Exploring Developers Rationale in the Linux Kernel

Mouna Dhaouadi
3rd year PhD Candidate
Université de Montréal, Canada
http://www-labs.iro.umontreal.ca/~dhaouadm/
October 21st, 2023
Motivation:

For a decision, software/system developer has rationale
‘Why’ reasoning for their decision

Rationale is *useful information* to:
understand the system,
learn from mistakes,
reuse solutions,
avoid conflicts
Real World Example

- Out-Of-Memory Killer (OOM-Killer) component
- Manually selected interesting commits from the Git history of OOM-Killer.
- Topic of "reclaiming used memory from the OOM victim".

Suren’s Challenge:
How does my decision impact previously established decisions? How to make sure I will not cause conflicts with existing rationales?

**Oom: give the dying task a higher priority**
August 9th, 2010

**memcg: give current access to memory reserves if it's trying to die**
March 23rd, 2011

**mm : introduce process_mrelease system call**
September 2nd, 2021

**mm, oom: introduce oom reaper**
March 25th, 2016

End-to-End Rationale Reconstruction, Mouna Dhaouadi, Bentley James Oakes, Michalis Famelis. The IEEE/ACM International Conference on Automated Software Engineering (ASE), Michigan, USA, 2022. NIER Track
How can we *structure*, *extract* and *manage* rationale information from code commits?
Proposed Solution

- A complete rationale extraction and management system

Rationale in Commits:

“Fix: Added check to prevent null pointer exception.”

versus

“changes”

Challenges:

◦ Implicit or unrecorded
◦ Ambiguous language
  • Especially past/future tenses
  • Non-native writers
◦ Need technical understanding
Plan

1. Motivation
2. Creating a Dataset
3. Information Inference Component
4. Response Generation Component
5. Conclusion and Future Work
Information Inference Component

- Need ground truth for the automatic classification
- Gain insights about the existing rationale

Dataset

Manual labelling of OOM-Killer commit sentences.

Procedure:
- Collect 418 commits
- Remove merge commits, filter code sentences
- Three authors label sentences
- Resolve conflicts in discussion
Codebook

- Obtained through piloting rounds and discussions
- **Multiple classifications per sentence**
- In disagreement, take classification union
- Fleiss kappa: Around 0.65 (fair to good agreement)
### Labelling Example

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Labelling</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm, oom: introduce independent oom killer ratelimit state</td>
<td>Decision</td>
</tr>
<tr>
<td>printk_ratelimit() uses the global ratelimit state for all printk</td>
<td>Supporting Facts</td>
</tr>
<tr>
<td>The oom killer should not be subjected to this state just because another subsystem or driver may be flooding the kernel log</td>
<td>Rationale</td>
</tr>
<tr>
<td>This patch introduces printk ratelimiting specifically for the oom killer</td>
<td>Decision</td>
</tr>
</tbody>
</table>
Labelling Insights

- 98.9% of commits contain rationale
- About 60% of sentences per commit contain rationale

Distribution of the sentences in the OOM dataset

Labelling Insights

Common Structure:
Decisions -> Supporting Facts -> Rationale -> Decisions

Distribution of the categories over the normalized positions of the sentences of the commit messages

Plan

1. Motivation
2. Creating a Dataset
3. Information Inference Component
4. Response Generation Component
5. Conclusion and Future Work
Information Inference Component

Kantara Pipeline

Decisions and Rationale Graph

Validation Mechanisms

Analysis Techniques

Retrieval System

Tool

Rationale Extraction

Rationale Structuring

Information Inference Component

Response Generation Component

Kantara: Information Extraction pipeline

End-to-End Rationale Reconstruction, Mouna Dhaouadi, Bentley James Oakes, Michalis Famelis. The IEEE/ACM International Conference on Automated Software Engineering (ASE), Michigan, USA, 2022. NIER Track
Prototype to evaluate feasibility of Kantara

- Evaluated on the Out-Of-Memory Killer (OOM-Killer) example
Automatic sentence classification

Binary classification: Logistic regression, decision tree, SVM
Multi-label classification: Random Forest, XGBoost, KNN

<table>
<thead>
<tr>
<th>XGBoost classification evaluation</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision</td>
<td>0.76</td>
<td>0.69</td>
<td>0.72</td>
</tr>
<tr>
<td>Rationale</td>
<td>0.62</td>
<td>0.41</td>
<td>0.49</td>
</tr>
<tr>
<td>Supporting Facts</td>
<td>0.64</td>
<td>0.68</td>
<td>0.66</td>
</tr>
</tbody>
</table>
Information Inference Component

Kantara Pipeline

Rationale Extraction

Decisions and Rationale Graph

Validation Mechanisms

Analysis Techniques

Retrieval System

Tool

Response Generation Component

Information Inference Component

Knowledge Graph

- Nodes: decisions and rationales
- Relationships: relatedness, similar, contradicts, history

"reduces the probability of such a lockup"

**Decision #D4**
Introduce OOM reaper.
Date = March 25th, 2016

"memory is freed in a more controllable way with CPU affinity and priority of the caller"

**Decision #D5**
Introduce process_mrelease system call that releases memory of a dying process from the context of the caller.
Date = September 2nd, 2021

"so that it may quickly exit and free its memory"

**Decision #D2**
give current access to memory reserves if it's trying to die.
Date = March 23rd, 2011

"so that it can exit() soon, freeing memory"

**Decision #D3**
I and the original author Luis agreed to disable this logic.
Date = April 14th, 2011

"Eventually, kernel may hang up when oom kill occur"

**Decision #D1**
give the dying task a higher priority.
Date = August 9th, 2010

"so that it may quickly exit and free its memory"

"Eventually, kernel may hang up when oom kill occur"

End-to-End Rationale Reconstruction, Mouna Dhaouadi, Bentley James Oakes, Michalis Famelis. The IEEE/ACM International Conference on Automated Software Engineering (ASE), Michigan, USA, 2022. NIER Track
Commit-level

Towards Understanding and Analyzing Rationale in Commit Messages using a Knowledge Graph Approach, Mouna Dhaouadi, Bentley James Oakes, Michalis Famelis. 5th Workshop on Artificial Intelligence and Model-driven Engineering (MDE-Intelligence), Sweden, 2023.
Plan

1. Motivation
2. Creating a Dataset
3. Information Inference Component
4. Response Generation Component
5. Conclusion and Future Work
Response Generation Component

Validation Mechanisms - prototype

Look for potential conflicts when a new decision is proposed.

**Decision #D5**
Introduce process_mrelease system call that releases memory of a dying process from the context of the caller.
Date = September 2nd, 2021

**Decision #D3**
I and the original author Luis agreed to disable this logic.
Date = April 14th, 2011

**Decision #D1**
give the dying task a higher priority.
Date = August 9th, 2010

**Decision #D2**
give current access to memory reserves if it's trying to die.
Date = March 23rd, 2011

---

Suren's Challenge:
How does my decision impact previously established decisions? How to make sure I will not cause conflicts with existing rationales?
Rationale Analysis and Visualisation

1. Defined the ontology in the OpenCasear framework

2. Populated the tool with our dataset

Towards Understanding and Analyzing Rationale in Commit Messages using a Knowledge Graph Approach, Mouna Dhaouadi, Bentley James Oakes, Michalis Famelis. 5th Workshop on Artificial Intelligence and Model-driven Engineering (MDE-Intelligence), Sweden, 2023.
Rationale Analysis and Visualisation

Analysis: Inferencing using SPARQL queries

```json
{
  "author": { "value": "Michal Hocko" },
  "commit_id": { "value": "c0" },
  "text": { "value": "mm: reduce noise in show_mem for lowmem allocations" },
  "isCommitWithRationale": { "value": "true" },
  "isSentenceRationale": { "value": "true" },
  "isSentenceDecision": { "value": "true" },
  "isSentenceSupporting": { "value": "false" }
}
```
Rationale Analysis and Visualisation

**Visualisation:** Visualizing the results of some queries

- Identify commits/subsystems/developers with insufficient rationale

Towards Understanding and Analyzing Rationale in Commit Messages using a Knowledge Graph Approach, Mouna Dhaouadi, Bentley James Oakes, Michalis Famelis. 5th Workshop on Artificial Intelligence and Model-driven Engineering (MDE-Intelligence), Sweden, 2023.
Plan

1. Motivation
2. Creating a Dataset
3. Information Inference Component
4. Response Generation Component
5. Conclusion and Future Work
Conclusion and Future Work

Summary:
A rationale extraction and management system
Linux Kernel as object of the study

Future work:
◦ Improve automatic classification
◦ Kantara as a pull request bot
◦ Building a retrieval system