ReachHover: Preserving Context in Data-flow Analysis

James Yoo\textsuperscript{a, b}  
Gail C. Murphy\textsuperscript{b}

\textsuperscript{a}University of Washington  
\textsuperscript{b}University of British Columbia
Who here has used an IDE?
Do You Ever Struggle To Maintain Context in Your IDE?
Do You Ever Struggle To Maintain Context in Your IDE?

Files

Classes

Methods

Locations
Preserving Context in IDEs

```java
@Override
protected void filter(CompletionResultSet resultSet) {
    try {
        if (resultSet != null) {
            if (filterPrefix != null) {
                resultSet.addAllItems(getFilteredData(results, filterPrefix));
                resultSet.setHasAdditionalItems(hasAdditionalItems);
            } else {
                Completion.get().hideDocumentation();
                Completion.get().hideCompletion();
            }
        }
    } catch (Exception ex) {
        Exceptions.printStackTrace(ex);
    }
    resultSet.finish();
}

private static boolean isJavaIdentifierPart(String text, boolean allowForDor) {
    for (int i = 0; i < text.length(); i++) {
        if (!Character.isJavaIdentifierPart(text.charAt(i)) || allowForDor && text.charAt(i) == '.') {
            return false;
        }
    }
    return true;
```
Preserving Context in IDEs

```java
@override
protected void filter(CompletionResultSet resultSet) {
    try {
        if (results != null) {
            if (filterPrefix != null) {
                resultSet.addAllItems(getFilteredData(results, filterPrefix));
                resultSet.setHasAdditionalItems(hasAdditionalItems);
            } else {
                Completion.get().hideDocumentation();
                Completion.get().hideCompletion();
            }
        }
        else if (queryType == TOOLTip_QUERY_TYPE) {
            resultSet.setToolTip(toolTip != null && toolTip.hasData() ? toolTip : null);
        }
        resultSet.setAnchorOffset(ancestorOffset);
        catch (Exception ex) {
            Exceptions.printStackTrace(ex);
        }
        resultSet.finish();
    }

private static boolean isJavaIdentifierPart(String text, boolean allowForDor) {
    for (int i = 0; i < text.length(); i++) {
        if (!Character.isJavaIdentifierPart(text.charAt(i)) || allowForDor && text.charAt(i) == '.') {
            return false;
        }
    }
    return true;
}
```
Preserving Context in IDEs

Present information on the periphery, away from the structure under investigation.
Preserving Context in IDEs
Preserving Context in IDEs

Context is not preserved across views of content
Preserving Context in IDEs

Context is not preserved across views of content

Views of content are separated and spatially distant
Contributions

• **Formative study**: what questions do programmers frequently ask?

• **ReachHover**: a context-preserving interface for data-flow questions

• **User study**: ReachHover helps developers answer data-flow questions
Contributions

- **Formative study**: what questions do programmers frequently ask?
Programmers care about data-flow
Programmers care about data-flow

Where does a value come from?
Programmers care about data-flow

Where does a value come from?

How is this object going to be modified?
Programmers care about data-flow

Where does a value come from?

How is this object going to be modified?

Which parts of this object are accessed?
Programmers care about data-flow

Where does a value come from?

How is this object going to be modified?

Which parts of this object are accessed?

How is this value used throughout the program?
Programmers care about data-flow

10 difficult questions from the literature
• Refactoring
• Data-flow
• ... 

Our work: Survey of 72 programmers
• How often do you ask yourself this question?

Most frequent: data-flow questions
• “Where does this value come from?”
• “How is this value modified?”
• ...
Contributions

• **Formative study**: what questions do programmers frequently ask?

  • Programmers *frequently* ask *data-flow questions*
Contributions

• **Formative study**: what questions do programmers frequently ask?

  • Programmers **frequently** ask **data-flow questions**

• **ReachHover**: a context-preserving interface for data-flow questions
ReachHover Live Demo
Contributions

• **Formative study**: what questions do programmers frequently ask?
  
  • Programmers *frequently* ask **data-flow questions**

  • **ReachHover**: a context-preserving interface for data-flow questions

  • **Open-source plugin** for the IntelliJ Platform
Contributions

• **Formative study**: what questions do programmers frequently ask?

  • Programmers *frequently* ask data-flow questions

• **ReachHover**: a context-preserving interface for data-flow questions

  • **Open-source plugin** for the *IntelliJ* Platform

• **User study**: ReachHover helps developers answer data-flow questions
Evaluating ReachHover: User Study

n = 20
~5 YoE average

randomized counter-balanced within-subjects

Task 1 (ReachHover)
Task 2 (Standard Tooling)

Task 1 (Standard Tooling)
Task 2 (ReachHover)

Task 2 (ReachHover)
Task 1 (Standard Tooling)
Evaluating ReachHover: Experimental Design

- **Task 1**: Questions along a **backward data-flow trace**
  - **Q1** Locations where a method argument is instantiated
  - **Q2** Where is a method argument used?

- **Task 2**: Questions along a **forward data-flow trace**
  - **Q1** Locations where a local variable is checked for null
  - **Q2** Where is a local variable used?
Evaluating ReachHover: Conclusions

• Programmers can accurately answer data-flow-related questions with ReachHover
Task 1: Backward Data-flow

```java
public void addBookmark(FileBookmarks fileBookmarks, BookmarkInfo bookmark) {
    checkModDuringFire();
    fileBookmarks.add(bookmark);
    BookmarkChange change = getBookmarkChange(bookmark);
    change.markAdded();
    bookmark.getProjectBookmarks().setModified(true);
    structureChange = true;
}
```

**T1.Q1** In how many locations does a call to the `BookmarkInfo` constructor exist?

- ReachHover: 10/10
- IntelliJ: 6/10
Task 1: Backward Data-flow

public void addBookmark(FileBookmarks fileBookmarks, BookmarkInfo bookmark) {
    checkModDuringFire();
    fileBookmarks.add(bookmark);
    BookmarkChange change = getBookmarkChange(bookmark);
    change.markAdded();
    bookmark.getFileBookmarks().getProjectBookmarks().setModified(true);
    structureChange = true;
}

T1.Q1 In how many locations does a call to the BookmarkInfo constructor exist?
• ReachHover: 10/10
• IntelliJ: 6/10
Task 1: Backward Data-flow

**T1.Q2** Provide the names of the methods where the value of *bookmark* is used

<table>
<thead>
<tr>
<th>Method</th>
<th>Correct Answers</th>
<th>Incorrect Answers</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unmodified IntelliJ</td>
<td>13</td>
<td>8</td>
<td>62%</td>
</tr>
<tr>
<td>ReachHover</td>
<td>14</td>
<td>2</td>
<td>88%</td>
</tr>
</tbody>
</table>
Task 2: Forward Data-flow

FindReplaceResult findReplaceResult = new FindReplaceResult(new int[]{-1, -1}, replacedString: "");
findReplaceResult.setErrorMsg(NbBundle.getMessage(DocumentFinder.class, resName: "pattern-error-dialog");
return findReplaceResult;

T2.Q1  How many times was findReplaceResult explicitly checked for a null value?

- ReachHover: 7/10
- IntelliJ: 7/10
Task 2: Forward Data-flow

FindReplaceResult findReplaceResult = new FindReplaceResult(new int[]{-1, -1}, · replacedString: "");
findReplaceResult.setResMsg(Bundle.getMessage(DocumentFinder.class, · resName: "pattern-error-diaeturn findReplaceResult;

T2.Q1 How many times was \texttt{findReplaceResult} explicitly checked for a null value?

- ReachHover: 7/10
- IntelliJ: 7/10
### Task 2: Forward Data-flow

**T2.Q2** Provide the names of the methods where the value of `findReplaceResult` is used

<table>
<thead>
<tr>
<th></th>
<th>Correct Answers</th>
<th>Incorrect Answers</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unmodified IntelliJ</td>
<td>22</td>
<td>10</td>
<td>69%</td>
</tr>
<tr>
<td>ReachHover</td>
<td>24</td>
<td>3</td>
<td>80%</td>
</tr>
</tbody>
</table>
Evaluating ReachHover: Conclusions

• Programmers can accurately answer data-flow-related questions with ReachHover

• ReachHover minimizes context-switching actions.
Evaluating ReachHover: Context

- Editor content changes as a proxy for context switching
- Lower average of editor content changes using ReachHover

(Lower is better)
Evaluating ReachHover: Conclusions

• Programmers accurately answer data-flow questions
• ReachHover reduces context-switching
• Users prefer context-preserving interface
Evaluating ReachHover: Qualitative Results

“Felt less cluttered and made the task easier”

“Was easier to use (maybe because I didn’t lose the context)”

“I preferred that I could look at the matches in the same modal window without affecting the open file”
Related Work: Alternative Interfaces

Code Bubbles

• An infinitely-zoomable 2D canvas
Related Work: Code Bubbles
Related Work: Alternative Interfaces

Code Bubbles
• An infinitely-zoomable 2D canvas

  Drastic re-design of a standard IDE

CodeRibbon
• A tiled “ribbon” of code views
Related Work: Code Ribbon
Related Work: Code Ribbon

Related Work:

Code Ribbon

```java
public boolean editorPopupEnabled = true;

public boolean allButtonsEnabled = true;

// Whether popup enabled or disabled on all buttons with mnemonics clicks.

// second click will trigger a notification, and so on.

// Whether to show notification when the IDE is in Presentation Mode

// Whether to show notification when the IDE is in Distraction Free Mode

public boolean disabledInDistractionFreeMode = false;
```

```java
/*
 * Copyright (c) 2019 Patrick Scheibe, Dmitry Kashin, Athielle.
 * Redistribution and use in source and binary forms, with or without modification, are
 * permitted provided that the following conditions are met:
 * 1. Redistributions of source code must retain the above copyright notice, this list
 *    of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright notice, this list
 *    of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

```java
package de.halirutan.keypromoter;
import com.intellij.openapi.application.ApplicationManager;
```
Preserving Context in IDEs

Context is not preserved across views of content

Views of content are separated and spatially distant
Contributions

• **Formative study**: what questions do programmers frequently ask?

  • Programmers *frequently* ask data-flow questions
Contributions

• **Formative study**: what questions do programmers frequently ask?
  
  • Programmers *frequently* ask **data-flow questions**

• **ReachHover**: a context-preserving interface for data-flow questions

• **Open-source plugin** for the **IntelliJ** Platform
Contributions

• **Formative study**: what questions do programmers frequently ask?
  
  • Programmers frequently ask **data-flow questions**

• **ReachHover**: a context-preserving interface for data-flow questions

  • **Open-source plugin** for the **IntelliJ Platform**

• **User study**: ReachHover helps developers answer data-flow questions

ReachHover source code