Assessing the Quality of the Steps to Reproduce in Bug Reports

27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'19)

Wednesday, August 28th, 2019
Users and developers report bugs when software does not behave as expected.
Users and developers *report bugs* when *software does not behave as expected*
Users and developers report bugs when software does not behave as expected.
COMPONENTS OF A BUG REPORT

MetaData

**Bug 101434** - [Contexts] performance: Slow cursor navigation in Text fields

- **Status**: VERIFIED FIXED
- **Alias**: None
- **Product**: Platform
- **Component**: UI (show other bugs)
- **Version**: 3.1
- **Hardware**: PC Linux-GTK
- **Importance**: P2 major (vote)
- **Target**: 3.1.1
- **Milestone**: 3.1.1
- **Assignee**: Douglas Pollock

- **Reported**: 2005-06-23 09:08 EDT by Tom Hofmann
- **Modified**: 2005-09-26 13:45 EDT (History)

**CC List**: 6 users

**See Also**

**Attachments**

- **updateShellKludge.zip** (5.07 KB, application/x-zip-compressed)
  - **2005-06-23 10:30 EDT**, Tom Hofmann
- **Patch to "org.eclipse.ui.workbench"** (3.96 KB, patch)

- **no flags**
- **Details**

- **no flags**
- **Details | Diff**
COMPONENTS OF A BUG REPORT

Attachments

**Attachments**

<table>
<thead>
<tr>
<th>Attachment Name</th>
<th>Size</th>
<th>Type</th>
<th>Flags</th>
<th>Details/Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>updateShellKludge.zip (3.07 KB)</td>
<td>no flags</td>
<td>Details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patch to &quot;org.eclipse.ui.workbench&quot;</td>
<td>3.96 KB</td>
<td>patch</td>
<td>no flags</td>
<td>Details/Diff</td>
</tr>
</tbody>
</table>

**Note**

You need to [log in](#) before you can comment on or make changes to this bug.

Tom Hofmann [ECA] 2005-06-23 09:08:52 EDT

Not sure whether this is my particular install or a new problem.

With the above build, cursor navigation in any text field is extremely slow.

Steps:
- open the Find dialog (alternatively: open the preference dialog, or the TeamXCreate Patch wizard)
- in a text field, enter some text, at least 20 characters
- press and hold the ARROW_LEFT / RIGHT keys
COMPONENTS OF A BUG REPORT

Description

Tom Hofmann – ECA 2005-06-23 09:08:52 EDT

N20056623-0010-gtk

Not sure whether this is my particular install or a new problem.

With the above build, cursor navigation in any text field is extremely slow.

Steps:
- open the Find dialog (alternatively: open the preference dialog, or the TeamXCreate Patch wizard)
- in a text field, enter some text, at least 20 characters
- press and hold the ARROW_LEFT / RIGHT keys

Expected: I see the caret move through the entered text
Actual: The cursor does not visibly change its position until after releasing the key plug some delay.

Notes:
- keyboard repeat rate is at the default
- the same works fine (+/- at least nowhere near the slowness of Text fields) in text editors
- other applications do not show this behavior

This is quite bad as the caret cannot be placed reliably this way. Will try to reproduce with an SWT-only example.
Steps to Reproduce (S2Rs)

Description

Not sure whether this is my particular install or a new problem.

With the above build, cursor navigation in any text field is extremely slow.

Steps:
- open the Find dialog (alternatively: open the preference dialog, or the TeamxCreate Patch wizard)
- in a text field, enter some text, at least 20 characters
- press and hold the ARROW_LEFT / _RIGHT keys

Expected: I see the caret move through the entered text
Actual: The cursor does not visibly change its position until after releasing the key plug some delay.

Notes:
- Keyboard repeat rate is at the default
- the same works fine (+/- at least nowhere near the slowness of Text fields) in text editors
- other applications do not show this behavior

This is quite bad as the caret cannot be placed reliably this way. Will try to reproduce with an SWT-only example.
**Steps to Reproduce (S2Rs)**

**Description**

Steps:
- open the Find dialog (alternatively: open the preference dialog, or the Team>Create Patch wizard)
- in a text field, enter some text, at least 20 characters
- press and hold the ARROW_LEFT / ARROW_RIGHT keys

This is quite bad as the caret cannot be placed reliably this way. Will try to reproduce with an SWT-only example.
Steps to reproduce are often...
Steps to reproduce are often…

Missing
Steps to Reproduce are Often…

Missing

Ambiguous
Steps to Reproduce are Often…

Missing

Ambiguous

Incomplete
Dear GitHub,

You have done so much to grow the open source community and make it really accessible to users. Somehow you have us chasing stars and filling up squares, improving the world’s software in the process.

However, many of us are frustrated. Those of us who run some of the most popular projects on GitHub feel completely ignored by you. We’ve gone through the only support channel that you have given us either to receive an empty response or even no response at all. We have no visibility into what has happened with our requests, or whether GitHub is working on them. Since our own work is usually done in the open and everyone has input into the process, it seems strange for us to be in the dark about one of our most important project dependencies.

The problems we most frequently have, and our best ideas for how to address them, are:

- Issues are often filed missing crucial information like reproduction steps or version tested. We’d like issues to gain custom fields, along with a mechanism (such as a mandatory issue template, perhaps powered by a newissue.md in root as a likely-simple solution) for ensuring they are filled out in every issue.

- Issues often accumulate content-less “:+1” comments which serve only to spam the maintainers and any others who might see them. We’d like comments and votes to be tied to code reviews, as a way to eliminate these comments and discuss real issues.

- Integrating our products with the GitHub GUI without breaking it, successfully. We’d like a way to easily see the impact of our tools or libraries on the GitHub GUI.

- We’re working on something based on GitHub which is going to impact the platform, and we’d like to do it in a way that is best for everyone — sadly, we’re currently running into friction.

We’re working on making the world’s software, and we’d like to have a better interaction with people who are doing the same. We’re working on making the world’s software, and we’d like to have a better interaction with people who are doing the same.

Thank you for your time to read this.

[Your Name]
The problems we most frequently have, and our best ideas for how to address them, are:

- Issues are often filed missing crucial information like reproduction steps or version tested. We’d like issues to gain custom fields, along with a mechanism (such as a mandatory issue template, perhaps powered by a newissue.md in root as a likely-simple solution) for ensuring they are filled out in every issue.

- Issues often accumulate content-less “+1” comments which serve only to spam the maintainers and any others subscribed to the issue. These +1s serve a valuable function in letting maintainers know how widespread an issue is, but their drawbacks are too great. We’d like issues to gain a first-class voting system, and for content-less comments like “+1” or “👍” or “me too” to trigger a warning and instructions on how to use the voting mechanism.

- Issues and pull requests are often created without any adherence to the CONTRIBUTING.md contribution guidelines, due to the inconspicuous nature of the “guidelines for contributing” link when creating an issue and the fact that it often contains a lot of information that isn’t relevant to opening issues (such as information about hacking on the project). Maintainers should be able to configure a file in the repo (interpreted as GFM) to be displayed at the top of the new issue / PR page instead of that link. Maintainers can choose to inline content there and / or link to other pages as appropriate.
The problems we most frequently have, and our best ideas for how to address them, are:

- Issues are often filed missing crucial information like reproduction steps or version tested. We’d like issues to gain custom fields, along with a mechanism (such as a mandatory issue template, perhaps powered by a newissue.md in root as a likely-simple solution) for ensuring they are filled out in every issue.

Issues are often filed missing crucial information like reproduction steps issue is, but their drawbacks are too great. We’d like issues to gain a first-class voting system, and for contentless comments like “+1” or “👍” or “me too” to trigger a warning and instructions on how to use the voting mechanism.

- Issues and pull requests are often created without any adherence to the CONTRIBUTING.md contribution guidelines, due to the inconspicuous nature of the “guidelines for contributing” link when creating an issue and the fact that it often contains a lot of information that isn’t relevant to opening issues (such as information about hacking on the project). Maintainers should be able to configure a file in the repo (interpreted as GFM) to be displayed at the top of the new issue / PR page instead of that link. Maintainers can choose to inline content there and / or link to other pages as appropriate.
THE IMPACT OF LOW-QUALITY S2Rs
THE IMPACT OF LOW-QUALITY S2Rs

Non-Reproducible Bugs
[Erfani Joorbachi et.al., MSR’14]

Unfixed Bugs
[Zimmerman et.al., ICSE’12]
THE IMPACT OF LOW-QUALITY S2Rs

Non-Reproducible Bugs
[Erfani Joorbachi et.al., MSR’14]

Unfixed Bugs
[Zimmerman et.al., ICSE’12]

Additional Bug Triage
[Breu, et.al., CSCW’10 & Zimmerman, et. al., ICSE’12]

Delay in Bug Resolution
[Guo, et. al., ICSE'10 & Zimmerman, et. al., ICSE’12]
Euler: Assessing the Quality of S2Rs

Bug Report → Euler → Quality Report → Reporters

<table>
<thead>
<tr>
<th>#</th>
<th>S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S₁</td>
<td>EM, MS</td>
</tr>
<tr>
<td>2</td>
<td>S₂</td>
<td>AS</td>
</tr>
<tr>
<td>3</td>
<td>S₃</td>
<td>VM</td>
</tr>
</tbody>
</table>
Euler is able to:
1) Automatically identify S2Rs
2) Associate the S2Rs with program information
3) Assess the quality of given S2Rs
Euler is able to:
1) Automatically identify S2Rs
2) Associate the S2Rs with program information
3) Assess the quality of given S2Rs

Euler: Assessing the Quality of S2Rs

Bug Report → Euler → Quality Report → Reporters

Euler is able to detect:
1) Missing Steps
2) Ambiguous Steps
3) Steps with Unexpected Vocabulary

<table>
<thead>
<tr>
<th>#</th>
<th>S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S₁</td>
<td>EM, MS</td>
</tr>
<tr>
<td>2</td>
<td>S₂</td>
<td>AS</td>
</tr>
<tr>
<td>3</td>
<td>S₃</td>
<td>VM</td>
</tr>
</tbody>
</table>
Euler’s Components & Workflow
Euler’s Components & Workflow

1) S2R Identification

Bug Report → Neural Sequence Labeling + Dependency Parsing → Reproduction Steps

- **S1**: [press] ‘Go to…’
- **S2**: [type] [number]
- **S3**: [press] ‘Create’

shows create list entry
Euler’s Components & Workflow

1) S2R Identification

- Bug Report
- Neural Sequence Labeling + Dependency Parsing
- Reproduction Steps

S1: [press] ‘Go to…’
S2: [type] [number]
S3: [press] ‘Create’
...

2) Execution Model Generation

- APK
- Automated Exploration
- Execution Model

Android App
Euler’s Components & Workflow

1) S2R Identification
- Bug Report
- Neural Sequence Labeling + Dependency Parsing
- Reproduction Steps
  - \( S_1 \): [press] ‘Go to...’
  - \( S_2 \): [type] [number]
  - \( S_3 \): [press] ‘Create’
  - ...

2) Execution Model Generation
- APK
- Automated Exploration
- Execution Model

3) Quality Assessment
- S2R Matching
- S2R Inference
- Random Exploration

Android App

Steps:
1. S2R Identification
2. Execution Model Generation
3. Quality Assessment
Euler's Components & Workflow

1) S2R Identification

Bug Report → Neural Sequence Labeling + Dependency Parsing → Reproduction Steps

S1: [press] ['Go to...']
S2: [type] [number]
S3: [press] ['Create'] ...

2) Execution Model Generation

APK → Automated Exploration → Execution Model

3) Quality Assessment

S2R Matching
S2R Inference
Random Exploration

Bug Report

Reproduction Steps

1) S2R Identification

Neural Sequence Labeling + Dependency Parsing

S1: [press] ['Go to...']
S2: [type] [number]
S3: [press] ['Create'] ...

2) Execution Model Generation

APK → Automated Exploration → Execution Model

3) Quality Assessment

S2R Matching
S2R Inference
Random Exploration

<table>
<thead>
<tr>
<th>#</th>
<th>S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S1</td>
<td>EM, MS</td>
</tr>
<tr>
<td>2</td>
<td>S2</td>
<td>AS</td>
</tr>
<tr>
<td>3</td>
<td>S3</td>
<td>VM</td>
</tr>
</tbody>
</table>
**Euler’s Components & Workflow**

1) **S2R Identification**
- Bug Report
- Neural Sequence Labeling + Dependency Parsing
- Reproduction Steps
  - S₁: [press] ‘Go to…’
  - S₂: [type] [number]
  - S₃: [press] ‘Create’

2) **Execution Model Generation**
- APK
- Automated Exploration
- Execution Model

3) **Quality Assessment**
- S2R Matching
- S2R Inference
- Random Exploration

**Table: Quality Annotations**

<table>
<thead>
<tr>
<th>#</th>
<th>S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S₁</td>
<td>EM, MS</td>
</tr>
<tr>
<td>2</td>
<td>S₂</td>
<td>AS</td>
</tr>
<tr>
<td>3</td>
<td>S₃</td>
<td>VM</td>
</tr>
</tbody>
</table>
Euler's Components & Workflow

1) S2R Identification

Bug Report → Neural Sequence Labeling + Dependency Parsing → Reproduction Steps

S1: [press] ['Go to...']
S2: [type] [number]
S3: [press] ['Create']
...

2) Execution Model Generation

APK → Android App → Automated Exploration → Execution Model

3) Quality Assessment

- S2R Matching
- S2R Inference
- Random Exploration

Reproduction Steps:

1) S2R Identification
2) Execution Model Generation
3) Quality Assessment

Bug Report: Reproduction Steps

- S1: [press] ['Go to...']
- S2: [type] [number]
- S3: [press] ['Create']
...

Neural Sequence Labeling + Dependency Parsing

Android App Execution Model
S2R IDENTIFICATION IN BUG REPORTS

Bug Report
... Usually when I press 'Go to...' and type a number and press 'Go', it simply refreshes ...

S2R Sentences
Usually when I press 'Go to...' and type a number and press 'Go', it simply refreshes ...

Individual S2R
S₁: [press] ['Go to...']
S₂: [type] [number]
S₃: [press] ['Go']
Usually when I press 'Go to...' and type a number and press 'Go', it simply refreshes ...

Approach: neural sequence labeling + dependency parsing
Bug Report:
Paragraphs, Sentences, Words

Paragraph 1:
Sentence 1.1
Sentence 2.2
...

Paragraph 2:
Sentence 2.1
Sentence 2.2
...

MODEL INPUT + DATASET LABELS
Beginning-Inside-Outside (BIO) tagging

Paragraph 1:
Sentence 1.1
Sentence 1.2 [B-S2R]
Sentence 1.3 [I-S2R]
Sentence 1.4 [I-S2R]
Sentence 1.5 [O]
Sentence 1.6 [O]

Paragraph 2:
Sentence 2.1
Sentence 2.2
...

Paragraph 1:
Sentence 1.1 [O]
Sentence 1.2 [B-S2R]
Sentence 1.3 [I-S2R]
Sentence 1.4 [I-S2R]
Sentence 1.5 [O]
Sentence 1.6 [O]
WORD REPRESENTATIONS

“Usually when I press 'Go to...' and ...”
"Usually when I press 'Go to...' and ..."
“Usually when I press 'Go to...' and ...”

Word embeddings (skip-gram)

Char embeddings (CNN)
“Usually when I press 'Go to...' and ...”

Word embeddings (skip-gram)

Char embeddings (CNN)

Usually
When
I

Usually
When
I

Bi-LSTM

Forward LSTM
Backward LSTM
**Sentence Representation & Inference**

Sentence 1

Sentence 2

... 

Sentence n

$w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ w_6 \ ...$
Sentence Representation & Inference

\[ \text{Avg}(w_k) + \text{Discourse Patterns} \]
Sentence Representation & Inference

```
<table>
<thead>
<tr>
<th>Sentence 1</th>
<th>Sentence 2</th>
<th>...</th>
<th>Sentence n</th>
</tr>
</thead>
<tbody>
<tr>
<td>w1</td>
<td>w2</td>
<td>w3</td>
<td>w4</td>
</tr>
</tbody>
</table>

Conditional Random Field (CRF)

Labels

Sentence 1 | O
Sentence 2 | B-S2R
... | ....
Sentence n | I-S2R
```
“When I **create** an **entry** for a **purchase**, the autocomplete list shows up”
"When I create an entry for a purchase, the autocomplete list shows up."

S2R: [create] [entry] [for] [purchase]
Euler’s Components & Workflow

1) S2R Identification
- Bug Report
  - Neural Sequence Labeling + Dependency Parsing
  - Reproduction Steps
  - $S_1$: [press] ‘Go to…’
  - $S_2$: [type] [number]
  - $S_3$: [press] ‘Create’

2) Execution Model Generation
- Android App
  - Automated Exploration
  - Execution Model

3) Quality Assessment
- S2R Matching
- S2R Inference
- Random Exploration

Table:

<table>
<thead>
<tr>
<th>#</th>
<th>S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$S_1$</td>
<td>EM, MS</td>
</tr>
<tr>
<td>2</td>
<td>$S_2$</td>
<td>AS</td>
</tr>
<tr>
<td>3</td>
<td>$S_3$</td>
<td>VM</td>
</tr>
</tbody>
</table>
EULER’S COMPONENTS & WORKFLOW

1) S2R Identification

Bug Report
Neural Sequence Labeling + Dependency Parsing
Reproduction Steps

2) Execution Model Generation

Android App
Automated Exploration
Execution Model

3) Quality Assessment

S2R Matching
S2R Inference
Random Exploration

1) S2R Identification
Neural Sequence Labeling + Dependency Parsing
Reproduction Steps

S1: [press] ‘Go to…’
S2: [type] [number]
S3: [press] ‘Create’

2) Execution Model Generation

Android App
Automated Exploration
Execution Model

3) Quality Assessment

S2R Matching
S2R Inference
Random Exploration

Bug Report
Reproduction Steps

1) S2R Identification
Neural Sequence Labeling + Dependency Parsing
Reproduction Steps

S1: [press] ‘Go to…’
S2: [type] [number]
S3: [press] ‘Create’

Android App
Automated Exploration
Execution Model

# | S2R | Quality Annotations
---|---|---
1 | S1 | EM
2 | S2 | AS
3 | S3 | VM
EXECUTION MODEL GENERATION

Application Installer
Execution Model Generation

Application Installer

Execution Engine
**Execution Model Generation**

Application Installer → Execution Engine → Execution Database
EXECUTION MODEL GENERATION

Application Installer → Execution Engine → Execution Database → Execution Model
EXECUTION DATABASE TRANSFORMATION

Execution database ➔ Execution Model (EM)

\[ EM = (V, E) \]
Execution Database Transformation

**Execution database**

<table>
<thead>
<tr>
<th>Execution Model (EM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Database icon]</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

\[ EM = (V, E) \]

\[ V: \text{App screens} \]

\[ v \in V = (GC, H) \]

*GC*: Set of GUI-components

*H*: Component hierarchy
Execution Database Transformation

**Execution database**

- **V**: App screens
  - $v \in V = (GC, H)$
  - $GC$: Set of GUI-components
  - $H$: Component hierarchy

**Execution Model (EM)**

- **E**: App interactions
  - $e \in E = (v_x, e, c, v_y)$
  - $v_x$: Source screen
  - $e$: Event (tap, swipe, etc.)
  - $c$: Component from $v_x$
  - $v_y$: Target screen

$EM = (V, E)$
Euler’s Components & Workflow

1) S2R Identification
   - Bug Report
   - Neural Sequence Labeling + Dependency Parsing
   - Reproduction Steps
   - S1: [press] ‘Go to…’
   - S2: [type] [number]
   - S3: [press] ‘Create’
   - …

2) Execution Model Generation
   - APK
   - Automated Exploration
   - Execution Model

3) Quality Assessment
   - S2R Matching
   - S2R Inference
   - Random Exploration

Bug Report
Neural Sequence Labeling + Dependency Parsing
Reproduction Steps
S1: [press] ‘Go to…’
S2: [type] [number]
S3: [press] ‘Create’
…

Android App
Automated Exploration
Execution Model

<table>
<thead>
<tr>
<th>#</th>
<th>S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S1</td>
<td>EM MS</td>
</tr>
<tr>
<td>2</td>
<td>S2</td>
<td>AS</td>
</tr>
<tr>
<td>3</td>
<td>S3</td>
<td>VM</td>
</tr>
</tbody>
</table>
Euler’s Components & Workflow

1) S2R Identification
   - Bug Report
   - Neural Sequence Labeling + Dependency Parsing
   - Reproduction Steps
   - S1: [press] ‘Go to…’
   - S2: [type] [number]
   - S3: [press] ‘Create’
   - …

2) Execution Model Generation
   - APK
   - Automated Exploration
   - Execution Model

3) Quality Assessment
   - S2R Matching
   - S2R Inference
   - Random Exploration

Bug Report
Reproduction Steps
1) S2R Identification
Neural Sequence Labeling + Dependency Parsing
Reproduction Steps
S1: [press] ‘Go to…’
S2: [type] [number]
S3: [press] ‘Create’
…

Android App
Automated Exploration
Execution Model

<table>
<thead>
<tr>
<th>#</th>
<th>S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S1</td>
<td>EM MS</td>
</tr>
<tr>
<td>2</td>
<td>S2</td>
<td>AS</td>
</tr>
<tr>
<td>3</td>
<td>S3</td>
<td>VM</td>
</tr>
</tbody>
</table>
COMPONENT RESOLUTION & QUERY FORMULATION

Individual S2R

S₁: [press] ['Go to…']
S₁: [enter] ['10'] [on] [price]
S₁: [set] [price] [to] ['10']

Queries

q₁: press + Go to...
q₂: Go to...
q₁: price

Matching Algorithm

Resolved Component
Matched S2Rs to app interactions

[press] ['Go to...']
[type] [number]
[press] ['Go']
S2R Execution & Inference

Matched S2Rs to app interactions

[press] ['Go to…‘]
[type] [number]
[press] ['Go’]

Execution Paths
S2R Execution & Inference

Matched S2Rs to app interactions:
- [press] ['Go to...'
- [type] [number]
- [press] ['Go']

Execution Paths:

Inferred Missing Steps:
- [press]['Go to...'
- [scroll down]
- [tap][‘Enter number’]
- [type][number]
- [scroll up]
- [press][‘Go’]
Matched S2Rs to app interactions

[p]press[/p][‘Go to…’]
[ty]type[/ty][number]
[p][press]‘Go’[/press]
[sc]scroll down[/sc]

Quality Report

<table>
<thead>
<tr>
<th>#</th>
<th>S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S₁</td>
<td>EM, MS</td>
</tr>
<tr>
<td>2</td>
<td>S₂</td>
<td>AS</td>
</tr>
<tr>
<td>3</td>
<td>S₃</td>
<td>VM</td>
</tr>
</tbody>
</table>
# EULER S2R QUALITY REPORT

### Legend for the Quality Annotations

- **EM**: Exact Match
- **AS**: Ambiguous Step
- **VM**: Vocabulary Mismatch
- **MS**: Missing Steps

<table>
<thead>
<tr>
<th>#</th>
<th>Identified S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
</table>
| 1  | Add favorites           | **EM**: This S2R matches the following app interaction: 1. Tap the "item fav (Add to favorites)" text view  
                                  | **MS**: There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the image button  
                                  | 2. Tap the "Chaos Communication Camp Opening" view                                                                                                 |
| 2  | Go into favorites       | **EM**: This S2R matches the following app interaction: 1. Tap the "item starred list (Show favorites)" text view  
                                  | **MS**: There are app interactions that are missing in the bug report and should be executed before this S2R: 1. Tap the "Navigate up" image button  
                                  | 2. Tap the image button  
                                  | 3. Tap the drop down list  
                                  | 4. Tap the list view                                                                                                                                |
| 3  | Select event            | **AS**: This S2R matches multiple actions (e.g., "long click" or "click").                                                                                                                                 |
| 4  | Remove event in event details screen | **VM**: The term "event in event details screen" does not match a GUI component from the app.                                                        |
| 5  | Hit BACK button to return | **EM**: This S2R matches the following app interaction: 1. Tap the back button                                                                 |


# EULER S2R Quality Report

Legend for the Quality Annotations

- Exact Match: EM
- Ambiguous Step: AS
- Vocabulary Mismatch: VM
- Missing Steps: MS

<table>
<thead>
<tr>
<th>#</th>
<th>Identified S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add favorites</td>
<td>EM: This S2R matches the following app interaction:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the &quot;item fav (Add to favorites)&quot; text view</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MS: There are app interactions that are missing in the bug report and should be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>executed before this S2R:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the image button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Tap the &quot;Chaos Communication Camp Opening&quot; view</td>
</tr>
<tr>
<td>2</td>
<td>Go into favorites</td>
<td>EM: This S2R matches the following app interaction:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the &quot;item starred list (Show favorites)&quot; text view</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MS: There are app interactions that are missing in the bug report and should be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>executed before this S2R:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the &quot;Navigate up&quot; image button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Tap the image button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Tap the drop down list</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Tap the list view</td>
</tr>
<tr>
<td>3</td>
<td>Select event</td>
<td>AS: This S2R matches multiple actions (e.g., &quot;long click&quot; or &quot;click&quot;).</td>
</tr>
<tr>
<td>4</td>
<td>Remove event in event details screen</td>
<td>VM: The term &quot;event in event details screen&quot; does not match a GUI component from the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>app.</td>
</tr>
<tr>
<td>5</td>
<td>Hit BACK button to return</td>
<td>EM: This S2R matches the following app interaction:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the back button</td>
</tr>
</tbody>
</table>
### Legend for the Quality Annotations

- **Exact Match**: EM
- **Ambiguous Step**: AS
- **Vocabulary Mismatch**: VM
- **Missing Steps**: MS

### Identified S2R | Quality Annotations
---
1. **Add favorites** | EM: This S2R matches the following app interaction:
   1. Tap the "item fav (Add to favorites)" text view
   MS: There are app interactions that are missing in the bug report and should be executed before this S2R:
   1. Tap the image button
   2. Tap the "Chaos Communication Camp Opening" view
2. **Go into favorites** | EM: This S2R matches the following app interaction:
   1. Tap the "item starred list (Show favorites)" text view
   MS: There are app interactions that are missing in the bug report and should be executed before this S2R:
   1. Tap the "Navigate up" image button
   2. Tap the image button
   3. Tap the drop down list
   4. Tap the list view
3. **Select event** | AS: This S2R matches multiple actions (e.g., "long click" or "click").
4. **Remove event in event details screen** | VM: The term "event in event details screen" does not match a GUI component from the app.
5. **Hit BACK button to return** | EM: This S2R matches the following app interaction:
   1. Tap the back button
## EULER S2R QUALITY REPORT

### Legend for the Quality Annotations

- **EM**: Exact Match
- **AS**: Ambiguous Step
- **VM**: Vocabulary Mismatch
- **MS**: Missing Steps

### Quality Annotations

<table>
<thead>
<tr>
<th>#</th>
<th>Identified S2R</th>
<th>Quality Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add favorites</td>
<td><strong>EM</strong> This S2R matches the following app interaction:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the &quot;item fav (Add to favorites)&quot; text view</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MS</strong> There are app interactions that are missing in the bug report and should be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>executed before this S2R:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the image button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Tap the &quot;Chaos Communication Camp Opening&quot; view</td>
</tr>
<tr>
<td>2</td>
<td>Go into favorites</td>
<td><strong>EM</strong> This S2R matches the following app interaction:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the &quot;item starred list (Show favorites)&quot; text view</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MS</strong> There are app interactions that are missing in the bug report and should be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>executed before this S2R:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the &quot;Navigate up&quot; image button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Tap the image button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Tap the drop down list</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Tap the list view</td>
</tr>
<tr>
<td>3</td>
<td>Select event</td>
<td><strong>AS</strong> This S2R matches multiple actions (e.g., &quot;long click&quot; or &quot;click&quot;).</td>
</tr>
<tr>
<td>4</td>
<td>Remove event in event details screen</td>
<td><strong>VM</strong> The term &quot;event in event details screen&quot; does not match a GUI component from the app.</td>
</tr>
<tr>
<td>5</td>
<td>Hit BACK button to return</td>
<td><strong>EM</strong> This S2R matches the following app interaction:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the back button</td>
</tr>
</tbody>
</table>
EULER S2R QUALITY REPORT

EULER's Quality Report

1 Bug Report Information

Application: Time Tracker v. 0.20 Bug report id: #46
Bug summary: Dialog Disappears on Device Rotation 3
Bug description: When I perform this sequence of events:
1. Click on OK
2. Click on "..." (More Options...)
3. Click on Add Task
4. Insert text "testtask"
5. Click on Add Task
6. Long Click on Element in List
7. Click on Edit Task
8. Orientation Change
   The dialog on screen disappears.

2 Quality Assessment of the Bug Report

Legend for the Quality Annotations

|-----------------|--------------------|------------------------|--------------------------|------------------|

<table>
<thead>
<tr>
<th># Identified Step to Reproduce (S2R)</th>
<th>Quality Annotations for the S2R</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Click on OK</td>
<td>EM This S2R matches the following app interaction:</td>
</tr>
<tr>
<td></td>
<td>1. Tap the &quot;button 1 (OK)&quot; button</td>
</tr>
<tr>
<td></td>
<td>MS There are app interactions that are missing in the bug report and should be executed before this S2R:</td>
</tr>
<tr>
<td></td>
<td>1. Tap the &quot;usage (Detailed instructions are at <a href="https://github.co">https://github.co</a>...)&quot; text view</td>
</tr>
<tr>
<td></td>
<td>2. Long tap the &quot;usage (Detailed instructions are at <a href="https://github.co">https://github.co</a>...)&quot; text view</td>
</tr>
<tr>
<td></td>
<td>3. Long tap the &quot;credits (Credits Based on TimeTracker+ by Sean E. Rus...)&quot; text view</td>
</tr>
<tr>
<td></td>
<td>4. Tap the &quot;credits (Credits Based on TimeTracker+ by Sean E. Rus...)&quot; text view</td>
</tr>
</tbody>
</table>
EULER's Quality Report

1 Bug Report Information

**Application:** Time Tracker v. 0.20 **Bug report id:** #46

**Bug summary:** Dialog Disappears on Device Rotation 3

**Bug description:** When I perform this sequence of events:

1. Click on OK
2. Click on "..." (More Options...)
3. Click on Add Task
4. Insert text "testtask"
5. Click on Add Task
6. Long Click on Element in List
7. Click on Edit Task
8. Orientation Change
   
The dialog on screen disappears.

2 Quality Assessment of the Bug Report

Legend for the Quality Annotations

- Exact Match (EM)
- Ambiguous Step (AB)
- Vocabulary Mismatch (VM)
- Incorrect Input Value (IV)
- Missing Steps (MS)

<table>
<thead>
<tr>
<th>#</th>
<th>Identified Step to Reproduce (S2R)</th>
<th>Quality Annotations for the S2R</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on OK</td>
<td>EM This S2R matches the following app interaction:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the &quot;button 1 (OK)&quot; button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MS There are app interactions that are missing in the bug report and should be executed before this S2R:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Tap the &quot;usage (Detailed instructions are at <a href="https://github.co">https://github.co</a>...)&quot; text view</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Long tap the &quot;usage (Detailed instructions are at <a href="https://github.co">https://github.co</a>...)&quot; text view</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Long tap the &quot;credits (Credits Based on TimeTracker+ by Sean E. Rus...)&quot; text view</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Tap the &quot;credits (Credits Based on TimeTracker+ by Sean E. Rus...)&quot; text view</td>
</tr>
</tbody>
</table>
Evaluating Euler

24 bug reports:
• 5 crashes
• 14 functional
• 5 look-n-feel

6 Android apps:
• GnuCash
• Mileage
• Schedule
• ...

12 evaluators:
• MS and PHD students
• B. Analyst
• ...

24 bug reports: 6 Android apps: 12 evaluators:
RESEARCH QUESTIONS

**RQ₁:** What is the accuracy of Euler in Identifying and Assessing S2Rs?

**RQ₂:** How useful are Euler's Reports?
Generating Ideal Reproduction Scenarios

Bug Reports
Generating Ideal Reproduction Scenarios

- Bug Reports
- Bug Replication
- Reproduction Scenarios
- Video Recordings
EXTERNAL EVALUATION OF EULER

Video Recordings

Quality Reports
EXTERNAL EVALUATION OF EULER

Video Recordings

Evaluators

Quality Reports

- Correct/Incorrect S2Rs and Annotations
- Usefulness/Quality of the Feedback
Euler’s Accuracy

- Identified S2Rs: **85/89 (98%)** are correctly identified
- Quality annotations: **91/124 (73%)** are correct
- Inferred missing S2Rs:
  - **92/158 (58%)** are correctly inferred
  - 5/7 steps for bug report (on avg)
Euler’s Accuracy

- Identified S2Rs: 85/89 (98%) are correctly identified
- Quality annotations: 91/124 (73%) are correct
- Inferred missing S2Rs:
  - 92/158 (58%) are correctly inferred
  - 5/7 steps for bug report (on avg)

Euler assesses the quality of S2Rs with high accuracy
Euler’s Accuracy

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp.</td>
<td>7%</td>
</tr>
<tr>
<td>Useful</td>
<td>15%</td>
</tr>
</tbody>
</table>

Response:
- Disagree
- Somewhat disagree
- Not agree nor disagree
- Somewhat agree
- Agree

- 87% Agree
- 83% Agree
- 1% Not agree nor disagree
- 6% Somewhat agree
Euler’s quality reports are both easy to understand and would prove helpful when writing bug reports.
PARTICIPANT FEEDBACK
“EULER’s S2Rs are pretty descriptive and would guide the user to complete better the bug description.”

“Developers/maintainers would find this tool *very* useful for their debugging process.”

“EULER correctly flags words such as ‘find’ and ‘fix’ that do not directly translate to an app action.”

“[The missing steps] help avoid the guessing part when reproducing the bug.”
Steps to Reproduce are Often…

- Missing
- Ambiguous
- Incomplete

**Steps to Reproduce are Often...**

- Missing
- Ambiguous
- Incomplete

---

**Euler’s Components & Workflow**

1) **S2R Identification**

   - Bug Report
   - Neural Sequence Labeling + Dependency Parsing
   - Reproduction Steps

2) **Execution Model Generation**

   - APK
   - Automated Exploration
   - Execution Model

3) **Quality Assessment**

   - S2R Matching
   - S2R Inference
   - Random Exploration

---

STEPS TO REPRODUCE ARE OFTEN...

- Missing
- Ambiguous
- Incomplete


EULER’S COMPONENTS & WORKFLOW

1) S2R Identification
   - Bug Report
   - Neural Sequence Labeling + Dependency Parsing
   - Reproduction Steps

2) Execution Model Generation
   - APK
   - Automated Exploration
   - Execution Model

3) Quality Assessment
   - S2R Matching
   - S2R Inference
   - Random Exploration

EXTERNAL EVALUATION OF EULER

- Video Recordings
- Evaluators
- Quality Reports

- Correct/Incorrect S2Rs and Annotations
- Usefulness/Quality of the Feedback
**Steps to Reproduce are Often…**

- Missing
- Ambiguous
- Incomplete


**Euler’s Components & Workflow**

1) S2R Identification
   - Bug Report
   - Neural Sequence Labeling + Dependency Parsing
   - Reproduction Steps

2) Execution Model Generation
   - APK
   - Automated Exploration
   - Execution Model

3) Quality Assessment
   - S2R Matching
   - S2R Inference
   - Random Exploration

### External Evaluation of Euler

- Video Recordings
- Evaluators
- Quality Reports

- Correct/Incorrect S2Rs and Annotations
- Usefulness/Quality of the Feedback

### Euler’s Accuracy

- Identified S2Rs: 85/89 (98%) are correctly identified
- Quality annotations: 91/124 (73%) are correct
- Inferred missing S2Rs:
  - 92/158 (58%) are correctly inferred

Euler assesses the quality of S2Rs with high accuracy
Steps to Reproduce are Often...

- Missing
- Ambiguous
- Incomplete

Euler’s Components & Workflow

1) S2R Identification
   - Bug Report
   - Neural Sequence Labeling + Dependency Parsing
   - S2R: [press] ['Go to...']
   - S2R: [type] ['number']
   - S2R: [press] ['Create']

2) Execution Model Generation
   - Android App
   - Automated Exploration
   - Execution Model

3) Quality Assessment
   - S2R Matching
   - S2R Inference
   - Random Exploration

External Evaluation of Euler

- Video Recordings
- Evaluators
- Quality Reports

- Correct/Incorrect S2Rs and Annotations
- Usefulness/Quality of the Feedback

Euler’s Accuracy

- Identified S2Rs: 85/89 (98%) are correctly identified
- Quality annotations: 91/124 (73%) are correct
- Inferred missing S2Rs:
  - 92/158 (58%) are correctly inferred
  - 5/7 steps for bug report (on avg)

Euler assesses the quality of S2Rs with high accuracy

Additional Slides
**Individual S2R**

$S_1$: [press] ['Go to...']

**App state (screen)**

$v_x \in V = (GC, H)$

**Resolved interaction**

$e_1 \in E = (v_x, e, c, v_y = null)$

$S_1$: [tap] on ['Go to...'] button
**Textual Similarity Measure**

\[
similarity(s_1, s_2) = \frac{|LCS(s_1, s_2)|}{\text{avg}(|s_1|, |s_2|)}
\]

- \( s_1 \) and \( s_2 \) are two **term sequences**
- \( LCS(s_1, s_2) \) is the **Longest Common Substring** (at term level)
- \( \text{avg}(|s_1|, |s_2|) \) is the **average length** of both sequences
GUI Component Resolution Example

Individual S2R

\[ S_1: \text{[search]} \ [\text{number}] \]

\[
\text{similarity('search number', 'search')} = \left| \text{'search'} \right| \left/ \text{avg(2,1)} \right. = \frac{1}{1.5} = 0.67
\]
SYSTEMATIC APP EXPLORATION

GUI Ripping → GUI hierarchy

Exploration Strategies

Top-down

Bottom-up

Execution database:
• App activities/screens
• Sequential GUI interactions
**Algorithm 1: Matching Algorithm**

**Data:** Query \((q)\), list of GUI components \((GC)\)

**Result:** Matched GUI component \((c)\), or mismatch/multiple-match result \((r)\)

```plaintext
Function execute_query(q, GC)
    gc_candidates = [];
    foreach c in GC do
        S = compute_similarity(q, c);
        if S ≥ 0.5 then
            gc_candidates.add(c)
    end
    if gc_candidates.size() == 1 then
        c = gc_candidates.first();
        return c;
    if gc_candidates.size() > 1 then
        return execute_heuristics(gc_candidates);
    if gc_candidates.size() == 0 then
        q' = reformulate_query(q);
        return execute_query(q', GC)
```
**Component Matching Algorithm**

For each component in the screen:

- Compute its textual similarity with the query
- If the similarity is high enough (>0.5)
  - The component is selected as candidate

Return the top candidate or reformulate the query using synonyms
**S2R Matching**

- **Individual S2R**
  - [press] ['Go to...']
  - [type] [number]
  - [press] ['Go']

- **Execution Model**

- **Matched S2Rs to app interactions**
  - [press] ['Go to...']
  - [type] [number]
  - [press] ['Go']
S2R Matching Algorithm

- Starting from a state/screen in the graph:
  - Select the adjacent states within n levels
  - Resolve the S2R on each state
  - Match the resolved actions against the graph transitions/interactions
  - Return the closest matched interaction to the starting state
S2R Matching Algorithm

• Starting from a state/screen in the graph:
  • Select the adjacent states within n levels
  • Resolve the S2R on each state
  • Match the resolved actions against the graph transitions/interactions
• Return the closest matched interaction to the starting state