Proactive Detection of Collaboration Conflicts

Yuriy Brun ♦️ Michael D. Ernst ♦️ Reid Holmes 🍀 David Notkin ♦️

♦️ University of Washington 🍀 University of Waterloo
Have you ever made a mistake while programming and only realized it later?

- design decision
- refactoring
- repeated someone else’s work
Speculative analysis: Predict the future and analyze it

current program
Speculative analysis: Predict the future and analyze it
Speculative analysis: Predict the future and analyze it
Speculative analysis: Predict the future and analyze it
Speculative analysis: Predict the future and analyze it
Speculative analysis: Predict the future and analyze it

speculate

speculative analysis

inform developer

current program

analyze

Utility evaluation

Contributions

References
Exploring the future

- Past version of the program
- Present version of the program
- Future version of the program

Speculative analysis is predictive.
Exploring the future

past version of the program

present version of the program

future version of the program

mining software repositories

regression testing

delta debugging

continuous testing

automated debugging

Speculative analysis is predictive.
Exploring the future

Continuous development

- execution [Henderson and Weiser 1985; Karinthi and Weiser 1987]
- compilation [Childers et al. 2003; Eclipse foundation 2011]
- testing [Saff and Ernst 2003, 2004]
- version control integration [Guimarães and Rito-Silva 2010]
Exploring the future

Continuous development
- execution [Henderson and Weiser 1985; Karinthi and Weiser 1987]
- compilation [Childers et al. 2003; Eclipse foundation 2011]
- testing [Saff and Ernst 2003, 2004]
- version control integration [Guimarães and Rito-Silva 2010]
Exploring the future

Continuous development

- execution [Henderson and Weiser 1985; Karinthi and Weiser 1987]
- compilation [Childers et al. 2003; Eclipse foundation 2011]
- testing [Saff and Ernst 2003, 2004]
- version control integration [Guimarães and Rito-Silva 2010]

Speculative analysis is predictive.
Contributions

- Speculative analysis
- Speculative analysis for collaborative development
  Crystal: prototype tool
- Utility of speculative analysis for collaborative development
Version-control terminology

Proactive conflict detection applies to both centralized and decentralized version control.

Terminology:

<table>
<thead>
<tr>
<th>Local commit:</th>
<th>decentralized</th>
<th>centralized</th>
</tr>
</thead>
<tbody>
<tr>
<td>commit</td>
<td>commit</td>
<td>save</td>
</tr>
<tr>
<td>incorporate:</td>
<td>push and pull</td>
<td>commit and update</td>
</tr>
</tbody>
</table>
The Gates conflict
The Gates conflict

The information was all there, but the developers didn’t know it.
The Gates conflict

The information was all there, but the developers didn’t know it.
The Gates conflict

The information was all there, but the developers didn’t know it.
The Gates conflict

The information was all there, but the developers didn’t know it.
The Gates conflict

The information was all there, but the developers didn’t know it.
The Gates conflict

The information was all there, but the developers didn’t know it.
The Gates conflict

The information was all there, but the developers didn't know it.
The Gates conflict

The information was all there, but the developers didn’t know it.
The Gates conflict
The Gates conflict

The information was all there, but the developers didn’t know it.
What could well-informed developers do?

- Avoid conflicts
What could well-informed developers do?

- Avoid conflicts
- Reduce conflict severity
Introducing Crystal: A proactive conflict detector

DEMO
Introducing Crystal: A proactive conflict detector

DEMO

http://crystalvc.googlecode.com
Speculative analysis in collaborative development

current program
speculate
analyze
inform developer

analyze
speculate
inform developer

current program
Reducing false positives in conflict prediction

Collaborative awareness

- Palantir [Sarma et al. 2003]
- FASTDash [Biehl et al. 2007]
- Syde [Hattori and Lanza 2010]
- CollabVS [Dewan and Hegde 2007]
- Safe-commit [Wloka et al. 2009]
- SourceTree [Streeting 2010]
Reducing false positives in conflict prediction

Collaborative awareness

- Palantír [Sarma et al. 2003]
- FASTDash [Biehl et al. 2007]
- Syde [Hattori and Lanza 2010]

- CollabVS [Dewan and Hegde 2007]
- Safe-commit [Wloka et al. 2009]
- SourceTree [Streeting 2010]

Crystal analyzes **concrete artifacts**, eliminating false positives and false negatives.
Utility of proactive collaborative conflict detection

- Are textual collaborative conflicts a real problem?
- How dangerous are safe merges?
- Do higher-order collaborative conflicts exist?
Are textual collaborative conflicts a real problem?

histories of 9 open-source projects:

- size: 26K–1.4MSLoC
- developers: 298
- versions: 140,000

Perl5, Rails, Git, jQuery, Voldemort, MaNGOS, Gallery3, Samba, Insoshi
Are textual collaborative conflicts a real problem?

histories of 9 open-source projects:

- size: 26K–1.4MSLoC
- developers: 298
- versions: 140,000

Perl5, Rails, Git, jQuery, Voldemort, MaNGOS, Gallery3, Samba, Insoshi
Are textual collaborative conflicts a real problem?

RQ1: How frequent are textual conflicts?

16% of the merges have textual conflicts.

RQ2: How long do textual conflicts persist?

Conflicts live a mean of 9.8 and median of 1.6 days. The worst case was over a year.

RQ3: How long do textually-safe merges persist?

Textually-safe merges live a mean of 11.0 and median of 1.9 days.
Are textual collaborative conflicts a real problem?

RQ1: How frequent are textual conflicts?
16% of the merges have textual conflicts.
Are textual collaborative conflicts a real problem?

RQ1: How frequent are textual conflicts?
16% of the merges have textual conflicts.

RQ2: How long do textual conflicts persist?
Conflicts live a mean of 9.8 and median of 1.6 days. The worst case was over a year.

Textually-safe merges live a mean of 11.0 and median of 1.9 days.
Are textual collaborative conflicts a real problem?

RQ1: How frequent are textual conflicts?
16% of the merges have textual conflicts.

RQ2: How long do textual conflicts persist?
Conflicts live a mean of 9.8 and median of 1.6 days. The worst case was over a year.
Are textual collaborative conflicts a real problem?

RQ1: How frequent are textual conflicts?
16% of the merges have textual conflicts.

RQ2: How long do textual conflicts persist?
Conflicts live a mean of 9.8 and median of 1.6 days. The worst case was over a year.

RQ3: How long do textually-safe merges persist?
Are textual collaborative conflicts a real problem?

RQ1: How frequent are textual conflicts?
16% of the merges have textual conflicts.

RQ2: How long do textual conflicts persist?
Conflicts live a mean of 9.8 and median of 1.6 days. The worst case was over a year.

RQ3: How long do textually-safe merges persist?
Textually-safe merges live a mean of 11.0 and median of 1.9 days.
How dangerous are safe merges?

RQ4: Where do textual conflicts come from?

93% of textual conflicts developed from safe merges.

20% of textually-safe merges developed into conflicts.
How dangerous are safe merges?

RQ4: Where do textual conflicts come from?

93% of textual conflicts developed from safe merges.
How dangerous are safe merges?

RQ4: Where do textual conflicts come from?
93% of textual conflicts developed from safe merges.

RQ5: Do textually-safe merges devolve into conflicts?
How dangerous are safe merges?

RQ4: Where do textual conflicts come from?

93% of textual conflicts developed from safe merges.

RQ5: Do textually-safe merges devolve into conflicts?

20% of textually-safe merges developed into conflicts.
Do higher-order collaborative conflicts exist?

RQ6: Does merged code fail to build or fail tests?

One in three conflicts are of higher-order.
Crystal is in the wild

“Crystal handles several projects and users effortlessly and presents the necessary information in a simple and understandable way.”
- a user

Microsoft Beacon

- A centralized version control-based tool.
- Microsoft product groups will use Beacon to help identify conflicts earlier in the development process.
- We will conduct user studies to measure effects on developers.
Contributions

- Introduced **speculative analysis** to guide future actions.
- Developed Crystal to **precisely** detect conflicts and **unobtrusively** inform developers.
- Analyzed 9 projects with over 140,000 versions: conflicts are **frequent** and **persistent**.

http://crystalvc.googlecode.com


