

NFM 2023 notification for paper 4279

NFM 2023 <nfm2023@easychair.org>

Tue 3/7/2023 6:10 AM

To: Ádám Zsófia <adamzsofi@edu.bme.hu>

Dear Zsófia Ádám:

We are delighted to inform you that your submission 4279: From Natural Language Requirements to the Verification of Programmable Logic Controllers: Integrating FRET into PLCverif has been accepted for presentation at NASA Formal Methods 2023. Congratulations!

There were 75 submissions to NFM this year, of which the program committee accepted 29 for presentation at the conference.

The final reviews for your paper are below. Please take the reviews into careful consideration as you prepare the final camera-ready version of your paper, and make sure to follow up on all the revisions requested by the referees. The camera-ready paper is due on March 27, 2023.

Congratulations again, and we look forward to seeing you at the University of Houston, Clear Lake in May! Registration information will soon be available on the conference webpage.

Swarat Chaudhuri, Kristin Yvonne Rozier, and James Dabney
NFM 2023 Program Chairs

SUBMISSION: 4279

TITLE: From Natural Language Requirements to the Verification of Programmable Logic Controllers: Integrating FRET into PLCverif

----- REVIEW 1 -----

SUBMISSION: 4279

TITLE: From Natural Language Requirements to the Verification of Programmable Logic Controllers: Integrating FRET into PLCverif

AUTHORS: Zsófia Ádám, Ignacio D. Lopez-Miguel, Anastasia Mavridou, Thomas Pressburger, Marcin Beś, Enrique Blanco Viñuela, Andreas Katis, Jean-Charles Tournier, Khanh V. Trinh and Borja Fernández Adiego

----- Overall evaluation -----

SCORE: 1 (weak accept)

----- TEXT:

The paper wants to mitigate the sentence constraints when specifying the requirements to the control systems. Specifically, the original methods European Organization for Nuclear Research (CERN) used to control magnetic systems is to pre-define many sentence templates so that the users are constrained in this limited corpus, or just directly utilize formal specifications. Here the authors want to mitigate this problem via directly mapping natural sentence into Linear Temporal Logic (LTL). They implemented the package FRET developed by NASA. In this package, the input sentences still own some constraints, but the corpus is much broader than the original case. Meanwhile, FRET can also verify the truth values of each component in LTL to check the semantic correctness.

The authors test their framework in two cases: safety PLC program and standard PLC program. The results turn to be great. In both two cases, the expression will be quite tedious without FRET.

Overall, this work is systematic and shows great performance. However, the paper can be improved if they can add the explanation of the methods FRET used to transform sentence into LTL, and the limitation of FRET. In other words, in what condition FRET will fail to translate sentence into LTL? Meanwhile, apart from FRET, is there any other methods can be used for this task too? If so, what is the advantage of FRET compared to others.

----- Correctness -----

SCORE: 4 (good)

----- Reproducibility -----

SCORE: 4 (good)

----- Eligible for Original Software Publication track in Science of Computer Programming -----

--

SELECTION: no

----- REVIEW 2 -----

SUBMISSION: 4279

TITLE: From Natural Language Requirements to the Verification of Programmable Logic Controllers: Integrating FRET into PLCVerif

AUTHORS: Zsófia Ádám, Ignacio D. Lopez-Miguel, Anastasia Mavridou, Thomas Pressburger, Marcin Beś, Enrique Blanco Viñuela, Andreas Katis, Jean-Charles Tournier, Khanh V. Trinh and Borja Fernández Adiego

----- Overall evaluation -----

SCORE: 1 (weak accept)

----- TEXT:

The authors present the integration NASA's FRET, a tool for supporting users in writing formal requirements, into Formal Verification of Programmable Logic Controller tool (PLCVerif).

The goal is twofold: 1. help the users to validate the formal requirements with higher confidence, and 2. improve the expressiveness.

The capabilities of the integration are described via two use cases involving CERN critical systems. The first one (Safety PCL Program of SM18 Cluster F) for showing validation capabilities, and the second one (OnOff Object) for showing the improved expressiveness.

Overall evaluation:

The paper is well written, the goals are clear as well as the support provided by the integration to the users.

The scientific novelty is limited, and the core of the work is the extension of FRET to support additional requirement patterns, and the architectural integration with PLCVerif.

The industrial applications of the integration show the quality and the usefulness of the proposed solution.

I have two main concerns, one about the content and one more about the form:

1. [Improved expressiveness]. It is not clear to me how FRET would help the user in creating new patterns. For example, in the second use case the authors claim that without FRET, formulating properties involving "next occurrence" would have required extra effort and LTL/CTL experts for the creation of new patterns. My understanding is that, in order to handle "next occurrence", also FRET needed to be extended with the extra operators $\text{nextOcc}(p, q)$ and $\text{prevOcc}(p, q)$, thus requiring extra effort and LTL/CTL experts.

A direct comparison with the previous approach (PCLVerif only) would greatly help the advantage of the proposed solution.

2. [Help Requirements Validation] Sometimes the authors state that the interactive simulator of FRET *ensures* that the automatically generated logical formulae correctly capture the intended semantics. It might be beneficial to relax this statement, since the simulator can only increase the confidence (not ensure) that the generated property is the desired one.

Minor Comments:

- Page 2, line 6

"... which are a set of pre-made templates with "blanks" where expressions containing variables of the PLC program can be added (e.g., "{1} is always true at the end of the PLC cycle")"

The meaning is clear, but the example in parenthesis might be confusing

- Page 6, Fig. 5 (a) should be expanded a bit to avoid overlap
- Appendix Pag. 8 Fig 6 "after laucnhing" -> "after launching"

----- Correctness -----

SCORE: 3 (fair)

----- Reproducibility -----

SCORE: 4 (good)

----- Eligible for Original Software Publication track in Science of Computer Programming -----

--

SELECTION: yes

----- REVIEW 3 -----

SUBMISSION: 4279

TITLE: From Natural Language Requirements to the Verification of Programmable Logic Controllers: Integrating FRET into PLCverif

AUTHORS: Zsófia Ádám, Ignacio D. Lopez-Miguel, Anastasia Mavridou, Thomas Pressburger, Marcin Beś, Enrique Blanco Viñuela, Andreas Katis, Jean-Charles Tournier, Khanh V. Trinh and Borja Fernández Adiego

----- Overall evaluation -----

SCORE: 1 (weak accept)

----- TEXT:

This is a short case study paper describing the experience of integrating a tool for requirements specification using a natural language (FRET) with a verifier for PLC programs (PLCverif). In PLCverif, the authors noticed several issues with the template-based approach to writing requirements: limited expressiveness due to the usage of templates and missing validation of requirements. Hence, to alleviate these problems, they integrate the FRET tool, which supports natural-language-based requirements specification, into PLCverif. The required extending FRET with a few additional constructs/predicates to support key PLC-related properties. Two case studies were performed with the integrated tools. The case studies, show, for example, how some key properties would require the additional of new templates to the "old" PLCverif, while that is not needed with FRET, thereby showing its usability.

Pros:

- Requirements specification that is approachable and easy to use by customers is one of the key problems to overcome if a verifier is to be used by non-expert users. Hence, extending a verifier to make requirements specification easier is a valuable contribution.
- The paper outlines the drawbacks of the current approach, as well as how FRET assists in alleviating them.
- Case studies appear to be real-world PLC programs, although I am not totally sure about that (are they just examples maybe?)

Cons:

- The biggest drawback of the paper in my opinion is that it is unclear if the case studies were done with end users, or also be experts. To me it seems that case studies were performed by the authors themselves, who already had the chosen properties in mind. Hence, the case studies are somewhat artificial since they do not show the benefits of using FRET in real-world settings (with non-expert users).
- The case studies are also pretty limited since they include the specification of only two properties.

----- Correctness -----

SCORE: 4 (good)

----- Reproducibility -----

SCORE: 4 (good)

----- Eligible for Original Software Publication track in Science of Computer Programming -----

--

SELECTION: no

----- REVIEW 4 -----

SUBMISSION: 4279

TITLE: From Natural Language Requirements to the Verification of Programmable Logic Controllers: Integrating FRET into PLCverif

AUTHORS: Zsófia Ádám, Ignacio D. Lopez-Miguel, Anastasia Mavridou, Thomas Pressburger, Marcin Beś, Enrique Blanco Viñuela, Andreas Katis, Jean-Charles Tournier, Khanh V. Trinh and Borja Fernández Adiego

----- Overall evaluation -----

SCORE: 2 (accept)

----- TEXT:

* Summary

The paper presents an integration of FRET into PLCverif, which enables users of PLCverif to specify requirements in a structured natural language (FRETish). This integration addresses several prior limitations of PLCverif and improves its usability. Specifically, FRET brings more expressive power to requirements, makes it easier to write them, and additionally provides a built-in simulator for

validation of complex requirements. The integration has been used in 2 CERN critical systems. These case studies utilized the improvements enabled by FRET: one utilized validation capabilities, while the other utilized the improved expressiveness of requirements.

* Pros

Relevant topic

Well written and explained

Described a useful integration of tools that improves usability of PLCverif

2 case studies that show improvements and impact

* Cons

The claim that FRETish enables (restricted) natural language requirements seems overstated. FRETish requirements seem to be a mix of restricted natural language and boolean expressions. This does not take away from the claims of improvements and impact.

* Questions/Suggestions to the authors

Section 1

"In this paper, we present the integration [of?] NASA Ames' Formal Requirements Elicitation Tool (FRET [1]) into PLCverif"

Section 3.1

Figure 2 (a): while the property has elements of natural language, it does not seem to be written in a "restricted/structured natural language". Perhaps, the claim about what FRETish is needs to be refined. Is it a combination of restricted natural language and boolean expressions?

Section 3.2

"Now PLCverif users can express a large variety of requirements in a natural language and validate these in the simulator." Similarly, to the comment above, this sentence does not seem accurate. The properties seem to be expressed in a mix of restricted/structured natural language and boolean expressions.

----- Correctness -----

SCORE: 4 (good)

----- Reproducibility -----

SCORE: 4 (good)

----- Eligible for Original Software Publication track in Science of Computer Programming -----

--

SELECTION: no