Rigorous Methods for Software Engineering
Coursework 1 Report

Design Decisions and Use of Cardinality Constraints

I decided to create a signature for the higher-level controls of the Water Tank Protection (WTP) system that represented overall control of all automated subsystems.
I created an abstract class representing the current water level which I extended for each possible reading. This helped constraint the system's predicates as the parameter representing the current water level could be type constrained to one of the extended signatures rather than the higher-level abstract signature.
The controller sub system also had fields to connect the alarm and drain if the predicate marked them as open or activated.

Cardinality constraints were used to constrain the outer bound number of sensors each WTP system could control. This was a '1 to many' relationship. Cardinality constraints were also used to control the lower bound number of sensor inputs that could be read by the controller when the drain valve was open. This was a 'many to 1/0' relationship.

Relationship between WTP Controller and the Sensor Readings

In my WTP model the water level is read by the sensors and sent to the WTP Controller. The controller uses this information to identify whether the alarm should be activated, or the drain opened. For the sensors to take a reading of the current water level, the higher-level controls need to activate this function. As the higher-level controls were controlling the sub-systems, the WTP Controller could therefore focus on only controlling the systems within the tank.

How did you Select the Scope? Use of Predicates

My WTP model ran a global scope of 3 on each individual predicate like so: run 'predicateName' for 3. predicateName is replaced by the name of one of four predicates. The analyser then checks for examples that have no more than 3 atoms in each set from the given predicate's constraints. I chose a scope of 3 because there was only ever a maximum of three possible readings: alarm on; alarm on and drain open; alarm off and drain closed.

I used predicates in my WTP model to describe the system's behaviour when the tank reached a particular water level. I added constraints to each predicate based upon the sensor's current readings, whether the alarm was activated, whether the drain was open and whether the console's reset mechanism was activated.

Suitability of Alloy for Specified Project

I do not feel Alloy was a suitable tool for this project. The ideal approach would have been a hierarchal structure which had the WTP system's higher-level controls on top and the automated subsystems underneath. In Alloy this would, however, have given rise to under-constrained instances and thus I had to avoid this approach.

Matthew Frankland (H00241997/mf48)
MEng Software Engineering