

Annual Canadian Celebration of Women
In Computing Conference (CAN-CWiC)

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

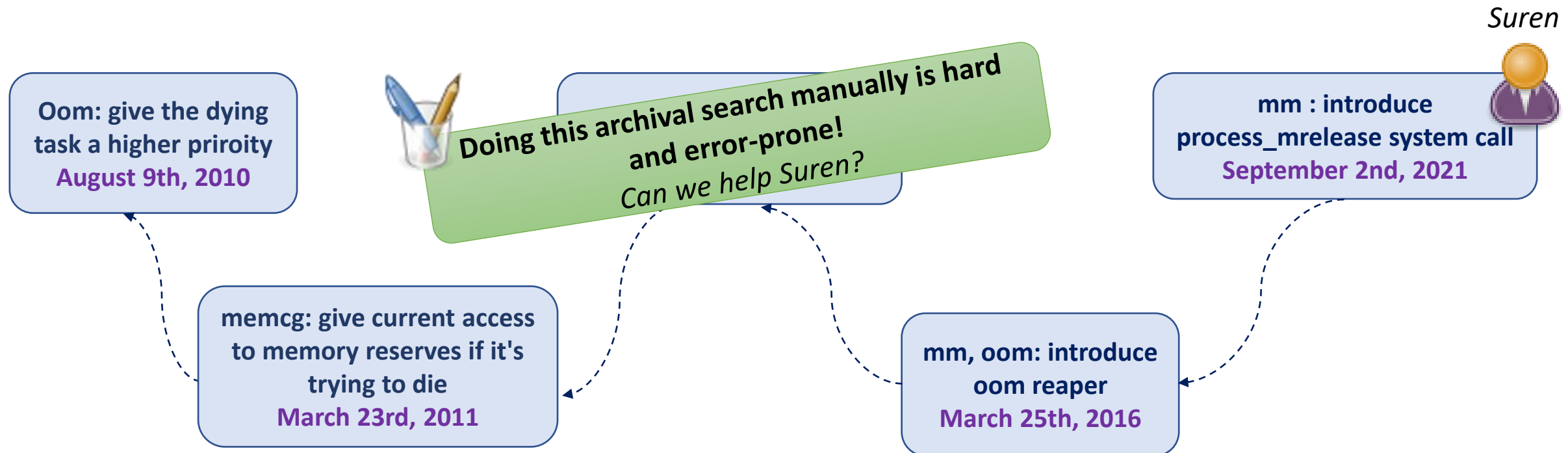
Rationale is *useful information* to :
understand the system,
learn from mistakes,
reuse solutions,
avoid conflicts

Real World Example

Suren's Challenge:

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

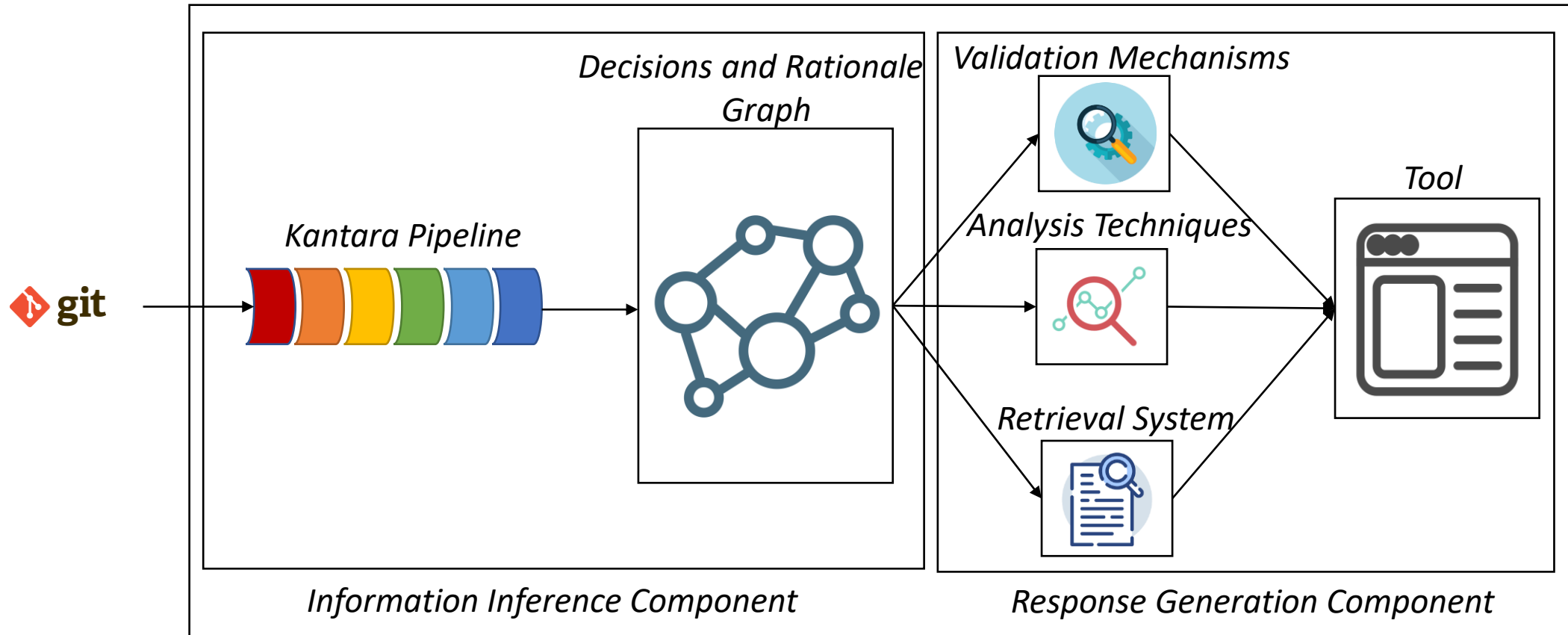
- Out-Of-Memory Killer (OOM-Killer) component
- Manually selected interesting commits from the Git history of OOM-Killer.
- Topic of "*reclaiming used memory from the OOM victim*".



Research question:

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.”

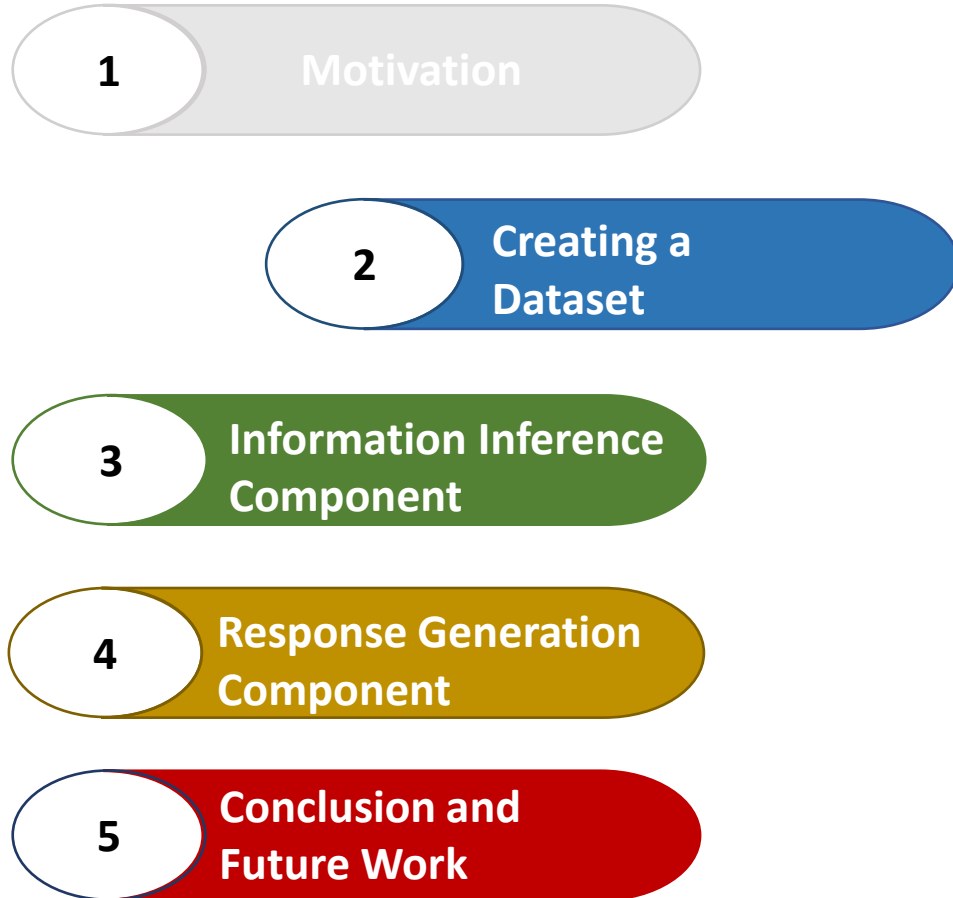
versus

“changes”

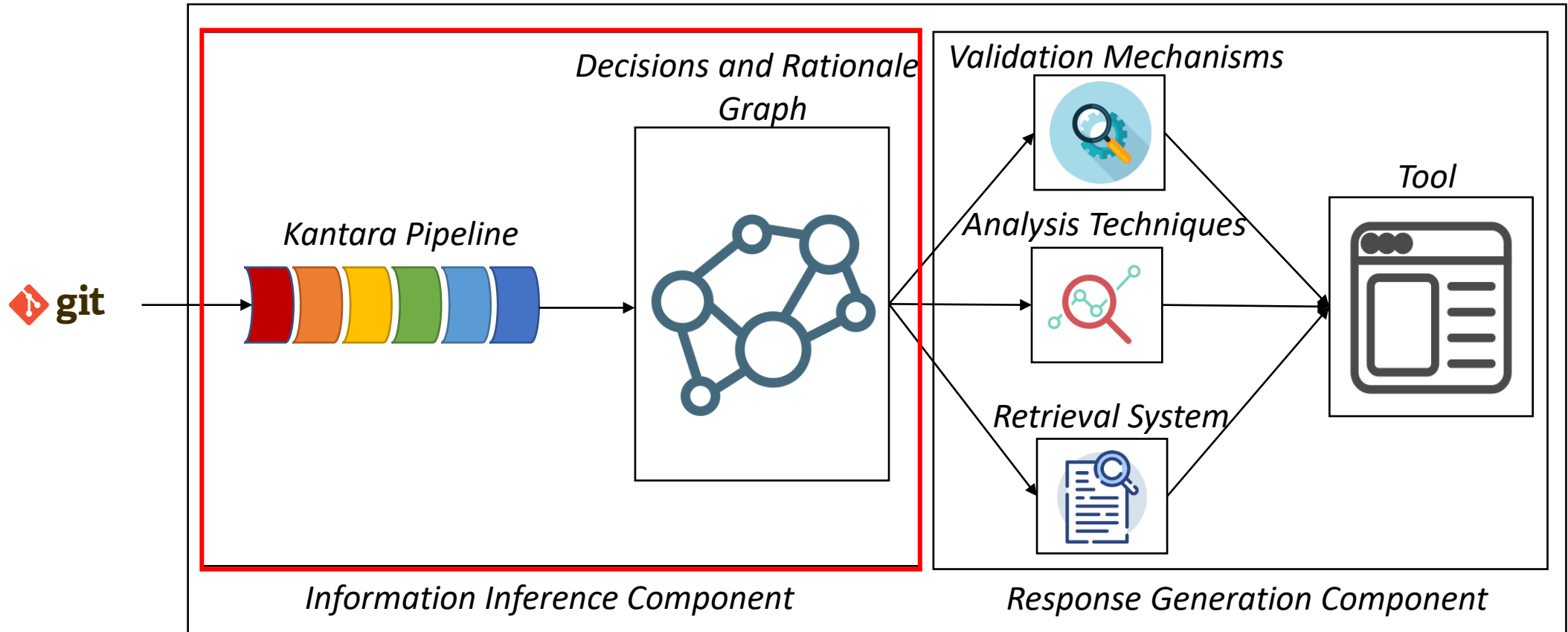
Challenges:

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

Plan



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

Label	Meaning
Decision	An action or a change that has been made, including a description of the patch behaviour
Rationale	Reason for a decision or value judgment
Supporting Facts	A narration of facts used to support a decision
Inapplicable	Pre-processing error or bad sentences (i.e., does not contain English sentences)

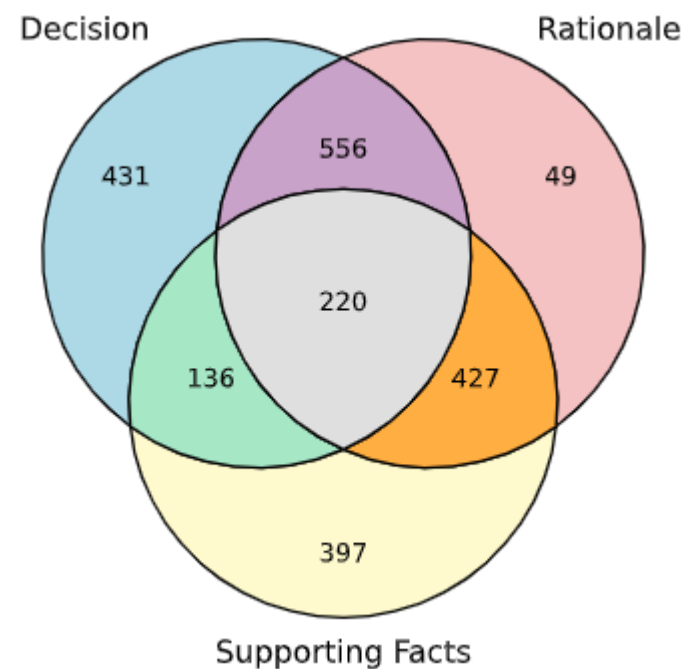
- Obtained through piloting rounds and discussions
- **Multiple classifications per sentence**
- In disagreement, take classification union
- Fleiss kappa: Around 0.65 (fair to good agreement)

Labelling Example

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

Labelling Insights

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

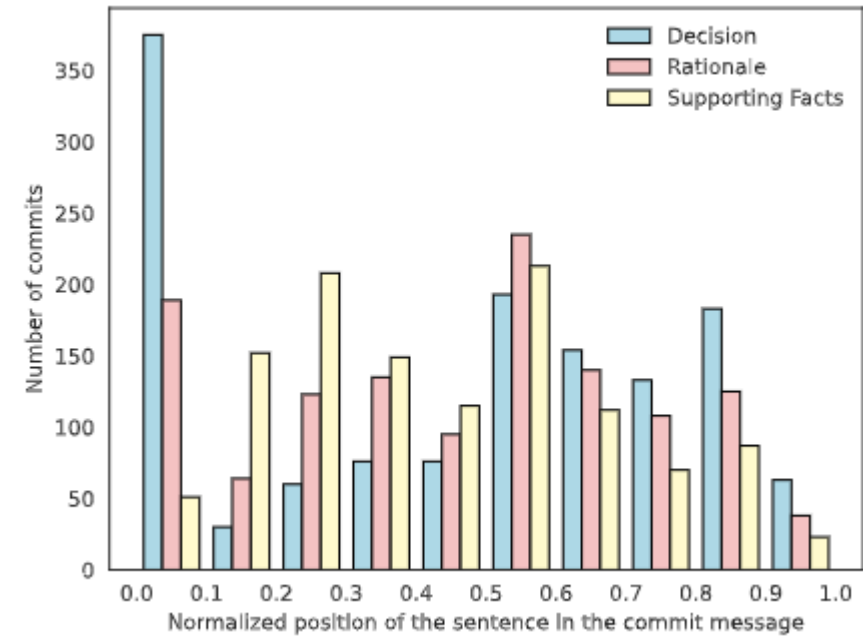


Distribution of the sentences in the OOM dataset

Labelling Insights

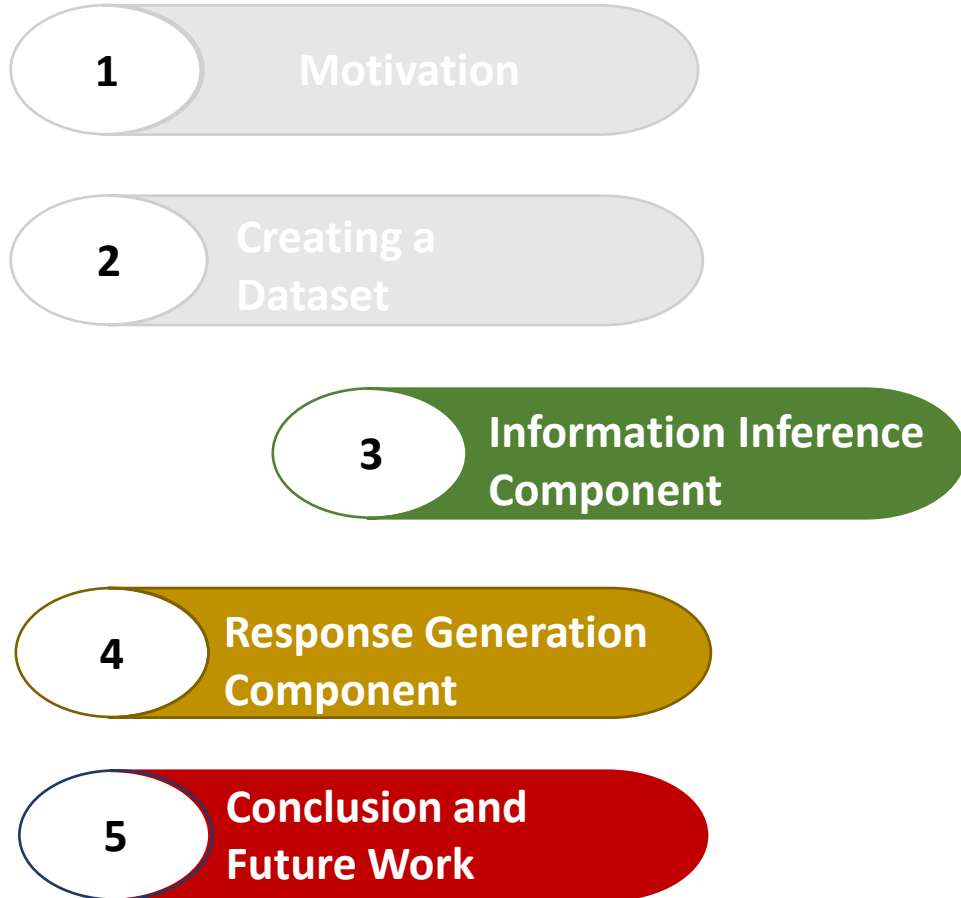
Common Structure:

Decisions -> Supporting Facts -> Rationale -> Decisions

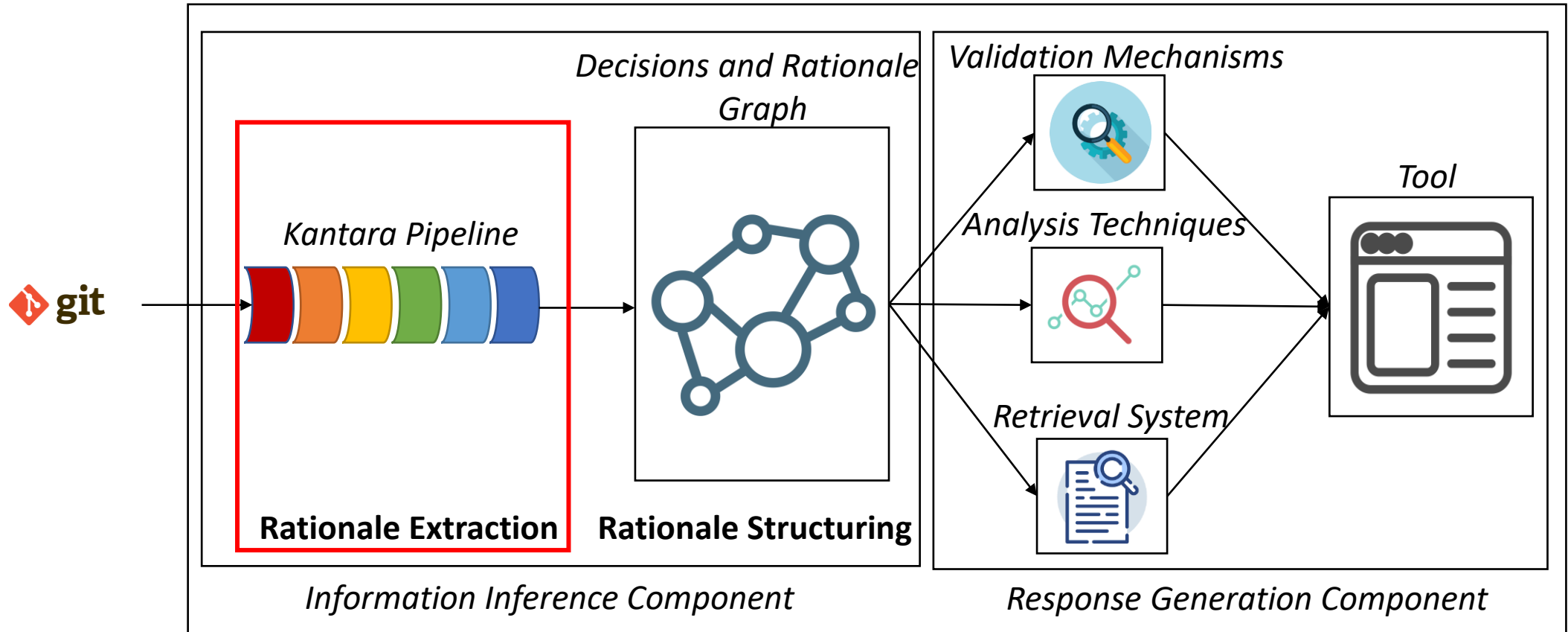


Distribution of the categories over the normalized positions of the sentences of the commit messages

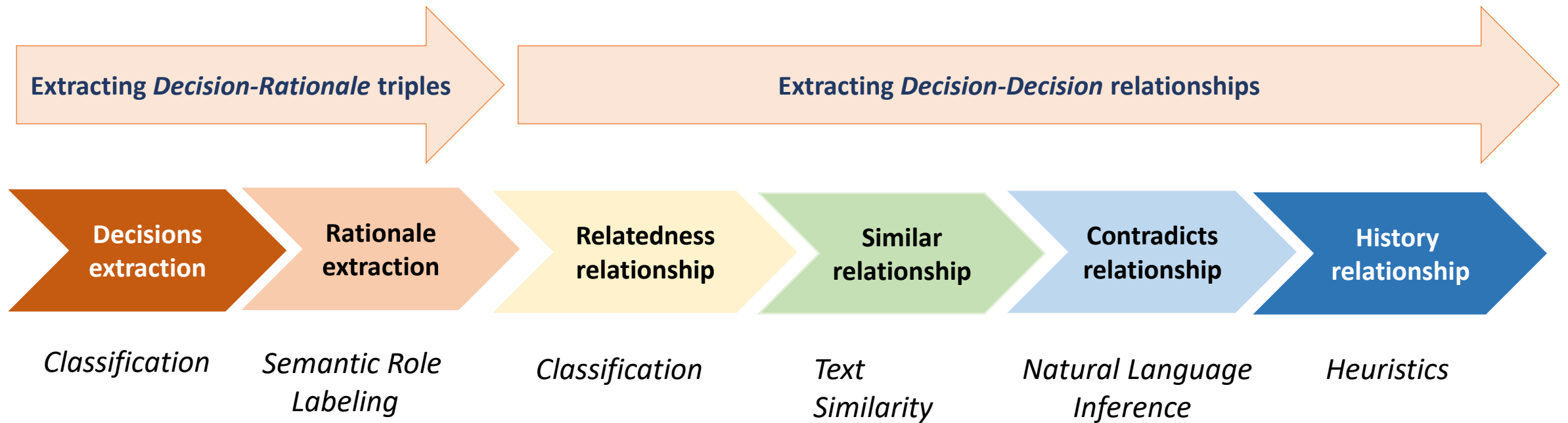
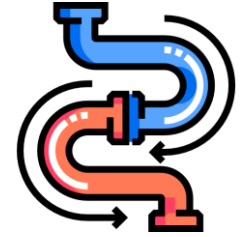
Plan



Information Inference Component



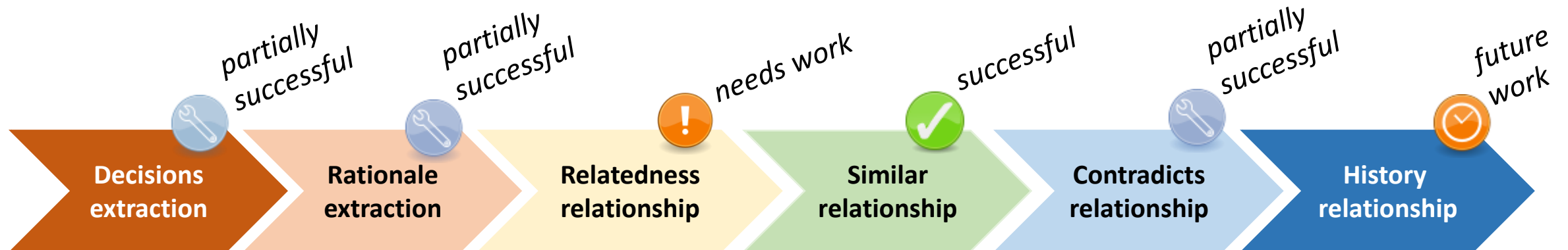
Kantara: Information Extraction pipeline



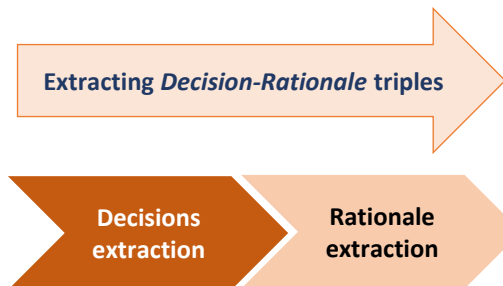
Prototype to evaluate feasibility of Kantara



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



Automatic sentence classification

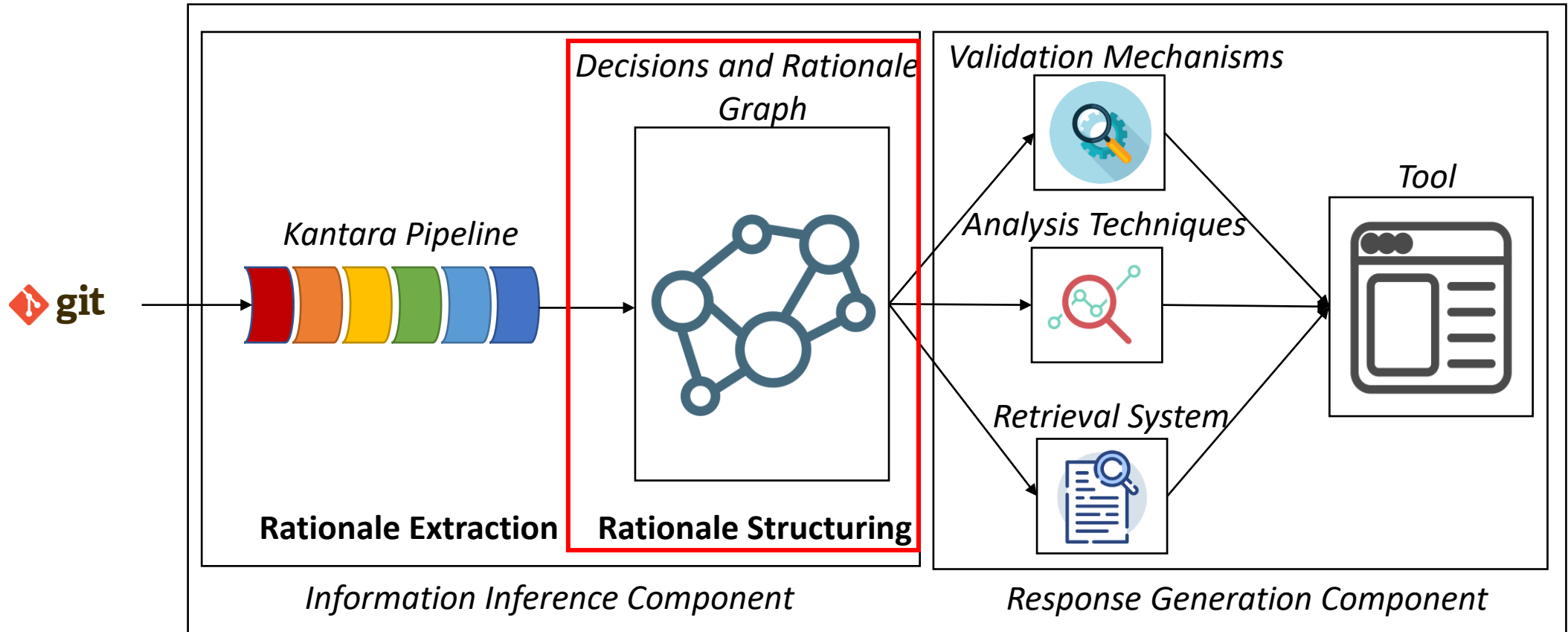


Binary classification: Logistic regression, decision tree, SVM

Multi-label classification: Random Forest, XGBoost, KNN

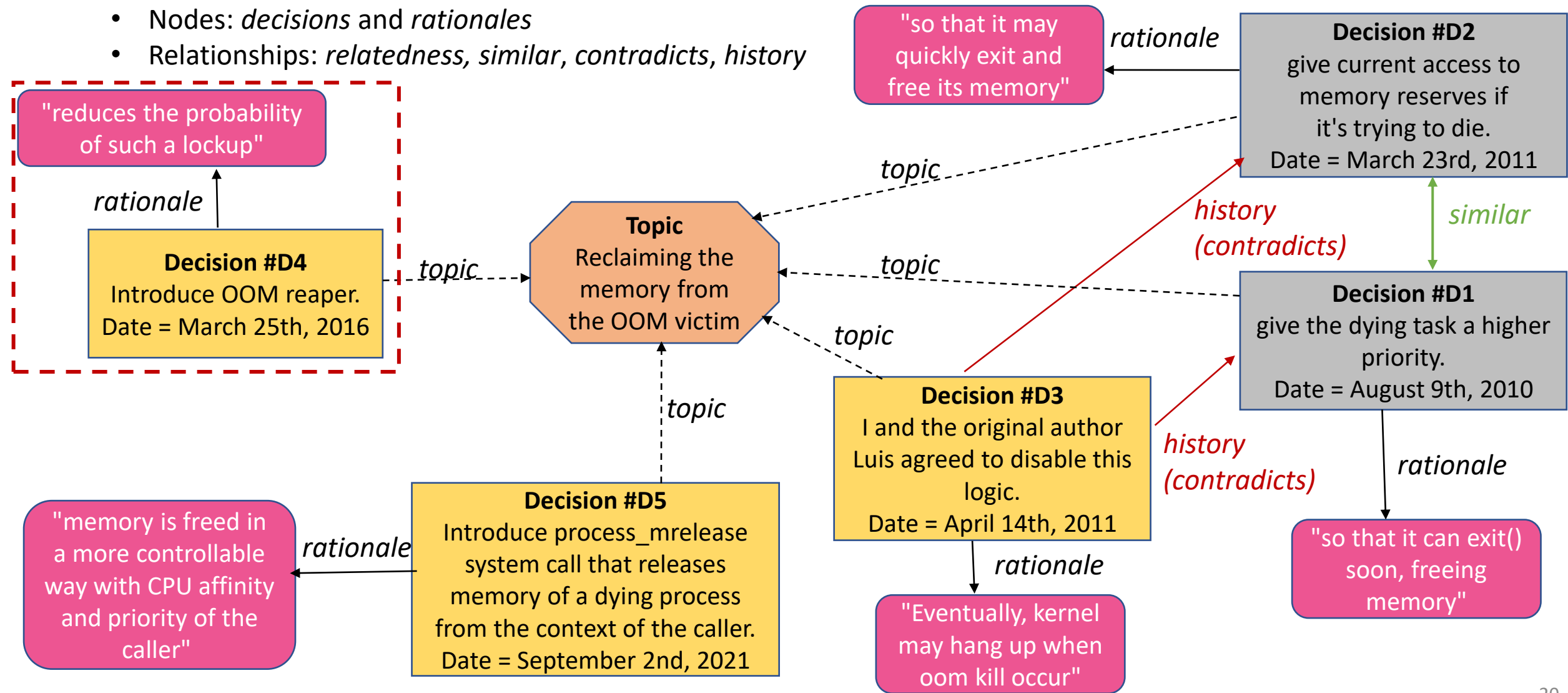
XGBoost classification evaluation			
Label	Precision	Recall	F1-score
Decision	0.76	0.69	0.72
Rationale	0.62	0.41	0.49
Supporting Facts	0.64	0.68	0.66

Information Inference Component

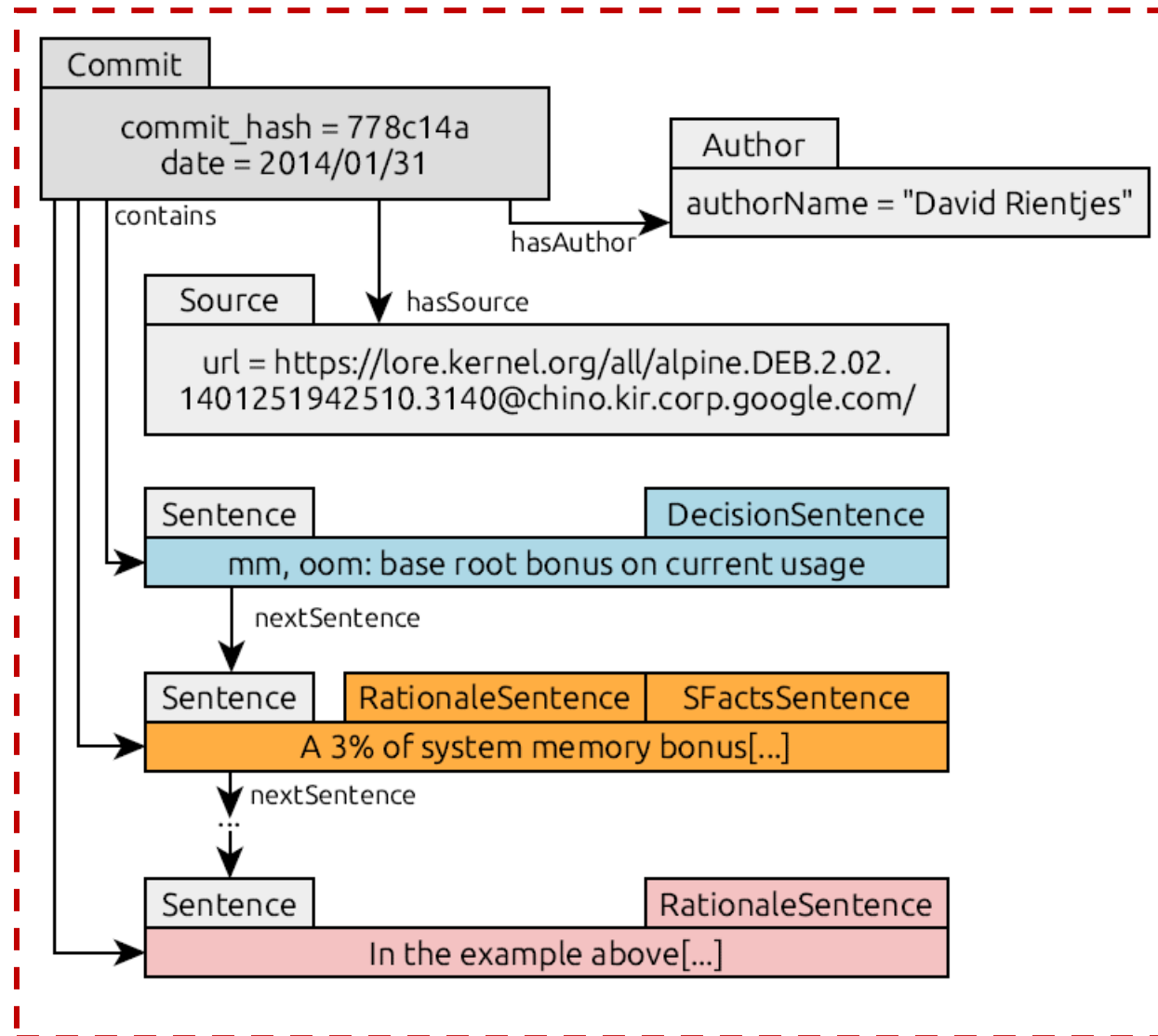


Knowledge Graph

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



Commit-level



Plan

1

Motivation

2

Creating a
Dataset

3

Information Inference
Component

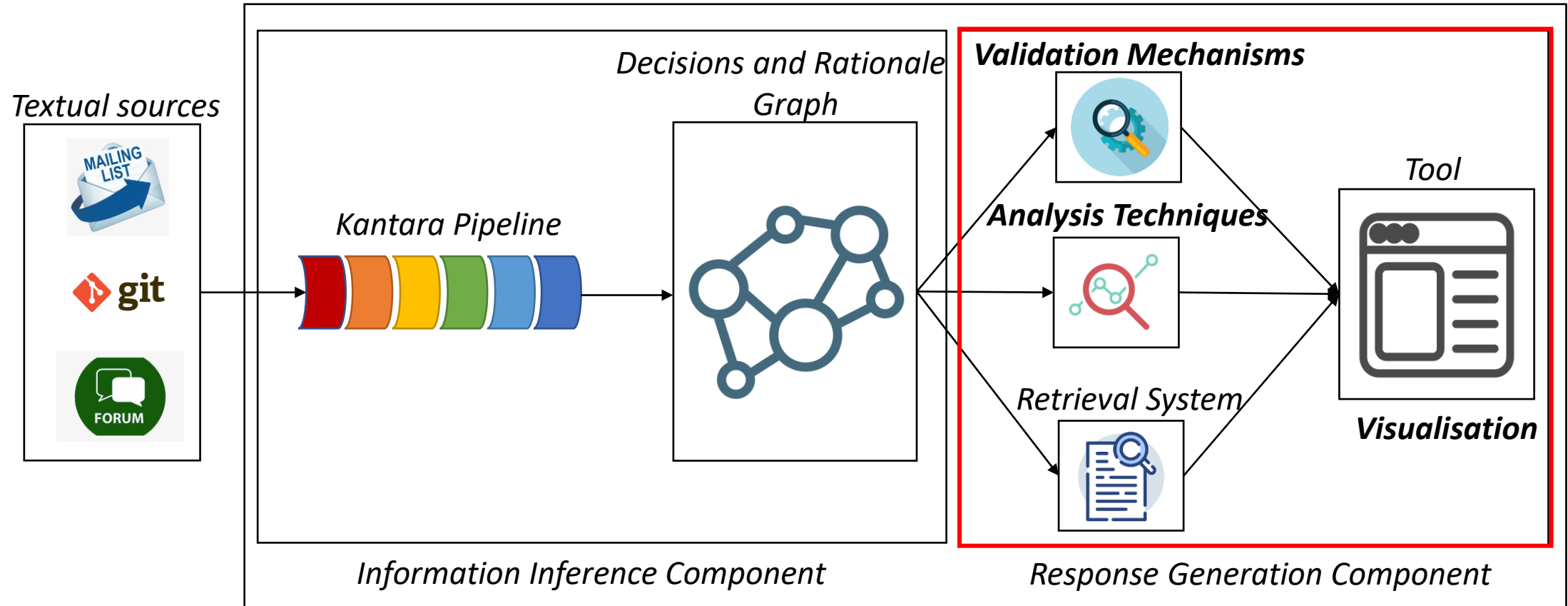
4

Response Generation
Component

5

Conclusion and
Future Work

Response Generation Component

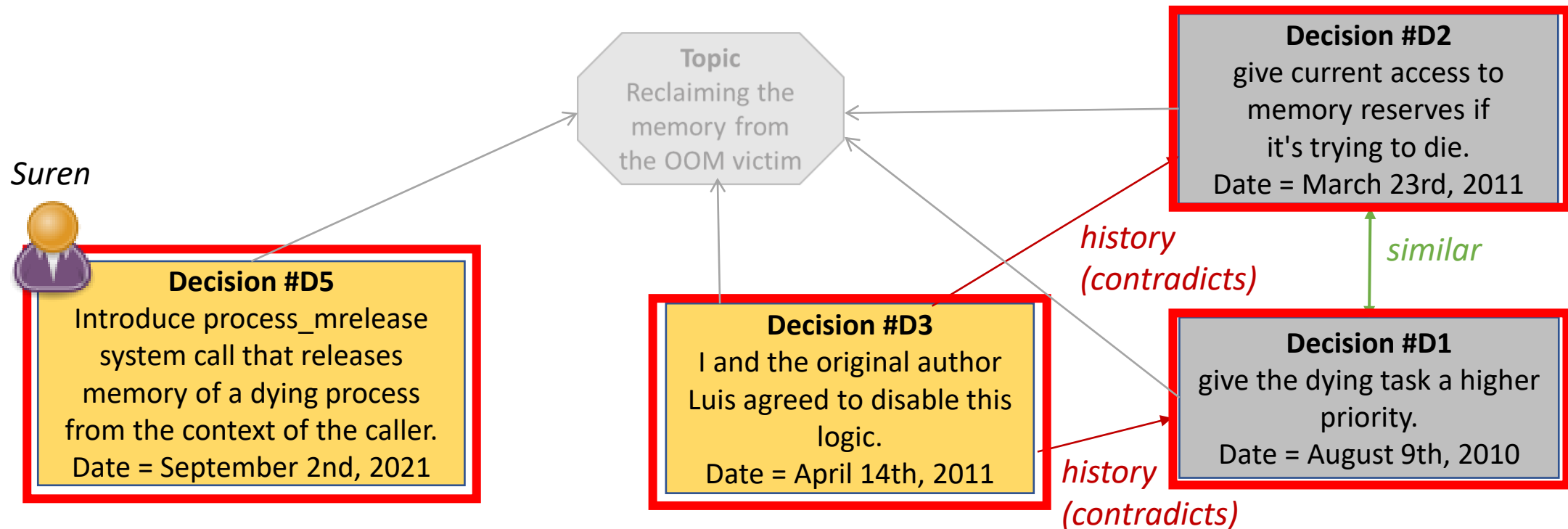


Validation Mechanisms - prototype

Suren's Challenge:

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

Look for potential conflicts when a new decision is proposed.



Rationale Analysis and Visualisation

1. Defined the ontology in the OpenCasear framework
2. Populated the tool with our dataset

```
extends <https://geodes.iro.umontreal.ca/rationale/vocab/authorV#> as authorV

@rdfs:label "Commit"
@rdfs:comment "The core Commit concept"
concept Commit < base:ContainedElement, base:IdentifiedThing

@rdfs:label "Sentence"
@rdfs:comment "Sentence concept"
concept Sentence < base:ContainedElement [
  restricts all relation base:isContainedIn to Commit
]

scalar property text [
  domain Sentence
  range xsd:string
  functional
]
]
TOWARDS UNDERSTANDING AND ANALYZING RATIONALE
```

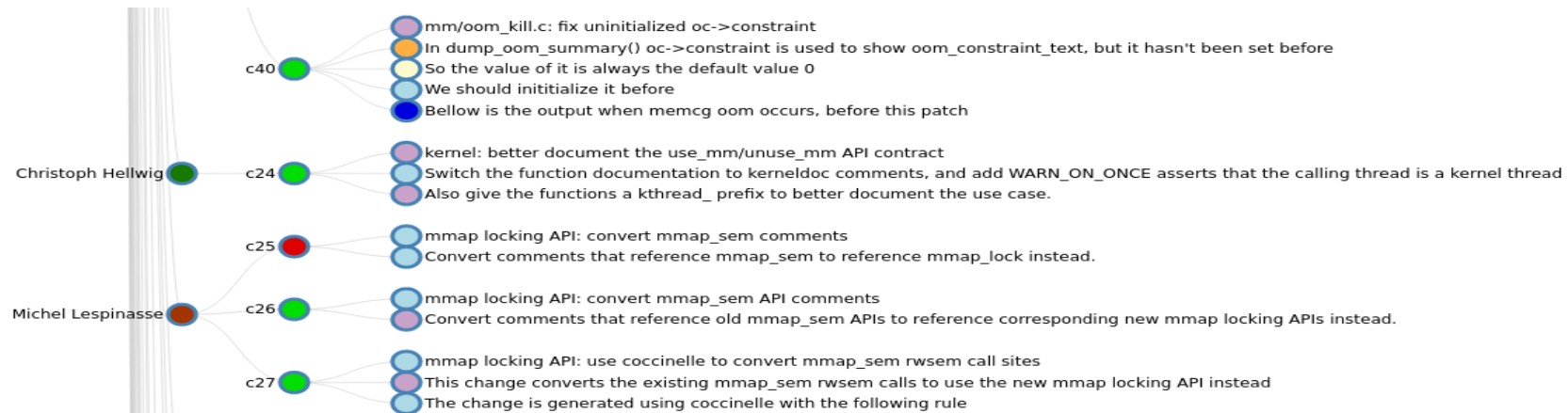
Rationale Analysis and Visualisation

Analysis: Inferencing using SPARQL queries

```
{
  "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

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