVA is under construction
  - Will handle Java source code in Subversion repository.
  - Will provide additional visualizations for in-depth exploration of specific instability regions.

Questions?

- The work completed to date was funded by a 2001 USENIX Student Research Grant.
- See http://www.cse.ucsc.edu/~jbevan for IVA progress and status updates.
- Email jbevan@cse.ucsc.edu with future questions.