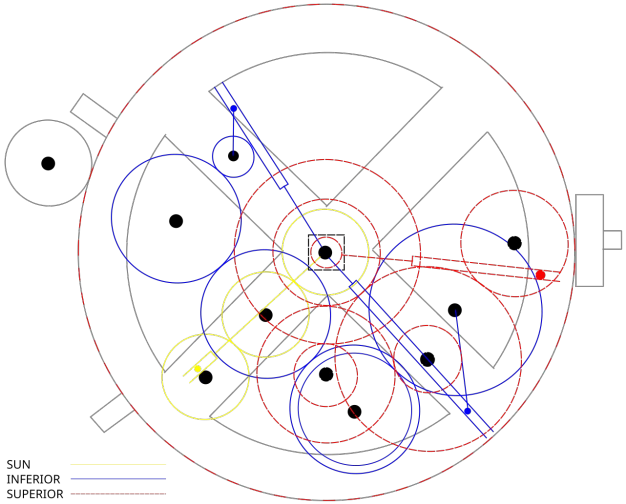
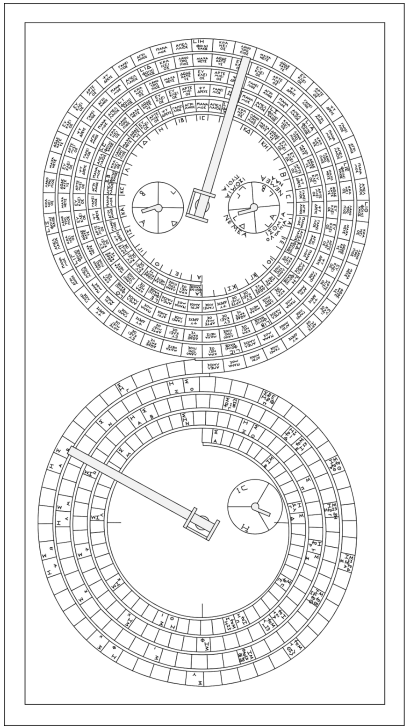
Automated Extraction and Analysis of Developer's Rationale in Open Source Software

Mouna Dhaouadi, Bentley Oakes and Michalis Famelis

FSE 2025 Rerouted Presentation





We basically *know* **what** the Antikythera Mechanism is.

We basically *know* **how** it works.

We can only *speculate* on **why** it was made.

We can only *speculate* on **why** it was made *that way.*

Design Knowledge **Erodes** and **Evaporates**

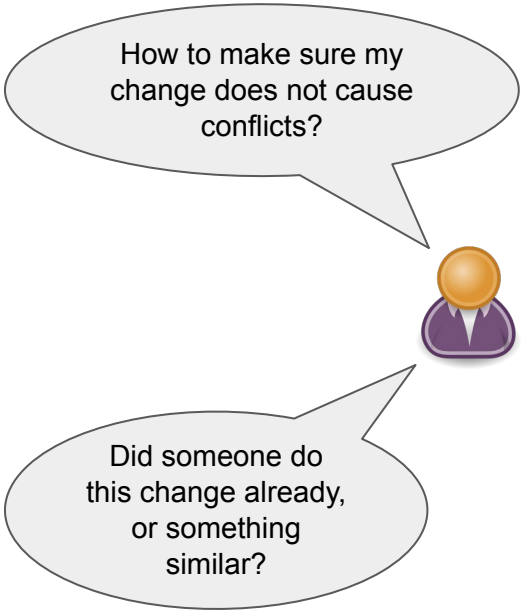
Arthur Evans reconstructed Knossos with reinforced concrete and colour beyond what could be securely known from the archaeological record.

Design Evaporation:

- People come and go
- People forget
- People miscommunicate
- The artifact remains

Design Erosion follows:

- System starts diverging from the original design
- New assumptions overwrite original intent



Robillard, FSE'16, Sustainable software design. https://doi.org/10.1145/2950290.2983983
Van Gurp and Bosch, JSS'02 Design erosion: problems and causes https://doi.org/10.1016/S0164-1212(01)00152-2

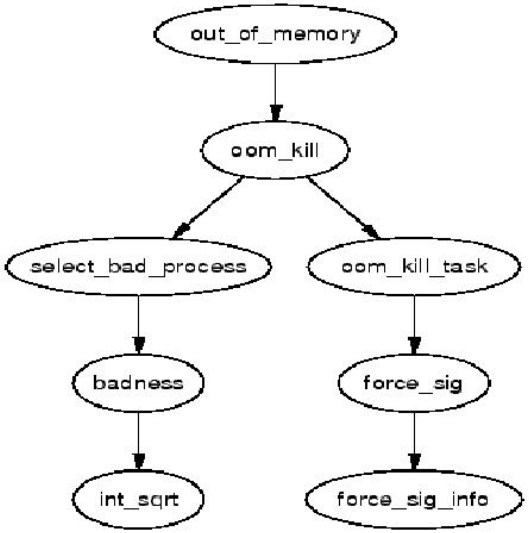
Linux Out-of-Memory Killer: An Example

Linux kernel:

- Development is through Git commits
- Culture for motivating/describing changes

Out-of-Memory Killer subsystem:

- When Linux runs out of memory, it calls OOM-Killer to avoid crashes
- Two broad steps:
  - Select “best” task to kill using heuristics
  - Force task to release memory and exit



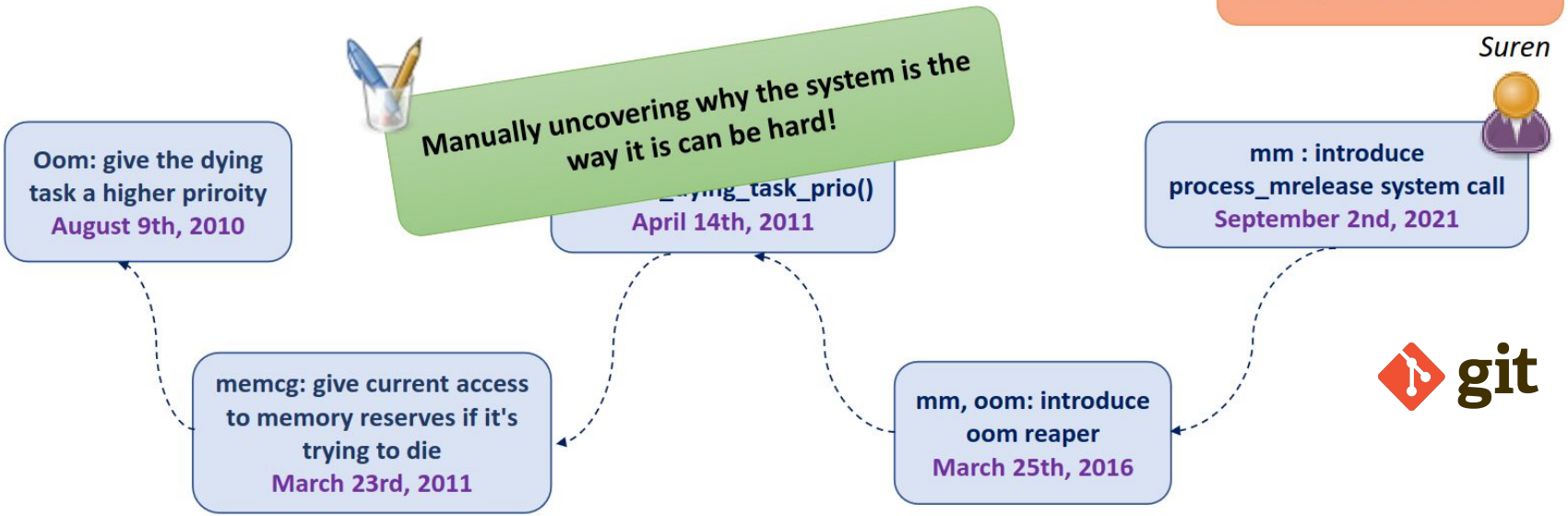
https://www.kernel.org/doc/gorman/html/understand/understand016.html

What is the impact of changes?

- Dev works on "*reclaiming used memory from the OOM victim*".
- Find interesting commits from the Git history of OOM-Killer.

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

Suren



Dhaouadi et al. , ASE'22, End-to-End Rationale Reconstruction https://doi.org/10.1145/3551349.3559547

Mining Commit Messages

- Commit Messages contain valuable information about developers rationale

- Rationale is the ***why*** behind decisions.

remove is_linear_pte() to simplify code

Decision **Rationale**

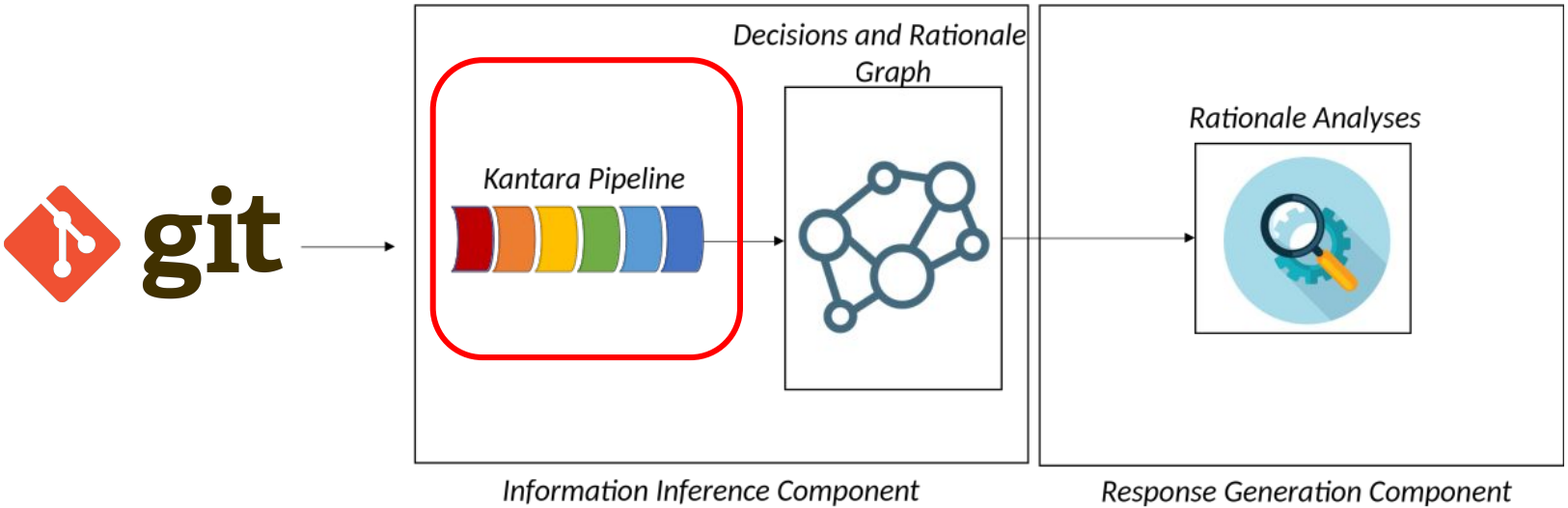
Learning from history:

- Reuse past solutions
- Avoid reasoning conflicts
- …

Dhaouadi et al. ICPC 2024 Rationale Dataset and Analysis for the Commit Messages of the Linux Kernel Out-of-Memory Killer https://doi.org/10.1145/3643916.3644413

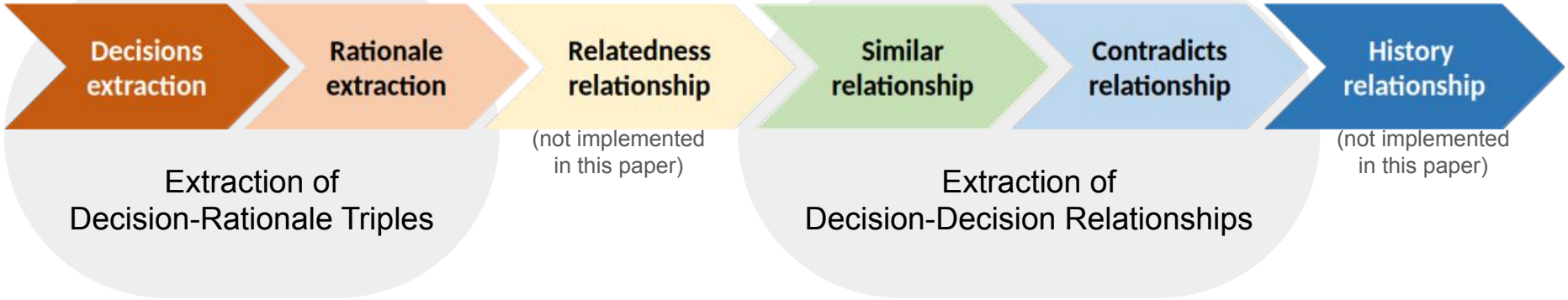
Rationale Extraction and Management

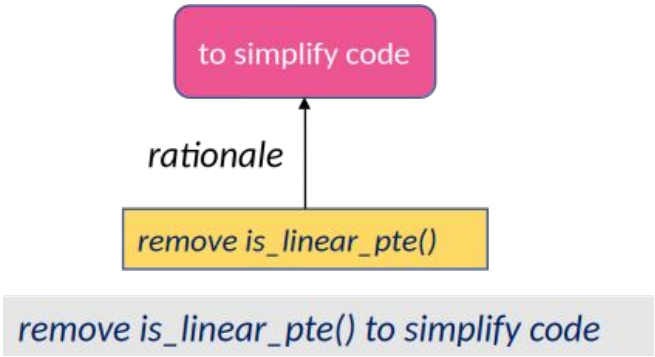
An On-Demand Developer Documentation (OD3) system:

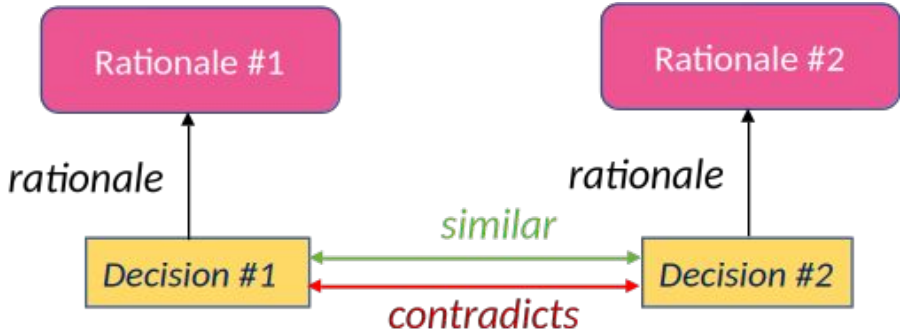


Robillard et al., ICSME'17, On-demand Developer Documentation https://doi.org/10.1109/ICSME.2017.17
Dhaouadi et al. , ASE'22, End-to-End Rationale Reconstruction https://doi.org/10.1145/3551349.3559547

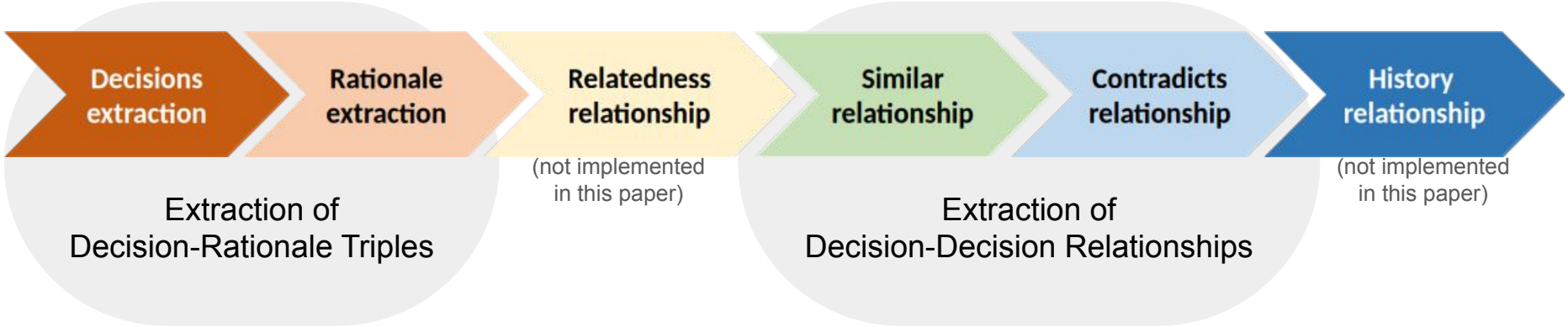
Kantara Pipeline







Kantara Pipeline on sentences containing **both** Decision and Rationale



Pinpointing:

- Bi-LSTM binary classifiers for D, R

Extraction:

- Attempted: Semantic Role Labeling
  misses more than half sentences with rationale
- GPT-3.5-turbo for D/R tuples

Pre-trained models:

- Similarity detection:
  distilbert-base-nli-mean-tokens
- Contradiction detection:
  roberta-large-mnli

Data Flow

Linux OOM Killer dataset:

- 404 commits
- 2234 manually labelled sentences
- Used to train D and R classifiers
  (10 fold cross validation, chose best)

Graph construction input:

- 774 sentences that have both D+R
- GPT-3.5-turbo extracts D/R tuples

Graph construction:

- 527 non-null D/R tuples
- Similarity/contradiction: compare 138 601 decision pairs
- Threshold 0.9

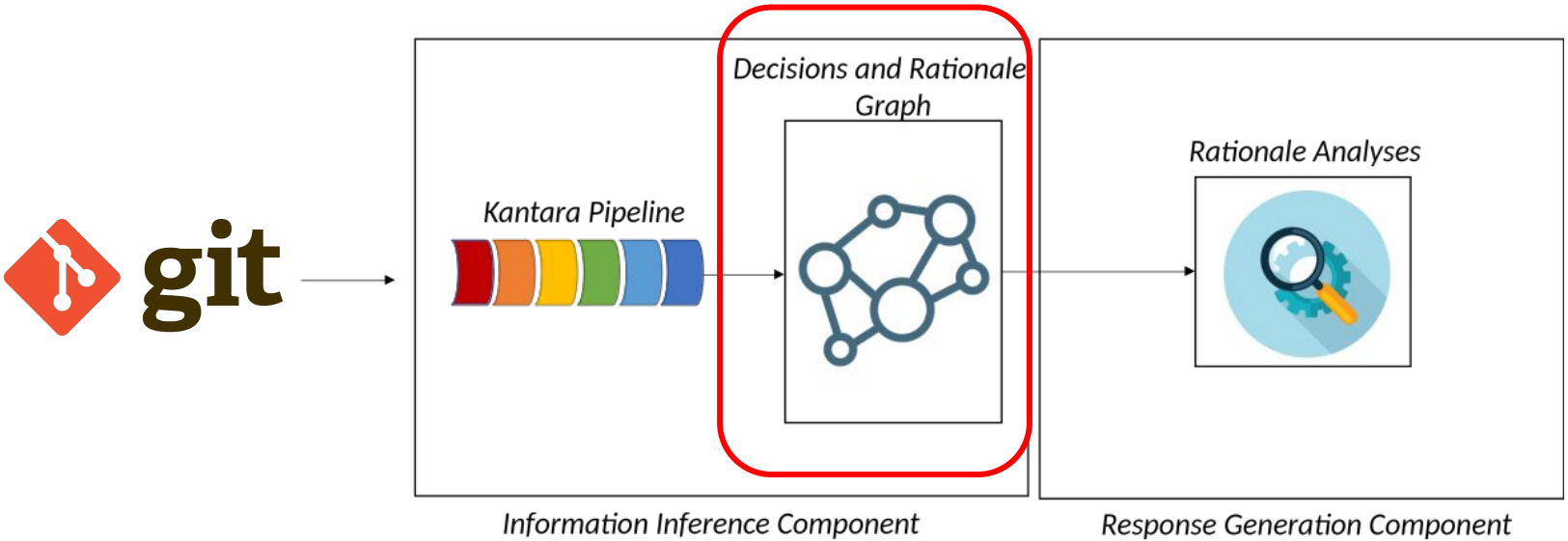
Generalization beyond the OOM Killer:

2 more Linux modules:
- slob.c
- acpi/button.c

Tian et al. open source dataset
- JUnit4
- Spring Boot
- Apache Dubbo
- Square Retrofit
- Square OkHttp

Rationale Extraction and Management

An On-Demand Developer Documentation (OD3) system:



Robillard et al., ICSME'17, On-demand Developer Documentation https://doi.org/10.1109/ICSME.2017.17
Dhaouadi et al. , ASE'22, End-to-End Rationale Reconstruction https://doi.org/10.1145/3551349.3559547

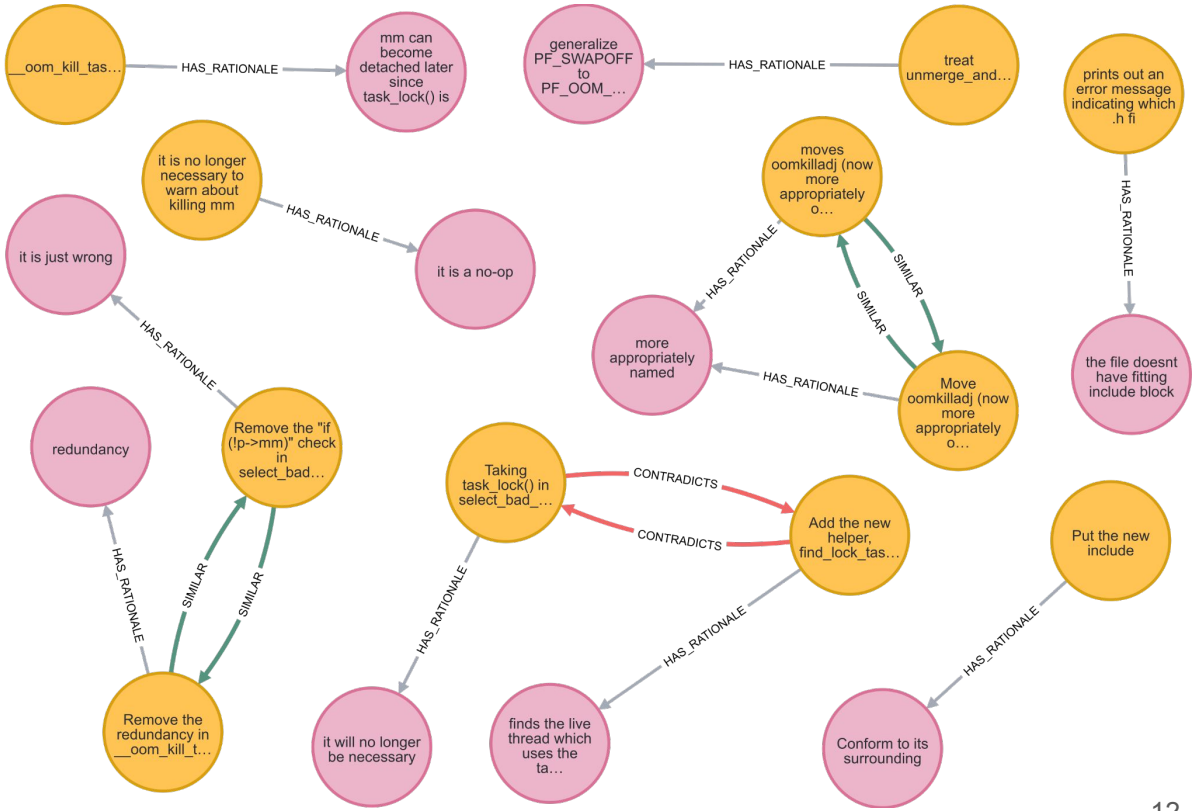
Decisions and Rationale Graph

For the OOM-Killer:

- 774 Decision+Rationale sentences
- 527 extracted decisions
- **138601 pairs** of decisions

- Compute relationships for each pair, with threshold 0.9

- 3 hours computation time



Learning from History: Reusing Solutions

Similar Decisions

Edges added when similarity score ≥ 0.9
29 similarity edges found for the OOM Killer

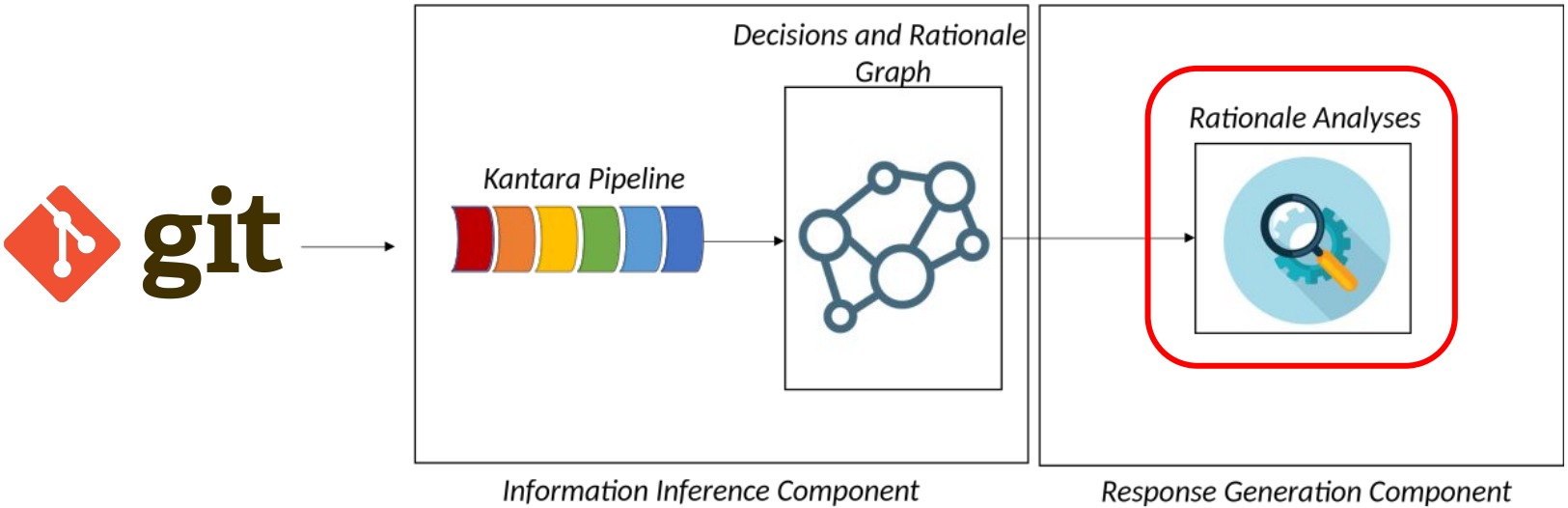
Learning from History: Avoiding Conflicts

Contradictory Decisions

Edges added when contradiction score ≥ 0.9
396 candidate contradiction edges found for the OOM Killer

Rationale Extraction and Management

An On-Demand Developer Documentation (OD3) system:

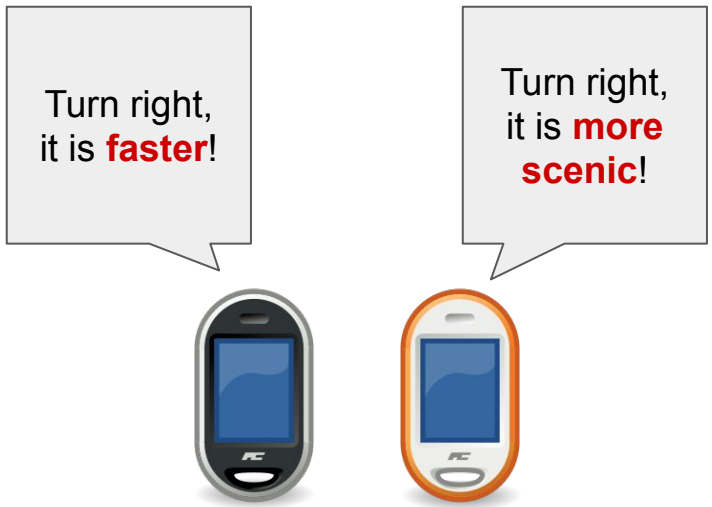


Robillard et al., ICSME'17, On-demand Developer Documentation https://doi.org/10.1109/ICSME.2017.17
Dhaouadi et al. , ASE'22, End-to-End Rationale Reconstruction https://doi.org/10.1145/3551349.3559547

Semantic Inconsistency Detection

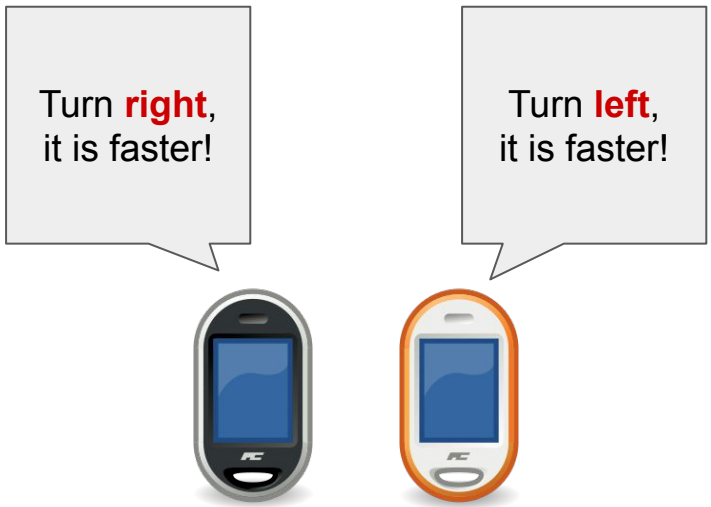
Analysis **M1**:

Similar Decisions and Contradicting Rationales



Analysis **M2**:

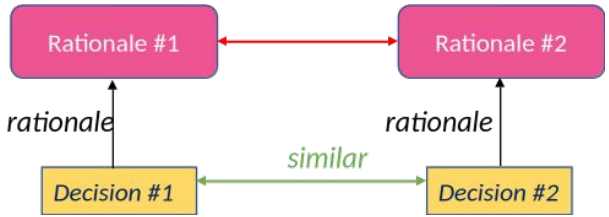
Contradictory Decisions and Similar Rationales



Benefits of Semantic Inconsistency Detection

Generate **inspection candidates** that reveal

- Miscommunications and inaccurate documentation
- Insufficient understanding of decision **impact**
- **Wrong** description of a patch in a commit
- Malicious attempt to hide **suspicious** contribs
- Conflicting **assumptions** across contributors

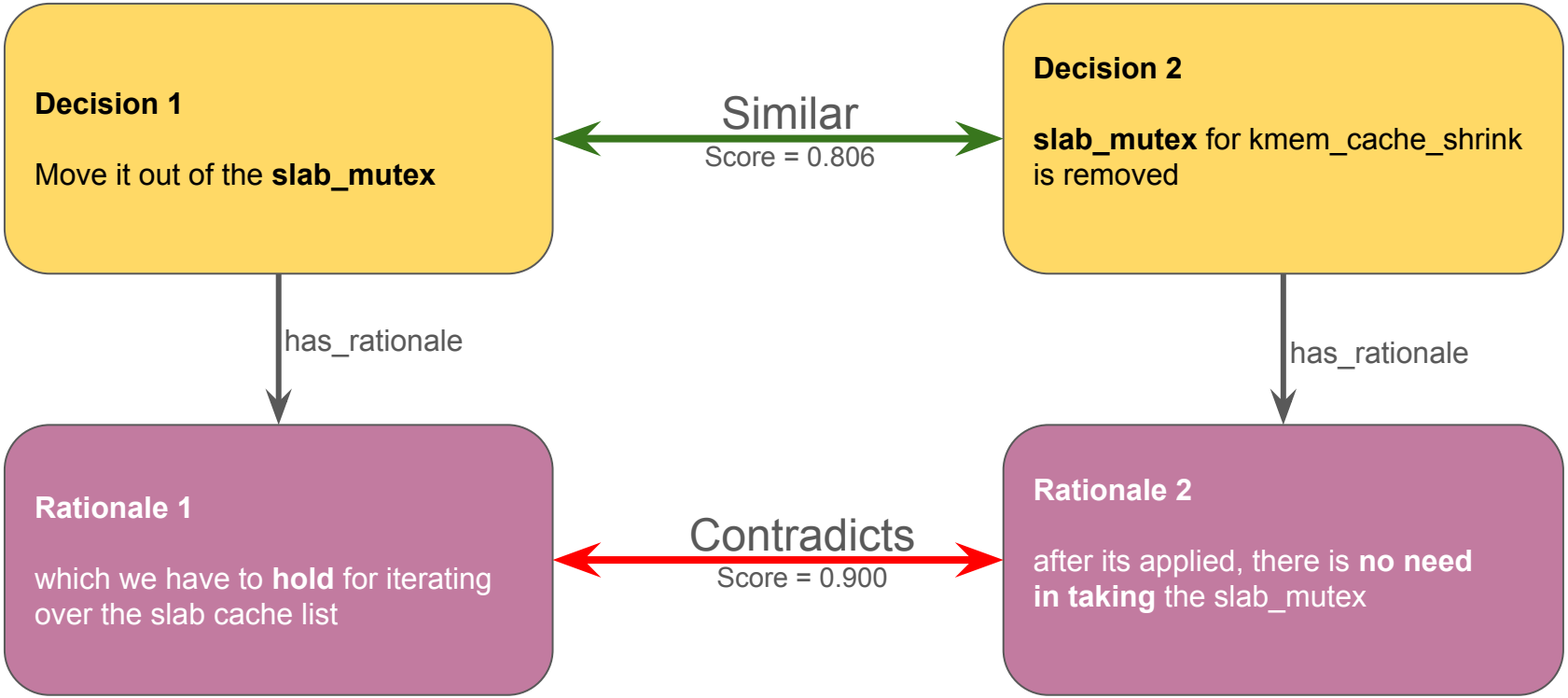


M1 analysis

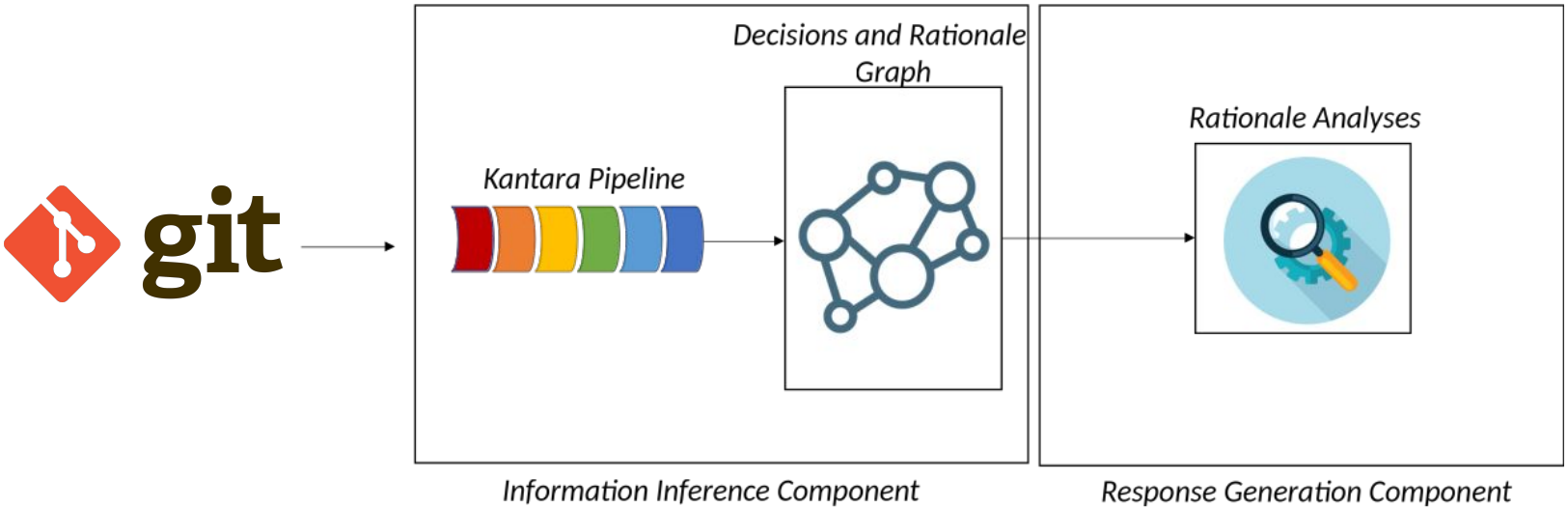
M1 example in the slob.c Linux module

Inspection candidate

Similar Decisions and Contradictory Rationales



But does it work?



Robillard et al., ICSME'17, On-demand Developer Documentation https://doi.org/10.1109/ICSME.2017.17
Dhaouadi et al. , ASE'22, End-to-End Rationale Reconstruction https://doi.org/10.1145/3551349.3559547

Evaluation Questions

1. Can we identify Decision and Rationale sentences?

2. Can we generate inspection candidates?

3. Can we generalize beyond the OOM Killer?

1. Can we identify Decision and Rationale sentences?

Yes

- 2234 **manually** labelled sentences from the OOM Killer
- Rationale: {1252 Y, 982 N}, Decision: {1343 Y, 891 N}
- Compared classic ML, deep learning, pretrained transformer models
- 10-fold cross-validation
- Best: **Bi-LSTM**.
- F1 scores: 0.93 for Decision, 0.91 for Rationale.
- Comparable to human raters.

2. Can we generate inspection candidates?

Yes

Offline / periodic

- 774 sentences from the OOM Killer, labelled with both D and R
- 527 non-null D/R tuples extracted using GPT-3.5-turbo
- 138 601 decision pairs compared by pretrained models on NVIDIA GeForce RTX 3090 GPU.
- About **3 hours** runtime
- Found 29 similarity edges, 396 candidate contradiction edges
- Manual inspection shows it found **interesting M1 and M2 examples** (see paper)

3. Can we generalize beyond the OOM Killer?
3a : unseen Linux modules

Other memory management module: **mm/slob.c**: 146 commits, 833 sentences

Unrelated module: **drivers/acpi/button.c**: 110 commits, 576 sentences

Preprocess like OOM Killer; then apply **Bi-LSTM**.

Manually validate sample of 20 commits.

Reasonable transfer

Humans agreed with predictions about **70–78%** of the time:

Reasonable transfer but not OOM-level certainty.

3. Can we generalize beyond the OOM Killer?
3b : other OSS projects

Classifier transfer:

Test classifiers on dataset by Tian et al.
("what"="D", "why"="R").
Overall F1 scores: 0.70 for D, 0.75 for R

Analysis:

Applied Kantara to D+R sentences
Found interesting cases:

- Number of M1 ∈ [3,14]
- Number of M2 ∈ [2, 15]

Project	D+R sentences
slob.c	252
button.c	191
JUnit4	145
Spring Boot	171
Apache Dubbo	138
Square Retrofit	151
Square OkHttp	137

Classifiers Reusable

Produced inspection candidates

Limitations and Takeaways

Kantara works for sentences with **both** D+R

We don’t handle **multi-sentence** rationale

Thresholds are heuristic

Extraction based on provider-**hosted**, **closed**-weight LLM (GPT-3.5-turbo)

LLM outputs not optimized and noisy

English-only / Linux-heavy

Further developer **validation** needed

Design evaporates and erodes but the past leaves traces.

Key ideas:

Use the traces to reconstruct decisions and rationales

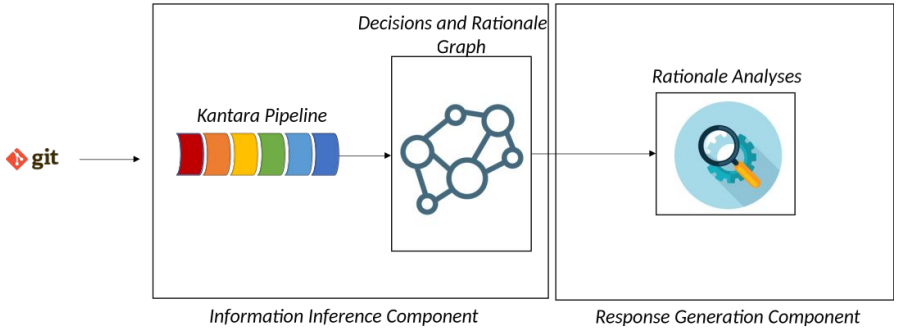
Surface candidate conflicts before new changes overwrite old intent

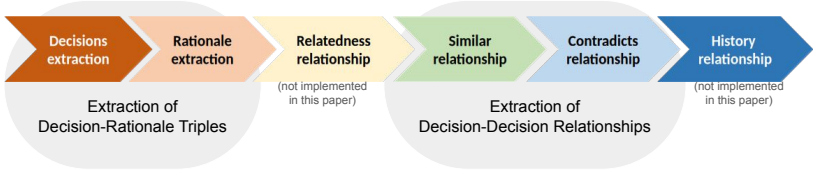
Automated Extraction and Analysis of Developer's Rationale in OS Software

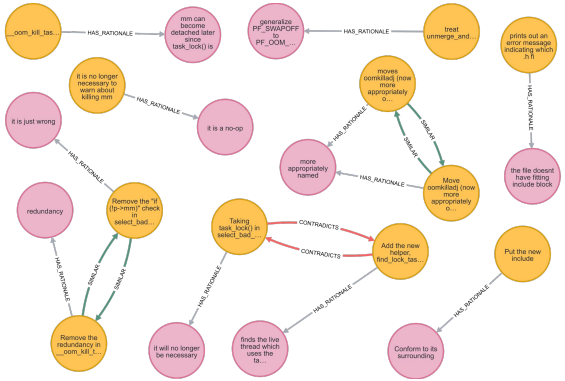
Mouna Dhaouadi

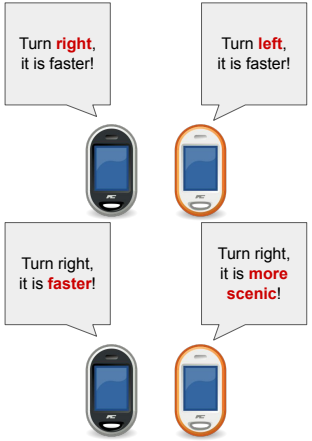
Bentley Oakes

Michalis Famelis









We can identify Decision and Rationale sentences

We can analyze the graph to find inspection candidates

We can generalize beyond the OOM Killer

Backup slides

Evaluation Setup

Question	Data	Evidence
1. Detect and extract?	OOM-Killer: 2,234 labelled sentences	**10-fold** cross validation; Compare to human **raters**
2. Build and analyze the graph?	OOM-Killer: 774 D+R sentences	**Time** to compute graph; identified tuples and interesting **cases**
3a. Generalize to other Linux modules?	slob.c: 833 sentences button.c: 576 sentences	Manual validation of **samples** of classified sentences
3b. Generalize to other OSS projects?	Tian et al. dataset	Compare to **ground** truth; Interesting **cases** found by graph analysis

Data used for Q3:

Project	D+R sentences
slob.c	252
button.c	191
JUnit4	145
Spring Boot	171
Apache Dubbo	138
Square Retrofit	151
Square OkHttp	137