Evaluation of Software Maintainability with Temporal Logic

\*

Abstract (Abstract1)

This template has been tailored for output on the US-letter paper size. The abstract is up to 150 words in length. Do not use symbols, special characters, footnotes, or math in paper title or abstract. All manuscripts must be in English. These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts. Please follow them. The template is used to format your paper and style the text. Do not change the margins, column widths, line spaces, and text fonts.

Keywords: Temporal Logic Verification, Release Notes, Reddit Posts.

# Introduction (*Heading 1*)

This template provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. We have specified only standard paper components. Other components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

# Related work

Lately, there's been a lot of interest in using temporal logic to check how software behaves over time, especially when it comes to safety and liveness. One recent example is a logic called Ticl, introduced by David Swasey and his team in 2024. It’s made to work well with proof assistants and help with programs that keep running or have ongoing effects — which are useful for long-term behavior, like what we’re focused on.

Back in 2015, Andrey Rybalchenko and others came up with a tool called T2. This one’s fully automated and works with LLVM IR. It can check whether the software finishes running or follows fair execution patterns. Basically, it showed that combining temporal logic with automation can really scale up verification in big systems.

Going further back, in 2007, Andreas Bauer’s group took a runtime approach. They turned temporal logic formulas into something called monadic difference logic, which allowed them to check temporal conditions in real time. They used something called difference decision diagrams — a clever trick that made things efficient.

In 2010, Rajeev Alur and his colleagues did a deep dive comparing different kinds of real-time temporal logics. They laid out which ones are more expressive and which ones are easier to compute. That survey is still one of the best references for picking the right logic for a task.

Of course, this field really got its boost from the classic work of Clarke, Emerson, and Sistla in 1985. They came up with model checking, which moved the field from doing handwritten proofs to more automated, system-based analysis. It was limited to finite models, sure, but it was a game changer.

More recently, in 2019, T. Ball and others explored how temporal logic fits into Java. Their work dealt with practical concerns in programming and verification, showing how to connect actual executions with more abstract models.

As for our work, it builds on all of this. We're using MLTL (Mission-Time Linear Temporal Logic) together with a tool called WEST to validate safety and maintenance-related behaviors over time. The aim is to support reliability — not just at one moment, but throughout a system’s lifecycle.

## Abbreviations and Acronyms

Do not use abbreviations in the title or heads unless they are unavoidable. Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Do not define abbreviations such as SMS, SI, MKS, and CGS.

## Units

* Use either SI (MKS) or CGS as primary units. Avoid combining SI and CGS units. (*bullet list*)
* Use a zero before decimal points: “0.39”, not “.39”. Use “cm3”, not “cc”.

## Equations

Equation could be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

 (1)

For the model analyzed in this paper, the stress intensity factor KI can be defined as

 (2)

Note that the equation is centered. Symbols in your equation have to be defined before or immediately following the equation. Use “Eq. 1” or “Equation 1”, not “(1)”, especially at the beginning of a sentence: “Equation 1 is . . .” To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence.

## Figures and Tables

#### Figures and Artwork: All figures should be numbered with Arabic numerals (1,2,...n). All photographs, schemas, graphs and diagrams are to be referred to as figures. Place figures at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures may span across both columns. Figure captions should be below the figures. Insert figures after they are cited in the text. Use the abbreviation “Fig. 1” in the text, and “Figure 1” at the beginning of a sentence.

1. Example of a figure caption. (*figure caption*)

Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader.

#### Tables: All tables should also be numbered with Arabic numerals (1,2,...n). Place tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large tables may span across both columns. Table heads should appear above the tables. Insert Tables after they are cited in the text. Use the abbreviation “Tab. 1” in the text, and “Table 1” at the beginning of a sentence.

1. an example of a table

| **Name** | **Colors** | | |
| --- | --- | --- | --- |
| ***Blue*** | ***White*** | ***Red*** |
| Flag | Yesa | Yes | Yes |

1. With fifty stars. (*Table footnote*)

# General Guidelines

Avoid hyphenation at the end of a line. Symbols denoting vectors and matrices should be indicated in bold type. Scalar variable names should normally be expressed using italics. Weights and measures should be expressed in SI units. Title your files in this order conferenceacrynom\_authorslastname.pdf

Section headings should be left justified, with the first letter capitalized and numbered consecutively, starting with the Introduction. Sub-section headings should be in capital and lower-case italic letters, numbered *A*, *B*, etc, and left justified, with second and subsequent lines indented. You may need to insert a page break to keep a heading with its text.

## Authors and Affiliations

Author names (Fachinger & al., 2004) should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

### Component heads: They identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract1”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

### Text heads: They organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Footnotes

Avoid using footnotes whenever it is possible. If you must use footnotes, place them at the bottom of the column on the page on which they are referenced. Use Times 8-point type, single-spaced. Number footnotes separately in superscripts. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

## Copyright Foms

You must submit the OkIP Copyright Form to have your paper published.

* 1. *Some Common Mistakes*
* The word “data” is plural, not singular.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

Below you have a reference (Young, 1989) to an excellent style manual for science writers.

## Important Notice

This template contains guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your paper prior to submission. Failure to remove template text from your paper may result in your paper not being published.

##### Acknowledgment *(Heading 5)*

Place in this section, acknowledgments of people or organizations that have contributed to support your work.

##### References

List all bibliographical references in 8-point Times New Roman, single-spaced, at the end of your paper. When referenced in the text, enclose the name of the author(s) in parenthesis with the year, for example (Daniel & Sam, 2014; Deal & Gaston, 2021).

Please use APA Style (Gerome, Nana, & Smith, 2015). Where appropriate, include the name(s) of authors and editors of referenced books (Alion, 2021) or book chapters (Clark, Woodley, & De Halas, 1962). Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” (Eliane, 2020). Papers that have been accepted for publication should be cited as “in press” (Nancy, in press). Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation (Yorozu & al., 1987).

Swasey, D., Ioannidis, E., Zakowski, Y., Angel, S., Zdancewic, S., & Birkedal, L. (2024). Structural temporal logic for mechanized program verification. arXiv preprint arXiv:2410.14906

Rybalchenko, A., Heizmann, M., Müller, P., & Podelski, A. (2015). T2: Temporal property verification. arXiv preprint arXiv:1512.08689

Andersen, H. R., & Kristoffersen, K. J. (2007). Temporal runtime verification using monadic difference logic. arXiv preprint arXiv:0705.4604

Alur, R., Etessami, K., & Yannakakis, M. (2010). Real-time and probabilistic temporal logics: An overview. arXiv preprint arXiv:1005.3200

Clarke, E. M., Emerson, E. A., & Sistla, A. P. (1985). Automatic verification of finite-state concurrent systems using temporal logic specifications. Lecture Notes in Computer Science (1985)

Ball, T., de Halleux, J., & Tillmann, N. (2019). Verifying temporal specifications of Java programs. Software Quality Journal (2019)