

1 Abstract

Explains briefly the paper and what it does.

2 Introduction

Introduce the scope.

- Event-based scheduling as a concept
 - Why is it useful?
- Network events and REST APIs as a concept
 -

2.1 Problem

Something about the MEOW monitor, and expanding it. Probably also why it would be a good idea(?).

- Why MEOW needs it
 - Further expansion of possibilities with the system

2.2 Background

Explaining the existing code and what it does and why. Also modules used and other important stuff to know.

- General MEOW_BASE structure:
 - Monitors check if patterns are triggered
 - * Patterns are triggered by an event. Currently only file event triggers are implemented.
 - Handlers perform recipes
 - * Recipes are actions that are taken when triggered. python code can be run as a recipe.
 - The conductor handles the job queue of rules
 - * Rules combine patterns and recipes (When this pattern is triggered, perform this recipe).
- Specific (but not too granular) implementation details of MEOW_BASE.
 - Class structure
- Other important stuff to know

3 Method

Explaining the code I wrote and why I made those choices.

4 Results

Does it work? How well?

4.1 Testing

4.2 Discussion

With the hindsight of the results, what could I have done better?

5 Future Work

What should someone do if they want to fix my mistakes, or expand on them further.

6 Conclusion

Did I succeed in what I wanted to do?