Enhancing Requirements Traceability and Runtime Monitoring through the Integration of FRET, WEST, and R2U2

Songyan Lai

Abstract

This thesis investigates the potential integration of three formal verification tools—FRET, WEST, and R2U2—to address gaps in requirements traceability, validation, and runtime monitoring in safety-critical systems. The study explores the capabilities of each tool, their interoperability, and the feasibility of a unified framework. The work is grounded in real-world applications, focusing on industries like aerospace. Challenges such as data transformation, scalability, and tool interfacing are identified, alongside recommendations for practical and industry-aligned solutions.

Introduction

Background:
Formal verification ensures that system requirements are correct, complete, and consistently implemented throughout the software lifecycle. Tools like FRET, WEST, and R2U2 play pivotal roles in different phases of this process:

- **FRET** translates natural language requirements into formal specifications.
- **WEST** validates Mission-Time Linear Temporal Logic (MLTL) formulas through regular expressions.
- **R2U2** monitors runtime system behavior to ensure compliance with predefined requirements.

Research Problem:
Despite their individual strengths, these tools operate in isolation, which can result in inefficiencies and traceability gaps. This research investigates whether integrating FRET, WEST, and R2U2 can enhance the software verification lifecycle, from design to deployment.

Research Questions:

1. Can FRET and WEST be seamlessly integrated, and what benefits would this offer?
2. Is it feasible to extend WEST's pre-deployment validation to R2U2's runtime monitoring environment?
3. What challenges arise in developing a unified framework, and how can they be overcome?

Literature Review

FRET and Requirements Traceability:
Developed by NASA and enhanced by Maynooth University's MU-FRET, FRET formalizes natural language requirements into temporal logic (Rozier, 2020). It improves stakeholder collaboration and reduces ambiguity, making it a cornerstone for integrating other tools.

WEST and MLTL Validation:
WEST validates MLTL formulas using regular expressions (Elwing et al., 2023). It ensures robust pre-deployment verification but requires consistent syntax and data transformation when paired with runtime tools.

R2U2 and Runtime Monitoring:
Designed for embedded systems, R2U2 supports LTL, MTL, and MLTL (Stark et al., 2018). Its real-time anomaly detection and probabilistic reasoning make it ideal for dynamic systems but necessitate alignment with FRET's and WEST's specifications for integration.

Comparative Studies and Integration Potential:
Previous studies highlight the modularity of these tools and their potential for integration, particularly in aerospace applications (Valu3s, 2024).

Methodology

1. **Tool Analysis:**
    - Conducted a comparative analysis of FRET, WEST, and R2U2 based on supported logics, validation approaches, and usability.
    - Reviewed practical applications in safety-critical systems.
2. **Integration Feasibility:**
    - Developed a prototype translator for converting FRET-generated specifications into WEST-compatible formats.
    - Explored the extension of WEST's outputs for runtime use in R2U2.
3. **Industry Feedback:**
    - Engaged with professionals in aerospace to evaluate practical challenges and expectations.

Findings

1. **Integration Feasibility:**
    - FRET's natural language translation simplifies requirement formalization, which WEST can validate pre-deployment.
    - WEST-validated formulas align with R2U2's runtime monitoring specifications, enabling a consistent verification process across the lifecycle.
2. **Challenges:**

   - Data format discrepancies between WEST and R2U2 require a robust transformation mechanism.
   - R2U2's scalability might limit its integration with more complex FRET-generated specifications.
3. **Industry Needs:**
   - Industry stakeholders prioritize tools that reduce manual effort, support real-time diagnostics, and align with regulatory standards.

Recommendations

1. **Short-Term Solutions:**
   - Develop middleware for translating outputs between tools. This is also what I am doing now.
   - Focus on domain-specific integrations (e.g., aerospace).
2. **Long-Term Vision:**
   - Create a unified framework integrating FRET, WEST, and R2U2.
   - Advocate for standardization in temporal logic syntax.
3. **User-Centric Design:**
   - Enhance tool interfaces to improve usability.
   - Provide comprehensive training and documentation.

Conclusion

Integrating FRET, WEST, and R2U2 can significantly enhance the software verification lifecycle, offering improved traceability, validation, and runtime monitoring. While challenges remain, a phased approach focusing on middleware development and industry collaboration provides a practical path forward. Future research should explore advanced AI-driven solutions and broader standardization efforts.

References

1. Rozier, K. Y. (2020). Applications of Temporal Logic in System Verification.
2. Elwing, J., et al. (2023). *Mission-Time LTL Validation via Regular Expressions*.
3. Stark, J., et al. (2018). Runtime Verification with R2U2.
4. Maynooth University (2024). *MU-FRET Enhancements for Safety-Critical Systems*.
5. Valu3s Project (2024). *Integration of Formal Verification Tools*.