Proof Validation – Chemical Plant Alarm Management System

Proof Obligation 2: Map apply - DEFAULT.Plant (Provable)

VDM-SL: (forall mk_Plant(schedule, alarms):Plant! & (forall a in set alarms & (forall peri in set (dom schedule) & peri in set dom schedule)))

This proof obligation ensures that any period (peri) used to access the schedule map within a Plant instance is always a valid key within that map's domain. It verifies that whenever the system attempts to retrieve data from the schedule using a period, that period must already exist as a key in the schedule map.

In the Java GUI code, the schedule map is implemented as a HashMap <String, Set<String>>. The schedule.containsKey(selectedPeriod) method call is exactly matches the VDM-SL check peri in set dom schedule. It ensures that before attempting to retrieve experts for a selected period using schedule.get(selectedPeriod), the code first verifies that the period actually exists as a key in the schedule HashMap.

Java GUI code:

```
if(schedule.containsKey(selectedPeriod)){
        Set<String> expertsOnDuty = schedule.get(selectedPeriod);
} else {
}
```

Status: Provable

- The VDM-SL tool was able to automatically prove this obligation. The "Provable" status indicates that the obligation is directly from the definitions of the Plant and Schedule types.
- The "trivial" proof status, means the proof is very basic and can clearly prove from the given values.
- This proof obligation ensures data integrity and prevent runtime errors.