What your IDE could do once you understand your code

Questions & Directions

Arie van Deursen
Delft University of Technology
Acknowledgments

• CHOOSE organizers

• WSE & VISSOFT organizers + audience
  – Ken Wong, Hausi Muller, Michele Lanza, Margaret-Anne Storey

• Members of the TU Delft Software Engineering Research Group:
  – Ali Mesbah, Martin Pinzger, Andy Zaidman, Bas Cornelissen, Anja Guzzi, Rini van Solingen, ...

• Jim Whitehead for good discussions synchronizing our talks

• [Flickr photos from faisco, elizabethsalib, andywon]
Today’s Structure

• Part I
  – Rethinking Program Understanding

• Part II
  – The move to the Web

• Part III
  – Web 2.0

• Part IV
  – Rethinking the IDE
Program Understanding

• The task of building mental models of the underlying software,
• at various abstraction levels, ranging from
  – models of the code itself to
  – ones of the underlying application domain,
• for software evolution purposes

M. Marin, A. van Deursen, and L. Moonen.
Identifying Crosscutting Concerns using Fan-in Analysis.
ACM Transactions on Software Engineering and Methodology 17(1):1-37, 2007
Program Comprehension Research

- Theories on program comprehension
- Tool support for program comprehension, helping developers to find
  - Hidden dependencies
  - Key classes
  - Feature implementations
  - ...

“Software engineers spend 50% of their time on program understanding”

Perhaps more accurately:
- 1% understanding
- 49% misunderstanding

Pie chart showing distribution of time spent on analysis & design, misunderstanding, and understanding.
Understanding as a “Defining Moment”?

• Deep thinking, and all that’s recorded is ...
  – The actual change

• Can the IDE help to record what we learn in our quest?
  – Can it distinguish right from wrong?
  – Will our team mates care?

• We need to rethink the IDE
Part II

The Great Move to the Web
Collaborative Spreadsheets

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
<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>1502</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>320</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>12</td>
<td>Baker, Richard H.</td>
<td>LA</td>
<td>6th</td>
<td>225-3901</td>
<td>341</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>225-2623</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>19</td>
<td>Bean, Melissa L.</td>
<td>IL</td>
<td>8th</td>
<td>225-3711</td>
<td>512</td>
</tr>
<tr>
<td>20</td>
<td>Beaufrez, Bob</td>
<td>CO</td>
<td>7th</td>
<td>225-2645</td>
<td>504</td>
</tr>
<tr>
<td>21</td>
<td>Becerra, Xavier</td>
<td>CA</td>
<td>31st</td>
<td>225-6235</td>
<td>1119</td>
</tr>
<tr>
<td>22</td>
<td>Berklev, Shelleve</td>
<td>NV</td>
<td>1st</td>
<td>225-5965</td>
<td>439</td>
</tr>
</tbody>
</table>
Wave.google.com

Here's a formula for you

\[
\frac{d}{dx} \left( \int_0^x f(u) \, du \right) = f(x).
\]

def main:
    print "Hello world!"

me (and kasyntsy): #!scala
object main {
    def main(Array[String] args) {
        println("Hello world!");
    }
}
Web-Based Accounting Software
### Step 1 of 3

Just type or copy-paste the words you want to use into the below box, one per line. Then click "Save" to proceed to print your bingo cards.

- A word list needs at least 24 words (you have typed 28) or 25 if you don’t use a free space.
- A word list can have as many words as you want.
- Every card will be made unique.
- Scroll down for a list of the words in order.

#### Edit Word List

**Title:** Baby Shower

<table>
<thead>
<tr>
<th>One Word/Phrase Per Line</th>
<th>Preview Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>milk, baby talk, bonnet, mommy, love</td>
<td>B I N G O</td>
</tr>
<tr>
<td>diaper bag, nap time, baby powder, rattle, blanket</td>
<td></td>
</tr>
<tr>
<td>giggles, crib, Free Space!, cute, bibs</td>
<td></td>
</tr>
<tr>
<td>booties, pacifier, high chair, cooking, car seat</td>
<td></td>
</tr>
</tbody>
</table>
“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
Ajax: The Technology Behind It


Jesse Garrett. *Ajax: A new approach to web applications*. Adaptive Path, 2005
[ Testing Ajax Automatically ]

• Ajax is **error prone:**
  – asynchronous, stateful, client side
  – DOM hacking, using Javascript...

• Atusa infrastructure:
  – Crawljax to infer all GUI states
  – Generation of test suite
  – Use of *structural invariants* as oracle
  – Use of *state comparators* to do
    regression testing

• Applied to various systems

---

Google Chrome OS

• Fastest route from “on”-button to browser

• All apps are web apps

• Open sourced last week
Part III -- Web 2.0

• Technology paradigm:
  – Ajax

• Usage paradigm
  – Architecture of participation

• An attitude:
  – Processes and tools are socially open
  – Content can be used in several different contexts (e.g., recommenders)

• Self-organizing means for informal communication
What's happening?

Latest: @taoxiease Congratulations!! 10 minutes ago

Home

1 new tweet.

Felielle Most questions in BOF session were of the form "I work for company x and you know what's cool, we created a DSL for problem y" #pdc09
3 minutes ago from twidroid

dalmaer RT @cwilso: "Gotta love ppl who think moving all state into the cloud makes their system stateless. It just makes your client stateless."
3 minutes ago from Tweetie

avandeurse @taoxiease Congratulations!!
10 minutes ago from web in reply to taoxiease

dalmaer RT @diveintomark: Q: Chrome OS "app store"? A: "Web has hundreds of millions of applications." Will work on aiding discoverability.
13 minutes ago from Tweetie

twitter Quick question: What's happening?
http://blog.twitter.com/200...
37 minutes ago from web

dirkriehe Weird question; how could Google certify Chrome OS
18 minutes ago from web

Home

@avandeurse
Direct Messages 16
Favorites
Retweets

Search

Lists

Lists are timelines you build yourself, consisting of friends, family, co-workers, sports teams, you name it.

New list

Trending Topics

#justbecause
New Moon
Voice Candy
#nottosayonfirstdate
#ChromeOS
<table>
<thead>
<tr>
<th>Votes</th>
<th>Answers</th>
<th>Views</th>
<th>Title</th>
<th>User</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>16</td>
<td>Increment digit value in String</td>
<td>Adrian Pronk 937 10</td>
<td>8s ago</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>PostgreSQL with NHibernate problem with LIKE and parameter.</td>
<td>pavel_mihailov 1</td>
<td>29s ago</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>2k</td>
<td>How to check that a string is a palindrome using regular expressions?</td>
<td>Stewart 176 7</td>
<td>41s ago</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>34</td>
<td>+100 Server-Side Configuration for GWT</td>
<td>Robert Munteanu 7,217</td>
<td>42s ago</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>49</td>
<td>Should a newbie programmer give up the thinking “I must Start with C” ? (as his first language)</td>
<td>giorgian 843 1</td>
<td>1m ago</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>149</td>
<td>Creating an array of zero width and zero height!?</td>
<td>quamrana 1,370 3</td>
<td>1m ago</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>12</td>
<td>I have never seen a software be released as jar, so how can I make a jar file</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
Stackoverflow Mechanisms

- Voting on questions & answers
- Spotting similar questions
- Adding comments
- Starring questions
- Picking the right answer
- Earning reputation points
- Earning badges
Part IV: IDE Implications

Example 1: CodeRun
Example II: Bespin
The “IDE in the Browser”

• Mozilla Bespin: HTML5 Canvas-based editing
• Heroku: Ruby-on-Rails deployment (+ IDE?)
• CodeRun: Cloud deployment + IDE (C#, JS, ...)
• Eclipse E4: project plans include:
  – "Eclipse in browser"; DOM-based access to workbench; REST-full IDE services
• WWWWorkspace
  – Browser-access to Eclipse OSGi services
• PHP Anywhere
Mixing all of these ....
... would it 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 to ...

- 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
• Microsoft Team Foundation Server
• CodeRun
• Bespin

• E. Shihab, Z. Jiang, and A. E. Hassan. On the use of Internet Relay Chat (IRC) meetings by developers of the GNOME GTK+ project. *MSR*, 2009.

… but much remains to be done!
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 to
  – Play a leading role in these directions
  – Leverage these trends in our methods & tools

How can maximize our impact?
What do we have to set in place right now?
Implications (I): Practice

• When will your project run in the browser?
  – How do you keep up with technology progress?
  – How will you involve your users?
  – [ How do you test your Ajax apps? ]

• Can you leverage the web to let your developers collaborate?
  – Can you merge social networking into the development process?
Implications (II):
Evaluating Research Results

• Unit of analysis:
  – the *team*
  – the *collaboration*

• *Community effort*
  – We must volunteer in each other’s experiments
  – Grant access to each other’s students
Implications (III):
Systems Studied should be Web Apps

• Need for compelling shared case studies
  – Web app sources harder to find on, e.g. source forge

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

• To be able to \textit{compare results} the community needs to work on common cases and tasks
Implications (IV): 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
Implications (V): Research Tools

• Successful research tools will
  – Foster collaboration
  – Run in the browser

• Can the browser force us to simplicity? (gmail)

• Can we exploit HTML5 and rich Javascript visualization libraries?

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?