CASES FOR EXPLAINABLE SOFTWARE SYSTEMS: CHARACTERISTICS AND EXAMPLES

Mersedeh Sadeghi, Verena Klös and Andreas Vogelsang

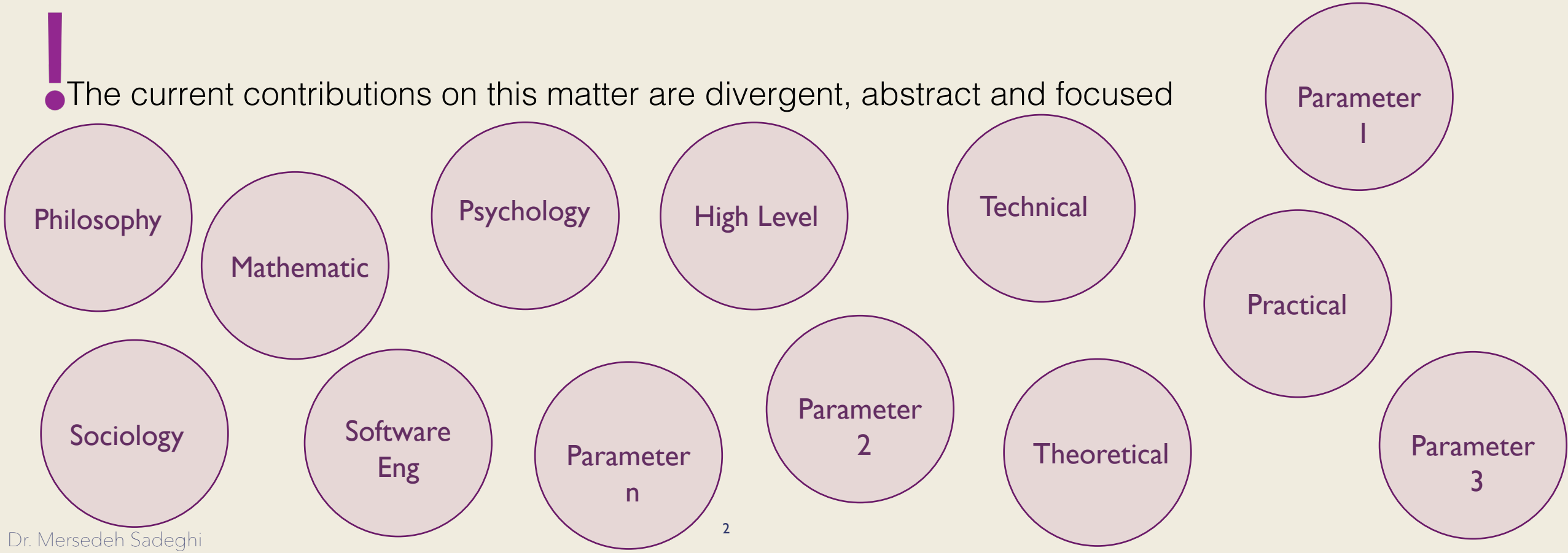
21.09.2021

Research Question

While there is a consensus on what needs explanation . . .

Little attention has been paid towards the characterization of the situations that demand explanations.

! The current contributions on this matter are divergent, abstract and focused

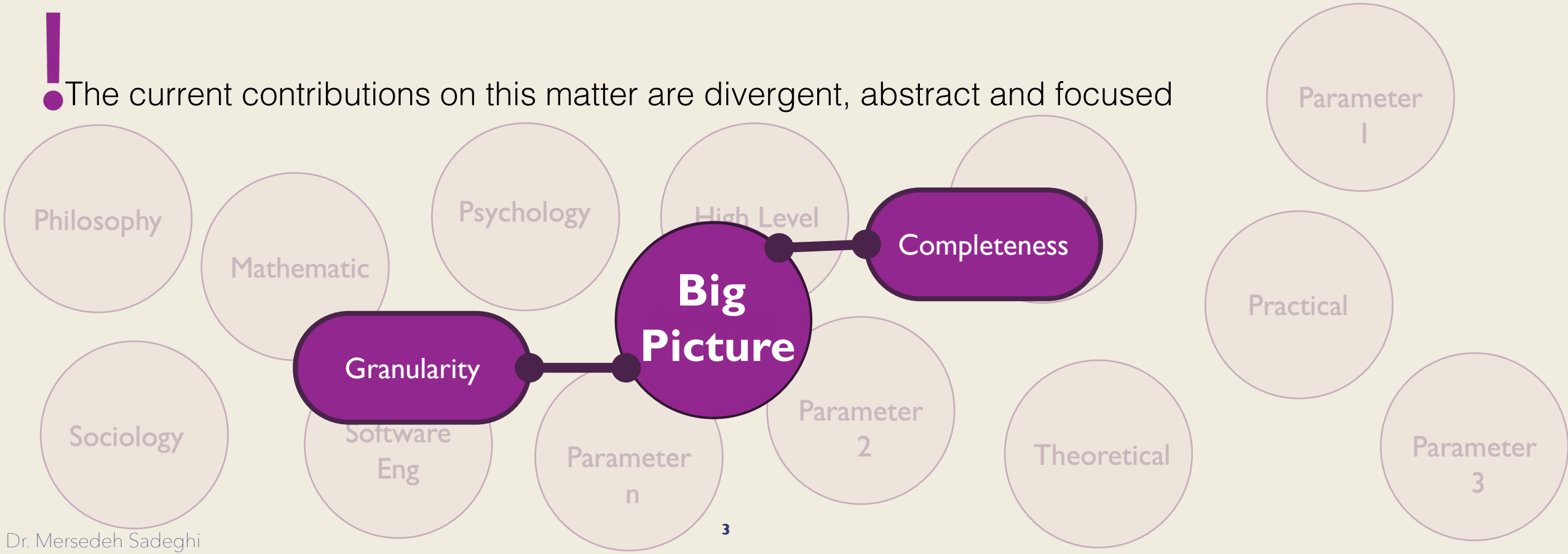


Research Question

While there is a consensus on what needs explanation . . .

Little attention has been paid towards the characterization of the situations that demand explanations.

- The current contributions on this matter are divergent, abstract and focused

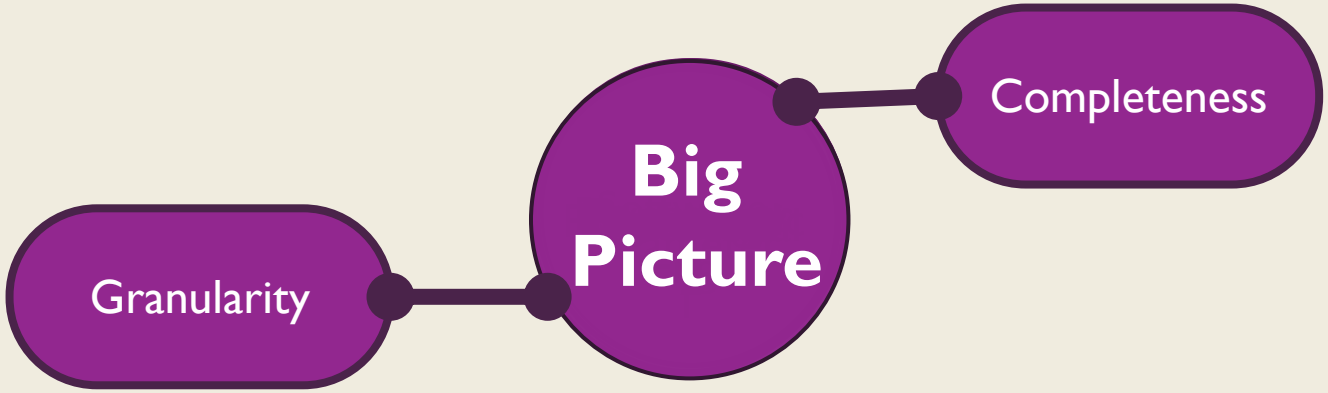


Research Question

While there is a consensus on what needs explanation . . .

Little attention has been paid towards the characterization of the situations that demand explanations.

Situation is combination of various factors and parameters that all together create an state of confusion



Research Question

While there is a consensus on what needs explanation . . .

Little attention has been paid towards the characterization of the situations that demand explanations.

Initial requirement for constructing an explainable system

What are the concrete examples of Explanation Cases?

Motivation

A systematic definition and classification of Explanation cases helps the community in various stages of Software Engineering of Explainable system:

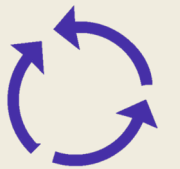
- Requirement Analysis
  - Requirement check list
- Design
  - Software Architecture, Mechanism
- Testing
  - Evaluation Criteria

Methodology

Further analysis, elaboration on and expansion of

- available argumentations and classifications
- use case scenarios from relevant publications

Results



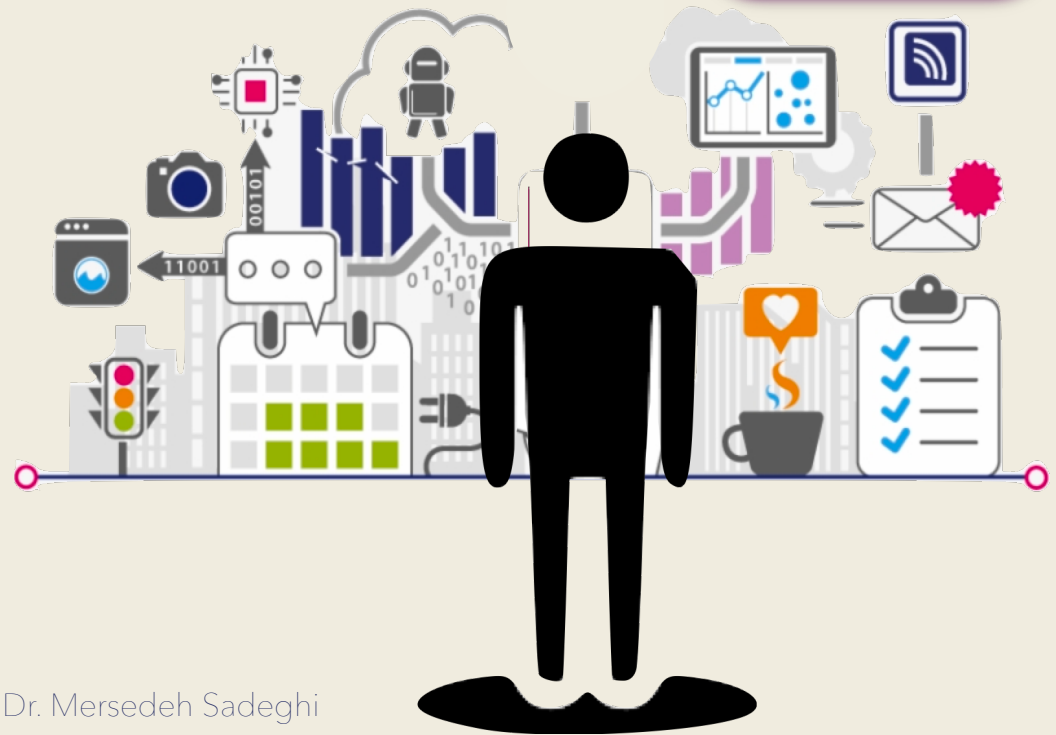
- Training
- Interaction
- Validation
- Debugging
  - Disobedience
    - Conflict
      - Goal-Order Conflict
      - Multi Users Conflict
    - Contextual Condition
    - System Condition
  - Failure
    - System Error
    - User Fault
  - Context Aware Behaviour
    - Suggestion
    - Autonomous Action

Results

Clarify

Convince

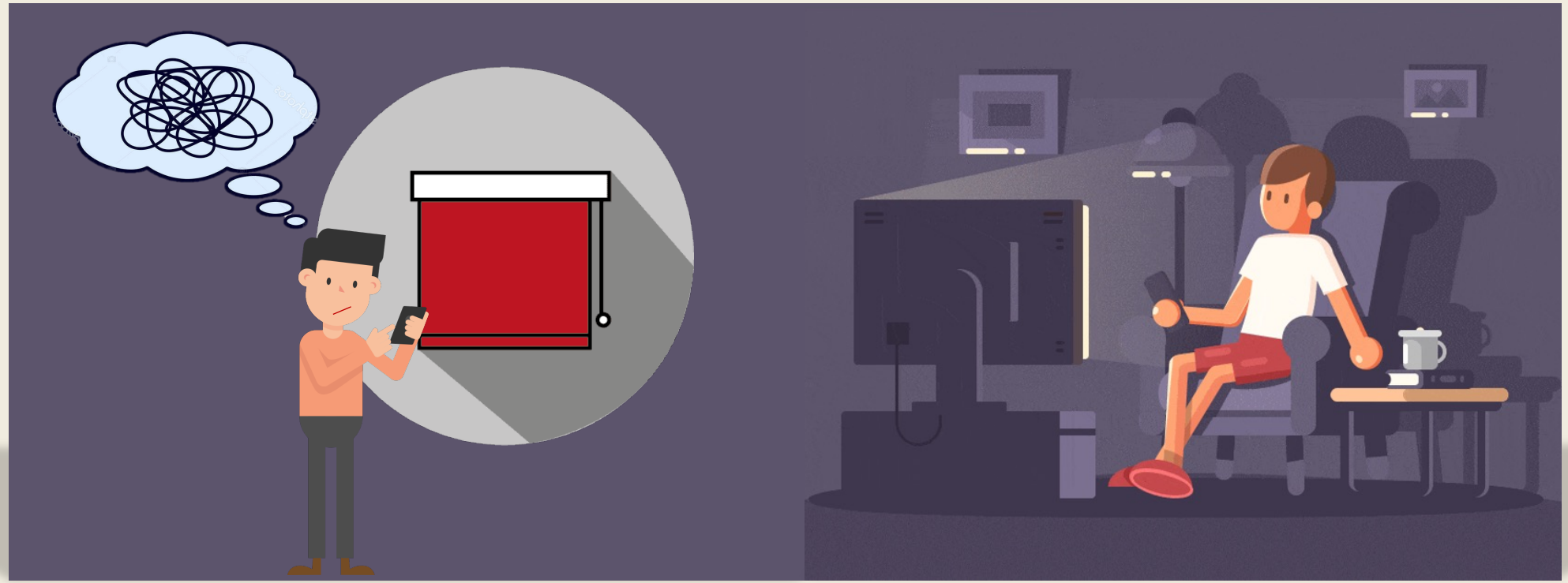
Instruct



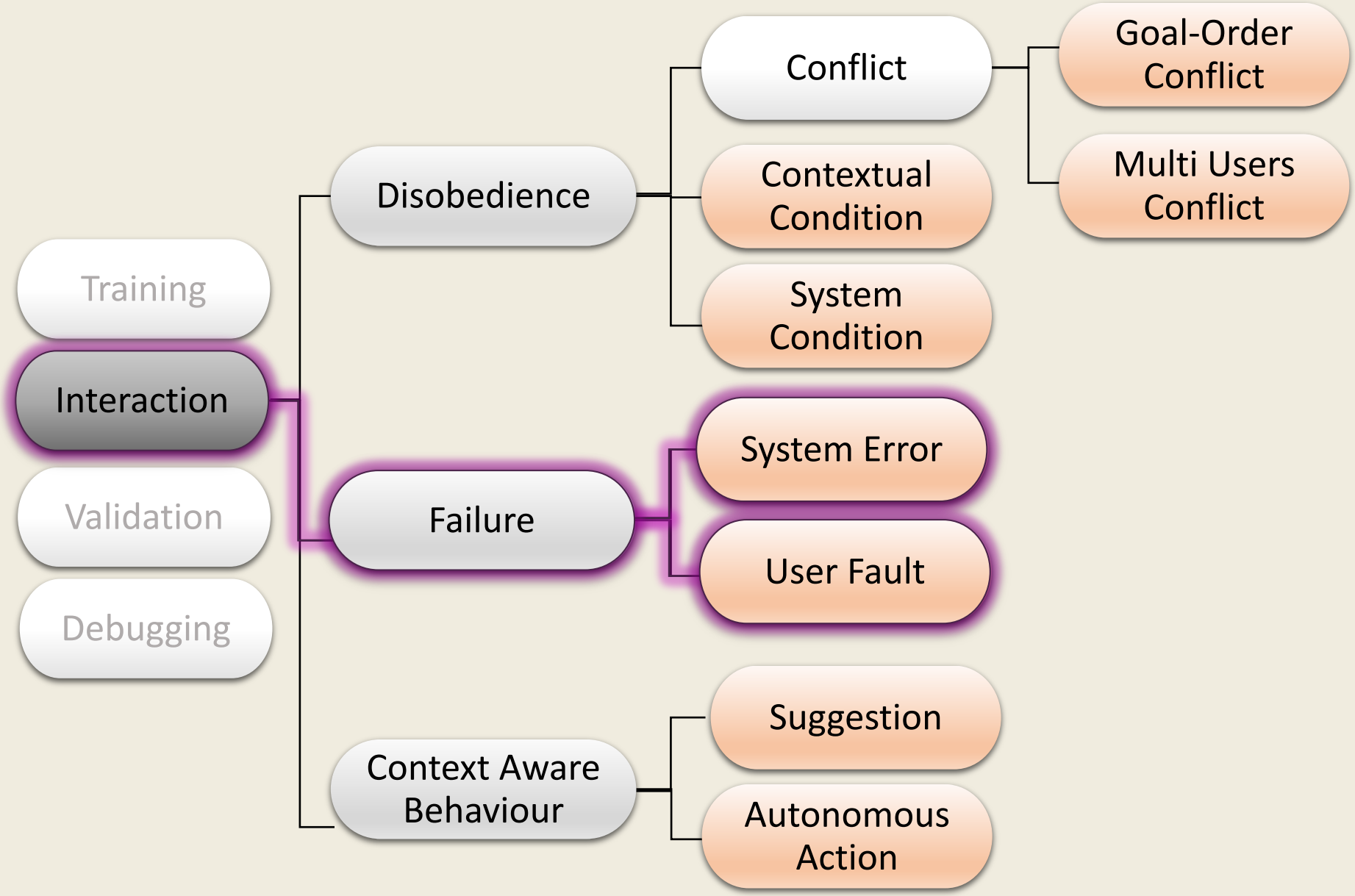
- Interaction
  - Disobedience
    - Conflict
      - Goal-Order Conflict
      - Multi Users Conflict
    - Contextual Condition
    - System Condition
  - Failure
    - System Error
    - User Fault
  - Context Aware Behaviour
    - Suggestion
    - Autonomous Action

Disobedience

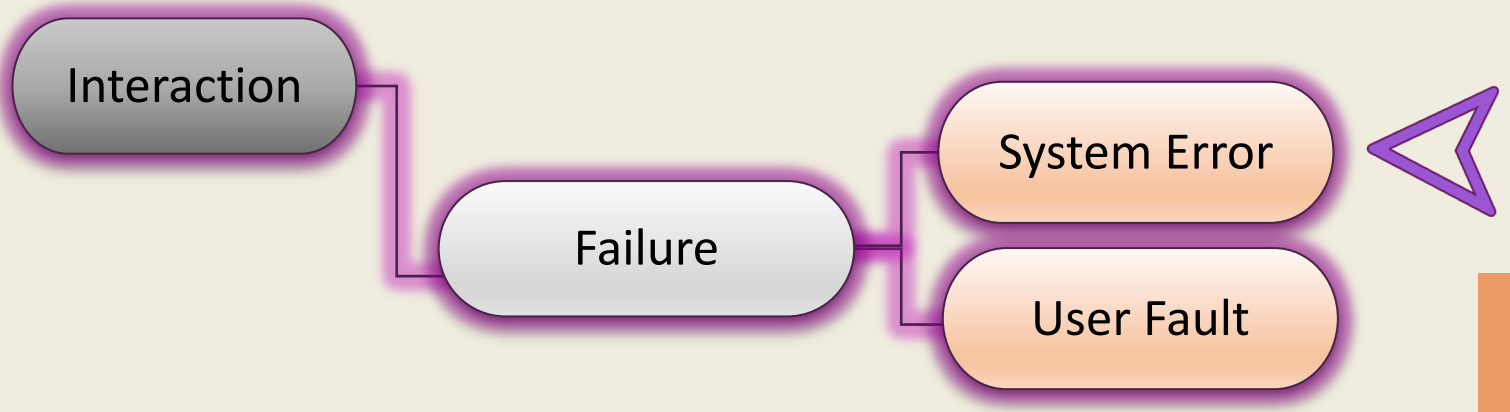
- Interaction
  - Disobedience
    - Conflict
      - Goal-Order Conflict
      - Multi Users Conflict
    - Contextual Condition
    - System Condition

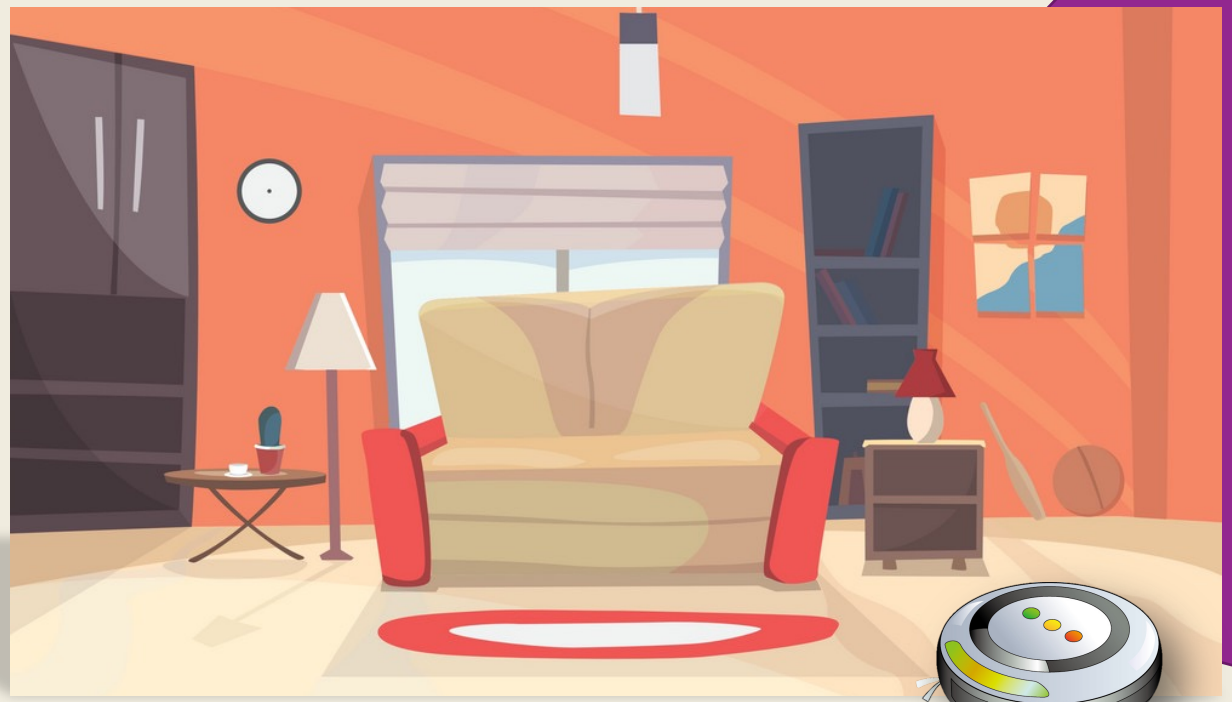


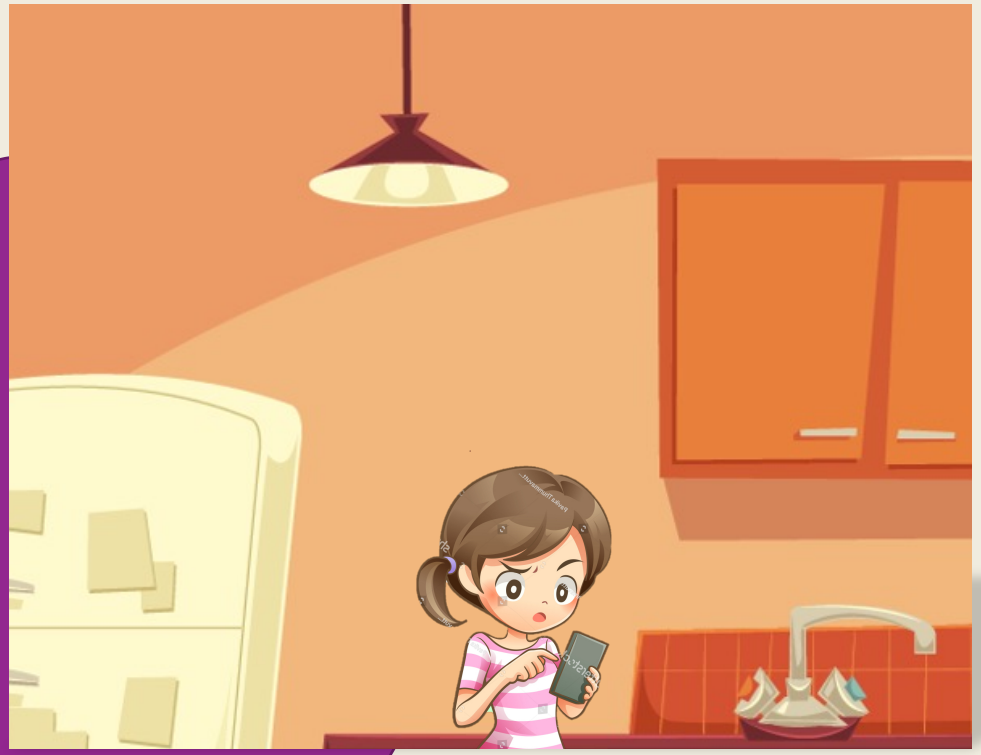
Failure


Training
Interaction
Validation
Debugging
Disobedience
Failure
Context Aware Behaviour
Conflict
Contextual Condition
System Condition
Goal-Order Conflict
Multi Users Conflict
System Error
User Fault
Suggestion
Autonomous Action

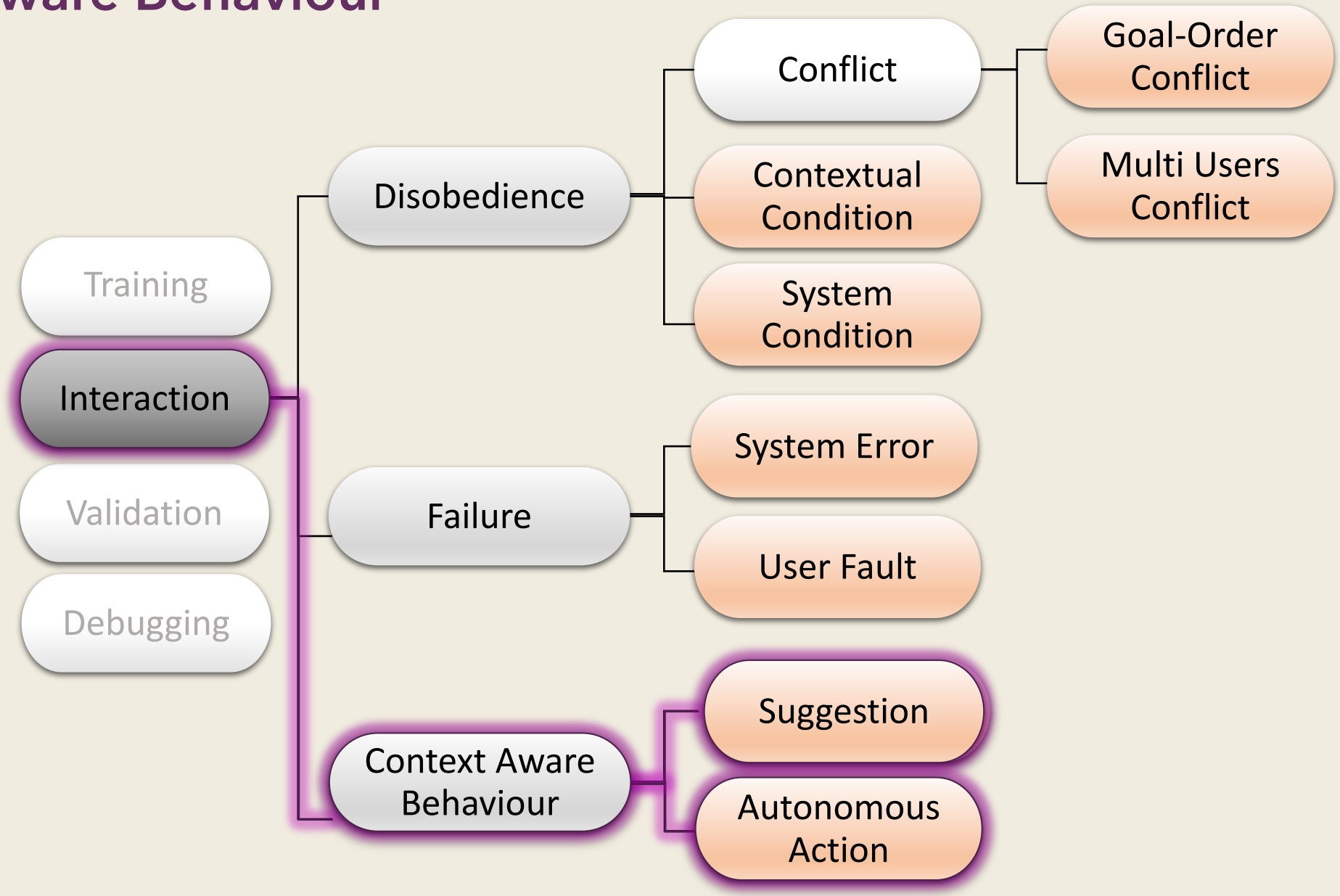
Failure







Context Aware Behaviour



Context Aware Behaviour

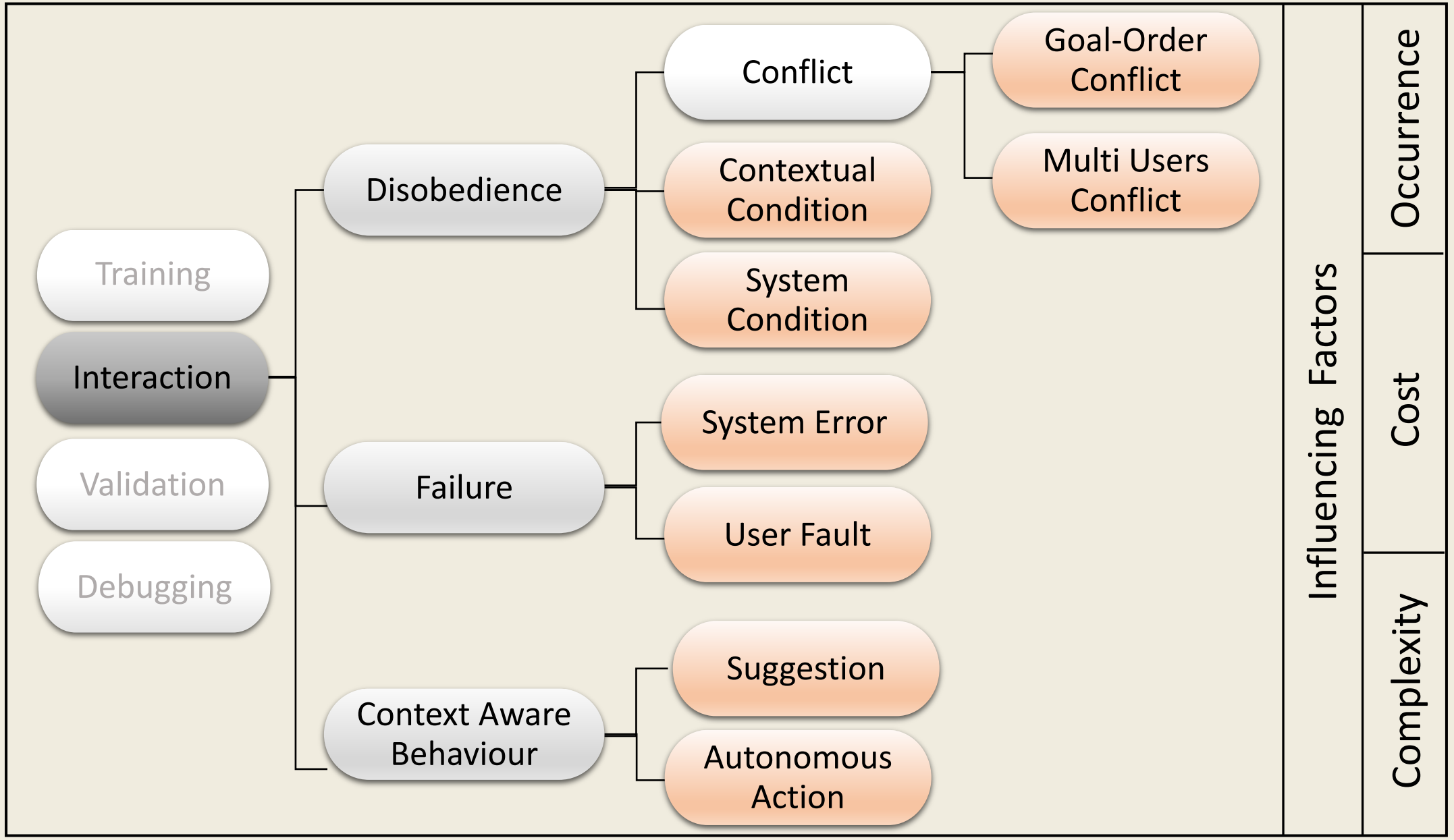
Interaction

Context Aware Behaviour

Suggestion

Autonomous Action

Influencing Factors



Conclusion

We presented our taxonomy of the explanation cases

But, it is a starting point to create Explainable Cases benchmarks.

We hope it will be extended with the help of the community

It helps software engineering of Explainable system

It helps the community aligning future research on explainable systems

https://github.com/mersedehSa/ExplanationCases

Thanks for your attention!