UNIVERSITY OF TORONTO

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Critical Systems Labs
Innovating Safely

AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases

Torin Viger, University of Toronto

Software Certification Consortium, Annapolis MD

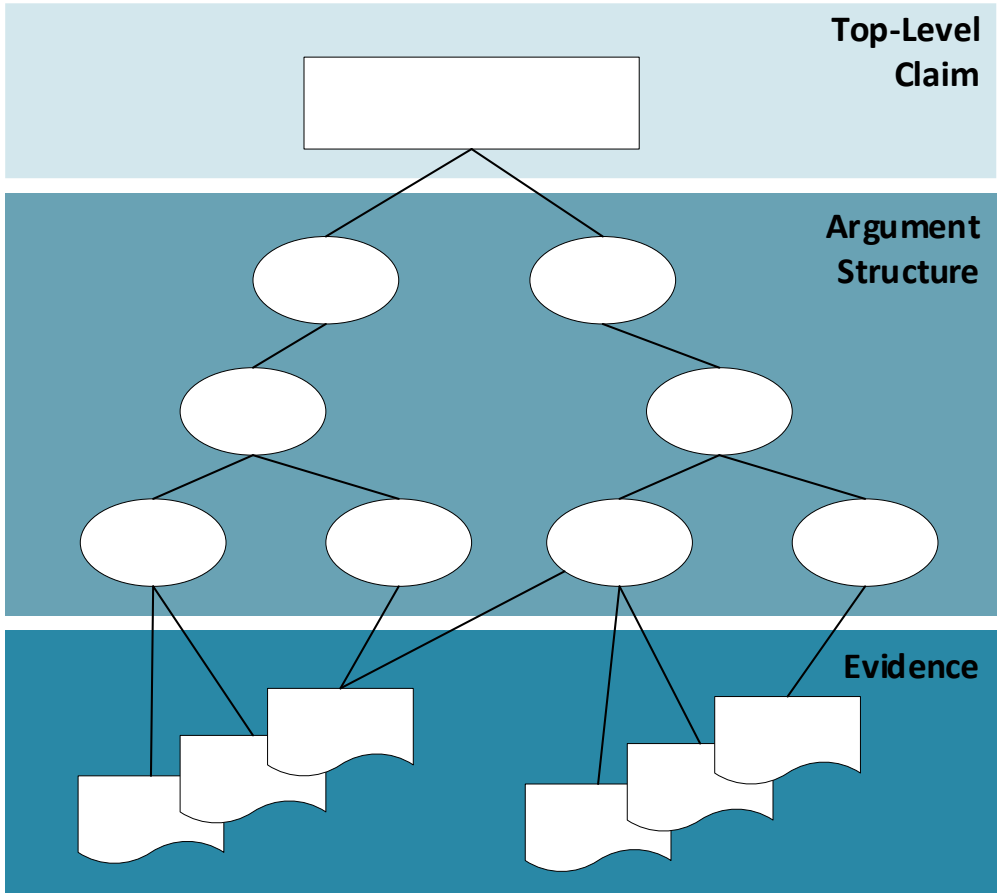
May 15th, 2025

Background: Safety Assurance and Assurance Cases

Top-level goal: The system is acceptably safe

Decompose over hazards, subsystems, environmental conditions etc.

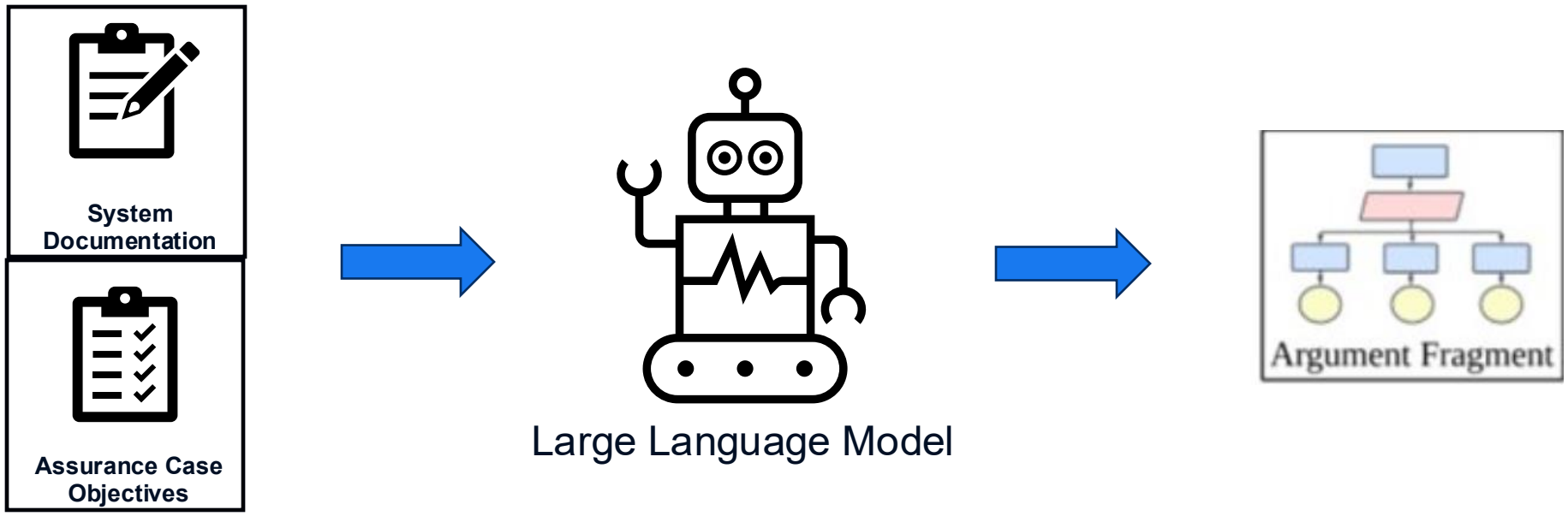
Present evidence that the argument's claims are satisfied



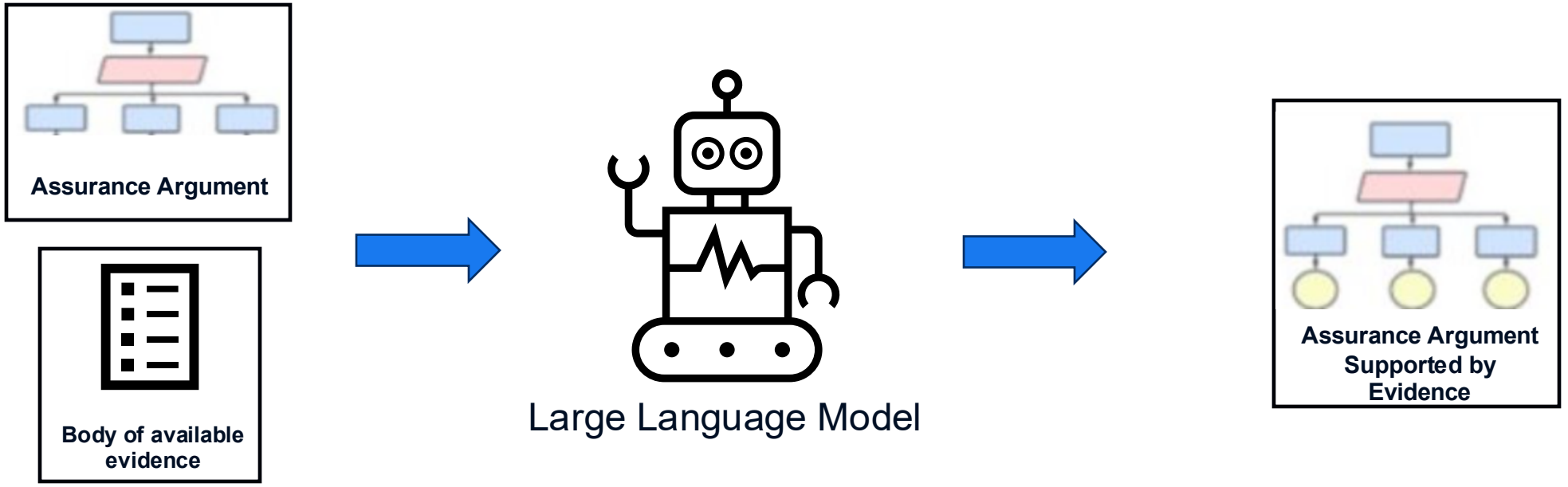
LLMs in Software Engineering and Assurance Case Development

- Large Language Models (LLMs) are increasingly used to support software engineering activities:
  - Writing and Reviewing Code
  - Creating Formal Specifications
- LLMs introduce new opportunities for supporting assurance case development
  - Assurance cases can be large, informal and error-prone
  - LLMs enable inexpensive analysis of informal arguments.

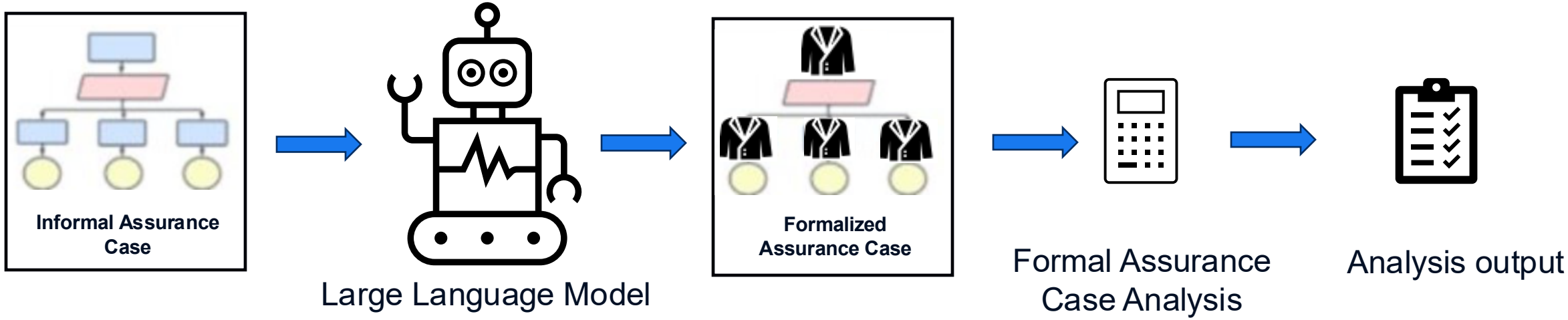
LLM Applications: Argument Creation



LLM Applications: Evidence Suggestion

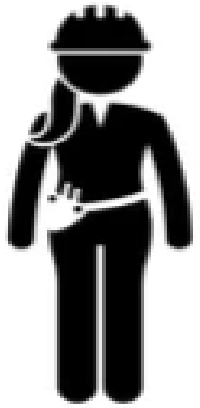


LLM Applications: Enabling Formal Analysis

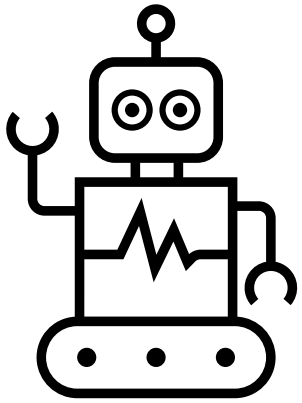


Risks of Large Language Models in Safety Assurance

- Hallucinations and non-determinism limit LLM use in safety-critical applications



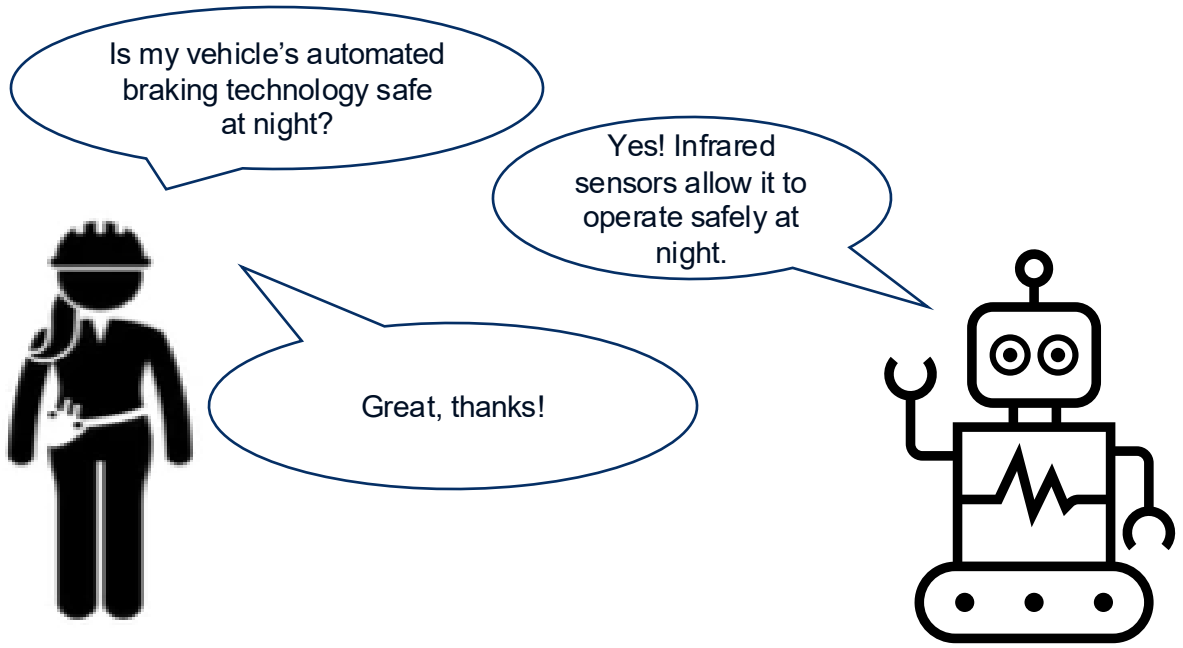
Safety Engineer



Large Language Model

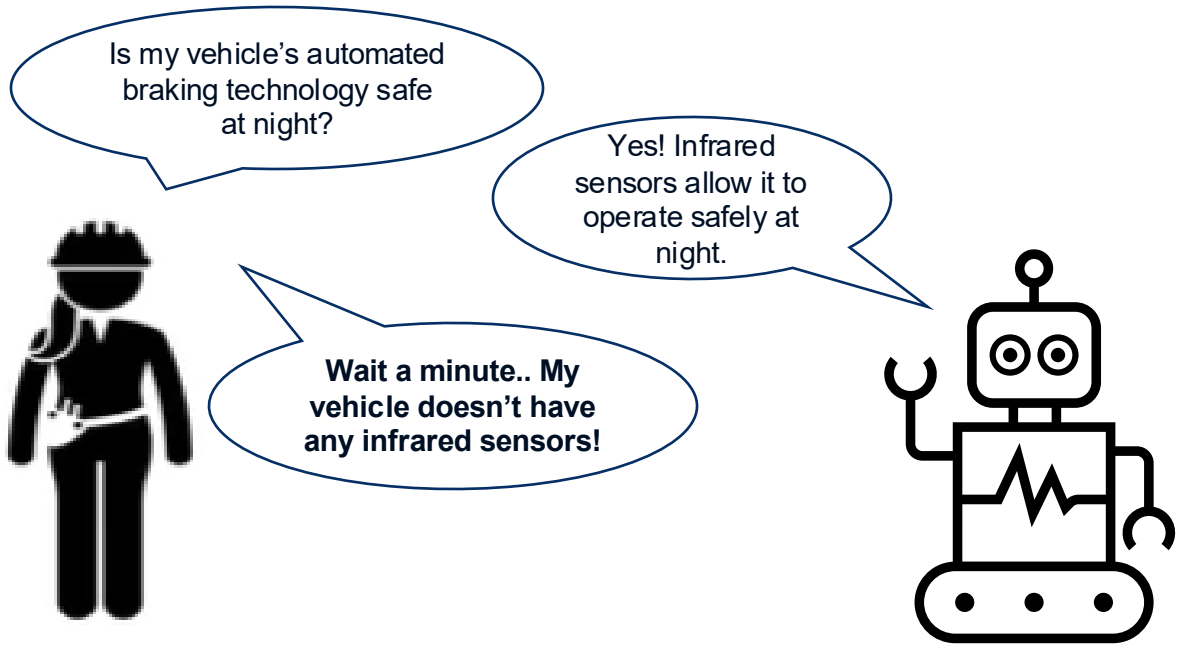
Motivation: Large Language Models in Safety Assurance

- Hallucinations and non-determinism limit LLM use in safety-critical applications



Motivation: Large Language Models in Safety Assurance

- Hallucinations and non-determinism limit LLM use in safety-critical applications



Our Work on LLM Assurance

- AI-Supported Eliminative Argumentation (AI-EA): LLMs for supporting **defeater identification** in ACs

[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

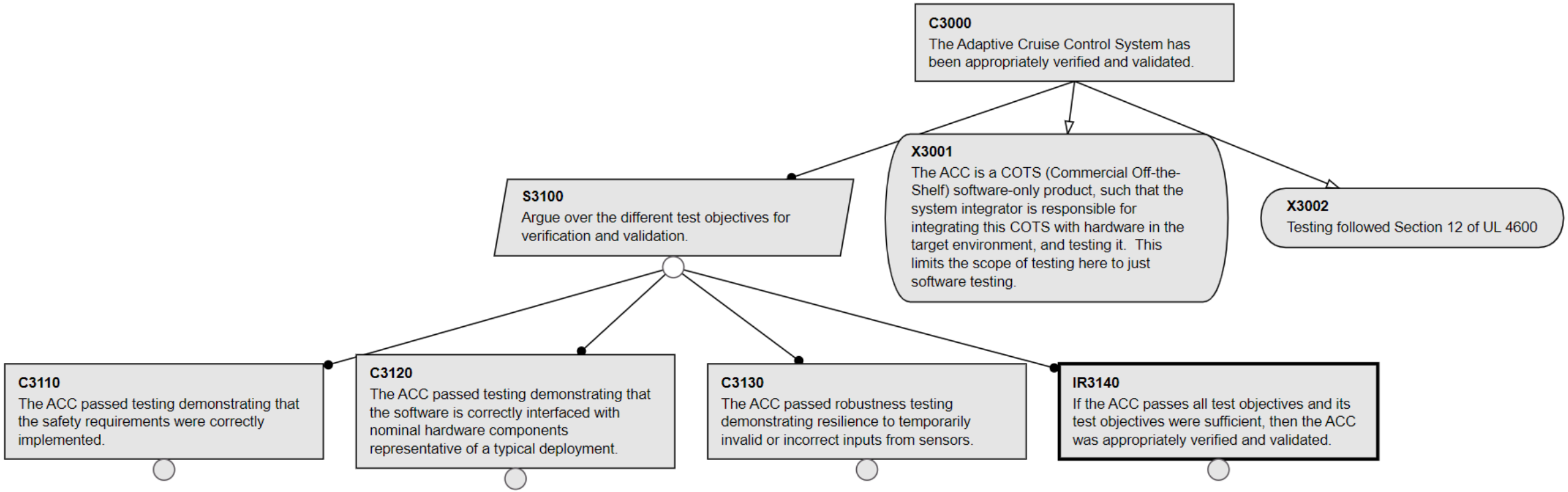
- LLMs for supporting **change impact assessment**

[4] Viger, T., Murphy, L., Diemert, S., Menghi, C., Chechik, M.: Supporting Change Impact Assessment with LLMs. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 203–204. IEEE (2024)

- Bow-Tie framework for documenting and analyzing the risks and benefits of LLM use in assurance tasks.

[5] S. Diemert, E. Cyffka, N. Anwari, O. Foster, T. Viger, and L. Millet. Balancing the Risks and Benefits of Using Large Language Models to Support Assurance Case Development. *To be published at SafeComp 2025.*

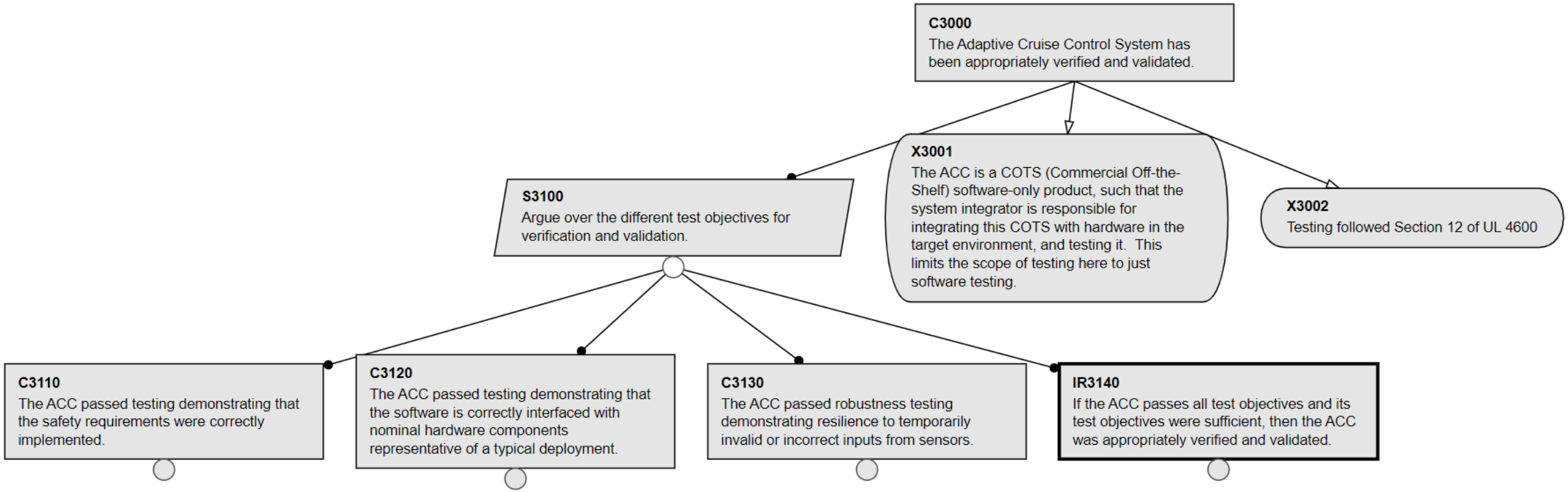
Running Example: Adaptive Cruise Control System



Eliminative Argumentation and Defeaters

- Engineers naturally have doubts about the systems they design. Assurance case methods should acknowledge this doubt rather than try to hide it
- Eliminative Argumentation (EA) incorporates the notion of doubt in assurance cases using explicit nodes called defeaters.
  - Confidence is increased by mitigating these defeaters
  - Unmitigated defeaters are left as *residual risks*

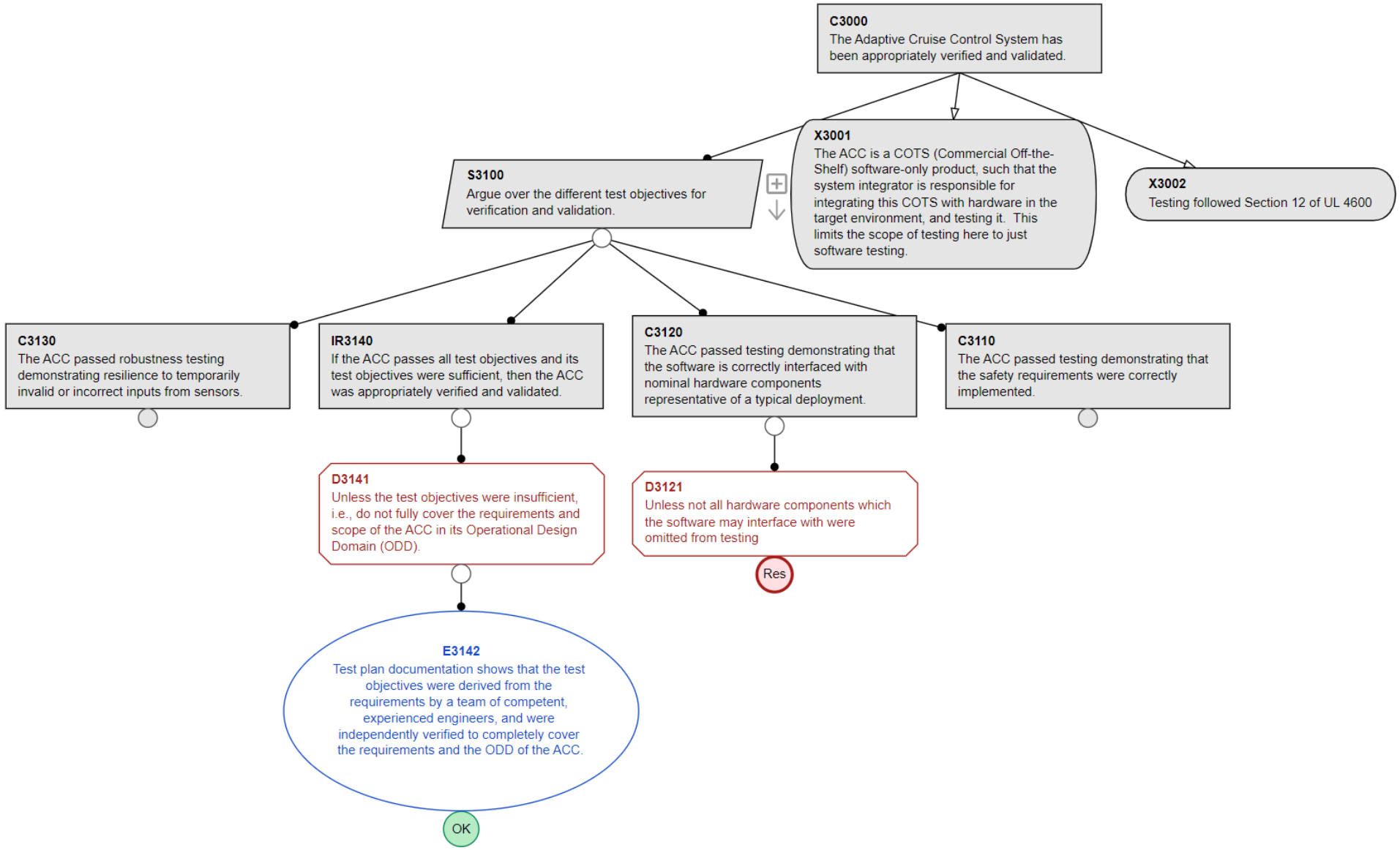
Assurance Case Example (EA): Adaptive Cruise Control System



What if an important hardware component was omitted from testing?

What if the test objectives are not sufficient?

Assurance Cases with Defeaters



Why Do We Need Support For Defeater Identification?

- Engineers may overlook defeaters due to blind spots or confirmation bias
- Manual effort is required to identify defeaters
  - Defeater generation is often done by experienced system engineers

LLM Support for Defeater Identification

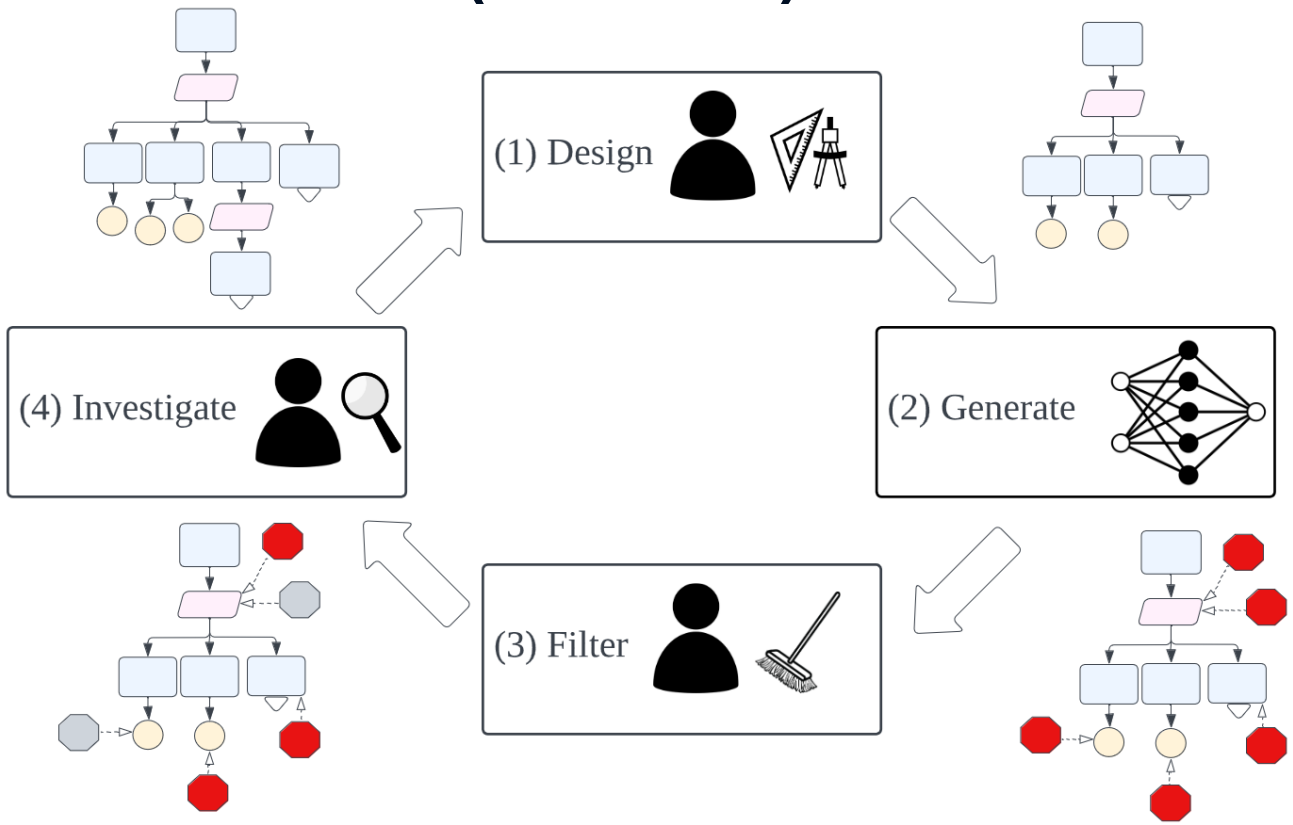
Benefits

- LLMs may identify defeaters omitted by engineers, e.g., due to blind spots, confirmation bias or lack of time/resources/knowledge
- Extremely fast and lightweight
- EA defeaters serve as questions that are reviewed by engineers, not as *conclusions* about the argument

Risks and their Mitigation

- LLM hallucinations create false **doubt** rather than false confidence
  - Leads to over-cautiousness, not safety concerns
- We propose AI-EA to **support**, not replace manual defeater generation

Our Framework: AI-Enabled Supported Argumentation (AI-EA)



[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

[2] Viger, T., Murphy, L., Diemert, S., Menghi, C., Di, A., Chechik, M.: Supporting Assurance Case Development Using Generative AI. In: SAFECOMP 2023, Position Paper (2023)

Challenge: What makes a defeater useful?

- **Key question:** How can we systematically evaluate whether LLM-generated defeaters are useful?
  - How can we evaluate whether ***any*** defeaters are useful.

Defeater 1: There might be an unconsidered factor somewhere in the ACC assurance case that undermines one of its claims

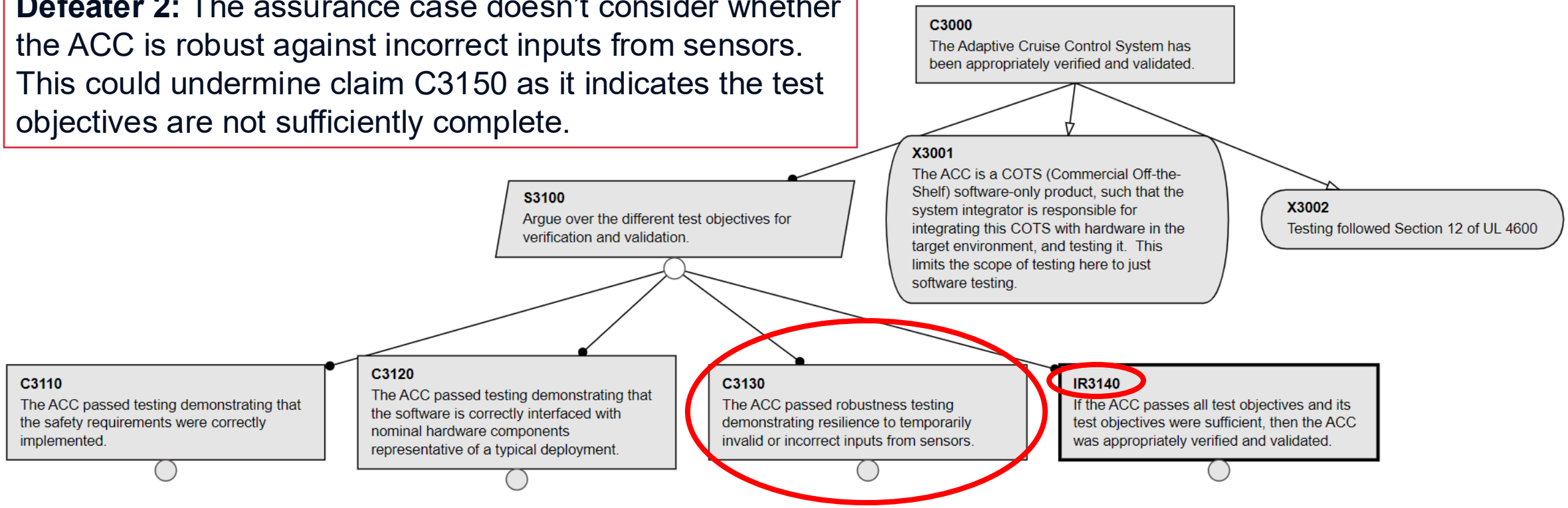
Defeater 2: The assurance case doesn't consider whether the ACC is robust against incorrect inputs from sensors. This could undermine claim 3150 as it indicates the test objectives are not sufficiently complete.

[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

[2] Viger, T., Murphy, L., Diemert, S., Menghi, C., Di, A., Chechik, M.: Supporting Assurance Case Development Using Generative AI. In: SAFECOMP 2023, Position Paper (2023)

Informativeness: Necessary but not Sufficient

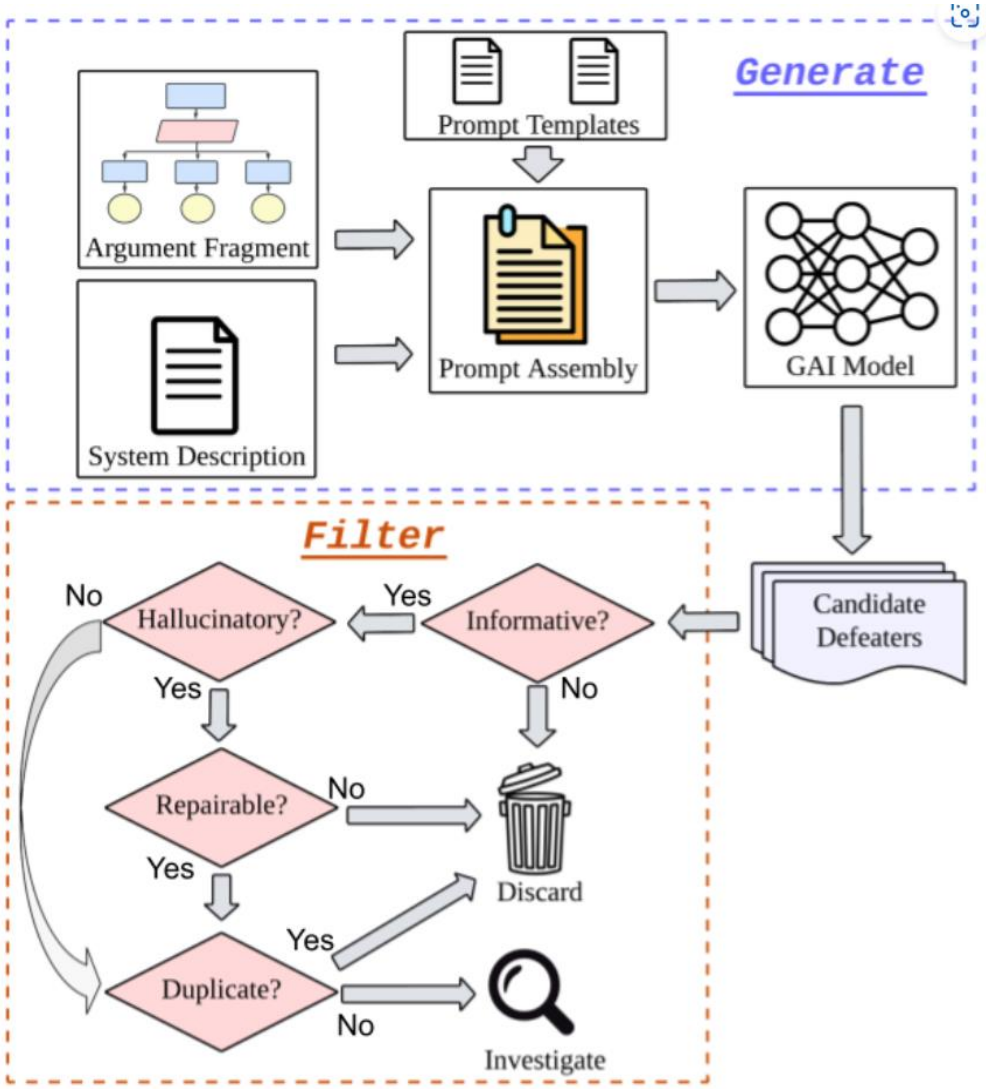
Defeater 2: The assurance case doesn't consider whether the ACC is robust against incorrect inputs from sensors. This could undermine claim C3150 as it indicates the test objectives are not sufficiently complete.



[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

[2] Viger, T., Murphy, L., Diemert, S., Menghi, C., Di, A., Chechik, M.: Supporting Assurance Case Development Using Generative AI. In: SAFECOMP 2023, Position Paper (2023)

AI-EA Workflow



[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

[2] Viger, T., Murphy, L., Diemert, S., Menghi, C., Di, A., Chechik, M.: Supporting Assurance Case Development Using Generative AI. In: SAFECOMP 2023, Position Paper (2023)

Evaluation of AI-EA

Two methods of empirical evaluation:

- Assessment of 171 defeaters by three reviewers
- User study in which eight practitioners used AI-EA to support Assurance Case development

- Full details of our evaluation and its results provided in our related work

[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

[3] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M., Anwari, N., Cyffka, E.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. Currently under Journal Review

Key Results

- Practitioners found that generated defeaters were clear, concise and easy to review, ~32% of generated defeaters were added to their ACs
  - Especially useful as a sanity check after a first pass
- ~52% of generated defeaters provided concrete information in all components (what, where, why)
- Only ~5% of generated defeaters contained unrepairable hallucinations (e.g., fully irrelevant/incoherent)
- During the user study, unique defeaters were identified both by humans(~23%) and by the LLM (~30%)
  - Highlights complimentary nature of LLM support

[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

[3] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M., Anwari, N., Cyffka, E.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. Currently under Journal Review

Try it out!

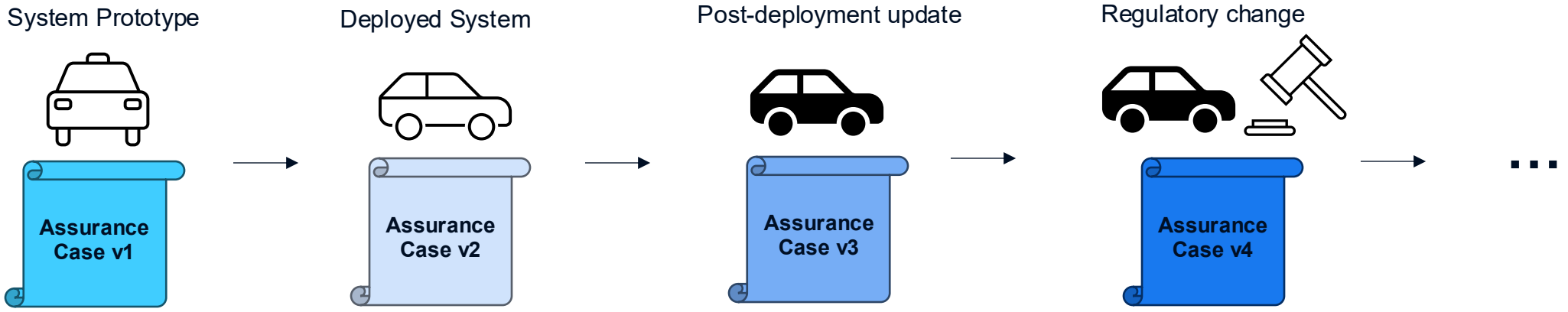
- It's essential to have reviewable and reproducible data for LLM evaluation. Our open-source artifact includes:
  - All 171 generated defeaters and our assessment of them
  - Software package for re-generating these results
  - Supports prompt modifications, can be used on new assurance cases
    - https://zenodo.org/records/13368055
- LLM Defeater generation has since been implemented in the assurance case development tool *Socrates*, developed by Critical Systems Labs, Inc. (https://criticalsystemslabs.com/)

[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

[3] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M., Anwari, N., Cyffka, E.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. Currently under Journal Review

Assurance Case Evolution

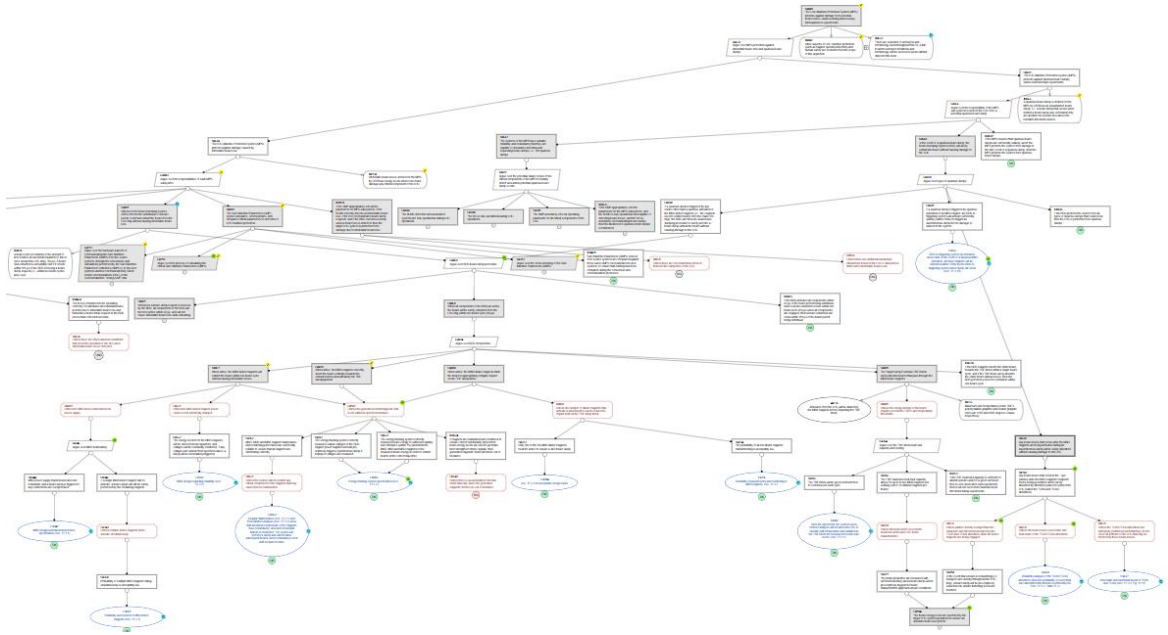
- Many critical systems evolve after they are deployed (e.g., over the air updates)
- As systems evolve, their assurance cases need to evolve too
  - ACs also evolve during development, when operational or regulatory environment changes etc.



[4] Viger, T., Murphy, L., Diemert, S., Menghi, C., Chechik, M.: Supporting Change Impact Assessment with LLMs. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 203–204. IEEE (2024)

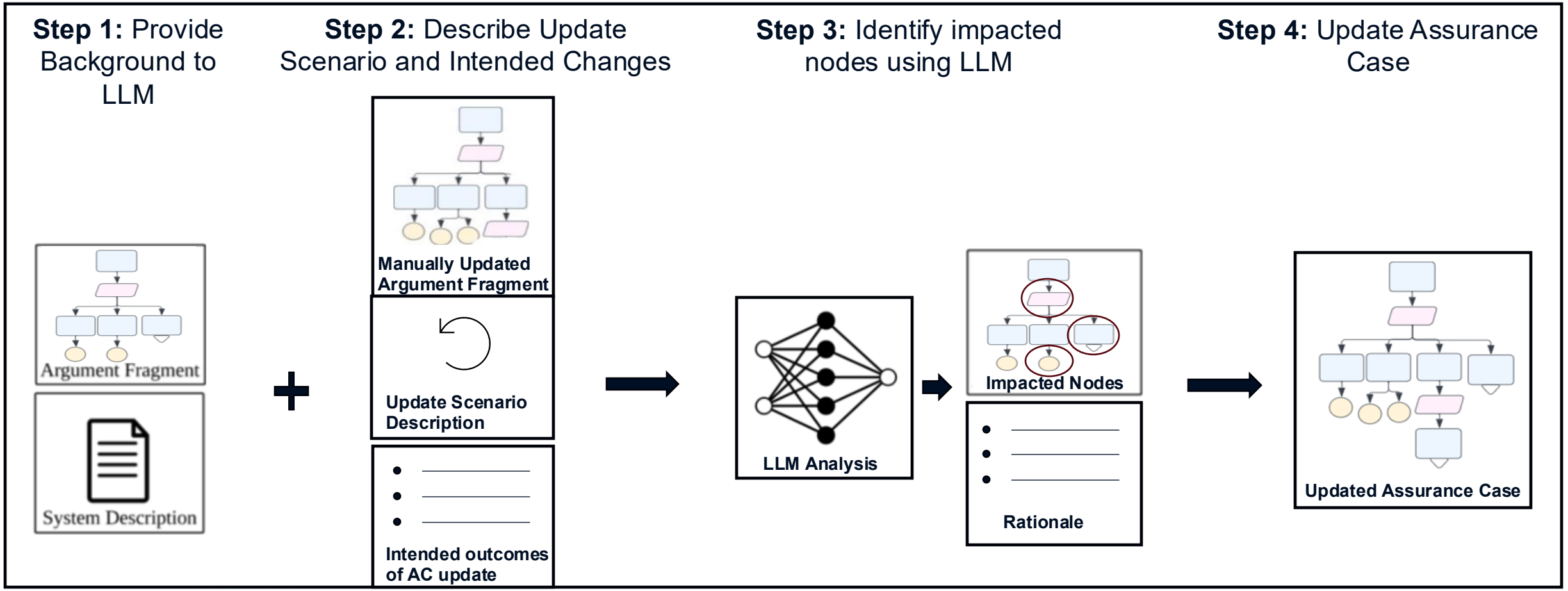
Challenges with Change Impact Assessment

- Assurance cases are often expressed in natural language
- Can contain thousands of interconnected nodes
- “Local” changes to a system may not be local in the assurance case
- Change impact assessment can be expensive and error-prone



[4] Viger, T., Murphy, L., Diemert, S., Menghi, C., Chechik, M.: Supporting Change Impact Assessment with LLMs. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 203–204. IEEE (2024)

Our Proposal: Supporting Change Impact with Large Language Models (LLMs)



[4] Viger, T., Murphy, L., Diemert, S., Menghi, C., Chechik, M.: Supporting Change Impact Assessment with LLMs. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 203–204. IEEE (2024)

Why is Change Impact Assessment well-suited to LLM support?

Benefits:

- Fast, lightweight support without requiring formalization / maintenance of traceability links
  - Complimentary to other approaches
- Applicable to a wide range of update scenarios
- Leverages semantics of assurance case nodes to inform impact assessment

Risks

- Recommending irrelevant nodes
- Omitting critical impacted nodes

[4] Viger, T., Murphy, L., Diemert, S., Menghi, C., Chechik, M.: Supporting Change Impact Assessment with LLMs. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 203–204. IEEE (2024)

Evaluation

Preliminary evaluation:

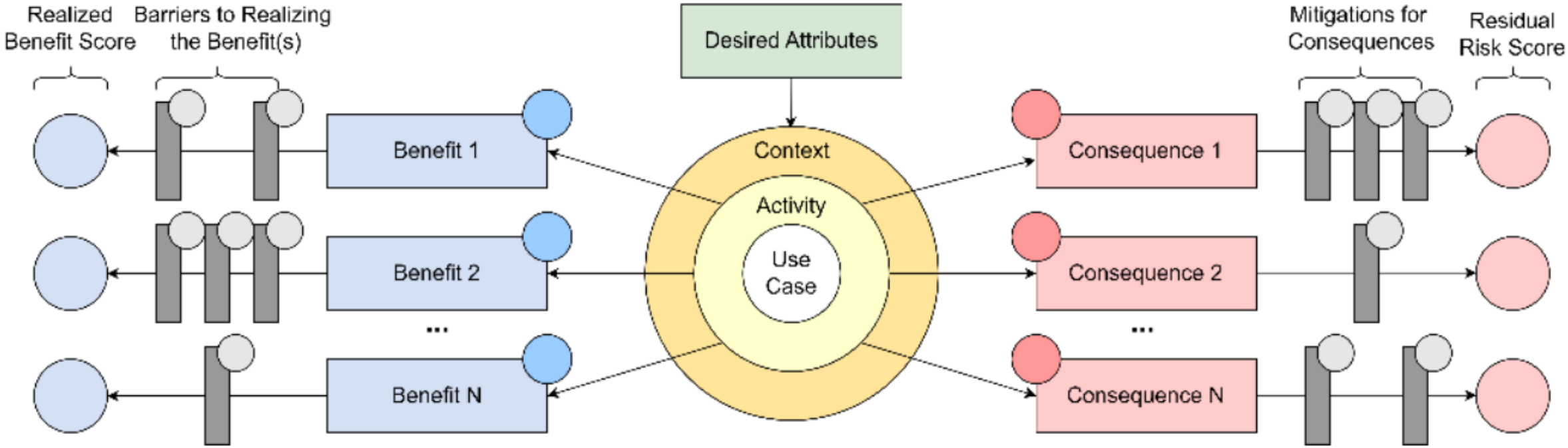
- LLMs are effective at identifying impacted nodes in highly constrained update scenarios (e.g., replacing all instances of one AC pattern with another, typographical changes)

Ongoing evaluation

- Evaluation over a wider range of update scenarios and assurance cases
  - What update scenarios can LLMs effectively support?
  - What data can be used to improve their effectiveness?

[4] Viger, T., Murphy, L., Diemert, S., Menghi, C., Chechik, M.: Supporting Change Impact Assessment with LLMs. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 203–204. IEEE (2024)

Managing LLM Risks and Benefits: The Bow-Tie Framework



[5] S. Diemert, E. Cyffka, N. Anwari, O. Foster, T. Viger, and L. Millet. Balancing the Risks and Benefits of Using Large Language Models to Support Assurance Case Development. *To be published at SafeComp 2025.*

Summary and Future Work

- Managing risks and benefits of LLM applications is essential
- Defeater generation and Change Impact Assessment represent two promising LLM applications where risks can be minimized by *supporting*, not *replacing*, existing methods
- We present AI-EA, a framework for generating and evaluating LLM defeaters
    - Empirical evaluation shows it is effective in supporting practitioners
- To support deeper analysis of future LLM applications, we introduce the bow-tie framework for

Ongoing and Future Work:

- Extended evaluation of LLM Change Impact Assessment
- Improved defeater generation with advanced prompting and defeater customization
- Application of the bow-tie framework to additional use-cases, e.g., supporting formalization of assurance cases.

Collaborators

Logan Murphy, University of Toronto
Alessio Di Sandro, University of Toronto
Aren Babikian, University of Toronto
Marsha Chechik, University of Toronto

Claudio Menghi, University of Bergamo

Simon Diemert, Critical Systems Labs, Inc.
Jeff Joyce, Critical Systems Labs, Inc.
Laure Millet, Critical Systems Labs, Inc.
Olivia Foster, Critical Systems Labs, Inc.
Erin Cyffka, Critical Systems Labs, Inc.
Naweed Anwari, Critical Systems Labs, Inc.

Sahar Kokaly, General Motors
Ramesh S, General Motors

Thank You!

[1] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE). pp. 284–294. IEEE (2024)

[2] Viger, T., Murphy, L., Diemert, S., Menghi, C., Di, A., Chechik, M.: Supporting Assurance Case Development Using Generative AI. In: SAFECOMP 2023, Position Paper (2023)

[3] Viger, T., Murphy, L., Diemert, S., Menghi, C., Joyce, J., Di Sandro, A., Chechik, M., Anwari, N., Cyffka, E.: AI-Supported Eliminative Argumentation: Practical Experience Generating Defeaters to Increase Confidence in Assurance Cases. Currently under Journal Review

[4] Viger, T., Murphy, L., Diemert, S., Menghi, C., Chechik, M.: Supporting Change Impact Assessment with LLMs. In: 2024 IEEE 35th International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 203–204. IEEE (2024)

[5] S. Diemert, E. Cyffka, N. Anwari, O. Foster, T. Viger, and L. Millet. Balancing the Risks and Benefits of Using Large Language Models to Support Assurance Case Development. *To be published at SafeComp 2025.*