Severe Weather Alert System

Dion Scheper, Terence Beijloos, Bart van Nimwegen

November 17, 2024

1 Context/Background

We are building a severe weather alert application that can be used by governments to alert civilians of incoming weather events. In the case of extremities like tsunami's, this system allows civilians to prepare for the possibly dangerous event. An efficient alerting system is essential to survive such events.

The flexibility of the SPL approach will ensure that the solution can evolve as new weather patterns emerge, new communication channels are introduced, or different geographic areas require specific adaptations.

2 Motivation

Creating a software product line for a severe weather app is a compelling idea because the domain offers significant variability, with many weather-related features like temperature, precipitation, and wind that can be tailored to different user needs.

Developing a minimum viable product is relatively straightforward, as the app could start with core functionalities like basic alerts for severe weather and expand later. Additionally, the project offers an opportunity to tackle the optional feature problem, where weather properties may seem independent but could have hidden dependencies during implementation.

Finally, the app isn't heavily reliant on external services; weather data can easily be mocked during development, making it easier to test and iterate.

3 Methodology

In this case we will be following a software development methodology. We will use these tools and processes:

- agile / remote
  We will work mostly remote, hosting interactive standups on discord where we can share our screens and discuss possible problems.
- git / github for version control of code
  This facilitates the version control so we can work simultaneously and merge code later.
- featureIDE
  We now have some experience with eclipse and the FeatureHouse plugin. This is a good starting point for our project.

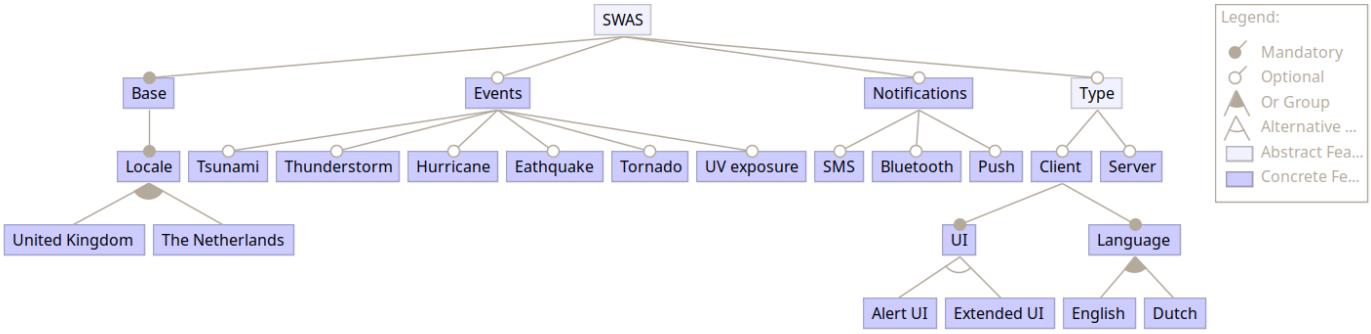
- optional: include aspectJ if possible
  The aspect oriented examples from the lectures were very interesting and seem to add a lot of value. If we have the chance we want to integrate this into our project.

4 Deliverables

We plan to use the following variability.

- configuration file
- translations
- compile time feature selection
- feature selection by the user

And we have drafted a feature model.



5 Area's of responsibility

We are working this project with three people. Everyone will have their own area of responsibility, for example it does not make sense to let two people start on the events as most events will share some base code.

- Dion
  Client/Server code to let the application interact and be notified.
- Terrence
  Events business logic
- Bart
  UI elements starting with the alert UI (simple)