What your IDE could do once you understand your code

Questions & Directions

Arie van Deursen
Delft University of Technology
Acknowledgments

• WSE & VISSOFT Organizers

• Members of the TU Delft Software Engineering Research Group, including
  – Ali Mesbah, Bas Cornelissen, Andy Zaidman, Martin Pinzger, Rini van Solingen, Cor-Paul Bezemer, ...

• Additional collaborators, including
  – Leon Moonen, Rainer Koschke, Danny Holten, Jack van Wijk, Bart van Rompaey, ...

• flickr photos from faisco, elizabethsalib, andywon
Today’s Structure

• Part I
  – What’s the problem? Documenting understanding

• Part II
  – Solution direction: Web 2.0?

• Part III
  – Evaluation: Begin with the end in mind

• Part IV
  – Conclusions: Towards a Knowledgeable IDE
M. Marin, A. van Deursen, and L. Moonen.
Identifying Crosscutting Concerns using Fan-in Analysis.

Program Understanding

• “40% of developing time is spent on program understanding”

• Probably more accurately:
  – 1% on understanding
  – 39% is spent on misunderstanding

• Should we tell the IDE the difference?
  – What could the IDE do with that knowledge?
Part II
The Great Move to the Web
Wave.google.com
Collaborative Spreadsheets

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Representative</td>
<td>State</td>
<td>District</td>
<td>Phone</td>
<td>Room</td>
</tr>
<tr>
<td>2.</td>
<td>Abercrombie, Neil</td>
<td>HI</td>
<td>1st</td>
<td>225-2726</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Ackerman, Gary L.</td>
<td>NY</td>
<td>5th</td>
<td>225-2601</td>
<td>2243</td>
</tr>
<tr>
<td>4.</td>
<td>Aderholt, Robert B.</td>
<td>AL</td>
<td>4th</td>
<td>225-4876</td>
<td>1433</td>
</tr>
<tr>
<td>5.</td>
<td>Akin, W. Todd</td>
<td>MO</td>
<td>2nd</td>
<td>225-2561</td>
<td>117</td>
</tr>
<tr>
<td>6.</td>
<td>Alexander, Rodney</td>
<td>LA</td>
<td>5th</td>
<td>225-8490</td>
<td>316</td>
</tr>
<tr>
<td>7.</td>
<td>Allen, Thomas H</td>
<td>ME</td>
<td>1st</td>
<td>225-6116</td>
<td>1127</td>
</tr>
<tr>
<td>8.</td>
<td>Andrews, Robert E.</td>
<td>NJ</td>
<td>1st</td>
<td>225-6501</td>
<td>2439</td>
</tr>
<tr>
<td>9.</td>
<td>Baca, Joe</td>
<td>CA</td>
<td>43rd</td>
<td>225-6161</td>
<td>328</td>
</tr>
<tr>
<td>10.</td>
<td>Bachus, Spencer</td>
<td>AL</td>
<td>6th</td>
<td>225-4921</td>
<td>442</td>
</tr>
<tr>
<td>11.</td>
<td>Baird, Brian</td>
<td>WA</td>
<td>3rd</td>
<td>225-3536</td>
<td>1421</td>
</tr>
<tr>
<td>13.</td>
<td>Baldwin, Tammy</td>
<td>WI</td>
<td>2nd</td>
<td>225-2906</td>
<td>1022</td>
</tr>
<tr>
<td>14.</td>
<td>Barrett, J. Gresham</td>
<td>SC</td>
<td>3rd</td>
<td>225-5301</td>
<td>1523</td>
</tr>
<tr>
<td>15.</td>
<td>Barrow, John</td>
<td>GA</td>
<td>12th</td>
<td>225-2823</td>
<td>226</td>
</tr>
<tr>
<td>16.</td>
<td>Bartlett, Roscoe G.</td>
<td>MD</td>
<td>6th</td>
<td>225-2721</td>
<td>2412</td>
</tr>
<tr>
<td>17.</td>
<td>Barton, Joe</td>
<td>TX</td>
<td>6th</td>
<td>225-2002</td>
<td>2109</td>
</tr>
<tr>
<td>18.</td>
<td>Bass, Charles F.</td>
<td>NH</td>
<td>2nd</td>
<td>225-5206</td>
<td>2421</td>
</tr>
<tr>
<td>20.</td>
<td>Beauprez, Bob</td>
<td>CO</td>
<td>7th</td>
<td>225-2645</td>
<td>504</td>
</tr>
<tr>
<td>22.</td>
<td>Berkley, Shelly</td>
<td>NV</td>
<td>1st</td>
<td>225-5965</td>
<td>439</td>
</tr>
</tbody>
</table>
Web-Based Accounting Software
“Why I’m Done Making Desktop Applications”

- Better *conversion*
- Easier to support
- The end of piracy
- Interaction analytics
- Customization per user
- Short innovation cycles

Patrick Mckenzie

Google Chrome Frame

- Applications such as Google Waves require ultrafast Javascript, and HTML5 (canvas)
  - Not supported by IE6, IE7, IE8
- Released this Tuesday: Google Chrome Frame
  - Plugin for Internet Explorer
  - V8 Javascript Engine
  - HTML5 support
  - Essentially runs Chrome within IE
Web 2.0

• **Technology paradigm:**
  – Ajax: Asynchronous JavaScript and XML

• **Usage paradigm**
  – Architecture of participation

• **An attitude:**
  – Processes and tools are socially open
  – Content can be used in several different contexts

• *Self-organizing means for informal communication*
Twitter

What are you doing?

Latest: RT @taoxiease: Check out the #CSM 2009 presentation slides at https://sites.google.com/site/... (more to be added along ... about 16 hours ago

Home

aademonkeyrock RT @berkowitz: MYOB develops CRM system for small business- http://is.gd/3DI07 #crm
3 minutes ago from TweetDeck

grammarware @jurgenvinju thanks, I've just read your comment re-evaluating to check if implemented architecture is drifting away from the intended one.
12 minutes ago from TweetDeck in reply to jurgenvinju

grammarware @PaulKlint thanks. I didn't know that! The userid does indeed prove it is him. Active or not, being registered is enough for us to mention.
19 minutes ago from TweetDeck in reply to PaulKlint

dalmaer First Microsoft reaction to Chrome Frame. I think they're right. http://bit.ly/CrqsJ (via @johnolilby)
20 minutes ago from Tweetie

jurgenvinju @grammarware I have added my 5 cents to your discussion. Good topic.
20 minutes ago from Adum in reply to grammarware

fbahr Listening to @oopsla2008 interview podcast (episode 9) with @crazybob on Google #Guice and #GuiceBerry http://bit.ly/6aCVP #DI
21 minutes ago from web
Micro-blogging Using Twitter

• Short updates on what you’re doing
• Visible to all
• Follow friends, or be followed
• ReTweet tweets you like
• Respond to tweets you like
• “Trending” topics

• Follow what’s hot in your area of interest
• Follow what people are doing
Recent Questions

- Increment digit value in String
  - Votes: 0
  - Answer: 1
  - Views: 16
  - Tags: java, regex
  - Time: 8s ago

- PostgreSQL with NHibernate problem with LIKE and parameter.
  - Votes: 0
  - Answers: 0
  - Views: 0
  - Tags: nhibernate, postgres
  - Time: 29s ago

- How to check that a string is a palindrome using regular expressions?
  - Votes: 9
  - Answers: 13
  - Views: 2k
  - Tags: regex, palindrome, interview-questions
  - Time: 41s ago

- Server-Side Configuration for GWT
  - Votes: 1
  - Answer: 1
  - Views: 34
  - Tags: gwt, configuration
  - Time: 42s ago

- Should a newbie programmer give up the thinking “I must Start with C” ? (as his first language)
  - Votes: 1
  - Answers: 5
  - Views: 49
  - Tags: c, programming-languages, subjective, discussion
  - Time: 1m ago

- Creating an array of zero width and zero height!?
  - Votes: 2
  - Answers: 5
  - Views: 149
  - Tags: array, c++, constructor, class, homework
  - Time: 1m ago

- I have never seen a software be released as jar, so how can I make a jar of a software?
  - Votes: 1
  - Answers: 1
  - Views: 12
  - Time: 1m ago
That was an interview question that I was unable to answer:

How to check that a string is a palindrome using regular expressions?

p.s. There is already a question "How to check if the given string is palindrome?" and it gives a lot of answers in different languages, but no answer that uses regular expressions.

The answer to this question is that "it is impossible". More specifically, the interviewer is wondering if you paid attention in your computational theory class.

In your computational theory class you learned about finite state machines. A finite state machine is composed of nodes and edges. Each edge is annotated with a letter from a finite alphabet. One or more nodes are special "accepting" nodes and one node is the "start" node. As each letter is read from a given word we traverse the given edge in the machine. If we end up in an accepting state then we say that the machine "accepts" that word.

A regular expression can always be translated into an equivalent finite state machine. That is, one that...
Ask questions, provide answers, vote for the things you find helpful — and Stack Overflow will bestow badges upon you. Here's a list of all the badges, along with a count of how many users have earned each one so far:

- **Autobiographer** × 17314  
  Completed all user profile fields
- **Beta** × 2617  
  Actively participated in the Stack Overflow private beta
- **Citizen Patrol** × 3560  
  First flagged post
- **Civic Duty** × 2603  
  Voted 300 times
- **Cleanup** × 2433  
  First rollback
- **Commentator** × 15444  
  Left 10 comments
- **Critic** × 15646  
  First down vote
- **Disciplined** × 1039  
  Deleted own post with 3 or more upvotes
- **Editor** × 33580  
  First edit
- **Enlightened** × 5963  
  First answer was accepted with at least 10 up votes
- **woot!** × 3201  
  Visited the site each day for 30 days
- **Famous Question** × 145  
  Asked a question with 10,000 views
Stackoverflow Mechanisms

- Voting on questions & answers
- Spotting similar questions
- Adding comments
- Starring questions
- Picking the right answer
- Earning reputation points
- Earning badges
Will the IDE go on line too?

• Mozilla Bespin:
  – HTML5 Canvas-based editing

• Heroku:
  – Ruby-on-Rails deployment (+ IDE?)

• Eclipse E4 project plans include:
  – "Eclipse in browser"; DOM-based access to workbench; REST-full IDE services

• WWWorkspace
  – Browser-access to Eclipse OSGi services
public class SettingsServlet extends HttpServlet {
    private File users;
    static final long serialVersionUID = 1;

    public SettingsServlet() {
        users = Activator.bundleContext.getDataFile("users");
        users.mkdirs();
    }

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException
    String pathInfo = req.getRathInfo();

    3 problem(s) found in src/org/eclipse/e4/bespin/server/SettingsServlet.java.
    x line 23: The method getRathInfo() is undefined for the type HttpServletRequest
    ! line 88: The declared exception ServletException is not actually thrown by the method doDelete(HttpServletRequest, HttpServletResponse) from type SettingsServlet
    ! line 88: The declared exception IOException is not actually thrown by the method doDelete(HttpServletRequest, HttpServletResponse) from type SettingsServlet
Mixing all of these ....
... would it then be possible to...

• Tell the IDE simply what you’re doing
• Follow the activities of your co-developers
• Hit the “YES” button to tell the IDE that you understand
• Learn from the IDE which paths others took to understand their code
• Appreciate the value of comments by seeing the author’s reputation
... or...

- Compose your IDE plugins the way you select widgets in, e.g., pageflakes?
- Share ... execution traces?
- Share ... breakpoints?
- Do ... joint editing?
- Share ... configurations and workspaces?
- Identify (local) “trending topics” in current development?
With that in place...

• The IDE just ... *knows*
  – It *learns* from interactions,
  – and encourages the developers to add information at any time

• At moments of enlightenment
  – The developer *knows*
  – and is forced by the IDE to share his/her understanding
  – which is merged with stored interaction information
Many Starting Points Exist ...

- IBM Jazz, MS Team Foundation Server, Bespin
- DeLine: Del.icio.us development tools (CHASE’08)
- Begel et al: Codebook (ICSE NIER’09)
- Treude & Storey: Tagging (ICSE’09)
- Shihab et al: IRC channels (MSR’09)
- Fritz/Murphy: Interaction mining (ESEC/FSE’07)
- .... and a lot more!

- .... But still much work remains to be done!
Part III
Begin With the End in Mind

• J. Ziman, 1978: *Reliable Knowledge: An Exploration of the Grounds for Belief in Science*

• **Consensual knowledge:**
  – The ultimate goal of science is a consensus of rational opinion

• Consensual statement:
  – Fully tested and accepted by overwhelming majority of well-informed scientists

• Consensible statement:
  – Has the potential for eventually contributing to a consensus
Comprehension & Dynamic Analysis: A Systematic Survey

• A person **understands** a program:
  – Ability to explain the program, its structure, its behavior, its effects on its operation context, and its relationships to its application domain
  – in terms that are qualitatively different from the tokens used to construct the source code of the program

• **Dynamic analysis:**
  – Properties of running system
Analyzed 172 Papers

- IWPC/ICPC: 33
- WCRE: 26
- CSMR: 20
- ICSM: 19
- SOFTVIS: 11
- JSME: 9
- JSS: 9
- TSE: 5
- OOPSLA: 5
- ASE: 5
- ICSE: 4
- SEKE: 3
- FSE: 3
- CASCON: 3
- Other: 21
<table>
<thead>
<tr>
<th>Facet</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>survey</td>
<td>a survey or comparative evaluation of existing solutions fulfilling a common goal.</td>
</tr>
<tr>
<td></td>
<td>design/arch.</td>
<td>the recovery of high-level designs or architectures.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td>the reconstruction of specific views, e.g., UML sequence diagrams.</td>
</tr>
<tr>
<td></td>
<td>features</td>
<td>the analysis of features, concepts, or concerns, or relating these to source code.</td>
</tr>
<tr>
<td></td>
<td>trace analysis</td>
<td>the understanding or compaction of execution traces.</td>
</tr>
<tr>
<td></td>
<td>behavior</td>
<td>the analysis of a system’s behavior or communications, e.g., protocol or state machine recovery.</td>
</tr>
<tr>
<td></td>
<td>general</td>
<td>gaining a general, non-specific knowledge of a program.</td>
</tr>
<tr>
<td>Target</td>
<td>legacy</td>
<td>legacy software, if classified as such by the author(s).</td>
</tr>
<tr>
<td></td>
<td>procedural</td>
<td>programs written in procedural languages.</td>
</tr>
<tr>
<td></td>
<td>oo</td>
<td>programs written in object-oriented languages, with such features as late binding and polymorphism.</td>
</tr>
<tr>
<td></td>
<td>threads</td>
<td>multithreaded systems.</td>
</tr>
<tr>
<td></td>
<td>web</td>
<td>web applications.</td>
</tr>
<tr>
<td></td>
<td>distributed</td>
<td>distributed systems.</td>
</tr>
<tr>
<td>Method</td>
<td>vis. (std.)</td>
<td>standard, widely used visualization techniques, e.g., graphs or UML.</td>
</tr>
<tr>
<td></td>
<td>vis. (adv.)</td>
<td>advanced visualization techniques, e.g., polymetric views or information murals.</td>
</tr>
<tr>
<td></td>
<td>slicing</td>
<td>dynamic slicing techniques.</td>
</tr>
<tr>
<td></td>
<td>filtering</td>
<td>filtering techniques or selective tracing, e.g., utility filtering.</td>
</tr>
<tr>
<td></td>
<td>metrics</td>
<td>the use of metrics.</td>
</tr>
<tr>
<td></td>
<td>static</td>
<td>information obtained through static analyses, e.g., from source code or documentation.</td>
</tr>
<tr>
<td></td>
<td>patt. det.</td>
<td>algorithms for the detection of design patterns or recurrent patterns.</td>
</tr>
<tr>
<td></td>
<td>compr./summ.</td>
<td>compression, summarization, and clustering techniques.</td>
</tr>
<tr>
<td></td>
<td>heuristics</td>
<td>the use of heuristics, e.g., probabilistic ranking or sampling.</td>
</tr>
<tr>
<td></td>
<td>fca</td>
<td>formal concept analysis.</td>
</tr>
<tr>
<td></td>
<td>querying</td>
<td>querying techniques.</td>
</tr>
<tr>
<td></td>
<td>online</td>
<td>online analysis, as opposed to post mortem (trace) analysis.</td>
</tr>
<tr>
<td></td>
<td>mult. traces</td>
<td>the analysis or comparison of multiple traces.</td>
</tr>
<tr>
<td>Evaluation</td>
<td>preliminary</td>
<td>evaluations of a preliminary nature, e.g., toy examples.</td>
</tr>
<tr>
<td></td>
<td>case study</td>
<td>case studies on medium-/large-scale open source systems (10K+ LOC) or traces (100K+ events).</td>
</tr>
<tr>
<td></td>
<td>industrial</td>
<td>evaluations on industrial systems.</td>
</tr>
<tr>
<td></td>
<td>comparison</td>
<td>comparisons of the authors’ approach with existing solutions.</td>
</tr>
<tr>
<td></td>
<td>human subj.</td>
<td>the involvement of human subjects, i.e., controlled experiments &amp; questionnaires.</td>
</tr>
<tr>
<td></td>
<td>quantitative</td>
<td>assessments of quantitative aspects, e.g., speed, recall, or trace reduction rate.</td>
</tr>
<tr>
<td></td>
<td>unknown/none</td>
<td>no evaluation, or evaluations on systems of unspecified size or complexity.</td>
</tr>
</tbody>
</table>
Attribute Occurrences

- regular
- quantitative
- preliminary
- comparison
- industrial
- unknown/none
- human subj.
Example: Evaluating ExtraVis Trace Visualization
Hierarchical Edge Bundles

Evaluation Through Case Studies

• Assessed how/why ExtraVis could be helpful in three dynamic analysis use cases:
  – Trace exploration, feature location, phase identification

• Selected three subject systems:
  – 1 industrial, 2 open source

• Exploration, theory building

Featue Location

Goal:
Establishing relations between feature invocations and the corresponding source code elements.

Execution scenario (150,000 events)

- Create new drawing
- Draw five types of figures
Feature Selection
Feature Analysis
Feature Analysis

figures.EllipseFigure
Analytical Generalization

• Helpful to understand user-triggerable functions.
  – Massive sequence chart helps to reveal recurring patterns
  – Circular bundle view helps to understand more detailed interactions

• Effective use of “screen real estate”

• Threats to (external) validity:
  – size of trace, presence of treads, role of identifiers, triggering the features, applied filtering in some cases.
ExtraVis Controlled Experiment

• Baseline: Eclipse

• Partition 34 subjects (academic + industry) into two groups:
  – Control group: Eclipse
  – Experimental group: Eclipse + ExtraVis

• Perform eight comprehension tasks on Checkstyle source code

• Measure time spent, and correctness of answers

B. Cornelissen, A. Zaidman, B. van Rompaey, and A. van Deursen
Trace Visualization for Program Comprehension: A Controlled Experiment.
In Proc. 17th International Conference on Program Comprehension (ICPC), 2009
Pacione’s nine principal comprehension activities.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Investigating the functionality of (a part of) the system</td>
</tr>
<tr>
<td>A2</td>
<td>Adding to or changing the system’s functionality</td>
</tr>
<tr>
<td>A3</td>
<td>Investigating the internal structure of an artifact</td>
</tr>
<tr>
<td>A4</td>
<td>Investigating dependencies between artifacts</td>
</tr>
<tr>
<td>A5</td>
<td>Investigating run-time interactions in the system</td>
</tr>
<tr>
<td>A6</td>
<td>Investigating how much an artifact is used</td>
</tr>
<tr>
<td>A7</td>
<td>Investigating patterns in the system’s execution</td>
</tr>
<tr>
<td>A8</td>
<td>Assessing the quality of the system’s design</td>
</tr>
<tr>
<td>A9</td>
<td>Understanding the domain of the system</td>
</tr>
</tbody>
</table>

Task design (cont’d)

• Example: Activity A1
  – “Investigating the functionality of (part of) the system”
  – is covered by
    • Task 1: globally understanding the main stages in a typical scenario
    • Task 4.1: providing a detailed description of the violation handling process
    • Task 4.2: determining whether check X reports violations

• Example: Activity A4
  – “Investigating dependencies between artifacts”
  – is covered by
    • Task 2.1: identifying three classes with high fanin and low fanout
    • Task 2.2: identifying a class in package X with a strong coupling to package Y
    • Task 3.2: listing the identifiers of all interactions between classes X and Y
    • Task 3.3: listing the identifiers of additional interactions in case of class Z
Correctness Results

Mean score Eclipse: 17.88
Mean score Ecl + ExtraVis: 12.47
Score Improvement: 43.38%
p value: < 0.001
Timing Results

Mean time Eclipse: 77.00
Mean time Ecl + ExtraVis: 59.94
Time Saved: 22.16%
p-value: 0.002
Findings

• 22% faster, 43% better
  – For the given comprehension tasks

• ExtraVis offers
  – global structural insight;
  – global and detailed behavioral insight;

• ExtraVis works best with a goal-oriented strategy
Threats to Validity

• Construct validity:
  – Can we distinguish use of traces in general from use of ExtraVis in particular?

• Internal validity:
  – Is background knowledge evenly distributed across groups?
  – Were subjects motivated?
  – Are the tasks biased? Are they too difficult? Too easy?
  – Was the time set realistic?
  – Did we grade the tasks correctly?

• External validity:
  – Are the tasks, checkstyle, and the subjects representative?
How do we Evaluate our Results?

<table>
<thead>
<tr>
<th>Through realistic, effort-taking, case studies</th>
<th>Through controlled experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitting analytical generalization</td>
<td>Smaller tasks by more people</td>
</tr>
<tr>
<td></td>
<td>Statistical generalization</td>
</tr>
</tbody>
</table>

- All done with the most critical attitude
- Reflected in at least the “threats to validity”
Part IV
Five Years from now...

Collaboration as in Web 2.0 and the move of applications to the web will be universal

We want our research to
– Play a leading role in these directions
– Leverage these trends in our methods & tools

What do we need in order to conduct the most critical and insightful evaluations?
What do we have to set in place right now?
Empirical Studies including Collaboration

- Visualizations are means for collaboration
  - Future tools will include collaboration support
- How can we evaluate this?
  - Case studies will be more complicated
- Community effort
  - Volunteer in each other’s experiments
  - Grant access to each other’s students
Access to Shared Web Applications

• We need compelling shared case studies
  – Web app sources harder to find on source forge

• Which maintenance tasks do we want to speed up and improve?
  – What role does collaboration play in there?

• To be able to compare results the community needs to work on common cases and tasks
Shared Analysis Infrastructures

• Analyzing web applications is highly challenging
  – Multitude of combined technologies
  – Static analysis insufficient
  – Ajax adds significant complexity

• **Community effort to build web analysis tools**
  – Our open source Crawljax infrastructure can be a starting point
Picasso Visualizations

• ExtraVis is heavy weight visualization artillery
• What about light-weight visualizations?
• Can the browser force us to simplicity? (gmail)
• Can we exploit HTML5 and rich Javascript visualization libraries?
• How does collaboration fit in?
Rethinking the IDE

• What is the essential IDE that is so simple that it can run in a browser?

• Shouldn’t collaboration be the central feature of an IDE?

• How can the IDE encourage developers to share their thoughts?

• How can the IDE distinguish right from wrong?