

Rule-based employee prioritization in Fairpark

Degree programme : BSc in Computer Science | Specialisation : Digital Business Systems

Thesis advisor : Prof. Rolf Jufer

Expert : Dr. Andreas Spichiger

Industrial partner : mobilid e, Gen ve

Fairpark is an online solution for parking and mobility management from mobilid e. A core feature is the prioritization system which sorts employees according to their right to a resource based on pre defined criteria. To enable mobilid e to react faster and more agile to customer requests these criteria are made dynamically adaptable with the help of a rule engine.

Introduction

Imagine: You are the manager of a company that employs 1000 people but only has 200 parking spaces available. How should the spaces be fairly distributed among the employees? Which employees really need to come to work by car and which can do without? To solve this there are mobility management solutions such as Fairpark from mobilid e. Its prioritization system allows companies to weight different criteria of their employees (travel time, mobility type, hierarchy within the company, etc.) to sort the employees according to their right to a certain resource. These criteria are currently hard coded in Fairpark. To enable mobilid e to react faster and more agile to customer requests, the criteria are made dynamically adaptable by using a rule engine.

Goal

The idea is that a mobilid e employee creates rules for a client company that are precisely geared to the client's needs. For example, if the client wants their employees to be prioritized higher if they work in department X, then the following rule can be created: „if users department = X, then add 1000 points to the user score“. All rules are evaluated against each of the client's employees which then can be sorted by their score to get a priority list. To turn this idea into reality, an agile software development project is carried out with the aim of creating a functional, adaptable and fully integrated prototype. The main objectives are to evaluate a suitable rule engine, integrate it into Fairpark, implement a graphical interface to manage the rules and ensure the correct execution of the rules.

Solution

To guarantee that all requirements are implemented in the best possible way, the created rule engine is a self-written solution that is seamlessly integrated into the Fairpark codebase. The main features are:

- Rule management user interface: Create rules with

an infinite amount of conditions (if part) and calculations (then part)

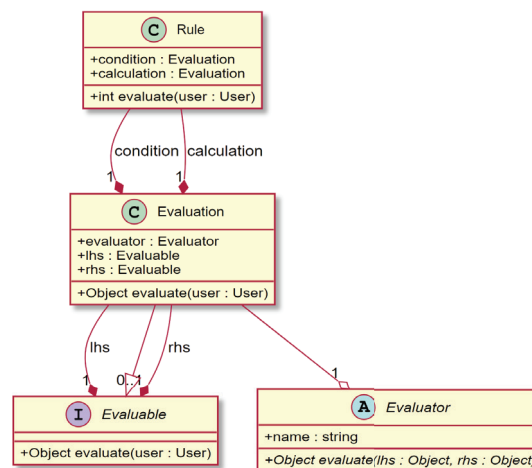
- Different evaluable components: Use fixed user attributes, generic user attributes, a manually entered value or even a pre-defined function inside the conditions and calculations
- Silent integration: Switch between the old and new solution as desired
- Sandbox testing: Test the rules against real users without altering the score of them
- Error handling: Input validation, type safety, loop detection, runtime testing and error recovery strategies
- Traceability: change log, test output log, score breakdown
- Adaptability: More types, evaluators and evaluable components can be added anytime by just adding more subclasses



Andreas Ramon Erb
andreas.erb@gmx.ch

Conclusion

The solution demonstrates the feasibility and stability of rule-based prioritization. It opens up new possibilities for mobilid e and can now be extended as desired thanks to its adaptability.



simplified UML class diagram of the rule engine core