

Empowering Rationale-aware Commit-based Development

Ph.D. Thesis Defense

13/02/2026

Mouna Dhaouadi

PhD candidate

Prof. Michalis Famelis
Prof. Jian-Yun Nie



Context - Rationale in Commit Messages

- Commit Messages contain information about developers' rationale behind a certain change
- Rationale == justification == reason == **why** behind decisions
- Expressed in natural language

remove is_linear_pte() to simplify code

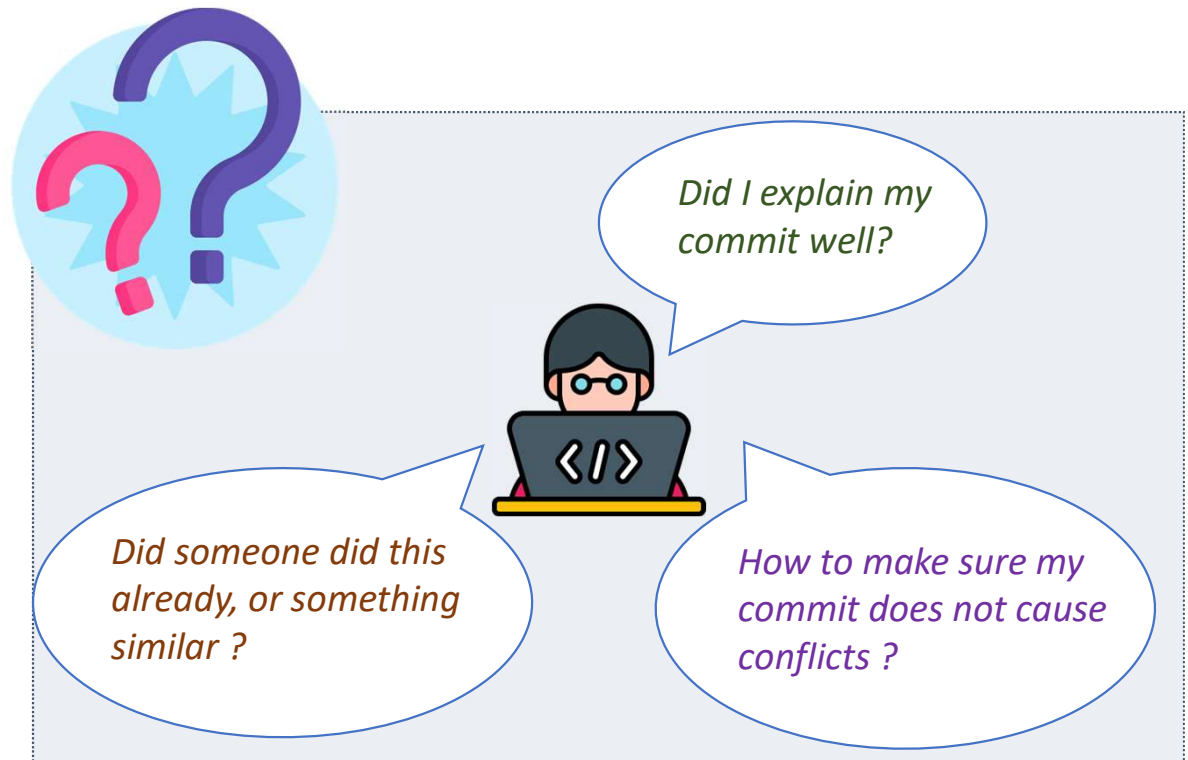
Decision

Rationale

Context - Problem

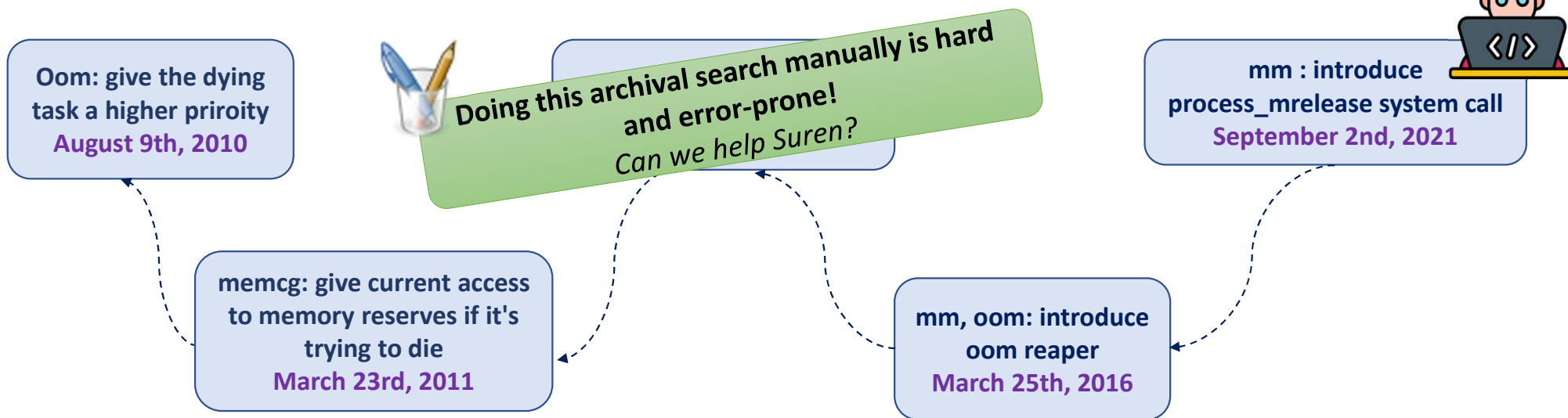
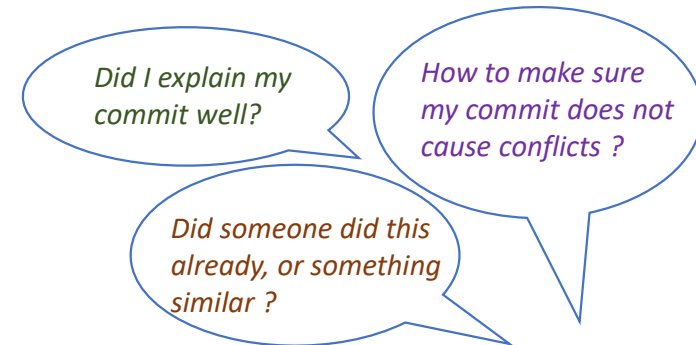
When committing a change, developers **need to**:

- Explain well for future reference
- Not repeat past solutions
- Not cause conflicts



Context – Real-world example

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



Context - Real-world problem

Did someone did this already, or something similar ?

- Code review emails are used to **retrieve rationales** for **past** decisions [Sutherland et al., 2009]
- The most **frequent hard-to-answer** questions dealt with **rationale** [Latoza et al., 2010]
- Interviews with **27** professional developers and managers :
 - developers ignoring the rationale had to **guess the intent** behind a certain code fragment
 - developers had a **hard time trying to locate the relevant information**
 - knowledge **recreation** activities **required** considerable **time and effort**[Robillard et al., 2021]
- Practitioners from **17** companies expressed **difficulties to find and retrieve** decision knowledge [Kleebaum et al., 2019]

Did I explain my commit well?

Research Vision: Empower Rationale-aware Commit-based Development

Assist decision-makers with an automated rationale extraction and management system



Rationale information is **useful**:

-  **understand the system**
-  **reuse solutions**
-  **avoid conflicts**
- learn from mistakes
-

Research Impact



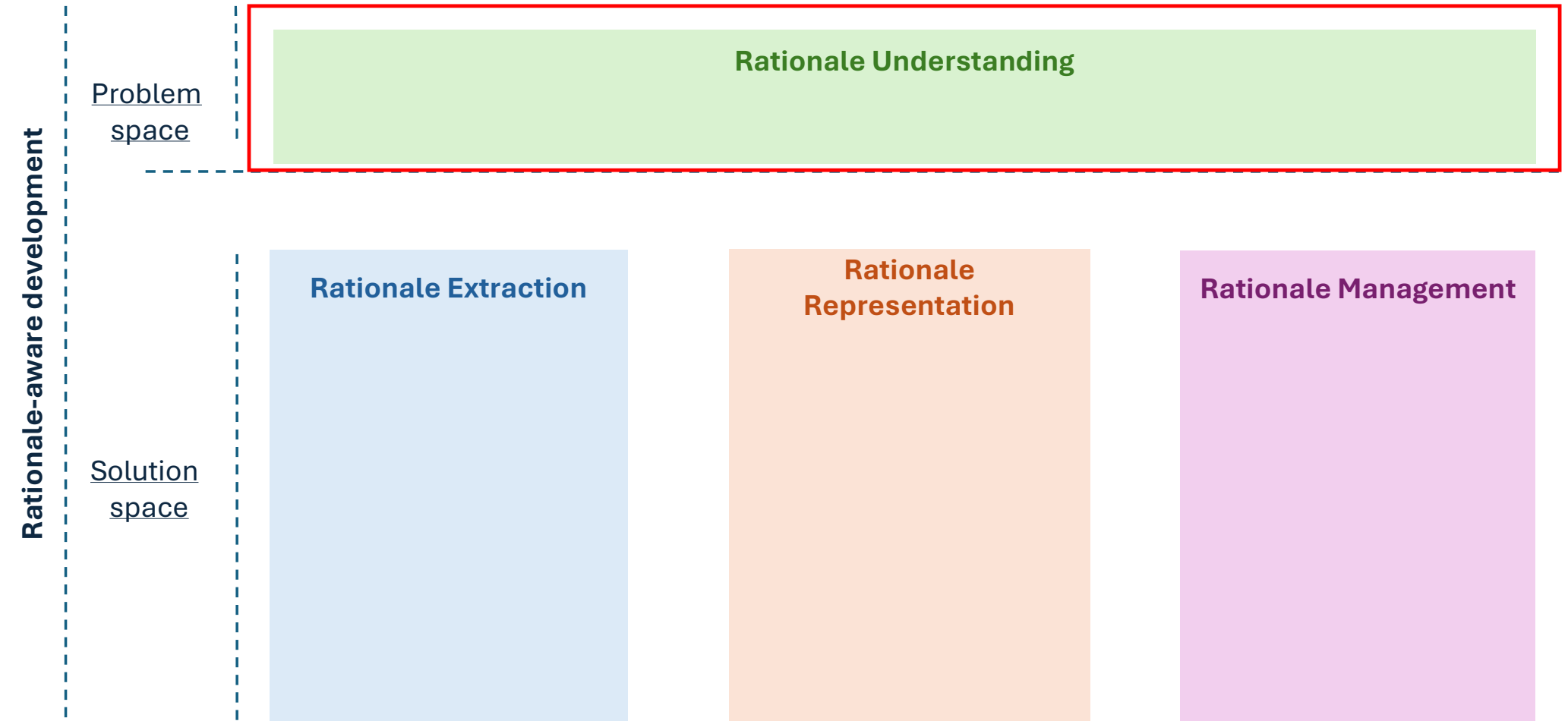
Solving this problem would have **positive impact**:

- Project:
 - Increase longevity
 - Increase quality
- Developers:
 - Improve productivity
 - Save time/effort



Historical background

- **Design rationale** research is an old research direction that aims at improving the quality of design
- The seminal work: the IBIS schema [Noble., 1988]
- Since then, a lot of design rationale work has been introduced (e.g., PHI[McCall., 1991], SEURAT[Burge., 2005])
- Researchers also proposed **software engineering rationale** [Dutoit., 2006]
- Several work about rationale in **software architecture knowledge** [Jansen., 2005]
- However, the **capture** and the **cost-benefit** problems hindered this line of research.  
- With the advances of *Natural Language Processing (NLP)* and *Machine Learning (ML)*, this line of research has been revisited recently, since it became “possible” to capture the information automatically
- **Big** corpus of work on representing, extracting and managing rationale.



Challenge 1: Limited Understanding of Rationale Practices

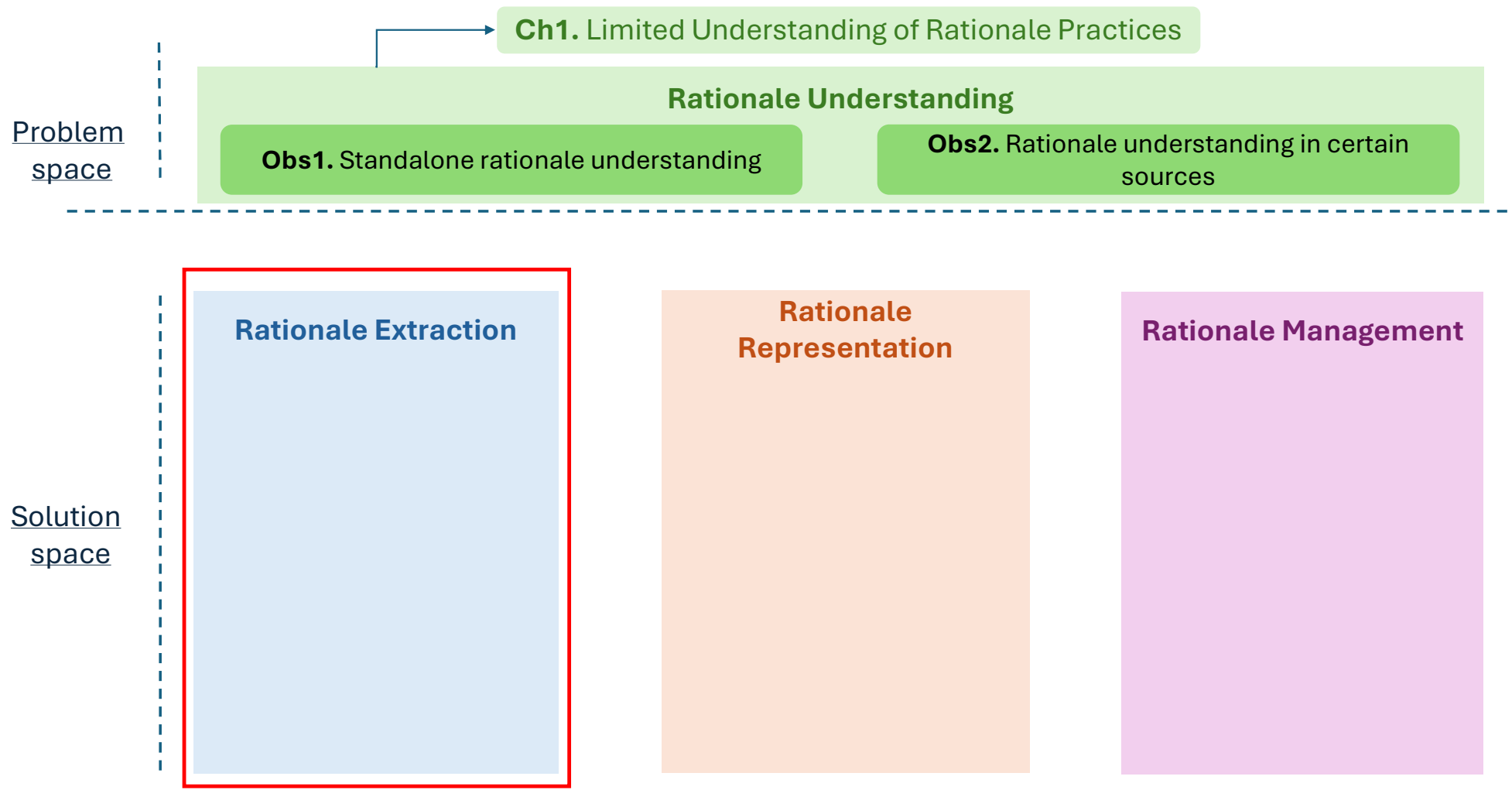
Understanding of Rationale Practices

- Frequency and completeness of rationale in chat messages [Alkadhi., 2017 Alkadhi., 2018]
- Developers' perspective of rationale for code commits [Alsafwan., 2019, Alsafwan., 2022]



**The understanding of developers' rationale practices is limited to:
standalone rationale (Observation 1)
and to certain sources (Observation 2)**





Challenge 2: Schema-specific Rationale Extraction

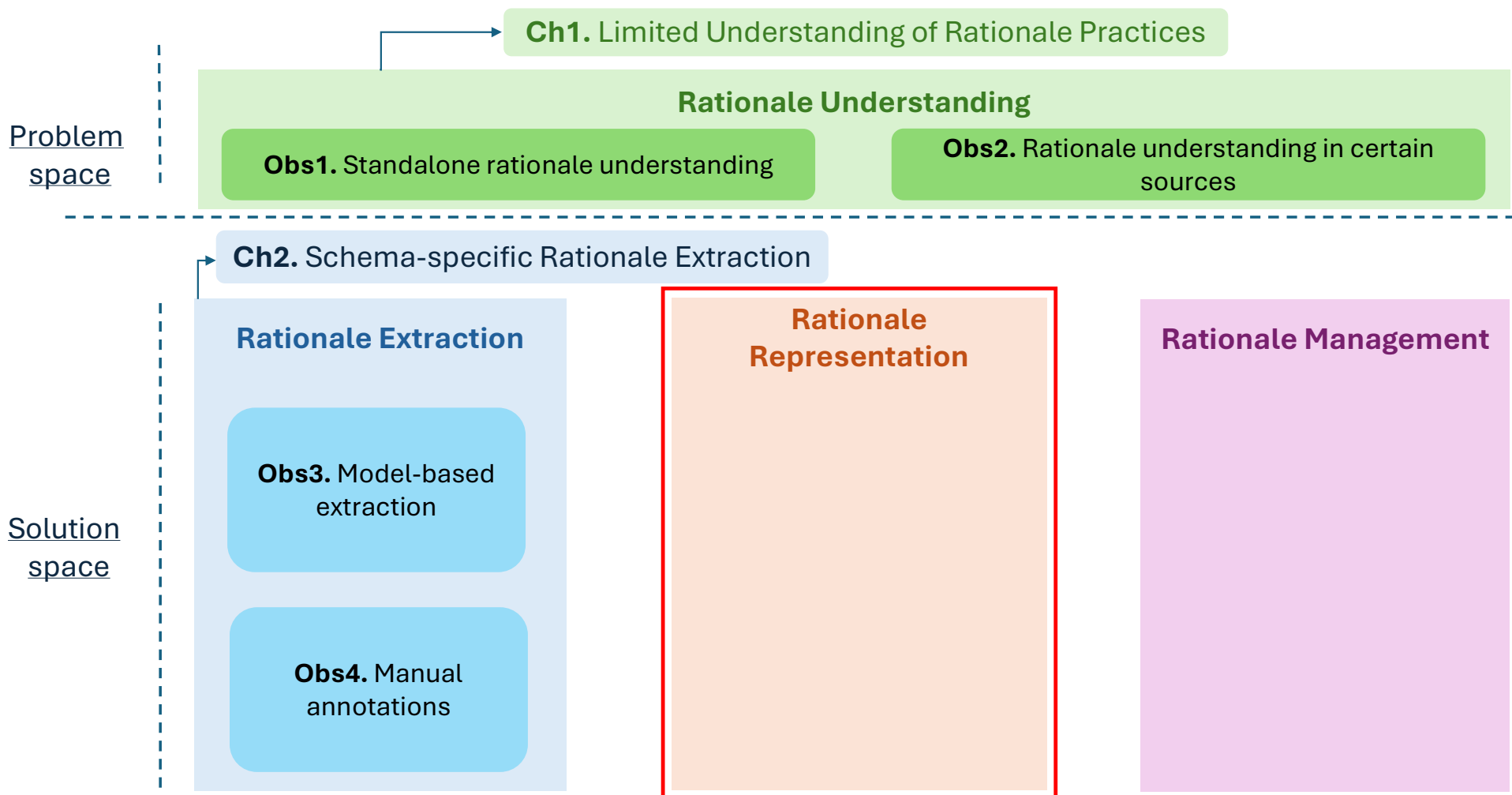
Rationale extraction

- Various rationale extraction approaches [Liang. 2012, Rogers., 2012, Lester., 2020, McCall., 2018, Alkadhi., 2017, Li., 2020, Zhao., 2024, Sharma., 2021]



The automated extraction techniques are model-based (Observation 3) and rely on manually annotated datasets (Observation 4)





Challenge 3: Inflexibility of Rationale Representations

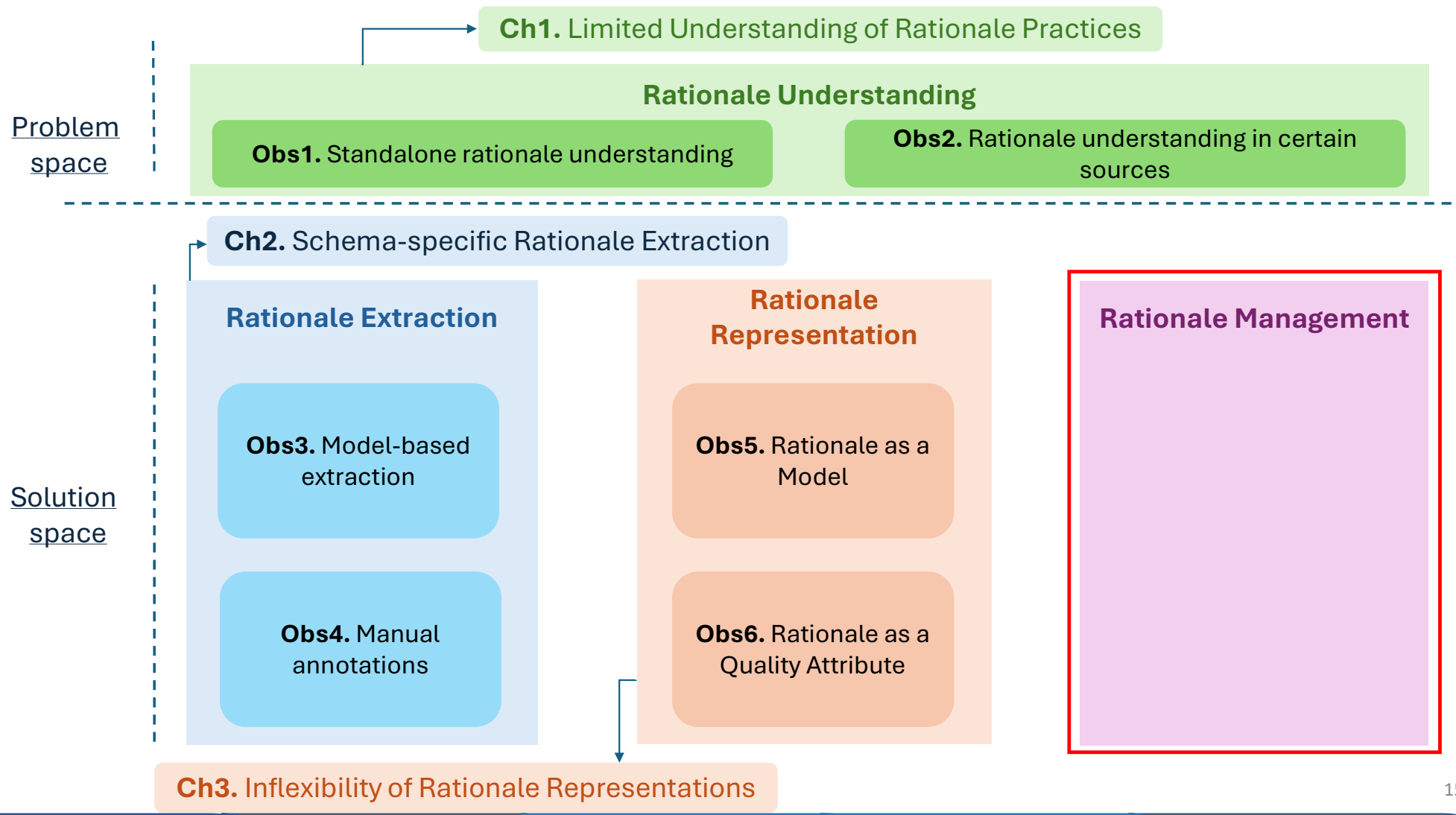
Rationale representation

- Various rationale representations [Mahabaleshwar., 2020, Kleebaum., 2020, Ullah., 2023, Yue., 2023]



**Rationale is represented as a model (Observation 5)
or as a quality attribute (Observation 6)**





Challenge 4: Limited Semantic Inference Techniques

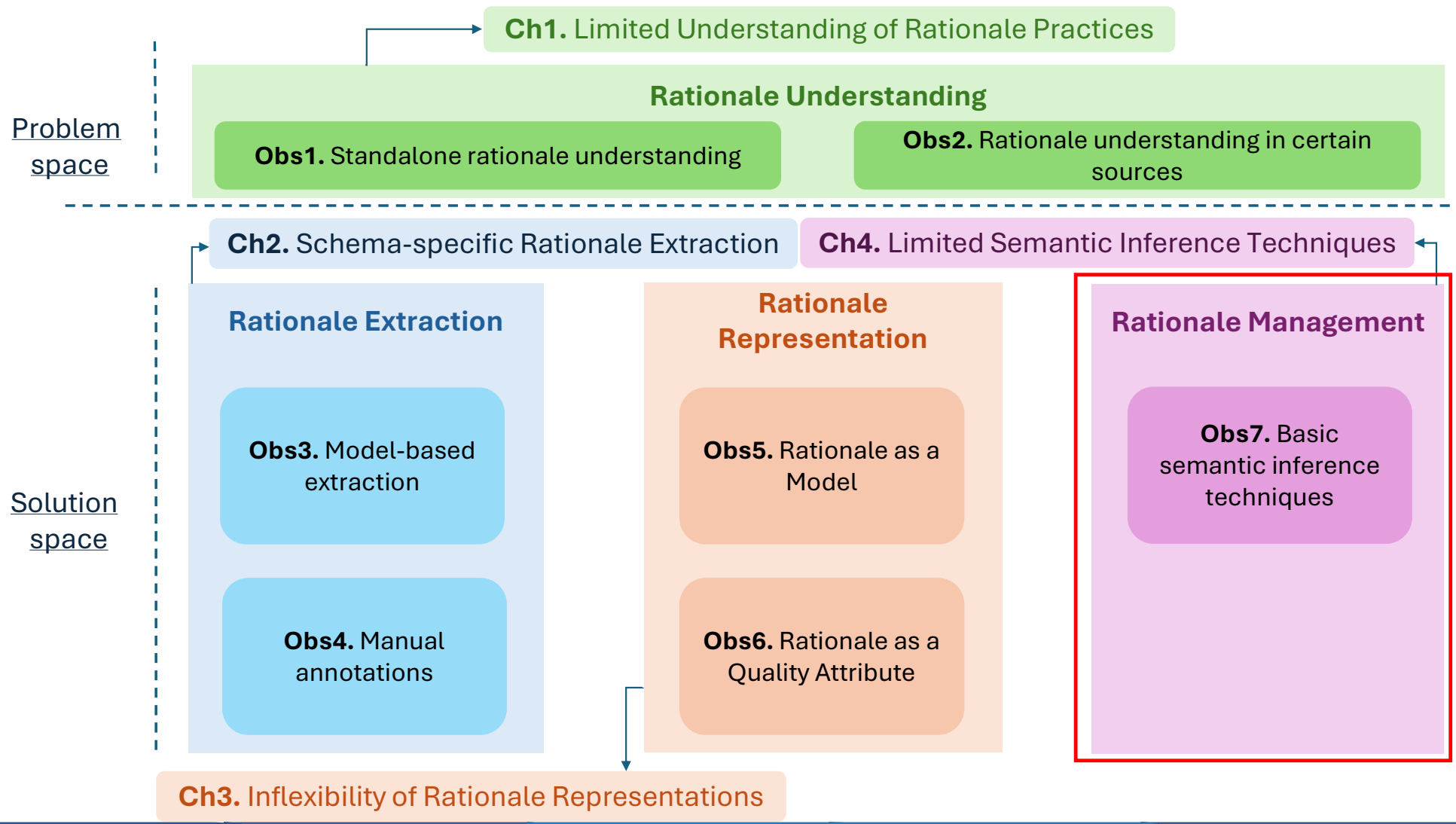
Rationale management

- Rule-based mechanisms [**Burge., 2003**]
- Relevancy-based rationale clustering [**Mahabaleshwar., 2020**]



Research on inferring semantic information from rationale is limited to basic techniques (Observation 7)





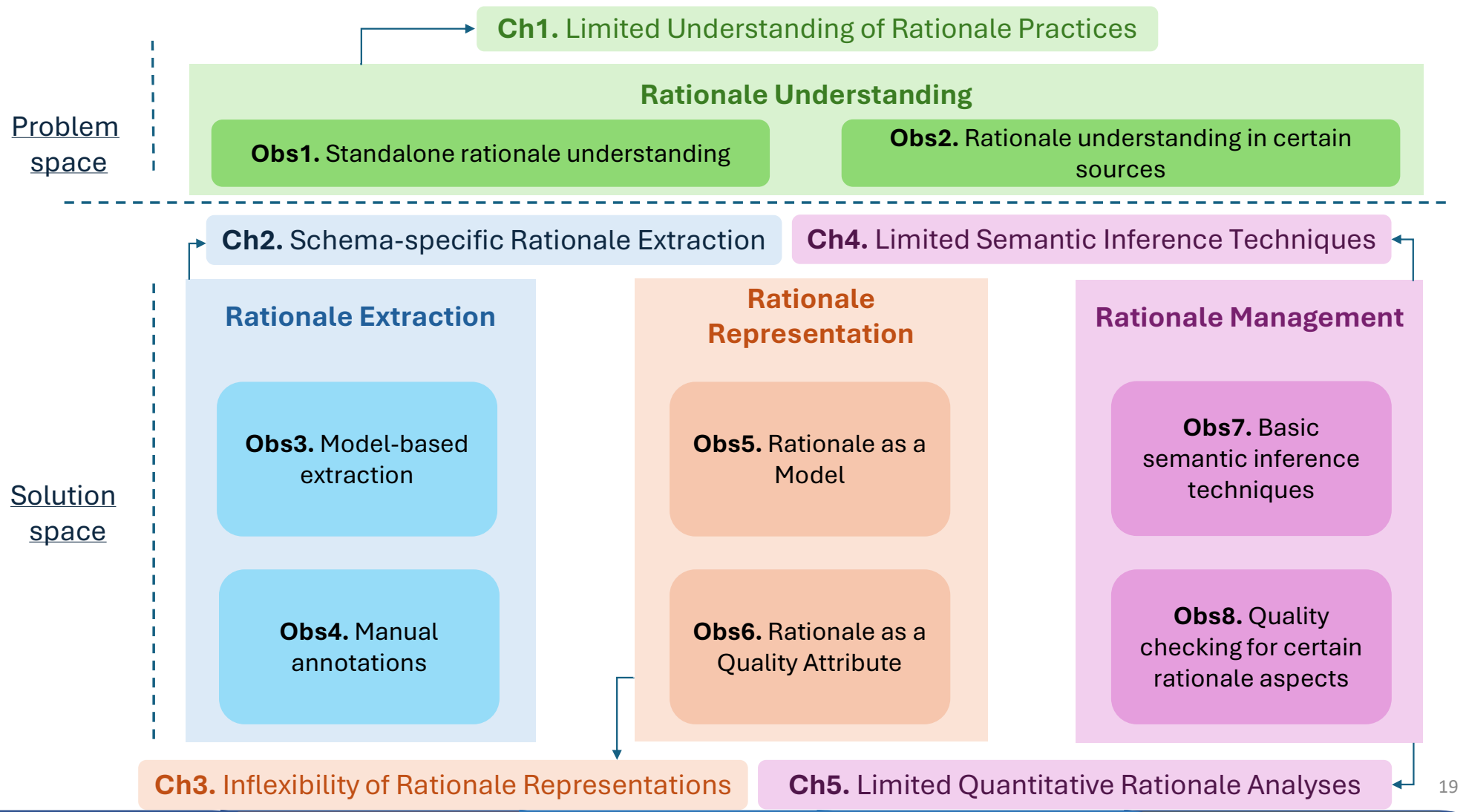
Challenge 5: Limited Quantitative Rationale Analyses

Rationale management

- Quality checking (completeness, requirements coverage, traceability) [Kleebaum., 2020, Kleebaum., 2021]
- A *good* message classification criteria [Tian., 2022]
- Correlation between quality evolution and defect proneness [Li., 2023]

Research on rationale quality is limited to certain aspects (Observation 8)





Thesis Statement



Automated **evidence-based quantitative** and **semantic** rationale **analyses**
can provide **effective support** for
rationale-aware commit-based development.

Developers needs when committing



21

Problem
space

Rationale Understanding

C1. Empirical understanding of rationale practices

Solution
space

Commit
messages



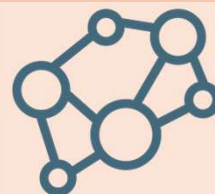
Rationale Extraction

C2. Kantara
pipeline



Rationale Representation

C3. Rationale and
decision graph



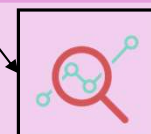
Information Inference Component

Rationale Management

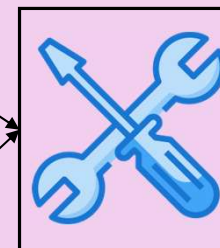
C4. Semantic
Analyses



C5. Quality
Analyses



Tools



Response Generation Component

Introduction

Related Work

Contributions (C1 - C2 - C3 - C4 - C5 - Tools)

Conclusion

Contributions



Ch1. Limited understanding of rationale practices



C1. Empirical understanding of rationale practices

(Chapter 4)
(Chapter 7)

Ch2. Schema-specific rationale extraction



C2. Kantara pipeline

(Chapter 2)
(Chapter 3)
(Chapter 5)

Ch3. Inflexibility of rationale representations



C3. Rationale and decision graph

(Chapter 2)
(Chapter 3)

Ch4. Limited semantic inference techniques



C4. Semantic analyses

(Chapter 2)
(Chapter 5)

Ch5. Limited quantitative rationale analyses



C5. Quality analyses

(Chapter 3)
(Chapter 4)
(Chapter 6)

Ch1. Limited Understanding of Rationale Practices

Problem
space

Rationale Understanding

C1. Empirical understanding of rationale practices

Solution
space

Commit
messages



Rationale
Extraction

C2. Kantara
pipeline



Rationale
Representation

C3. Rationale and
decision graph

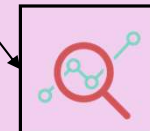


Rationale Management

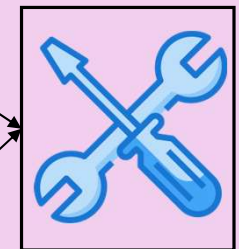
C4. Semantic
Analyses



C5. Quality
Analyses



Tools



Information Inference Component

Response Generation Component

Validate developers needs (C1)

Better understand:

1. The experience of developers *needing* and *finding* **similar** and **conflicting** rationale information in commit messages.
2. The experience of developers *needing* to **assess**, and **refining** rationale information in commit messages.



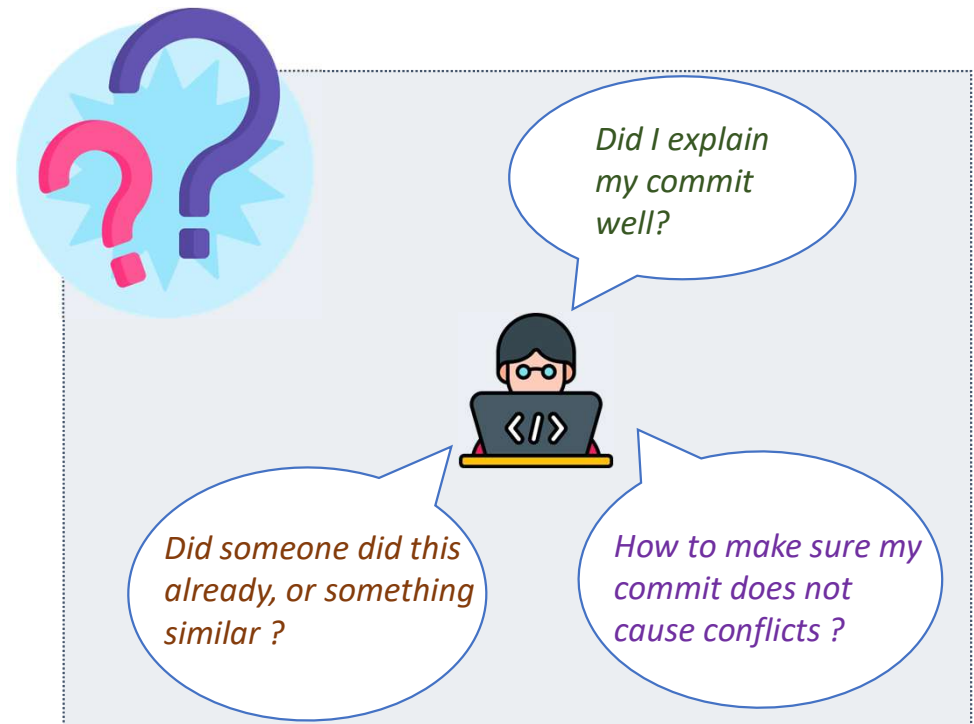
Validate developers needs (C1)

- Empirical study:

- Survey



- Semi-Structured Interviews



Survey Design



RQ1: What is the experience of developers *needing and finding* **similar** rationale information in commit messages?

Did someone did this already, or something similar ?

RQ2: What is the experience of developers *needing and finding* **conflicting** rationale information in commit messages?

How to make sure my commit does not cause conflicts ?

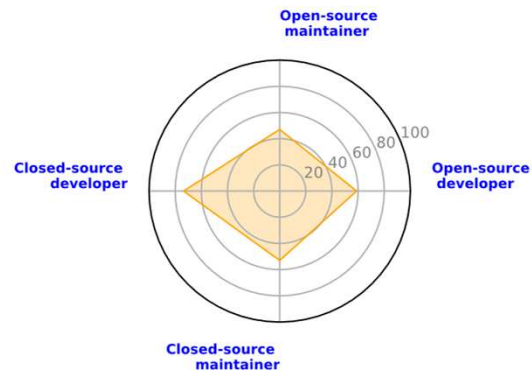
RQ3: What is the experience of developers *needing to* **assess, and refining** rationale information in commit messages

Did I explain my commit well?

Demographics: Role vs Activity



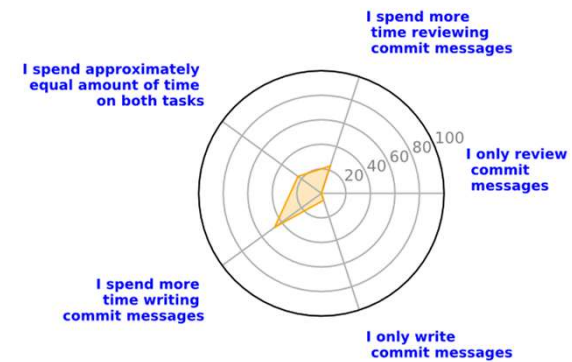
Developer Role



- 26 both “maintainer” and “developers”
 - 8 only developers
 - 5/8 “only write” or “more write”
 - 3/8 “equal time”
- ⇒ **High alignment** between *writing* activity and *developer* role

⇒ *Developers (18): “only write”, “more write”*

Developer Activity



- 8 “only review” or “more review”
 - 100% both “developer” and “maintainer”
- 8 “equal time” :
 - 3 only “developer”
 - 5 both “developer” and “maintainer”

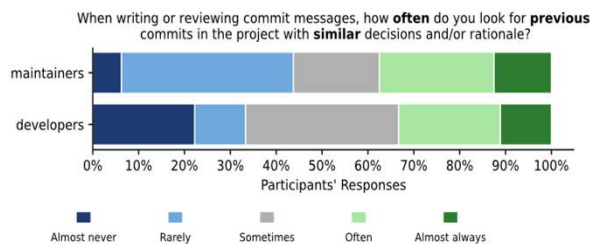
⇒ *Maintainers (16) : “only review”, “more review” or “equal time”*

Survey Results

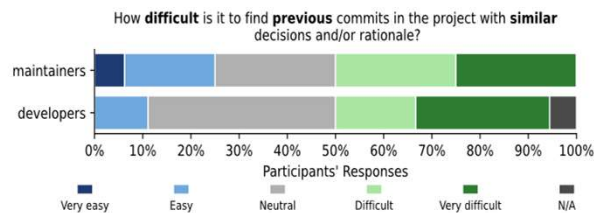
Did someone did this already, or something similar ?



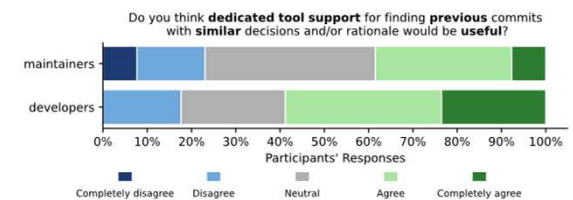
RQ1: What is the experience of developers *needing* and *finding* similar rationale information in commit messages?



Frequency of need (Q7)



Difficulty of finding (Q9)



Need for **dedicated** tool support

Finding **previous** commits in the project with **similar** decisions and/or rationale is **important**, but could be **difficult to acquire** and is **time consuming** in the current practice, making **dedicated tool support** useful.

Survey Results

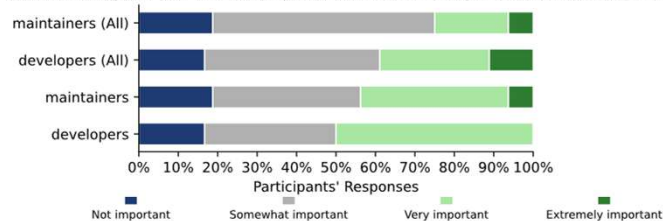
How to make sure
my commit does not
cause conflicts ?



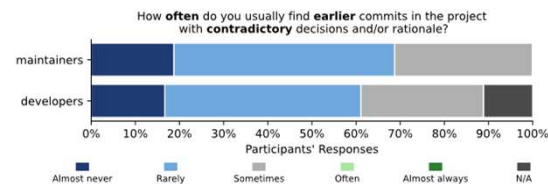
RQ2: What is the experience of developers *needing* and *finding* **conflicting** rationale information in commit messages?

When writing or reviewing commit messages, how **important** is knowing if the decision and/or rationale of the commit **might contradict** with the decisions and/or rationale of **earlier** commits in the project for the **completion** of your work?

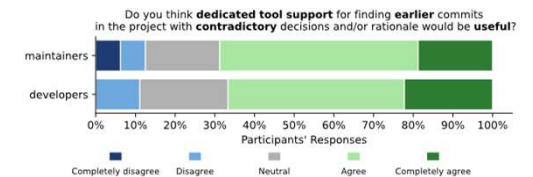
(All) How **important** it is to be able to identify **all earlier** commits in the history of the commit messages of your software project(s) with **contradictory** decisions and/or rationale?



Importance of need (Q18,Q22)



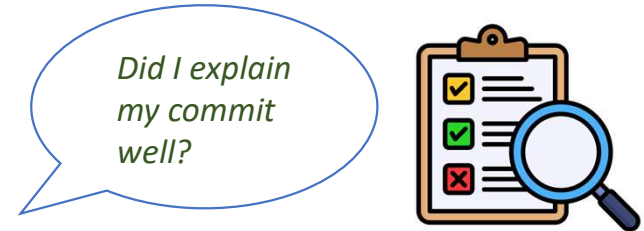
Frequency of finding (Q20)



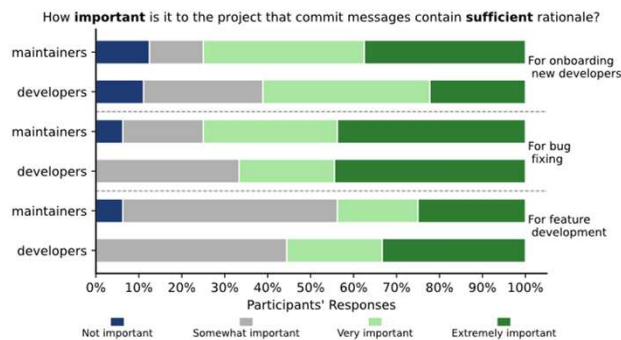
Need for **dedicated** tool support

Knowing if the decision and/or rationale of the commit **might contradict** with the decisions and/or rationale of **earlier** commits in the project is **important**, but is **difficult to acquire** and could be **time consuming** in the current practice, making **dedicated tool support** useful.

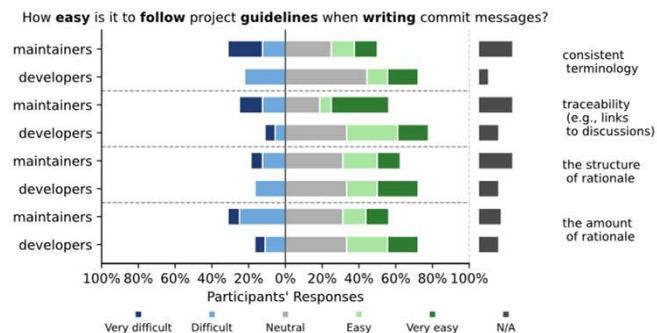
Survey Results



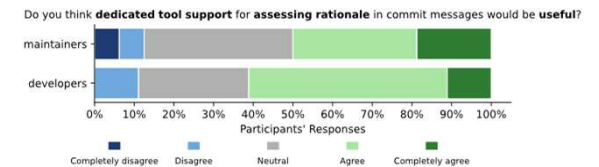
RQ3: What is the experience of developers *needing to assess, and refining* rationale information in commit messages?



Importance of sufficiency (Q27)



Difficulty of compliance (Q29)



Need for **dedicated** tool support

Having **sufficient, compliant and consistent** rationale in commit messages is **important** but **difficult to maintain** and is **refinement-prone** in the current practice, making **dedicated tool support useful**.

Survey Results (Continued)



Dev vs Main.

Maintainers face a **greater** and **more frequent need** to retrieve **similar** rationale information from past commits.

Developers show a **stronger need** to detect **contradictory** rationale.

Maintainers emphasize traceability, consistency, and knowledge preservation, whereas **developers** emphasize efficiency, feature progress, and tool-assisted support.

RQ1 vs RQ2.

Identifying **contradictory** rationale information is a **more urgent and critical need** for developers and maintainers than retrieving **similar** rationale.

Survey Results – Rationale Requirements



We derive rationale-aware tool requirements from our findings:

Understanding the project :

- (a) identifying project expectations when it comes to maintaining the quality (consistency, compliance, sufficiency) of rationale information
- (b) retrieving the accumulated rationale knowledge from the project history

Reviewing work :

- (a) ensuring the work's conformance with project expectations about rationale quality
- (b) contextualizing the work with respect to existing project knowledge

Follow-up Interviews Design



- **Interviews:**
 - 4 participants
 - 13 questions

Q1	We study rationale in code commit messages. For us, rationale is the “why” behind the changes made in a commit. How does this align with your view of rationale ?
Q2	When writing code commits, how do you explain your commit’s relationship with previous commits in the project ?
Q3	When writing code commits, are you looking at other commit messages? What are you looking for?
Q4	When writing a commit, if you were looking for related commits (similar decisions, opposite decision, similar rationale, etc), how would you do this? Please describe the process.
Q5	When reviewing code commits, how do you understand the relationships with previous commits ?
Q6	When reviewing code commits, are you looking at other commit messages? What are you looking for?
Q7	When reviewing a commit, if you were looking for related commits (similar decisions, opposite decision, similar rationale, etc), how would you do this? Please describe the process.
Q8	What is good rationale in code commits for you?
Q9	Do you think the definition of good rationale is the same for developers and maintainers?
Q10	What’s bad rationale in code commits for you?
Q11	What’s important in your project to ensure commit messages quality? Any guidelines or processes ? Any tools such as linters?
Q12	What would be a useful tool, if any, to find related commits for you?
Q13	What would be a useful tool, if any, to help you write better rationale in your commit messages?

Follow-up Interviews Results



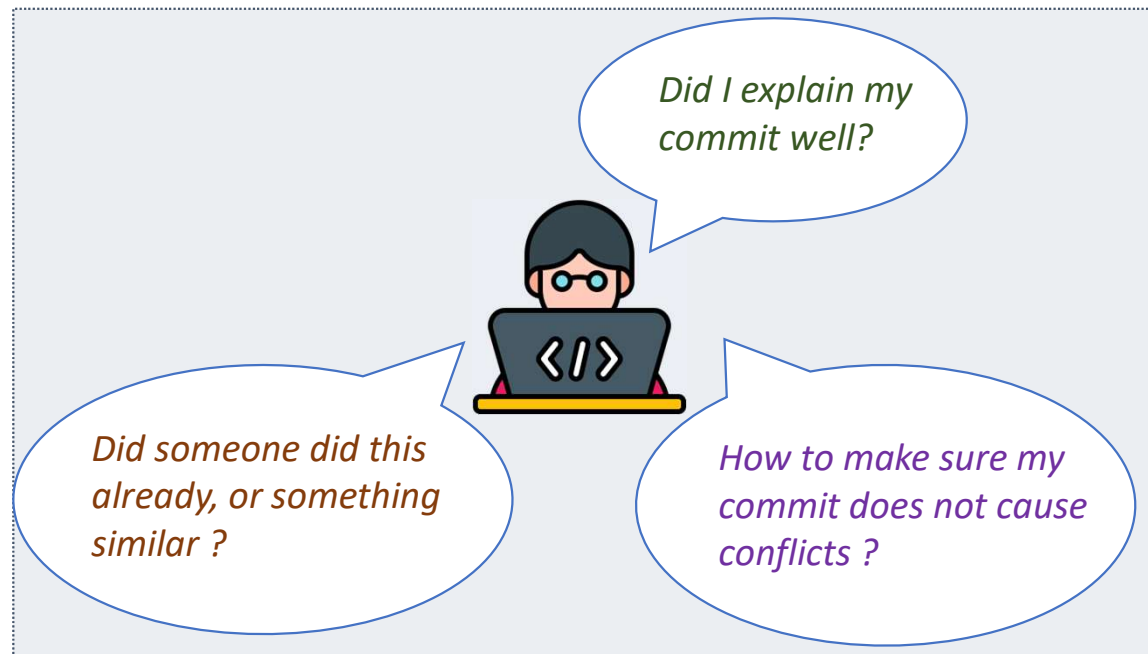
The interviews validated several survey findings:

- The **importance**, the **necessity**, and the **difficulty** of finding contextual information for a given commit.
- The **importance** of rationale compliance, consistency, and sufficiency within commit messages.
- Rationale information often needs to be **refined** when writing commit messages.
- The **insufficiency** of current practice and the **need** for dedicated tool support for finding related commits.

More insights:

Rationale embedded in code comments **vs** external documentation.

Developers needs



36

Thesis Statement



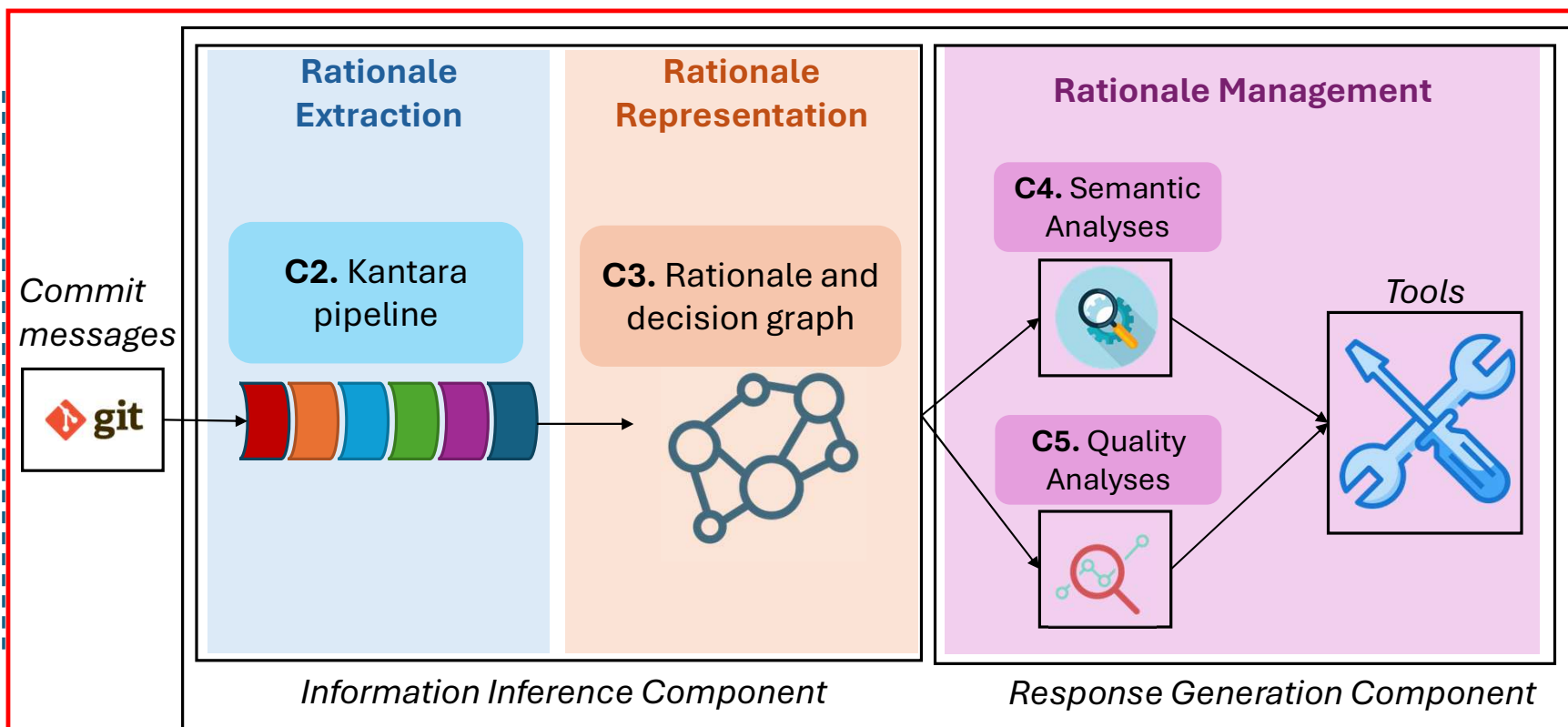
Automated **evidence-based** quantitative and semantic rationale **analyses**
can provide **effective support** for
rationale-aware commit-based **development**.

Problem
space

Rationale Understanding

C1. Empirical understanding of rationale practices

Solution
space



38

Introduction

Related Work

Contributions (C1 - C2 - C3 - C4 - C5 - Tools)

Conclusion

Solution-space



How can we *extract, structure and manage* rationale information from code commits?

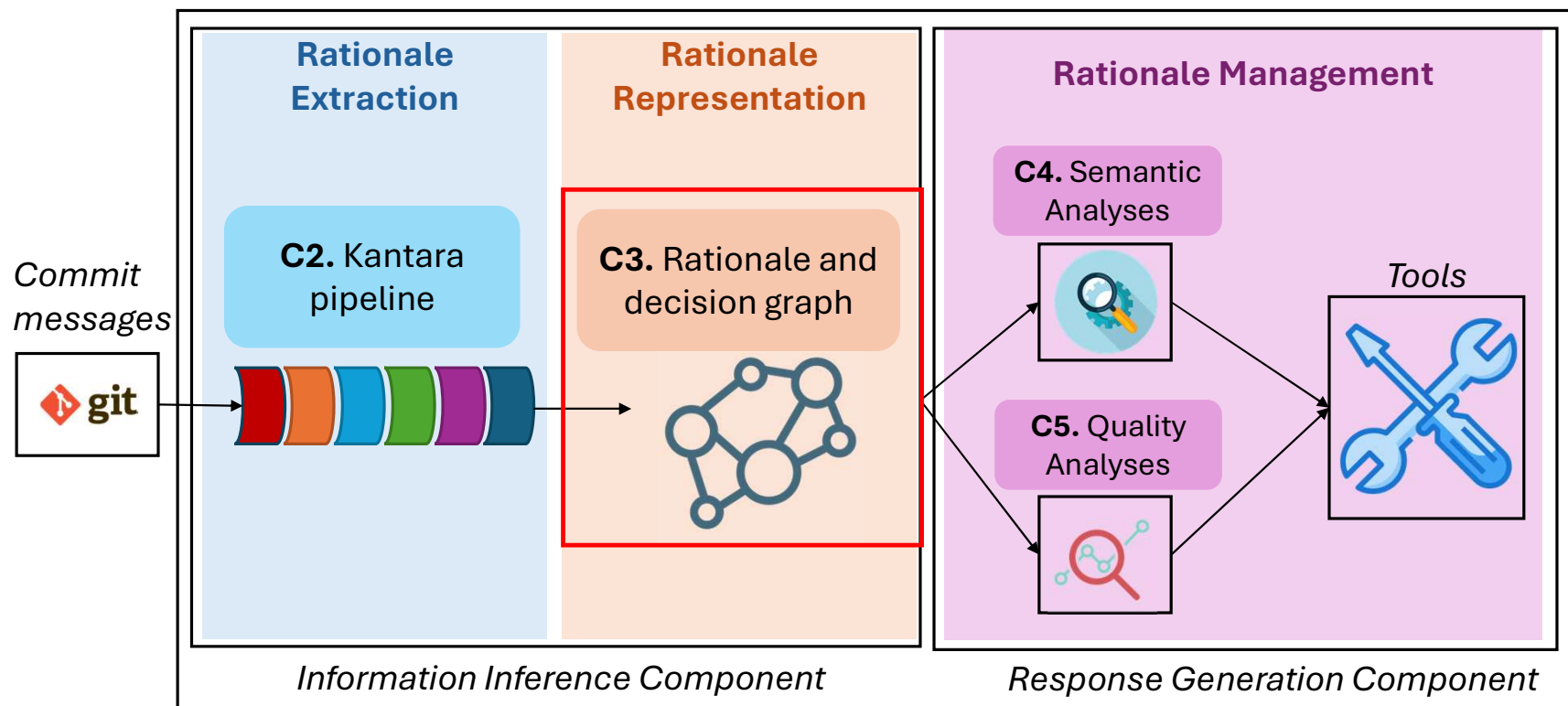


39

Solution-space

Ch3. Inflexibility of rationale representations

How can we extract, structure and manage rationale information from code commits?

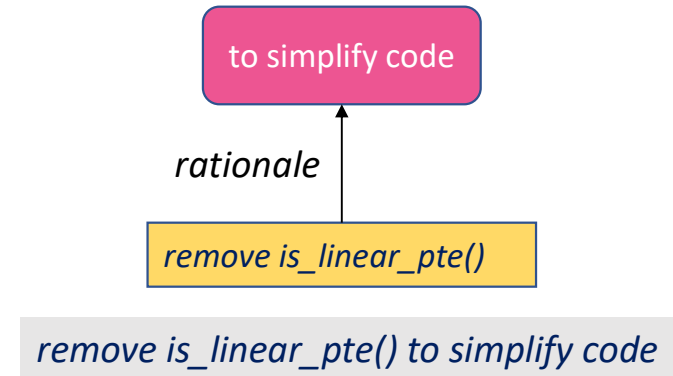
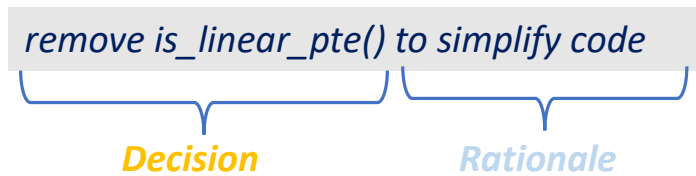


40

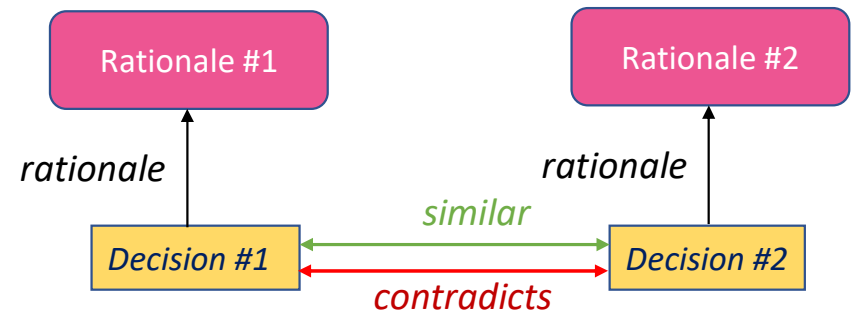
Structure: Rationale and Decisions Graph



ASE NIER'22

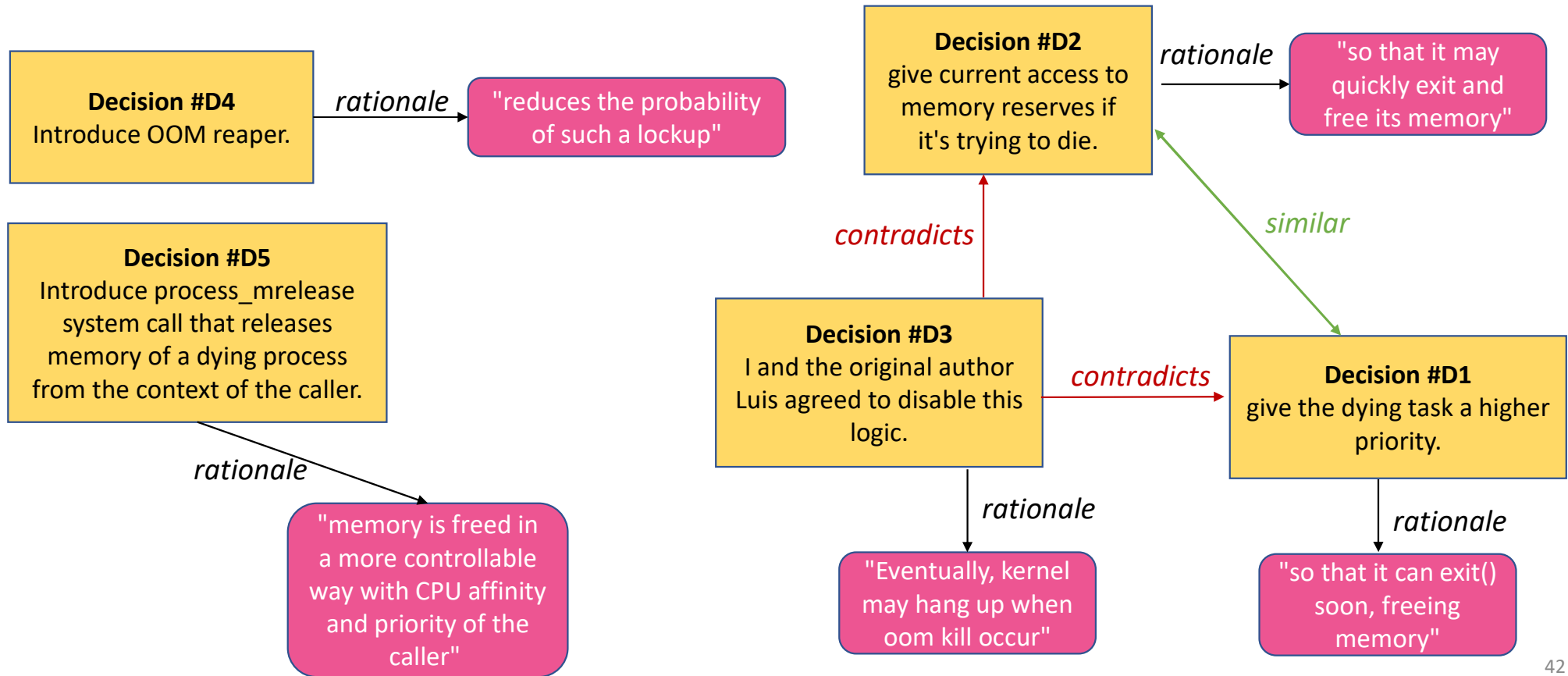
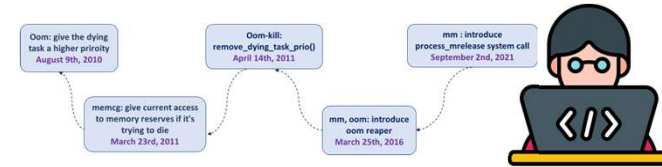


- **Nodes:** Pairs (*decision* and *rationale*)
- **Edges:** relationships between **decisions**



41

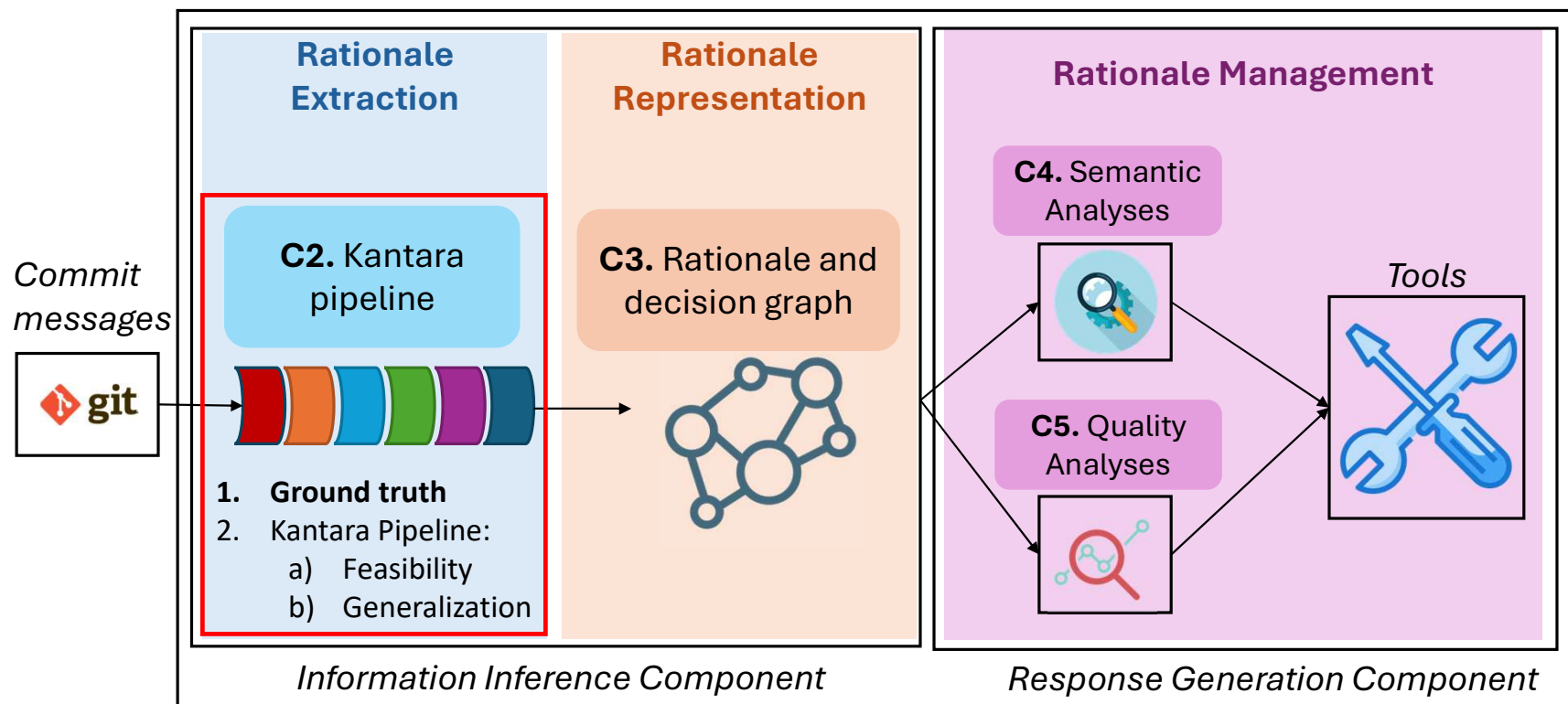
Rationale and Decision Graph (C3)



42

Solution-space

How can we extract *structure* and *manage* rationale information from code commits?





ICPC'24

Out-Of-Memory (OOM) Killer Dataset

- Collect commit history of the OOM-Killer
- Codebook:
 - 4 labels: “Decision”, “Rationale”, “Supporting Facts”, “Inapplicable”.
- Batch annotations of the sentences
 - 3 annotators manually label sentences
 - Multiple classifications per sentence
 - In disagreement, take classification union
 - Fleiss kappa: ~ 0.66

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)

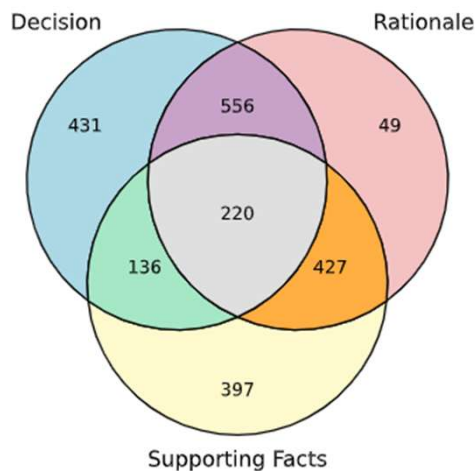


Out-Of-Memory (OOM) Killer Dataset

Column	Value
commit number	4
commit ID	C_kwDOACN7MtoAKGExOWNhZDA2OTE1OTdlYjc5YzEyM2I4YTE5YTlmYWJhNWFiN2Q5MGU
author name	A [REDACTED] M [REDACTED]
committer name	akpm
message	mm/oom_kill.c: fix vm_oom_kill_table[] ifdeffery arm allnoconfig: mm/oom_kill.c:60:25: warning: 'vm_oom_kill_table' defined but not used [-Wunused-variable] 60 static struct ctl_table vm_oom_kill_table[] = Cc: Luis Chamberlain <mcgrof@kernel.org> Signed-off-by: Andrew M [REDACTED] <akpm@linux-foundation.org>
URL	https://api.github.com/repos/torvalds/linux/git/commits/a19cad0691597eb79c123b8a19a9faba5ab7d90e
message_preprocessed	mm/oom_kill.c: fix vm_oom_kill_table[] ifdeffery
Decision	yes
Rationale	yes
Supporting Facts	no

Example dataset entry showing the structure

Examples



Sentence

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

Labelling

Decision
 Supporting Facts
 Rationale
 Decision

Sentence

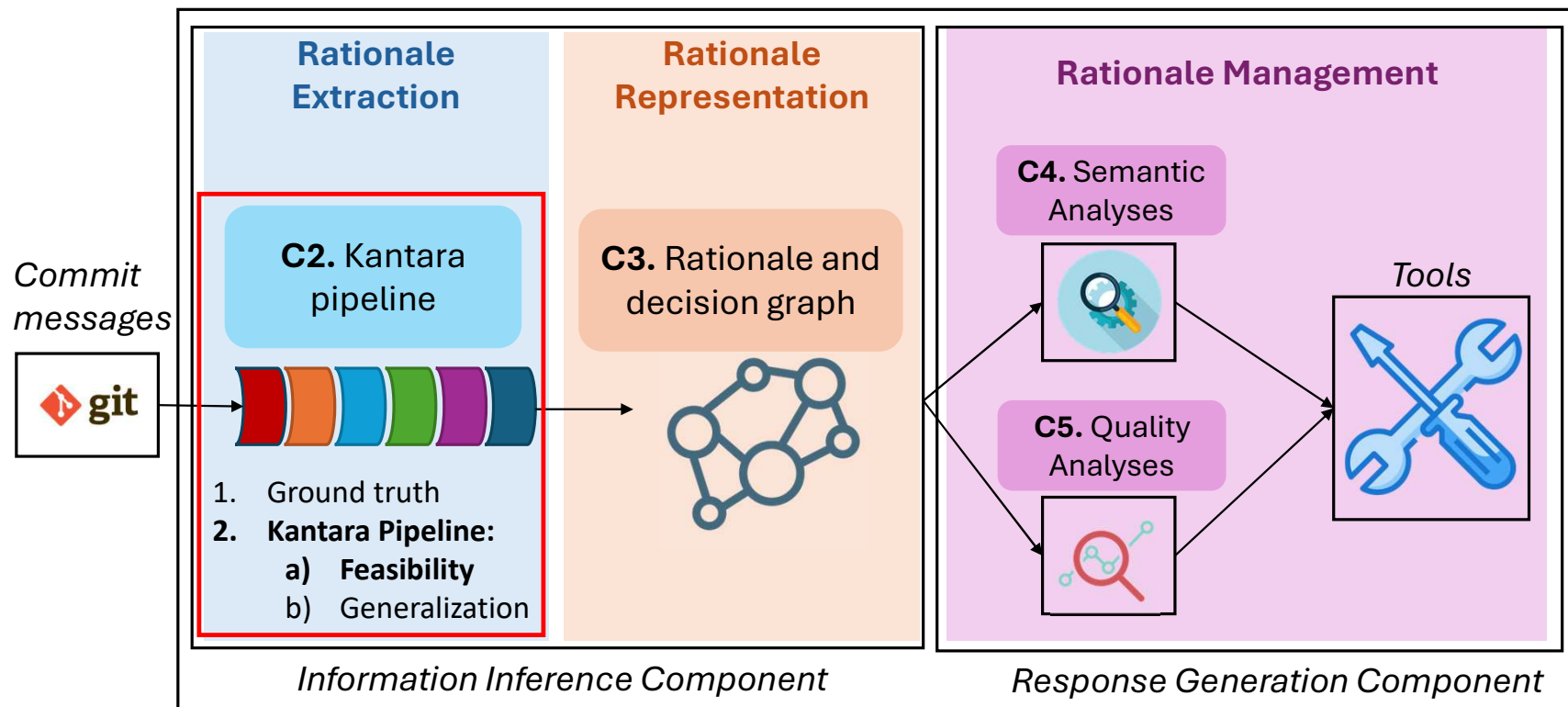
tlb: mmu_gather: Remove start/end arguments from tlb_gather_mmu()
 The 'start' and 'end' arguments to tlb_gather_mmu() are no longer needed now that there is a separate function for 'fullmm' flushing
 Remove the unused arguments and update all callers.

Labelling

Decision
 Rationale, Supporting Facts
 Decision, Rationale



Extraction – Kantara pipeline (C2)



47

Kantara pipeline (C2)

Kantara **only** works on sentences containing **both** Decision and Rationale!



ASE NIER'22

Extracting Decision-Rationale triples

Extracting Decision-Decision Relationships

Decisions
extraction

Rationale
extraction

Relatedness
relationship

Similar
relationship

Contradicts
relationship

History
relationship

to simplify code

rationale

`remove is_linear_pte()`

Rationale #1

rationale

Decision #1

Rationale #2

rationale

Decision #2

similar

contradicts

`remove is_linear_pte()` to simplify code

Introduction

Related Work

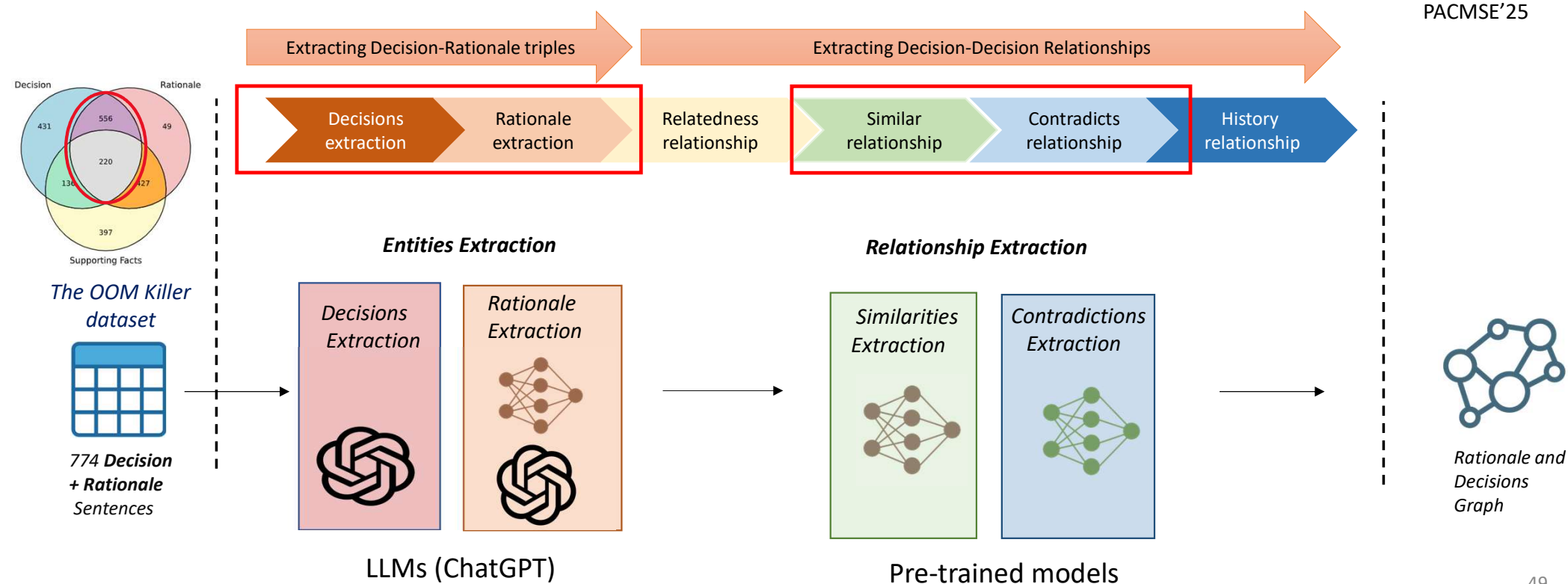
Contributions (C1 - **C2** - C3 - C4 - C5 - Tools)

Conclusion

Kantara pipeline (C2)



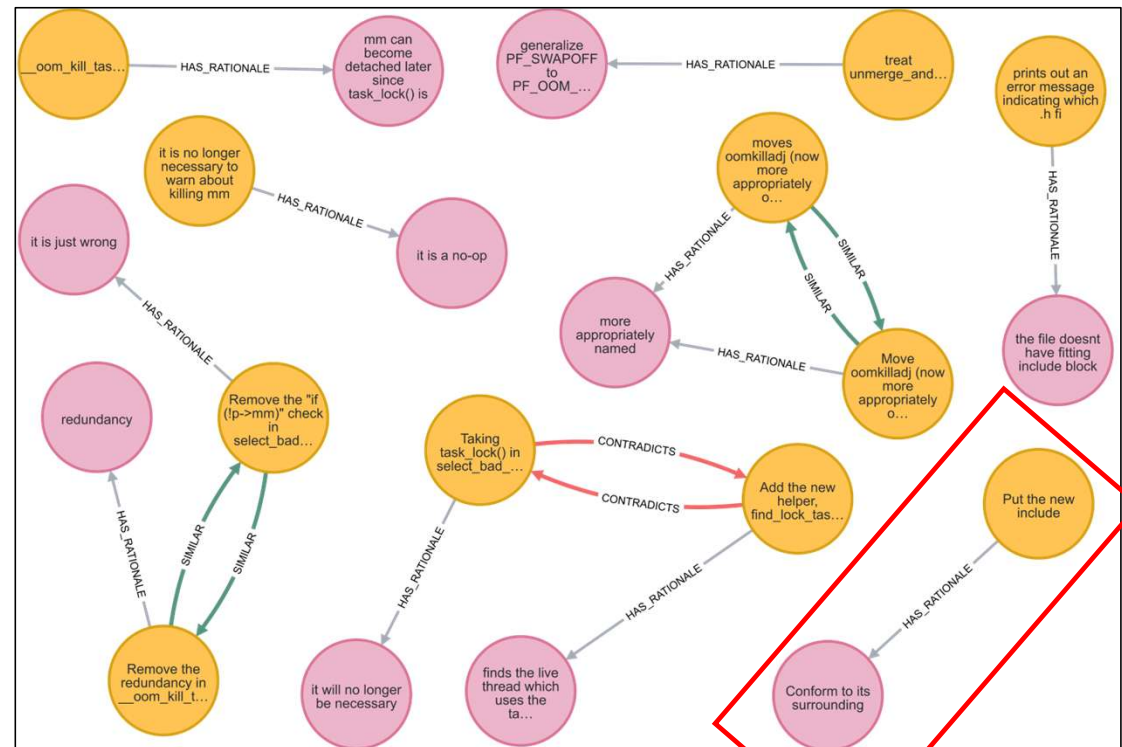
PACMSE'25



Kantara output – OOM-Killer

- 774 **Decision + Rationale** Sentences
- 527 extracted decisions:
⇒ 138601 **pairs** of decisions

⇒ Computed these relationships between **each pair** of the extracted **decisions** with threshold of **0.9**
- 29 **similarities**
- 396 **contradictions**
- **Performance:** ~ 3 hours.

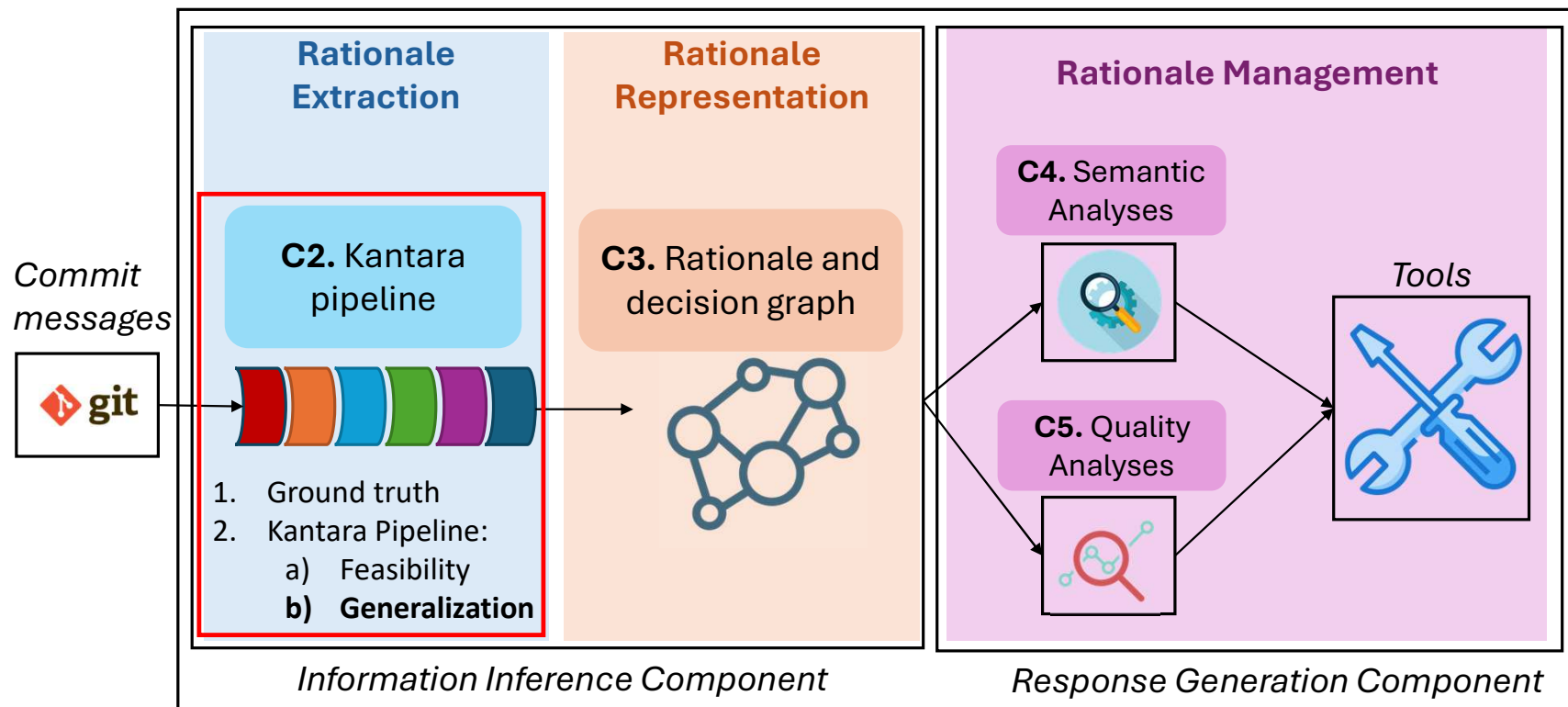


Part of the *Rationale and Decisions* graph for the OOM-Killer module of the Linux Kernel.

50



Extraction – Kantara pipeline (C2)



51



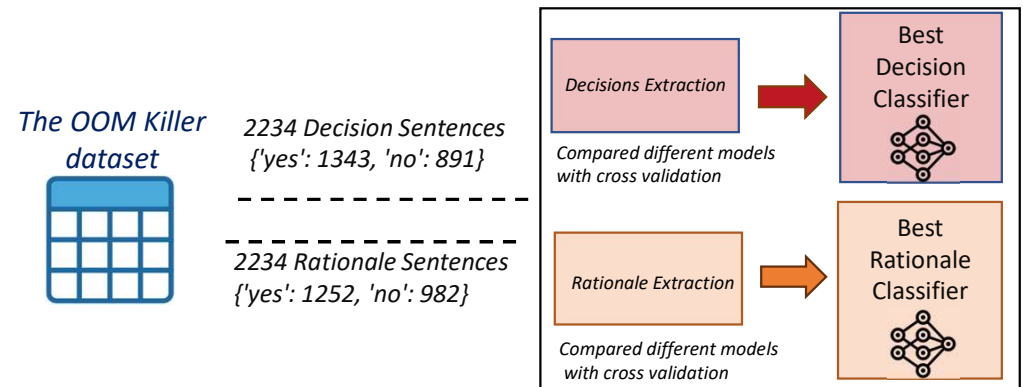
How to Generalize?

- Kantara **only** works on sentences containing **both** Decision and Rationale!
- We need sentences that are both decision and rationale *remove is_linear_pte() to simplify code*
=> Two binary classification tasks: **Decision-containing** and **Rationale-containing** sentence

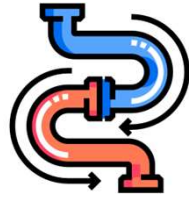
Several Binary classifiers:

1. Trained on the OOM-Killer dataset
 2. Evaluated with 10-fold cross validation on the OOM-Killer dataset
- => Choose **Best** Models

Automating the extraction of decision and rationale sentences



Extractors generalizable?



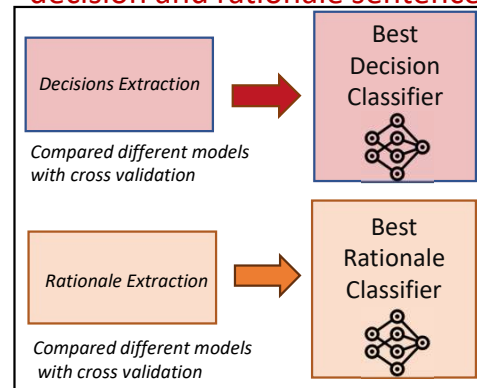
The OOM Killer dataset



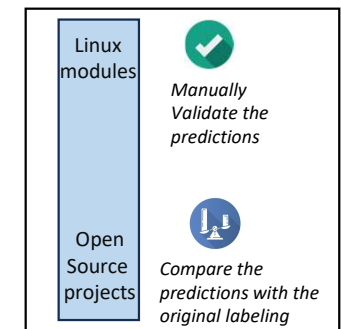
2234 Decision Sentences
{'yes': 1343, 'no': 891}

2234 Rationale Sentences
{'yes': 1252, 'no': 982}

Automating the extraction of decision and rationale sentences

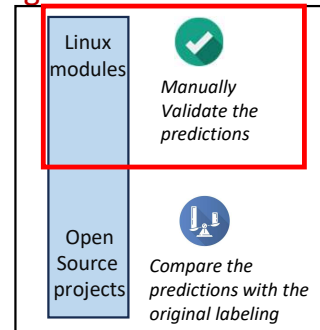


Investigation of generalization



- 2 Linux modules: *slob.c* and *button.c*
- 5 Github projects: *Junit*, *Retrofit*, *SpringBoot*, *Dubbo*, *Okhttp*

4. Investigation of generalization



Extractors generalizable? - Validation

Linux: *slob.c* and *button.c*

- post-label commit-based sampling
- random 20 commits per file / 91 sentences per file
- individually indicated whether or not we agree with the predictions of the classifiers

Table 9. Bi-LSTM inter-rater agreement.

Rater	mm/slob.c		drivers/acpi/button.c	
	Decision	Rationale	Decision	Rationale
Rater 1	70.3%	69.2%	73.6%	80.2%
Rater 2	73.6%	70.3%	71.4%	80.2%
Rater 3	73.6%	80.2%	65.9%	73.6%
Average	72.5%	73.6%	70.3%	78%
Inter-rater agreement	79.1%	65.9%	75.8%	60.4%

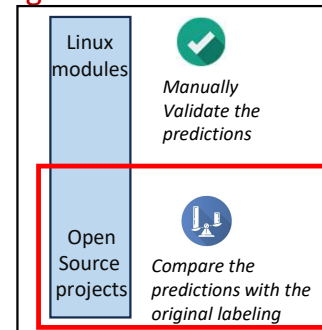
Extractors generalizable? - Validation

Other OS projects:

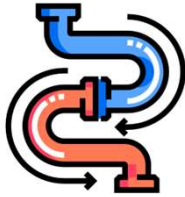
- Dataset of [Tian et al. *What Makes a Good Commit Message?*, ICSE'22]
- 1649 commit messages from five OSS projects.
- Labels: *Why_and_What*, *No_What*, *No_Why*, and *Neither*.
- Align labels to compare [**Why = Rationale** and **What = Decision**].

Project	Decision				Rationale			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
JUnit4	0.60	0.81	0.65	0.72	0.57	0.58	0.75	0.66
Apache Dubbo	0.57	0.72	0.70	0.71	0.66	0.77	0.78	0.78
Square Retrofit	0.64	0.92	0.66	0.77	0.57	0.55	0.80	0.65
Square OkHttp	0.57	0.87	0.59	0.71	0.62	0.71	0.77	0.74
Spring-boot	0.46	0.83	0.47	0.60	0.74	0.96	0.75	0.84
All commits	0.56	0.83	0.60	0.70	0.64	0.73	0.77	0.75

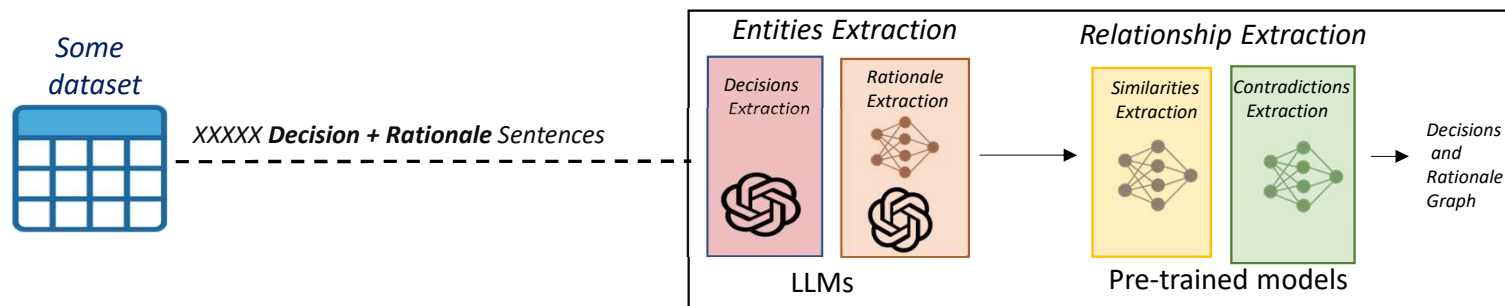
4. Investigation of generalization

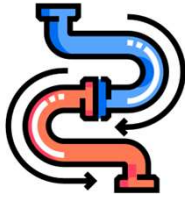


Kantara can generalize

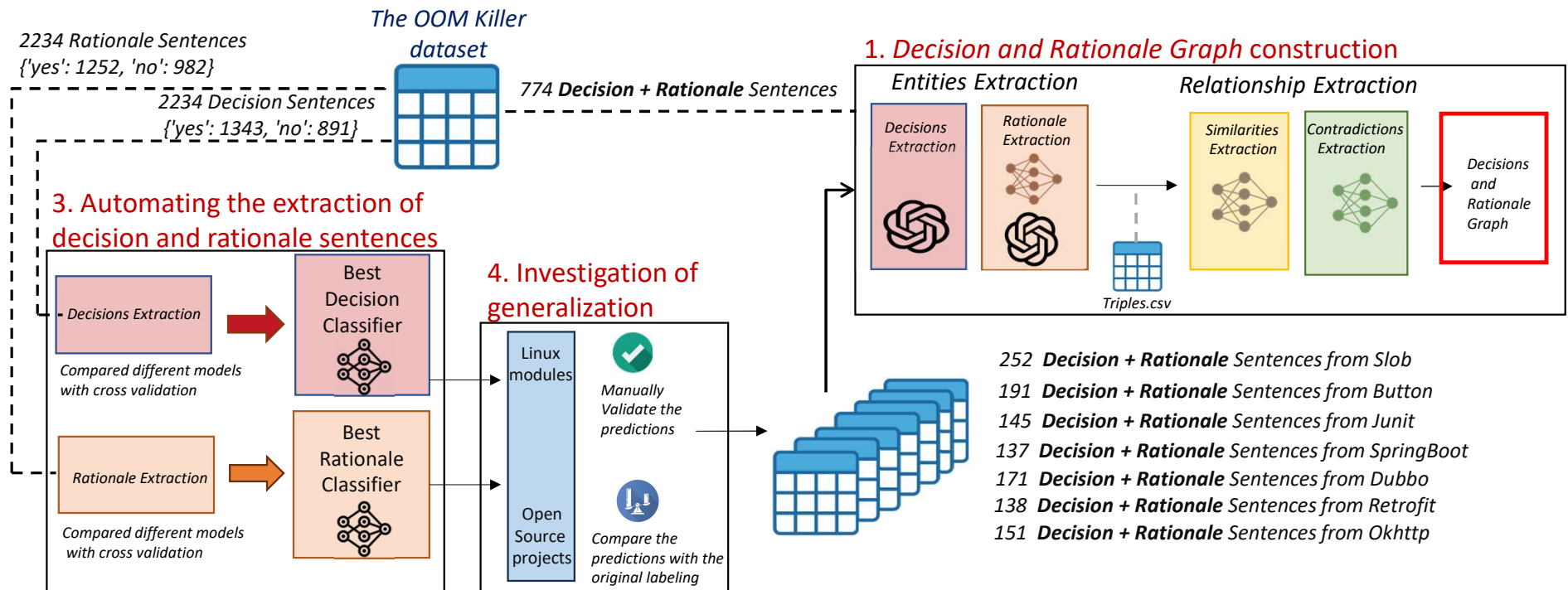


LLMs and pre-trained models + Reusable classifiers = Kantara can **generalize**





Application to Other Contexts

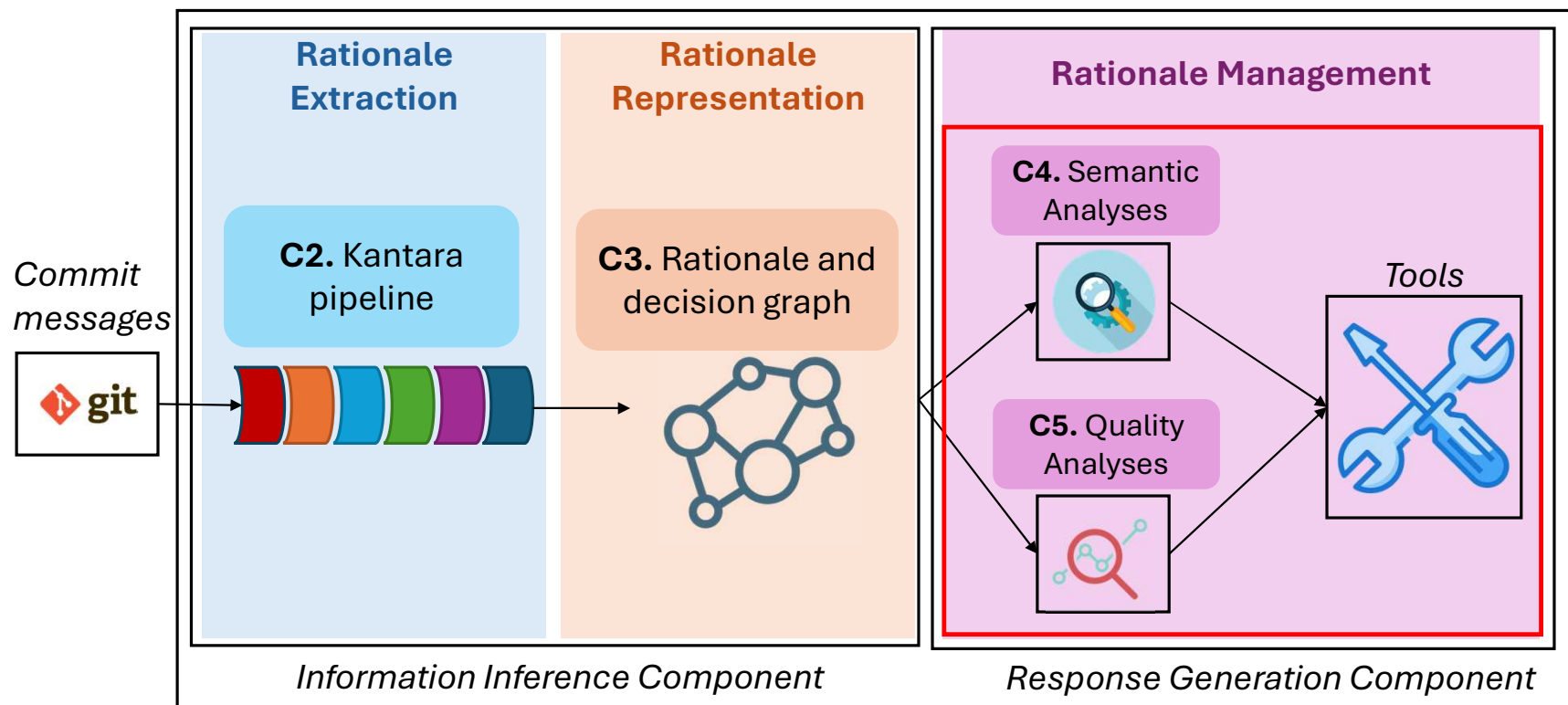


Solution-space

Ch4. Limited semantic inference techniques

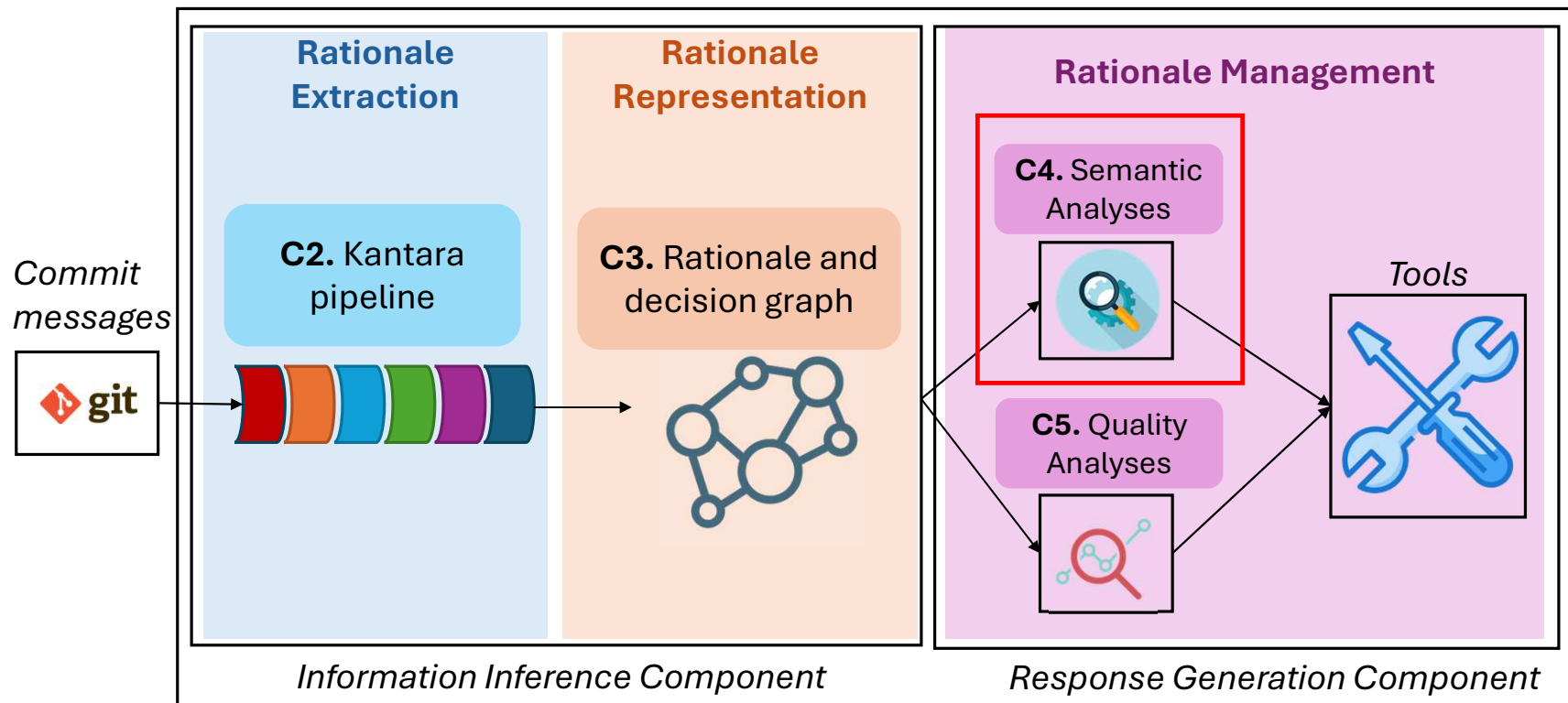
Ch5. Limited quantitative rationale analyses

How can we *extract, structure and* **manage** rationale information from code commits?



58

Mangement- Semantic Analyses (C4)



ASE NIER'22



PACMSE'25



Semantic Rationale Analyses (C4)

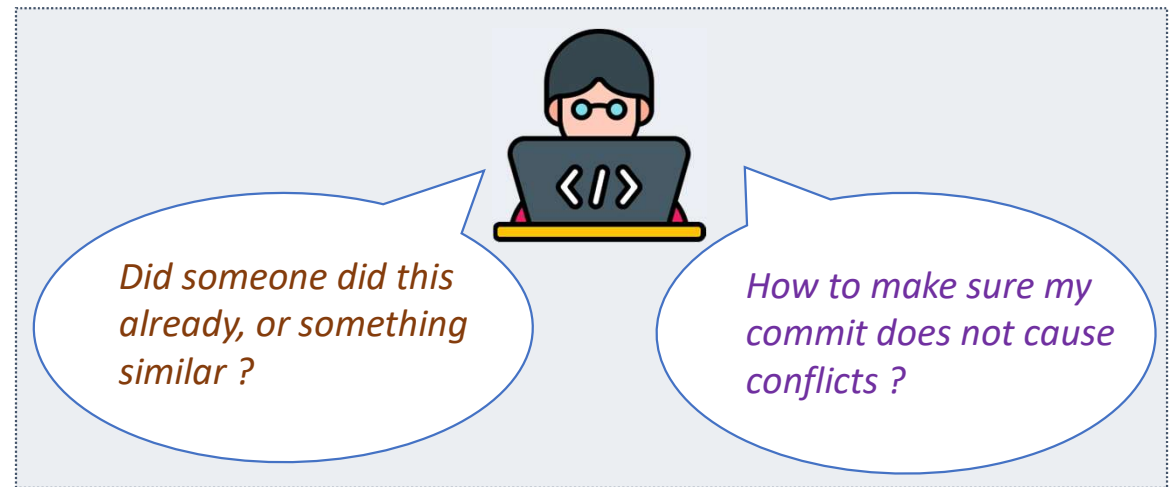
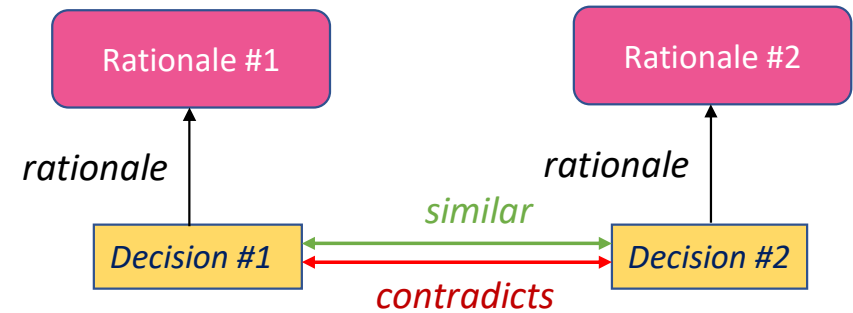
Decision-Decision Relationships



- Two **similar** Decisions
 - Reuse past solutions



- Two **Contradictory** Decisions
 - Avoid potential conflicts and reasoning problems



60

Semantic Rationale Analyses (C4)



- Reuse past solutions
 - Two similar Decisions

Did someone did this already, or something similar ?

Nº	Decision 1	Decision 2	Relationship
1	Prepare for new header dependencies before moving code to <linux/sched/coredump.h>	Prepare for new header dependencies before moving code to <linux/sched/mm.h>	Similar (0.99)
2	Create a trivial placeholder <linux/sched/coredump.h> file that just maps to <linux/sched.h> to make this patch obviously correct and bisectable	Create a trivial placeholder <linux/sched/mm.h> file to make this patch obviously correct and bisectable	Similar (0.94)
3	Remove unnecessary locking in exit_oom_victim()	remove unnecessary locking in oom_enable()	Similar (0.93)
4	moves oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct signal_struct	Move oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct mm_struct	Similar (0.97)

Semantic Rationale Analyses (C4)



- Avoid potential conflicts
 - Two **contradictory** Decisions

*How to make sure
my commit does not
cause conflicts ?*

Nº	Decision 1	Decision 2	Relationship
5	replace try_oom_reaper wake_oom_reaper	by This patch adds try_oom_reaper.	Contradicts (0.98)
6	push the re-check loop out of freeze_processes into check_frozen_processes	We are not checking whether the task is frozen	Contradicts (0.96)
7	giving the dying task an even higher priority	to avoid boosting the dying task priority in case of mem_cgroup OOM	Contradicts (0.97)
8	Add the new helper, find_lock_task_mm()	Taking task_lock() in select_bad_process() to check for OOM_DISABLE and in oom_kill_task() to check for threads sharing the same memory will be removed in the next patch in this series	Contradicts (0.91)
9	Move oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct mm_struct	make the magical value of -17 in /proc/<pid>/oom_adj defeat the oom-killer altogether	Contradicts (0.95)

62

Semantic Rationale Analyses (C4)



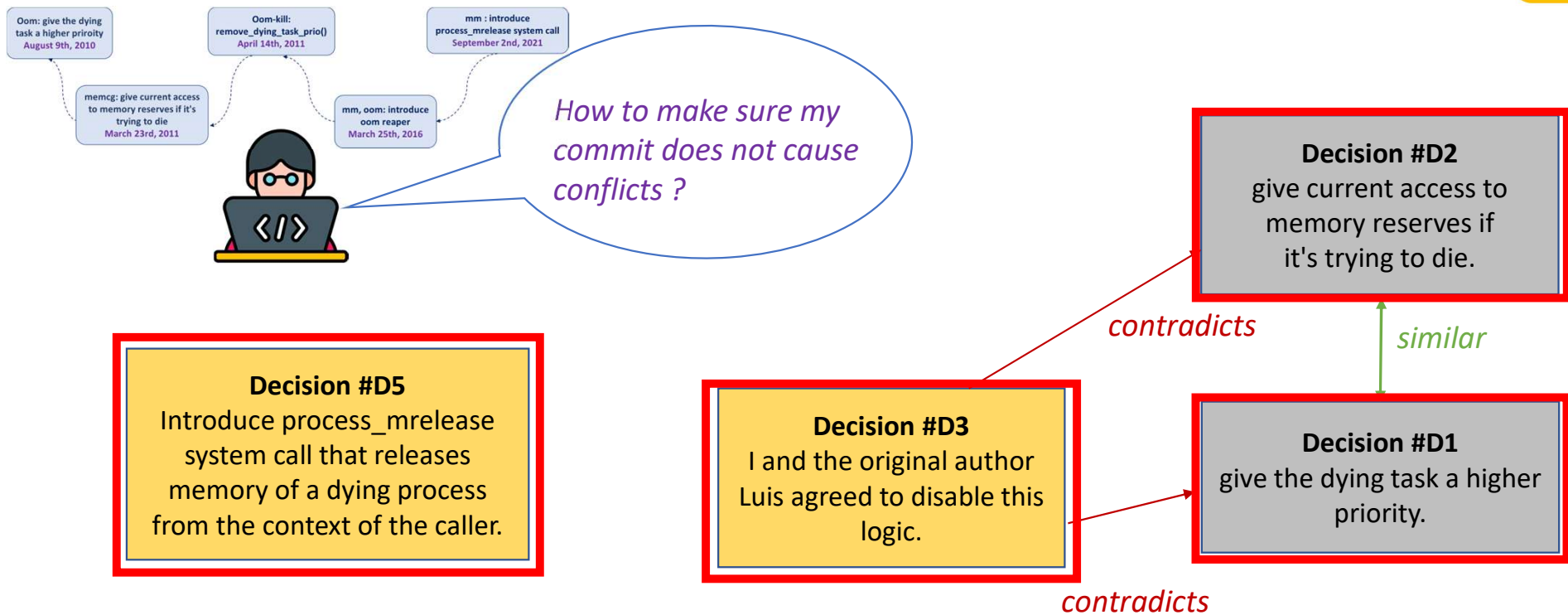
- Avoid potential conflicts
 - Two similar Decisions & Two contradictory Decisions

How to make sure my commit does not cause conflicts ?

Nº	Decision 1	Decision 2	Relationship
1	Prepare for new header dependencies before moving code to <linux/sched/coredump.h>	Prepare for new header dependencies before moving code to <linux/sched/mm.h>	Similar (0.99)
2	Create a trivial placeholder <linux/sched/coredump.h> file that just maps to <linux/sched.h> to make this patch obviously correct and bisectable	Create a trivial placeholder <linux/sched/mm.h> file to make this patch obviously correct and bisectable	Similar (0.94)
3	Remove unnecessary locking in exit_oom_victim()	remove unnecessary locking in oom_enable()	Similar (0.93)
4	moves oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct signal_struct	Move oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct mm_struct	Similar (0.97)
5	replace try_oom_reaper by wake_oom_reaper	This patch adds try_oom_reaper.	Contradicts (0.98)
6	push the re-check loop out of freeze_processes into check_frozen_processes	We are not checking whether the task is frozen	Contradicts (0.96)
7	giving the dying task an even higher priority	to avoid boosting the dying task priority in case of mem_cgroup OOM	Contradicts (0.97)
8	Add the new helper, find_lock_task_mm()	Taking task_lock() in select_bad_process() to check for OOM_DISABLE and in oom_kill_task() to check for threads sharing the same memory will be removed in the next patch in this series	Contradicts (0.91)
9	Move oomkilladj (now more appropriately named oom_adj) from struct task_struct to struct mm_struct	make the magical value of -17 in /proc/<pid>/oom_adj defeat the oom-killer altogether	Contradicts (0.95)

63

Semantic Rationale Analyses (C4)





Semantic Rationale Analyses (C4)

Feasibility & Generalization

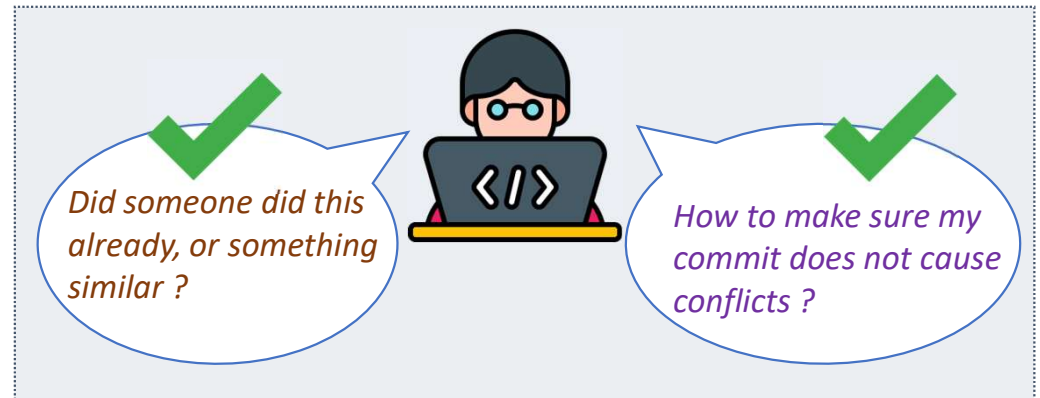
Automatically supporting **effective rationale analyses**, such as finding **potentially** similar and contradictory decisions, is **feasible** and with **reasonable** performance.



1. Reuse past solutions



2. Avoid reasoning conflicts



65

Semantic Rationale Analysis Tools



Rationale Search Engine

Select a Module

Springboot

Similarities

Similarities_file: similarities_oss_SPRINGBOOT_with_0.5_threshold_with_URL.csv

	Decision1	Rationale1	URL1
0	Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
1	Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
2	Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
3	Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
4	Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
5	Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
6	Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
7	Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring

Rationale Search Engine
(**Retrospective Analysis**)



Rationale RAG
(**Prospective Analysis**)



Rationale RAG

Select a Module

Springboot

Triples_file: ChatGPT_decision_rationale_oss_SPRINGBOOT_with_URL.csv

Enter your commit

Fix checkstyle violation

Similar commits

Enter the cosine similarity threshold:

0.6

4 similar commits

	Commit	Url
0	Fix checkstyle violation see the PR link	https://github.com/sp
1	Fix checkstyle errors fix	https://github.com/sp
2	Fix checkstyle header violation See <issue_link>	https://github.com/sp
3	Fix checkstyle violations nan	https://github.com/sp

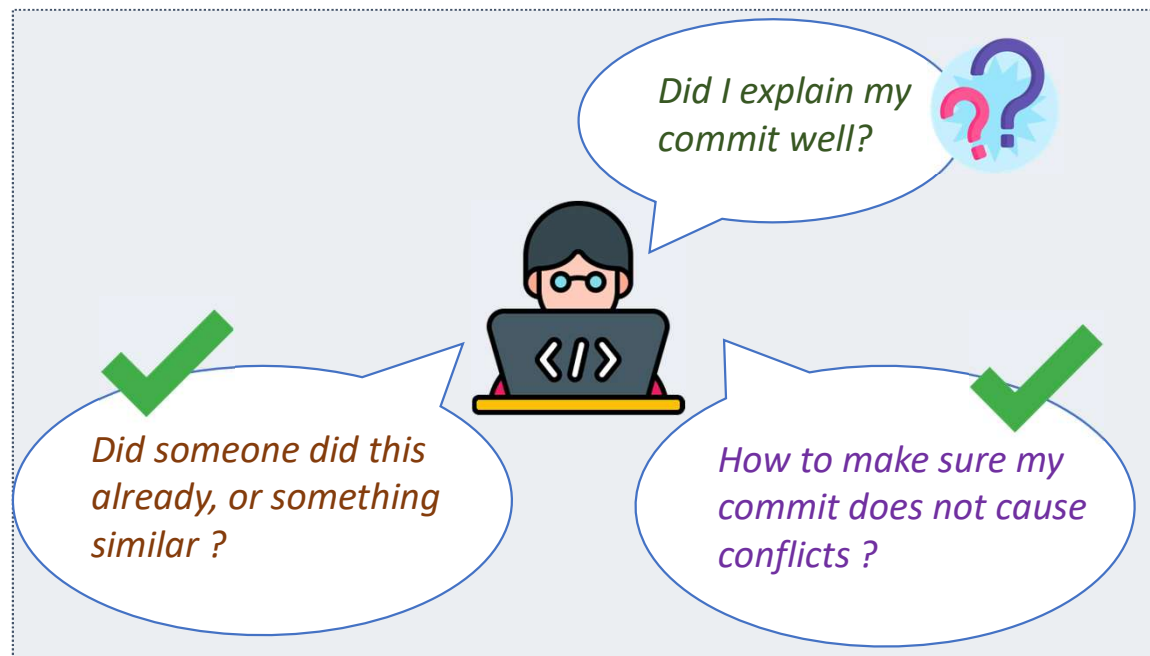
Get ChatGPT to double check these simila

Thesis Statement

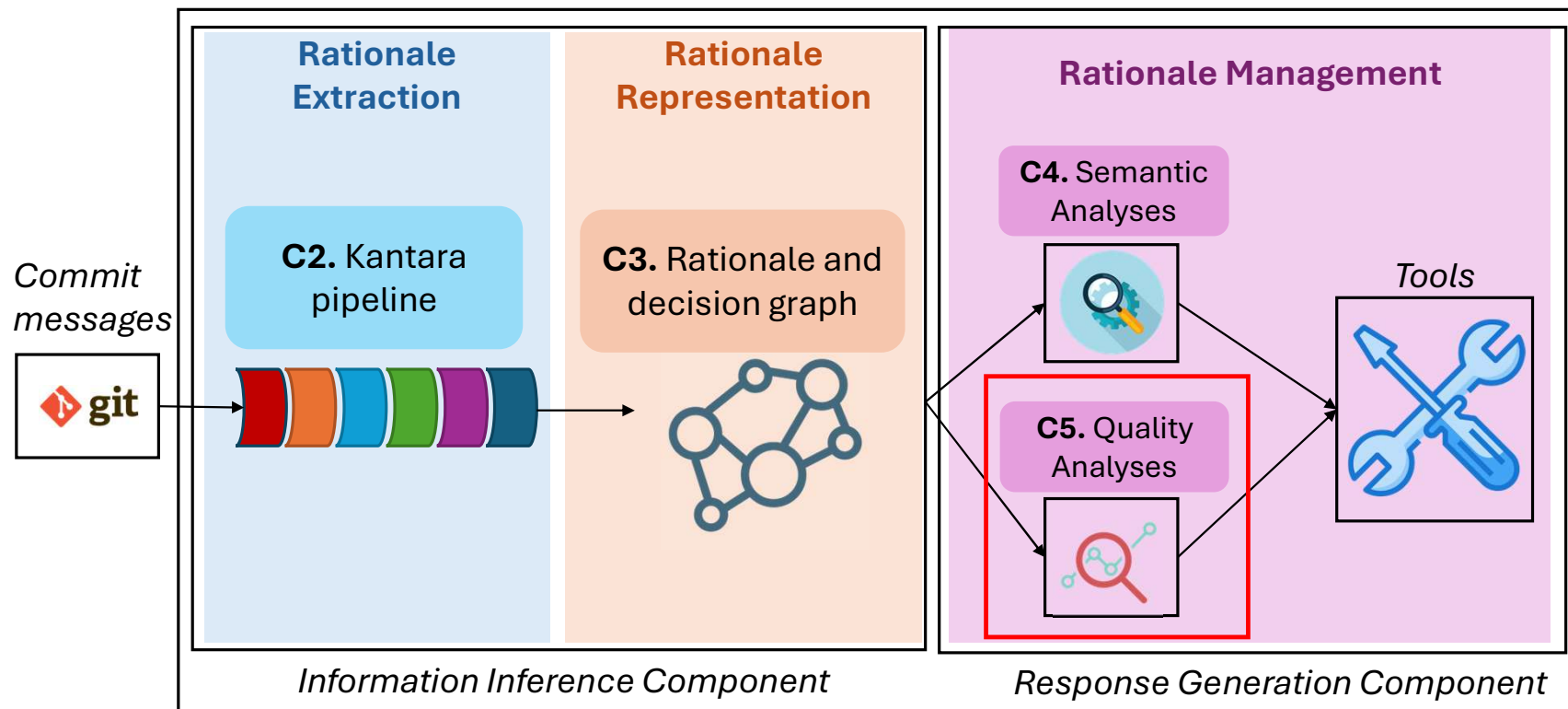


Automated **evidence-based quantitative** and **semantic** rationale **analyses**
can provide **effective support** for
rationale-aware commit-based development.

Third need?



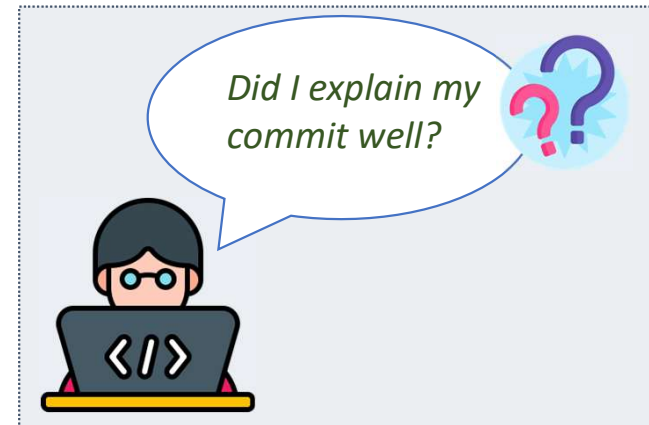
Mangement-Quality Analyses (C5)





Quantitative Analyses for Rationale Quality (C5)

1. How can we **understand** developers' rationale in **commit messages**?
2. How can we **help** developer's **write better commit messages**?



Quantitative Analyses for Rationale Quality



Presence of Rationale

- How **many** commits contain rationale?
- How **much** of the commit contains rationale?



Factors impacting Rationale

- Does the quantity of rationale reported depend on the commit message **size**?
- Does the quantity of rationale reported depend on the developer **experience**?



Evolution of rationale over time

- How does rationale **evolve** over time?
- How does rationale evolve over time for the five **core** contributors?



Structure of commit messages

- In what **order** do the categories mostly appear?

Quality Analyses: Rationale in the OOM-Killer



Presence

Commit messages **almost always** contain rationale information.

On average, around **60% of the message** contains rationale information



Impacting factors

The **quantity** of rationale information reported does **not** depend on the commit message **size** or developer **experience**.

Experienced developers have a rationale density around **60%**.



Evolution over time

Rationale density is **consistent** (~ 0.6).
Decision density is always **high** (> 0.5).
Supporting facts density is **lower** (< 0.6).

More experienced developers write **short** commit messages (fewer than eight sentences).



Structure of commit messages

Developers tend to **start and end** their commit messages with **Decisions**.

Rationale and Supporting Facts appear in the **middle** of the commit, with Supporting Facts usually preceding Rationale sentences

Quality Rationale Analyses



ICPC'24

1. How can we **understand** developers' rationale in **commit messages**?
2. How can we **help** developer's **write better commit messages**?

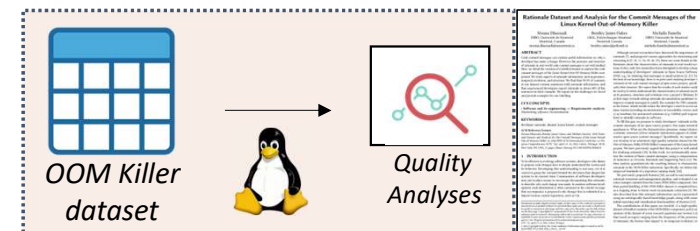
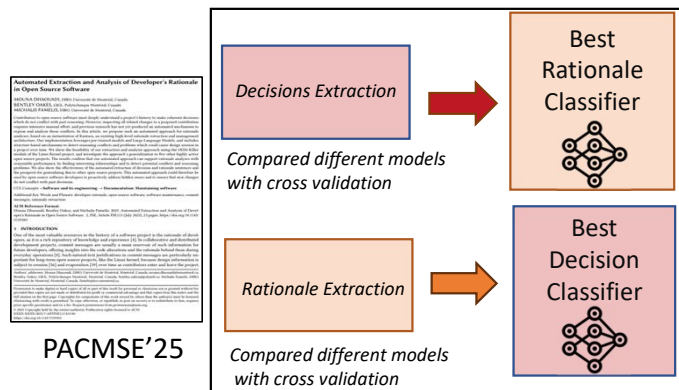
Quantitative analyses helps developers and maintainers to **evaluate** and **improve** their documentation practices, resulting in a **better understanding** of the system

How to Generalize?



Generalizable classifiers

- Two binary classifiers [Decision, Rationale]
 - Trained on OOM-Killer dataset
 - Validated on:
 - Two other Linux modules 
 - Five other open-source projects 





Rationale Quality Analysis Tool

The mm/slob.c module

Resulting dataset:

% preview of the labelled dataset

Number of commits: 146
Number of sentences: 833

Distribution

Decision only sentences: 233
Rationale only sentences: 162
Decision & Rationale sentences: 252
No Decision and No Rationale sentences: 186

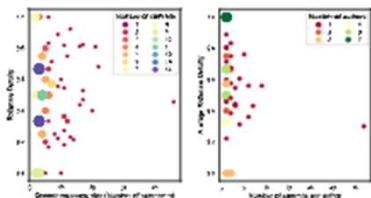
Word Clouds



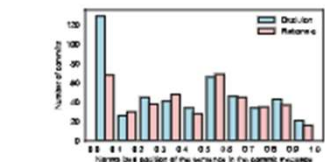
Rationale Presence

Total Number of commits: 146
Number of commits that contain rationale: 124
Rationale Percentage: 84.93%
Average Rationale Density: 0.56

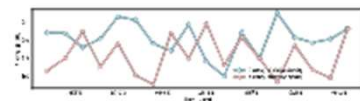
Rationale Factors



Commit Message Structure



Rationale Evolution



Module Analyzer (Retrospective Analysis)

- **Presence** of rationale
- **Factors** impacting rationale
- **Evolution** of rationale over time
- **Structure** of commit messages

Commit Message Analyzer (Prospective Analysis)

Commit Message Rationale Analyses

Enter your commit message

mm: make function get_attribute() public.

I changed the visibility of function get_attribute().
Now, bug number #333 is fixed.
Debugged by: Mouna

Start Commit Analysis

message_preprocessed	Decision	Rationale
0 mm: make function get_attribute() public.	1	0
0 I changed the visibility of function get_attribute().	1	0
0 Now, bug number #333 is fixed.	0	1

Number of sentences: 3

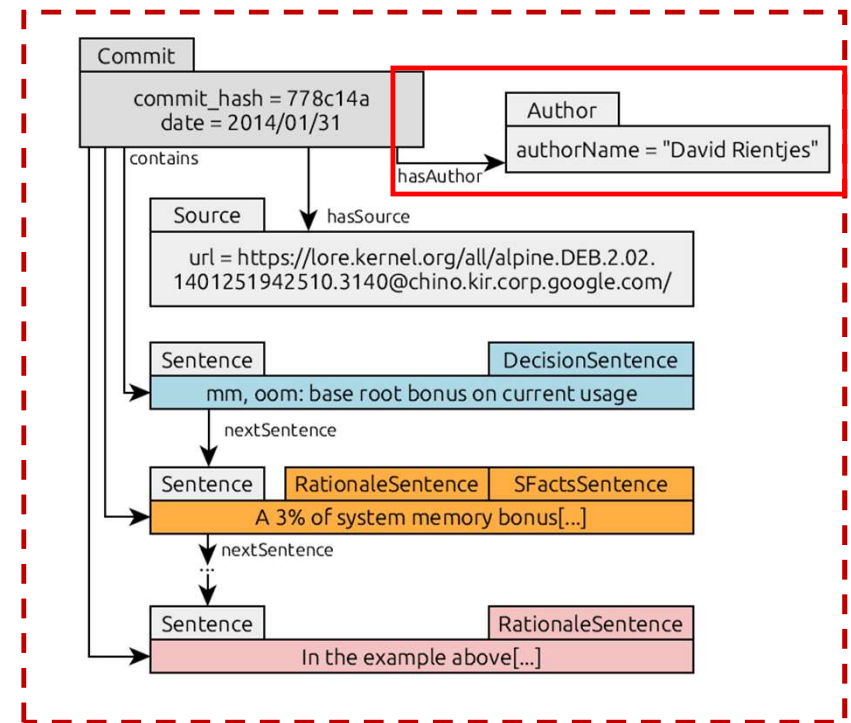
Your commit message's rationale density: 0.33

Your commit message's decision density: 0.67

Low Rationale! Please add justification for your changes.

More Quality Analyses

- Commit-level representation
 - **Author**
 - Date
 - Next-Sentence



Ontological-based approach



MDE-Intelligence'23

1. Defined the ontology in the OpenCAESAR framework
2. Populated the tool with our dataset 
3. Inferencing using SPARQL queries

```
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
]
```

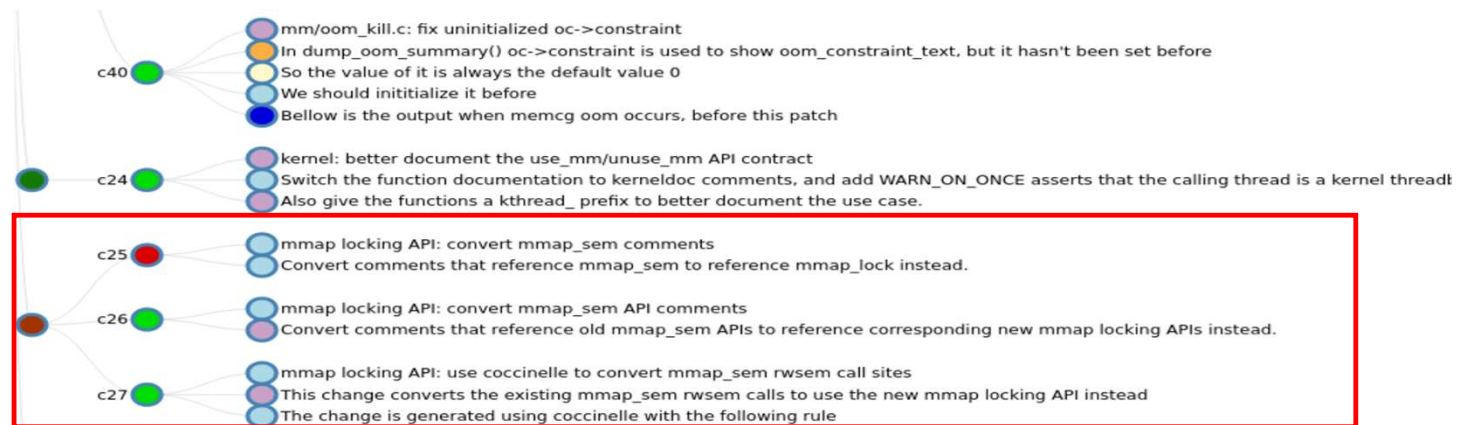
TOWARDS UNDERSTANDING AND ANALYZING RATIONALE

```
{
  "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" }
}
```

77

Rationale analyses and Visualization

Interactive Visualisation: Visualizing the results of some queries



- Identify commits/subsystems/**developers** with insufficient rationale

Thesis Statement



Automated **evidence-based quantitative** and **semantic** rationale **analyses**
can provide **effective support** for
rationale-aware commit-based development.

Different types of Rationale Analyses

- Not only semantic analyses



1. Reuse past solutions



2. Avoid reasoning conflicts



Did someone did this already, or something similar ?

How to make sure my commit does not cause conflicts ?

..... But also quantitative analyses..



Did I explain my commit well ?



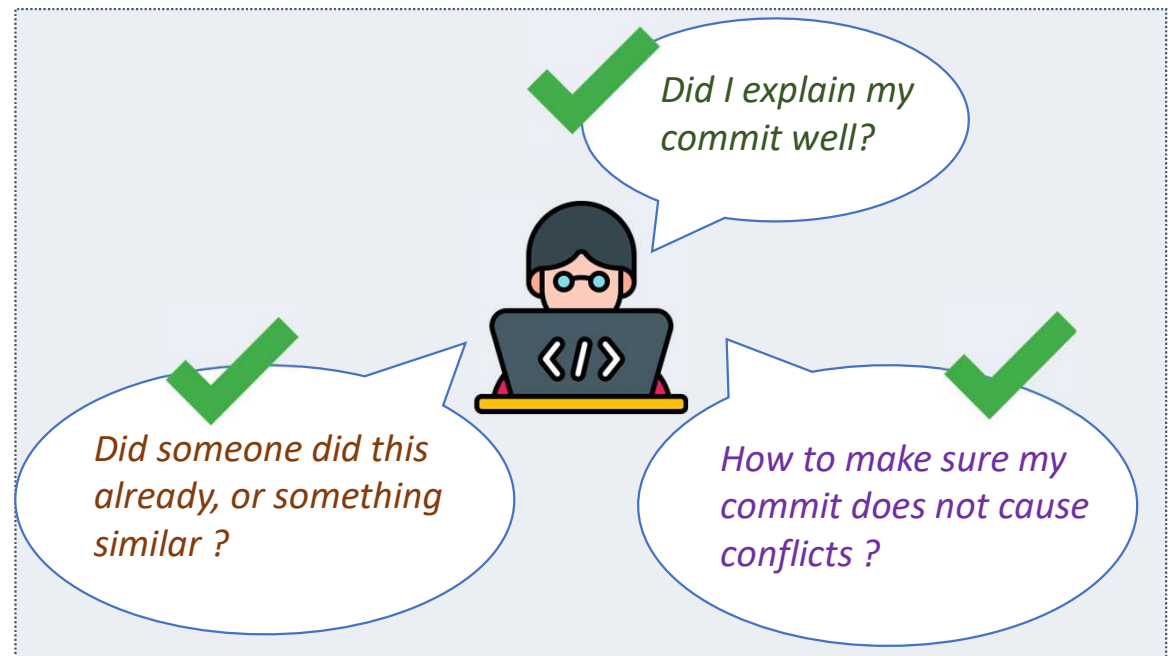
3. Evaluate and improve documentation practices, resulting in a better understanding of the system

80

Needs Addressed

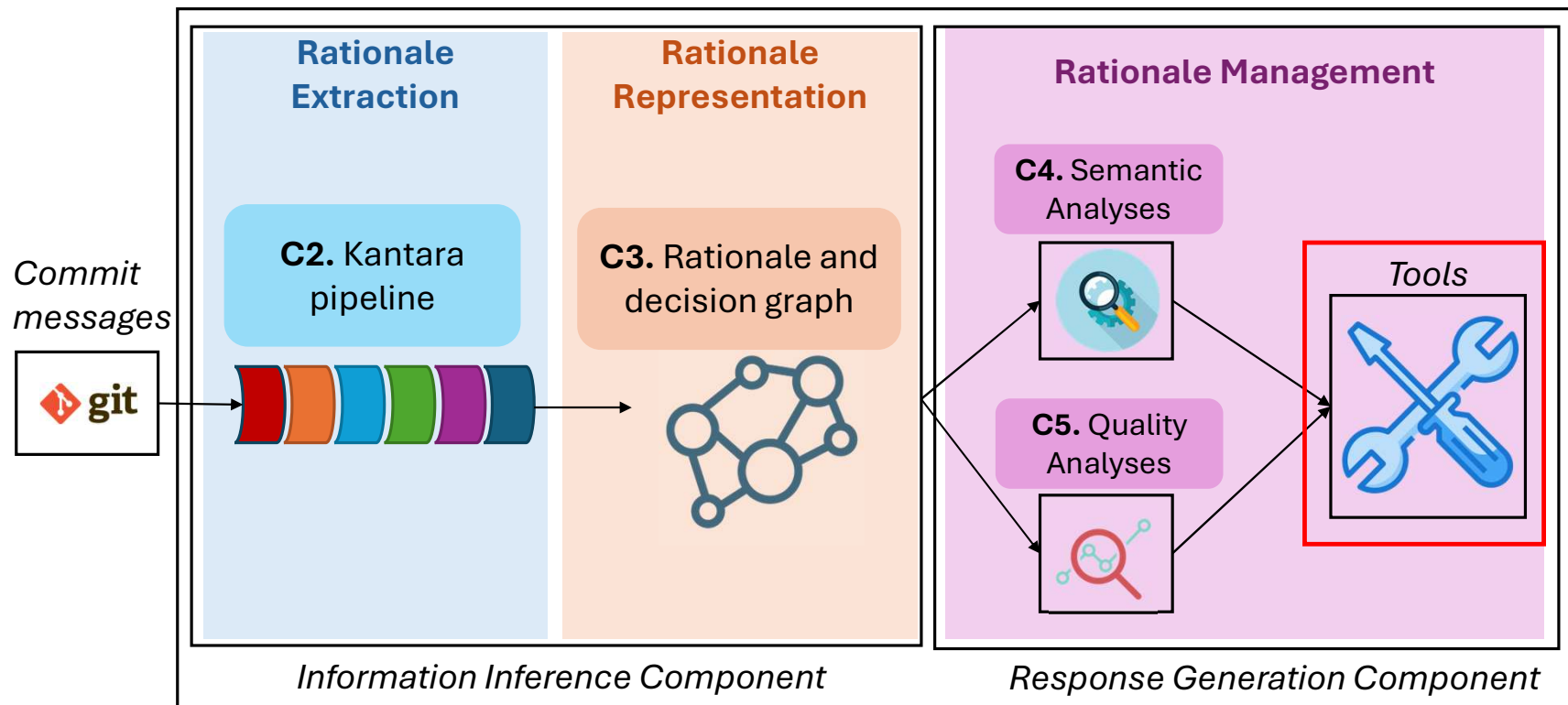
Semantic and quantitative
rationale analyses can help
developers and maintainers

- Better understand the system 
- Reuse past solutions 
- Avoid reasoning conflicts 



81

Mangement – Tools



Tool ideas for Rationale-aware development





Rationale Search Engine

Rationale Search Engine

Select a Module

Springboot

Similarities

Similarities_file: similarities_oss_SPRINGBOOT_with_0.5_threshold_with_URL.csv

Decision1	Rationale1	URL1
0 Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
1 Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
2 Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
3 Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
4 Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
5 Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
6 Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring
7 Fix type "idemp" -> "idemp"	See <pr_link>	https://github.com/spring



Rationale RAG

Rationale RAG

Select a Module

Springboot

Triples_file: ChatGPT_decision_rationale_oss_SPRINGBOOT_with_URL.csv

Enter your commit

Fix checkstyle violation

Similar commits

Enter the cosine similarity threshold:

0.6

4 similar commits

Commit1	Url
0 Fix checkstyle violation see the PR link	https://github.com/sp
1 Fix checkstyle errors fix	https://github.com/sp
2 Fix checkstyle header violation See <issue_link>	https://github.com/sp
3 Fix checkstyle violations nan	https://github.com/sp

Get ChatGPT to double check these simila



Module Analyzer

The mm/slob.c module

Resulting dataset:

% preview of the labelled dataset:

Number of commits: 146
Number of sentences: 833

Distribution

Decision only sentences: 233
Rationale only sentences: 162
Decision & Rationale sentences: 252
No Decision and No Rationale sentences: 186

Word Clouds

Rationale Presence

Total Number of commits: 146
Number of commits that contain rationale: 124
Rationale Percentage: 84.93%
Average Rationale Density: 0.56

Rationale Factors

Commit Message Structure

Rationale Evolution



Message Analyzer

Commit Message Rationale Analyses

Enter your commit message

mm: make function get_attribute() public.

I changed the visibility of function get_attribute().
Now, bug number #333 is fixed.
Debugged by: Mouna

Start Commit Analysis

message_preprocessed	Decision	Rationale
0 mm: make function get_attribute() public.	1	0
0 I changed the visibility of function get_attribute()	1	0
0 Now, bug number #333 is fixed.	0	1

Number of sentences: 3

Your commit message's rationale density: 0.33

Your commit message's decision density: 0.67

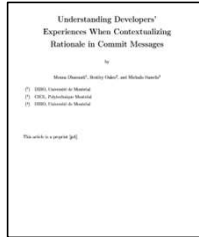
Low Rationale! Please add justification for your changes.

Semantic Analyses

Quality Analyses

83

Survey Results – Rationale Requirements (reminder)



Chapter 7

We derive rationale-aware tool **requirements** from our findings:

Understanding the project :

- (a) identifying project expectations when it comes to maintaining rationale quality
- (b) retrieving the accumulated rationale knowledge from the project history

Analytics-based design approach

Search-based design approach

Reviewing work :

- (a) ensuring the work's conformance with project expectations about rationale quality
- (b) contextualizing the work with respect to existing project knowledge

Automated conformance – based design approach

IR and RAG-based design approach

Automated conformance -based design approach



Message Analyzer

Commit Message Rationale Analyses

Enter your commit message

```

mm: make function get_attribute() public.

I changed the visibility of function get_attribute().
Now, bug number #133 is fixed.
Debugged-by: Mouna
  
```

[Start Commit Analysis](#)

message_unprocessed	Decision	Rationale
0 mm: make function get_attribute() public.	1	0
0 I changed the visibility of function get_attribute()	1	0
0 Now, bug number #133 is fixed.	0	1

Number of sentences: 3

Your commit message's rationale density: 0.33

Your commit message's decision density: 0.67

Low Rationale! Please add justification for your changes.

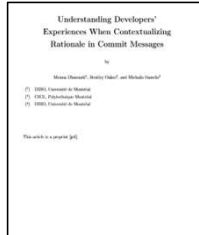
Quality Analyses

More Interviews

Tool ideas to support Rationale-aware development

- 4 pre-recorded videos
- 4 participants (same people from our last interviews)

Q: “What do you think of the design principles behind the tools ?”



Chapter 7



Desired functionalities

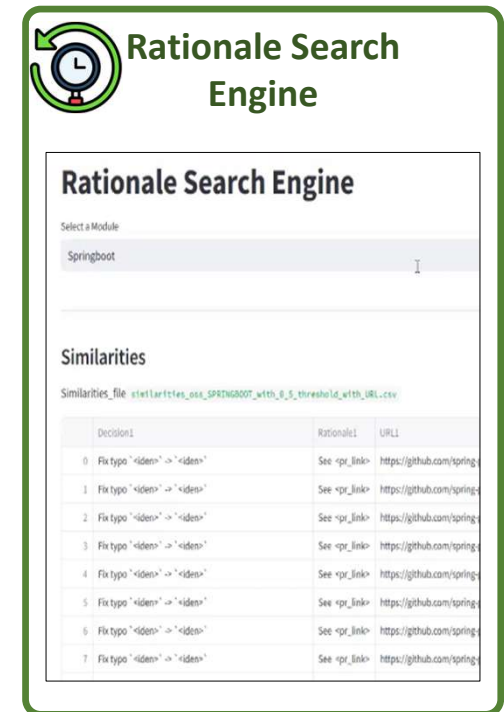
Search-based design approach

Understanding the project :

(a) identifying project expectations when it comes to maintaining the quality (consistency, compliance, sufficiency) of rationale information

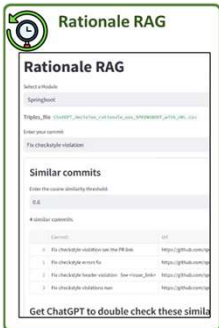
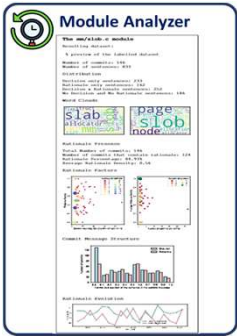
(b) retrieving the accumulated rationale knowledge from the project history

- Include sematic similarity search in the commit history
- Identify repeated changes
- Uncover different approaches and detect oscillations between different solutions
- Flag potentially problematic or malicious changes and inconsistencies
- Provide a user-friendly interface





Desired functionalities

Design approach	IR and RAG-based	Analytics-based	Automated conformance-based
Requirement	Contextualizing the work with respect to existing project knowledge	Identifying project expectations when it comes to maintaining the quality of rationale information	Ensuring the work's conformance with project expectations about rationale quality
Prototype	 Semantic Analyses	 Quality Analyses	 Quality Analyses
Desired functionalities	Automatically surface related and potential contradictory commits	- Help monitor project health. - Provide clear guidelines about current practices for new developers - Identify major shifts in development	- Complement cosmetic checks - Produce reliable output - Be integrated with the development workflow

88

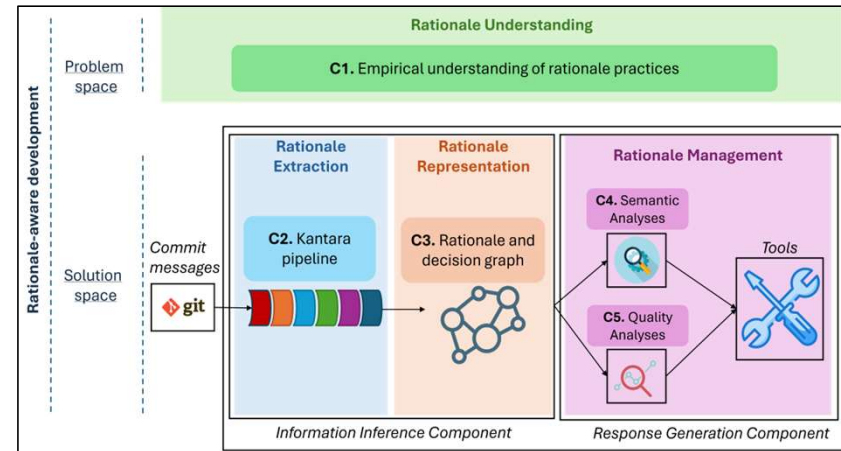
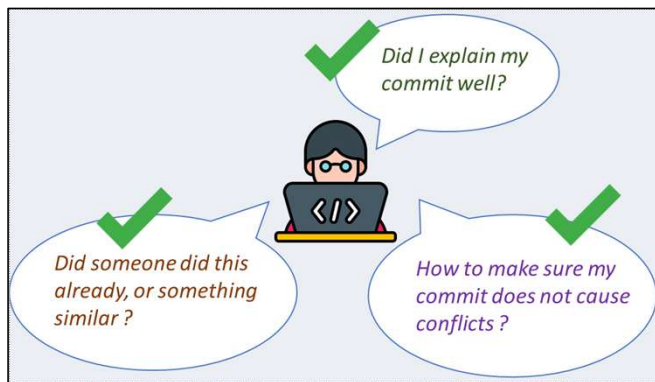
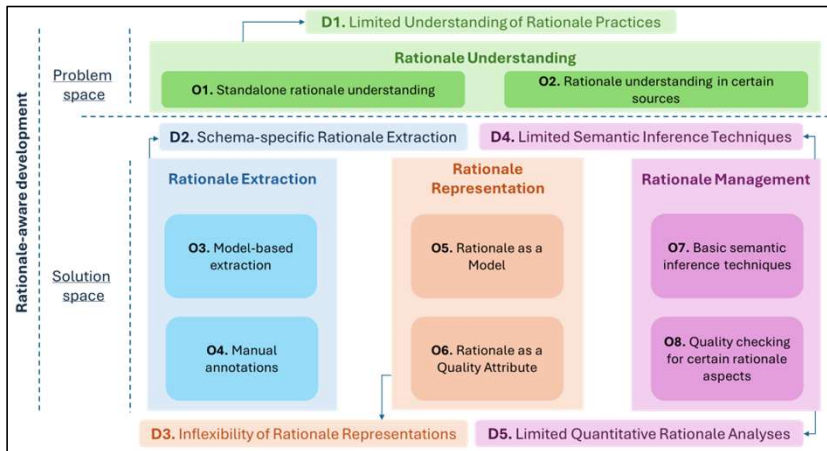
Thesis Statement



Automated **evidence-based quantitative** and **semantic rationale analyses**
can provide **effective support** for
rationale-aware commit-based development.



Conclusion



Research Impact

Solving this problem would have **positive impact**:

- **Project:**
 - Increase longevity
 - Increase quality
- **Developers:**
 - Improve productivity
 - Save time/effort



Limitations - Rationale in Commits

Fix: Added check to prevent null pointer exception.

versus

Changes

Limitations:

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

Future work

- Improve the approach:
 - Better rationale extraction (e.g., support rationale spanning multiple sentences)
- Enrich the graph:
 - More relationships, finer representation => more analyses
- Integration with development platforms (git-based)
- Integrate other data sources (discussions, code reviews, etc..)
- Application on more projects => Repository of Rationale Information
- Personalization / user profile
- Code change and rationale alignment

Bibliography 1/5

- Sutherland, Andrew et al. "**Can peer code reviews be exploited for later information needs?**" International Conference on Software Engineering, 2009.
- LaToza, Thomas D et al. "**Hard-to-answer questions about code.**" Evaluation and usability of Programming Languages and Tools, 2010.
- Robillard, Martin P. "**Turnover-induced knowledge loss in practice**". European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2021.
- Kleebaum, Anja et al. "**How do practitioners manage decision knowledge during continuous software engineering?**" International Conference on Software Engineering and Knowledge Engineering, 2019.
- Noble, Douglas et al. "**Issue-based information systems for design.**" 1988.
- McCall, Raymond J. "**Phi: A conceptual foundation for design hypermedia.**" Design studies, 1991.

Bibliography 2/5

- Burge, Janet E. "**Software engineering using design RATIONale.**" PhD thesis, 2005.
- Dutoit, Allen H et al. "**Rationale management in software engineering: Concepts and techniques.**" Rationale management in software engineering, 2006.
- Jansen, Anton et al. "**Software architecture as a set of architectural design decisions.**" Working IEEE/IFIP Conference on Software Architecture, 2005.
- Alkadhi, Ranaa et al. "**Rationale in development chat messages: an exploratory study, 2017**". MSR, 2017.
- Alkadhi, Ranaa et al. "**How Do Developers Discuss Rationale?**". SANER, 2018.
- Alsafwan, Khadijah et al. "**Decomposing the rationale of code commits: The software developer's perspective**". ESEC/FSE, 2019.

Bibliography 3/5

- Alsafwan, Khadijah et al. **"Developers' need for the rationale of code commits: An in-breadth and in-depth study"**. JSS, 2022.
- Liang, Yan, et al. **"Learning the "Whys": Discovering design rationale using text mining—An algorithm perspective."** Computer-Aided Design, 2012.
- Rogers, Benjamin et al. **"Exploring techniques for rationale extraction from existing documents."** International conference on software engineering, 2012.
- Lester, Miriam et al. **"Using evolutionary algorithms to select text features for mining design rationale."** AI EDAM, 2020.
- McCall, Raymond. **"Using argumentative, semantic grammar for capture of design rationale."** International Conference on Design Computing and Cognition, 2018.
- Alkadhi, Rana et al. **"React: An approach for capturing rationale in chat messages"**. ESEM, 2017.

Bibliography 4/5

- Li, Xueying et al. "**Automatic identification of decisions from the hibernate developer mailing list**". EASE, 2020.
- Zhao, Jiuang et al. "**DRMiner: Extracting latent design rationale from Jira issue logs**". ASE, 2024.
- Sharma, Pankajeshwara, et al. "**Extracting Rationale for Open Source Software Development Decisions — A Study of Python Email Archives**". ICSE, 2021.
- Mahabaleshwar, Manoj. "**Tool support for architectural decision making in large software intensive projects**", Ph.D. thesis, 2020.
- Kleebaum, Anja et al. "**Continuous management of requirement decisions using the ConDec tools.**", REFSQ, 2020.
- Ullah, Tahir et al. "**Exploring and mining rationale information for low-rating software applications**". Soft Computing, 2023.

Bibliography 5/5

- Yue., Gaofeng et al. **“Building a design-rationale-centric knowledge network to realize the internalization of explicit knowledge”**, Applied Sciences, 2023.
- Burge, Janet et al. **“Rationale support for maintenance of large-scale systems”**, ELISA, 2003.
- Kleebaum, Anja et al. **"Continuous Rationale Visualization."**, VISSOFT 2021.
- Tian, Yingchen et al. **"What makes a good commit message?"**, ICSE, 2022.
- Li, Jiawei et al. **"Commit message matters: Investigating impact and evolution of commitmessage quality"**, ICSE, 2023.