

Iteration 2 Requirements

CSCI3081

Program Design and Development

Entities

- Player
- Robot
- SuperBot
- Home Base
- Recharge Station
- Obstacle

Behavior

- WIN: Freeze all the robots
- LOSE: All robots become SuperBots OR run out of battery
- Robots autonomously avoid collision UNLESS distress call sensed
- Player-Robot collision: robot freezes
- Robot-FrozenRobot collision: unfreeze Robot
- Player-SuperBot collision: Player freezes
- Time unfreezes Player
- Robot-HomeBae collision: robot becomes SuperBot
- Bounce off of walls and other entities.

Sensors and Events

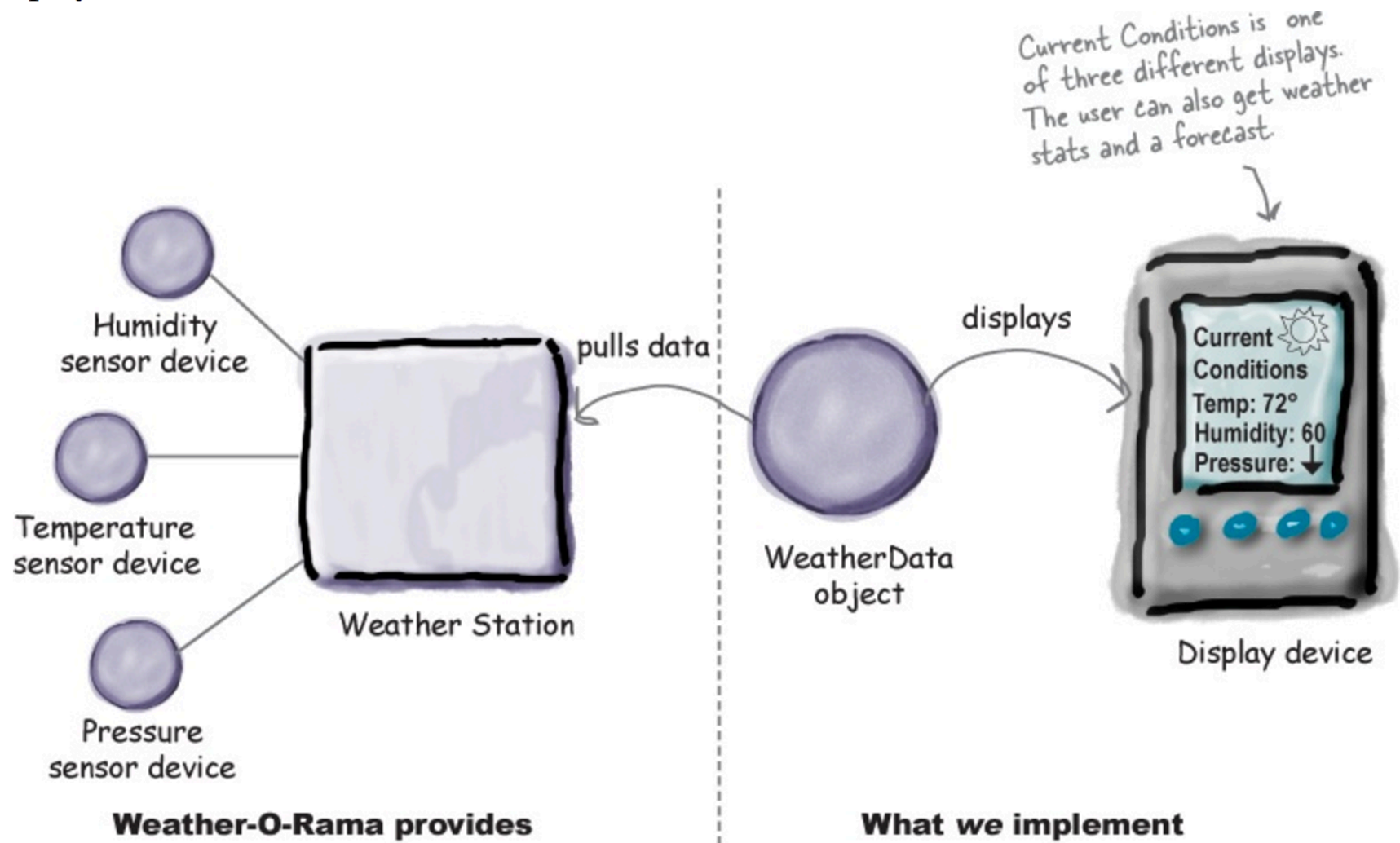
EVENTS

- Collision
- Proximity
- DistressCall
- TypeEmit

Sensors

- Touch
- Proximity
- Distress
- EntityType

Observer Pattern

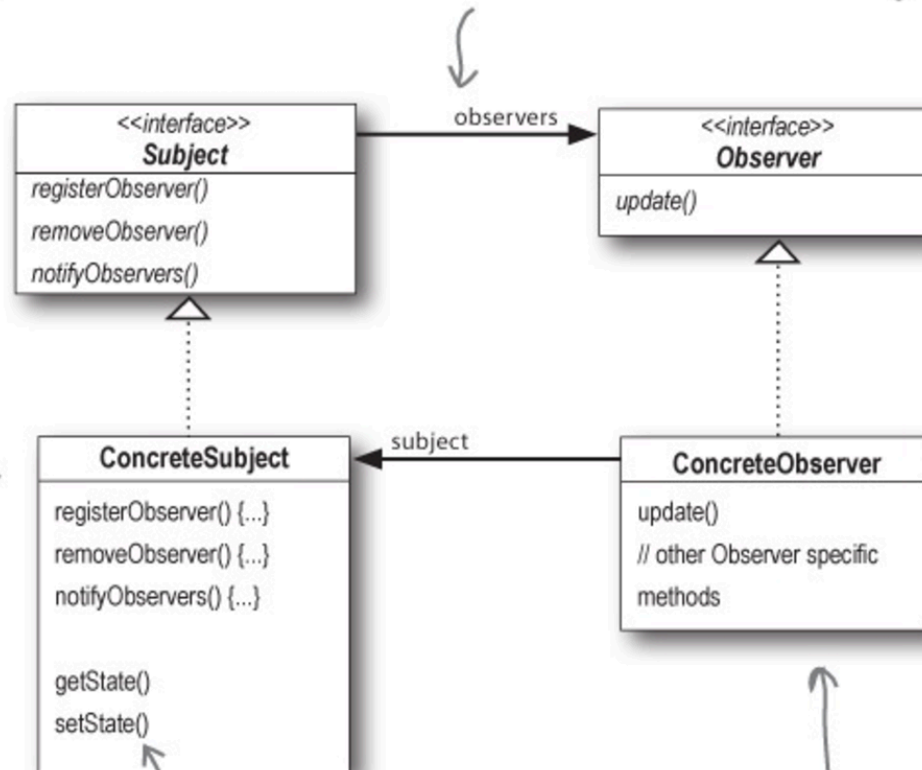


The Observer Pattern defined: the class diagram

Here's the Subject interface. Objects use this interface to register as observers and also to remove themselves from being observers.

Each subject can have many observers.

All potential observers need to implement the Observer interface. This interface just has one method, update(), that gets called when the Subject's state changes.



A concrete subject always implements the Subject interface. In addition to the register and remove methods, the concrete subject implements a `notifyObservers()` method that is used to update all the current observers whenever state changes.

The concrete subject may also have methods for setting and getting its state (more about this later).

Concrete observers can be any class that implements the Observer interface. Each observer registers with a concrete subject to receive updates.

Design Decisions

- Initialization of all entities.
- Sensor and Event class.
 - What's in the base class?
 - What's in each of the classes?
- Design of registration of sensors with arena (through entity, or directly from sensor)??
 - How do entities and sensor know about the arena?
- Should design of entities change?
 - Recharge and regular obstacles might move in the future.
- Motion Handle: Different uses of sensor information.
 - Strategy?
 - Template?
 - Something else?
- How to turn Robot into SuperBot?
- How are type and distress calls emitted and sensed in Arena?