

CS 525 - ASD

# Advanced Software Development

**MS.CS Program**  
Department of Computer Science  
Rene de Jong, MsC.



Maharishi University  
OF MANAGEMENT

# CS 525 - ASD

## Advanced Software Development

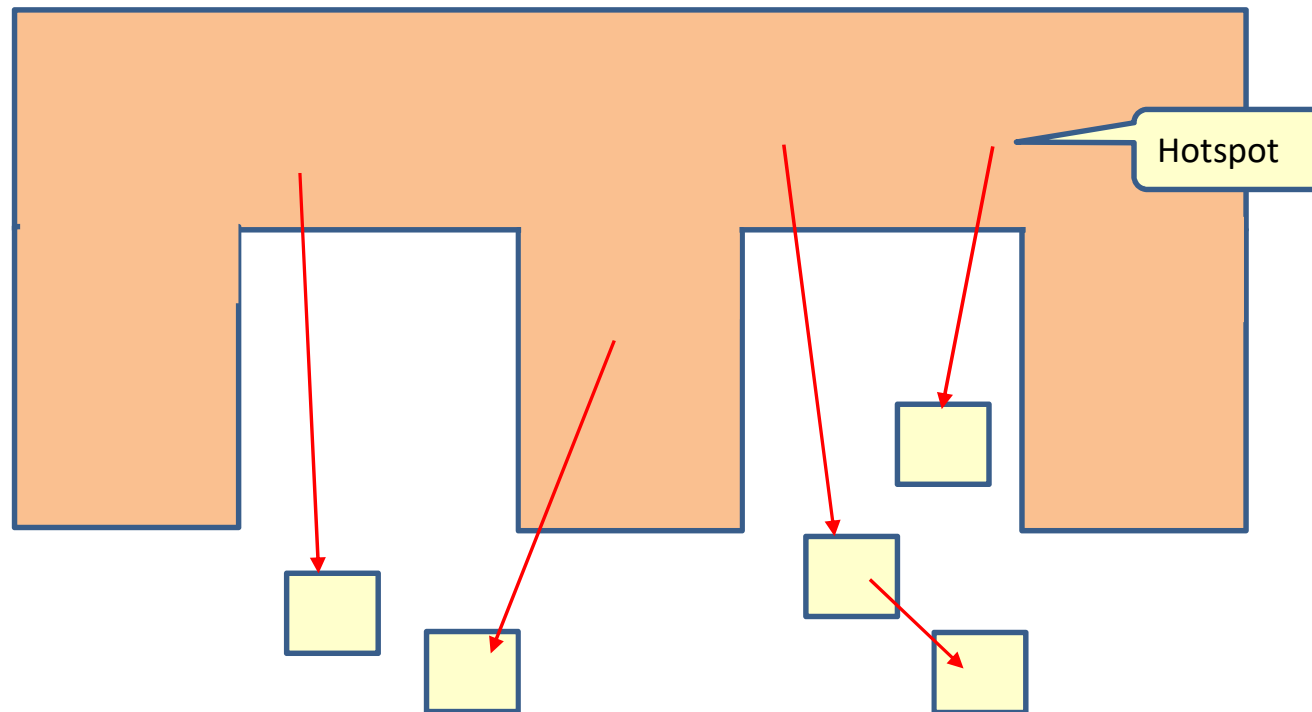
© 2019 Maharishi University of Management

**All course materials are copyright protected by international copyright laws and remain the property of the Maharishi University of Management. The materials are accessible only for the personal use of students enrolled in this course and only for the duration of the course. Any copying and distributing are not allowed and subject to legal action.**



Maharishi University  
OF MANAGEMENT

# Hotspot (plugin point)

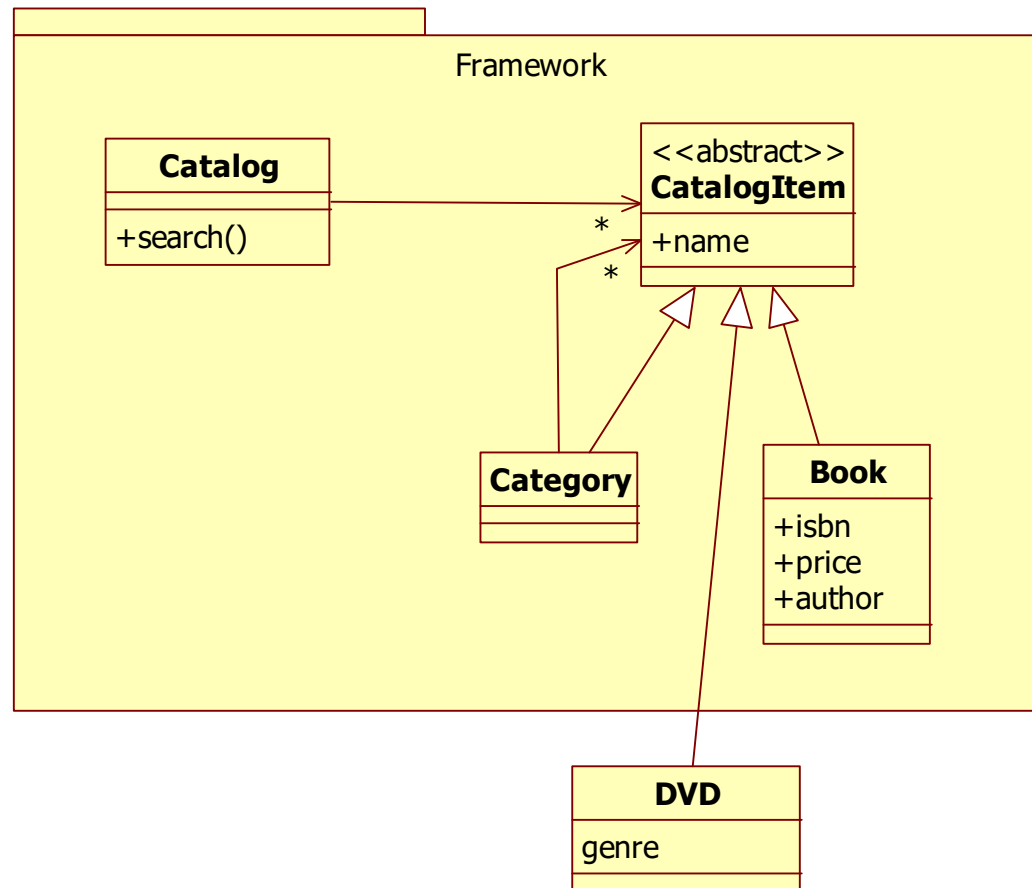


# How to make hotspots?

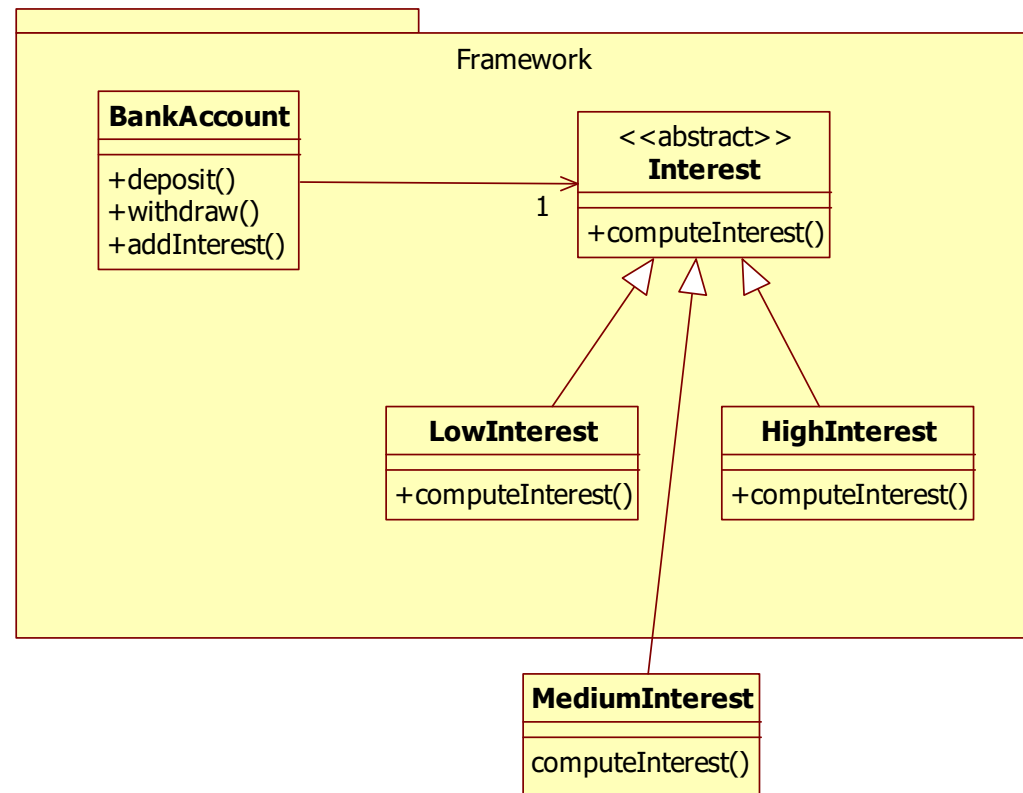
---

- Plugin new algorithms
  - Strategy pattern, Chain of responsibility pattern
- Plugin new state behavior
  - State pattern
- Plugin new listeners
  - Observer pattern
- Translate between your code and FW code
  - Adapter pattern
- Plugin new actions
  - Command pattern
- Plugin new traversal algorithm
  - Iterator pattern
- Create new objects
  - Factory
- Add classes to a tree structure
  - Composite pattern

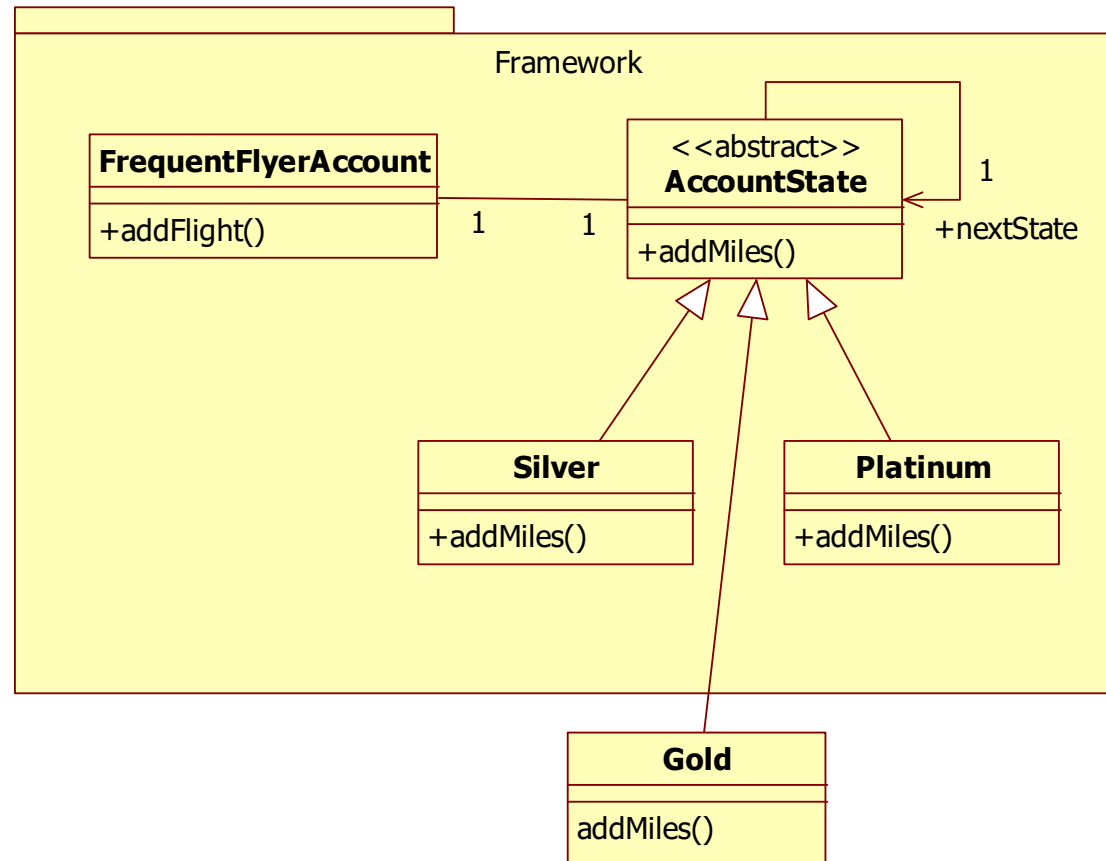
# Plugin points: Composite pattern



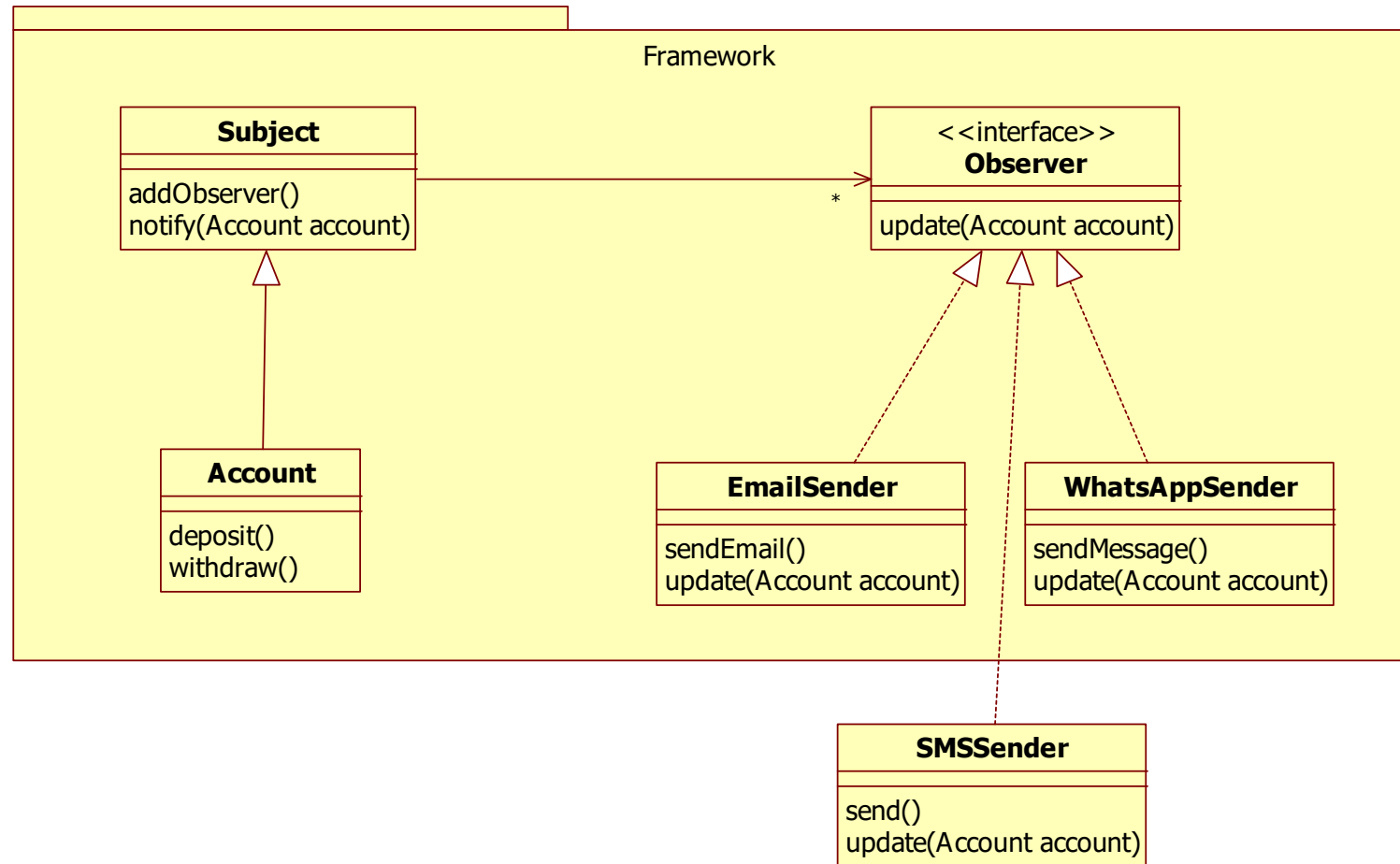
# Plugin points: Strategy pattern



# Plugin points: State pattern

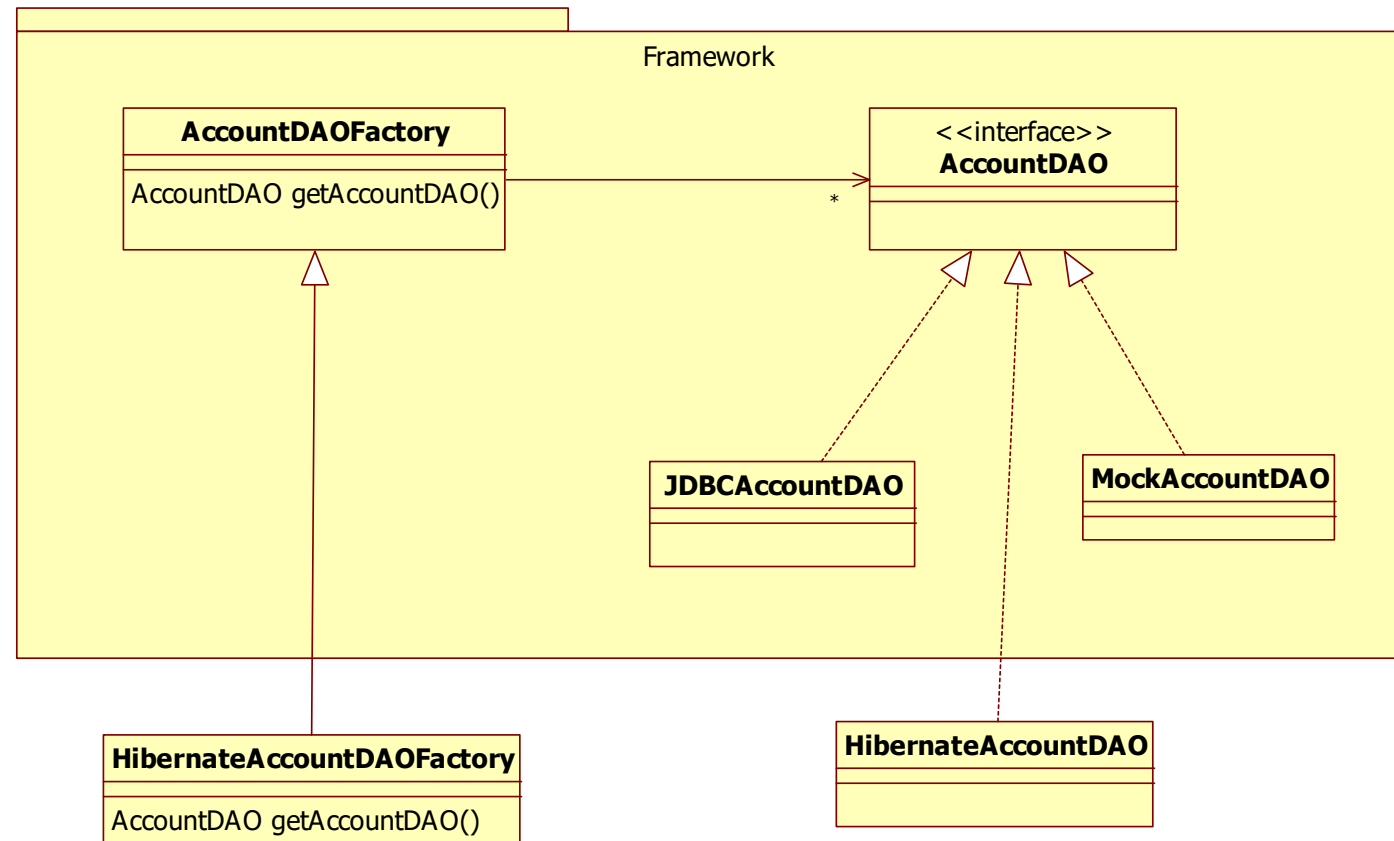


# Plugin points: Observer pattern

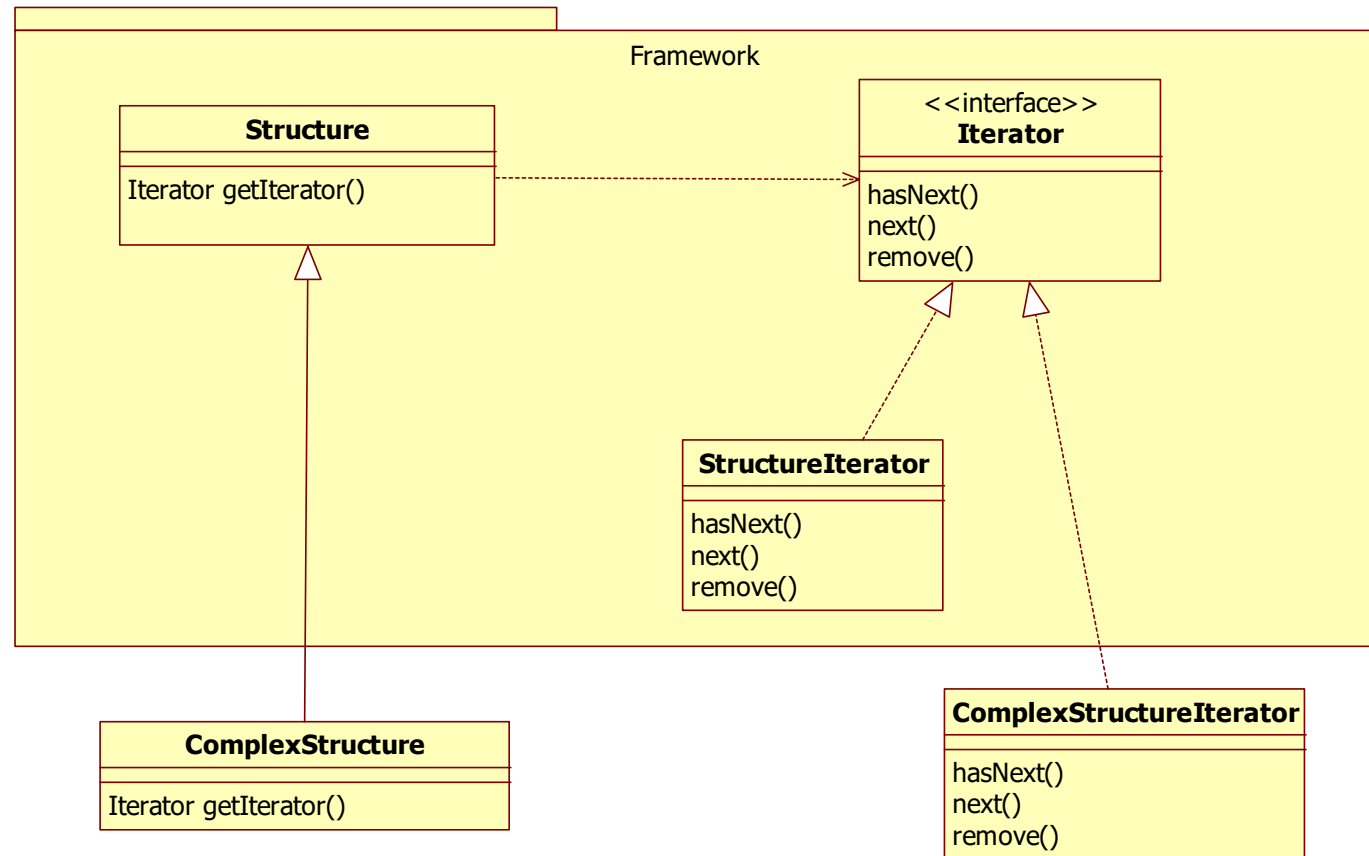




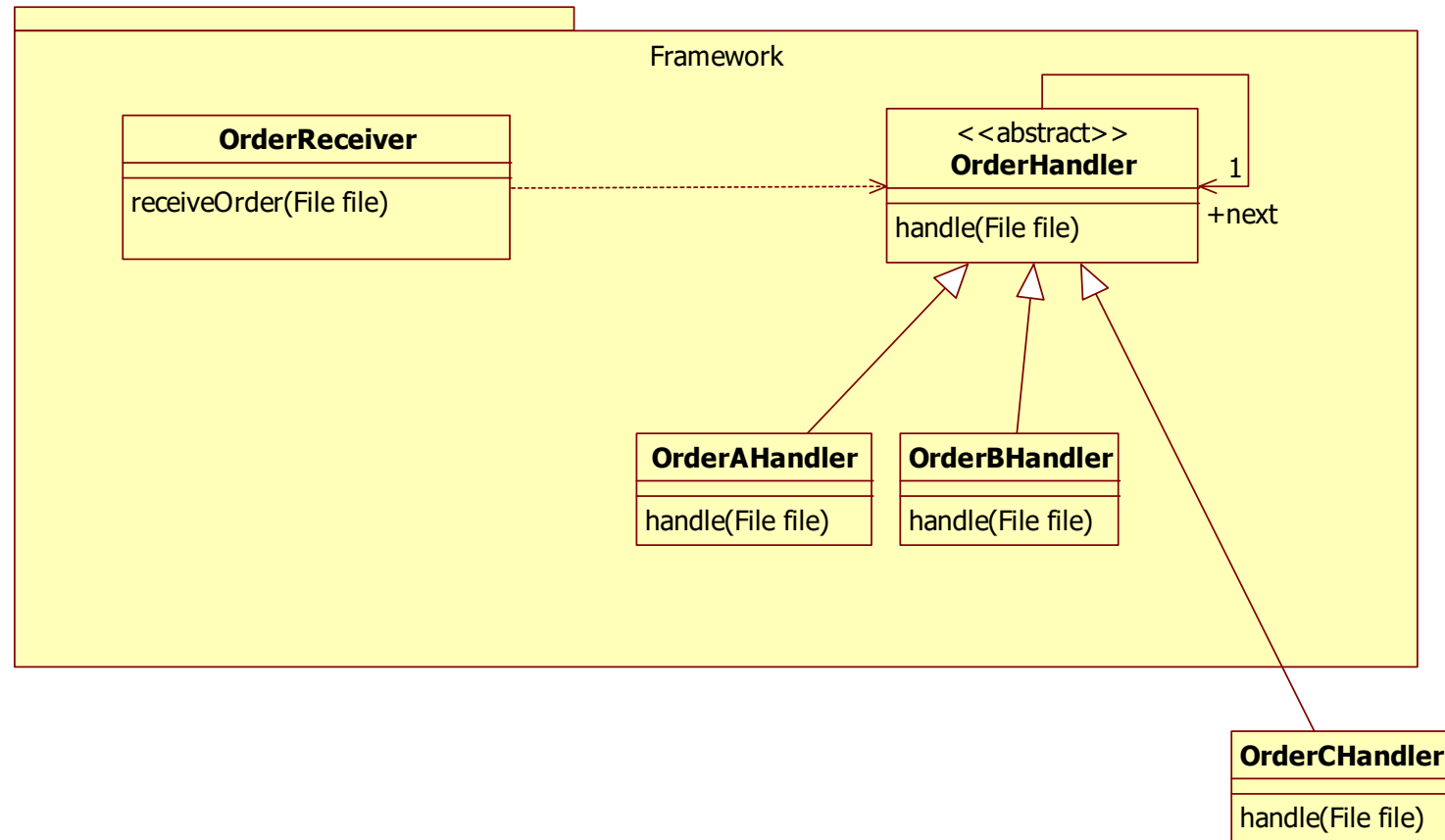
# Plugin points: Factory pattern



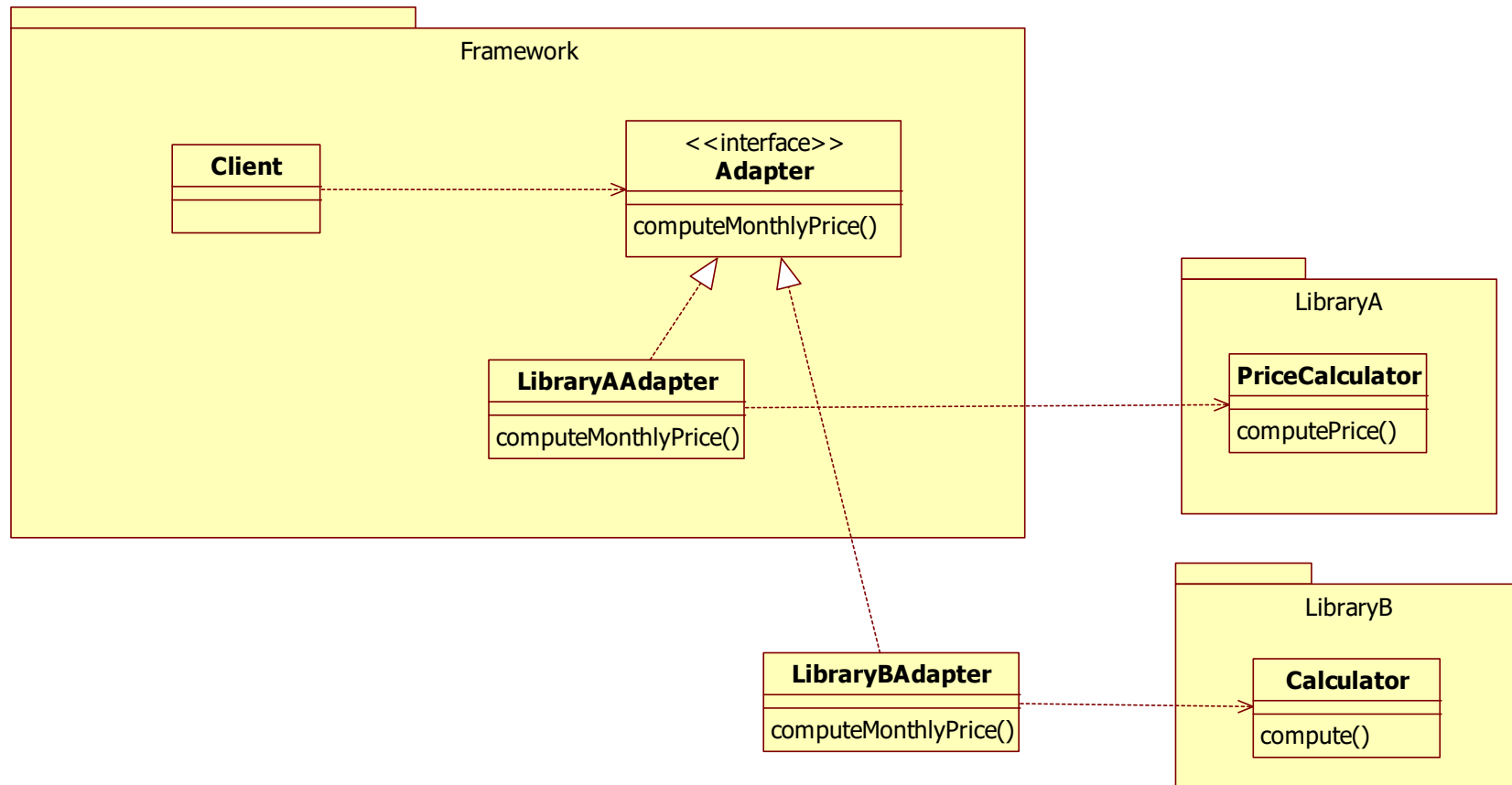
# Plugin points: Iterator pattern



# Plugin points: COR pattern

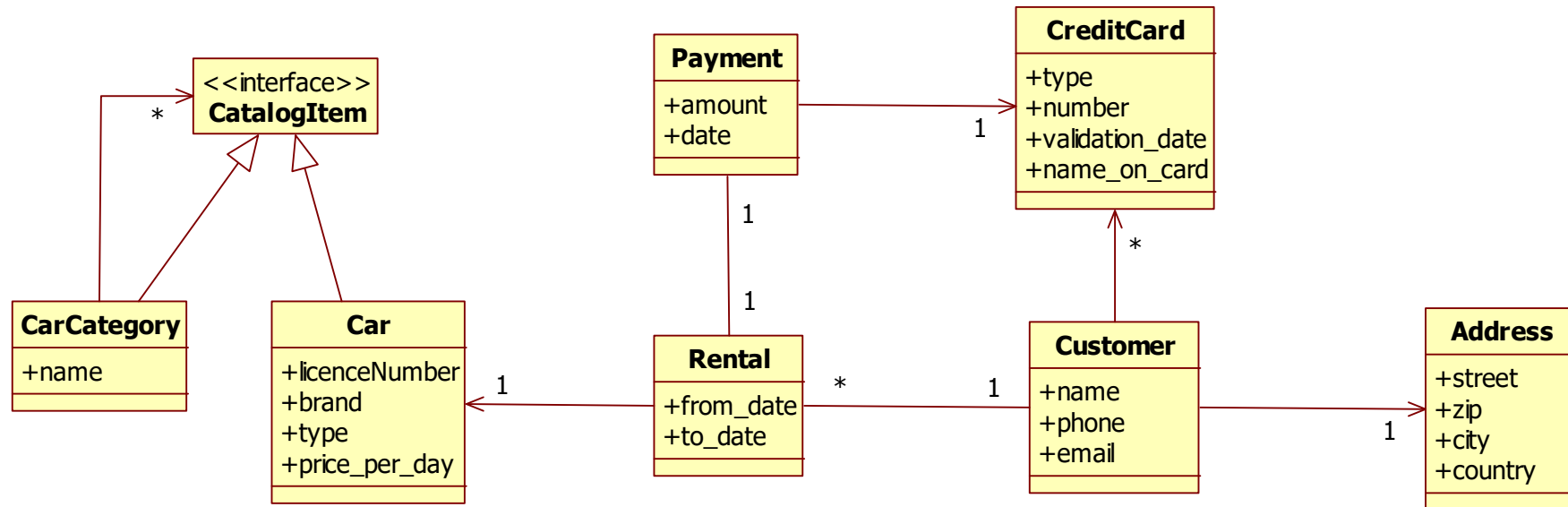


# Plugin points: Adapter pattern

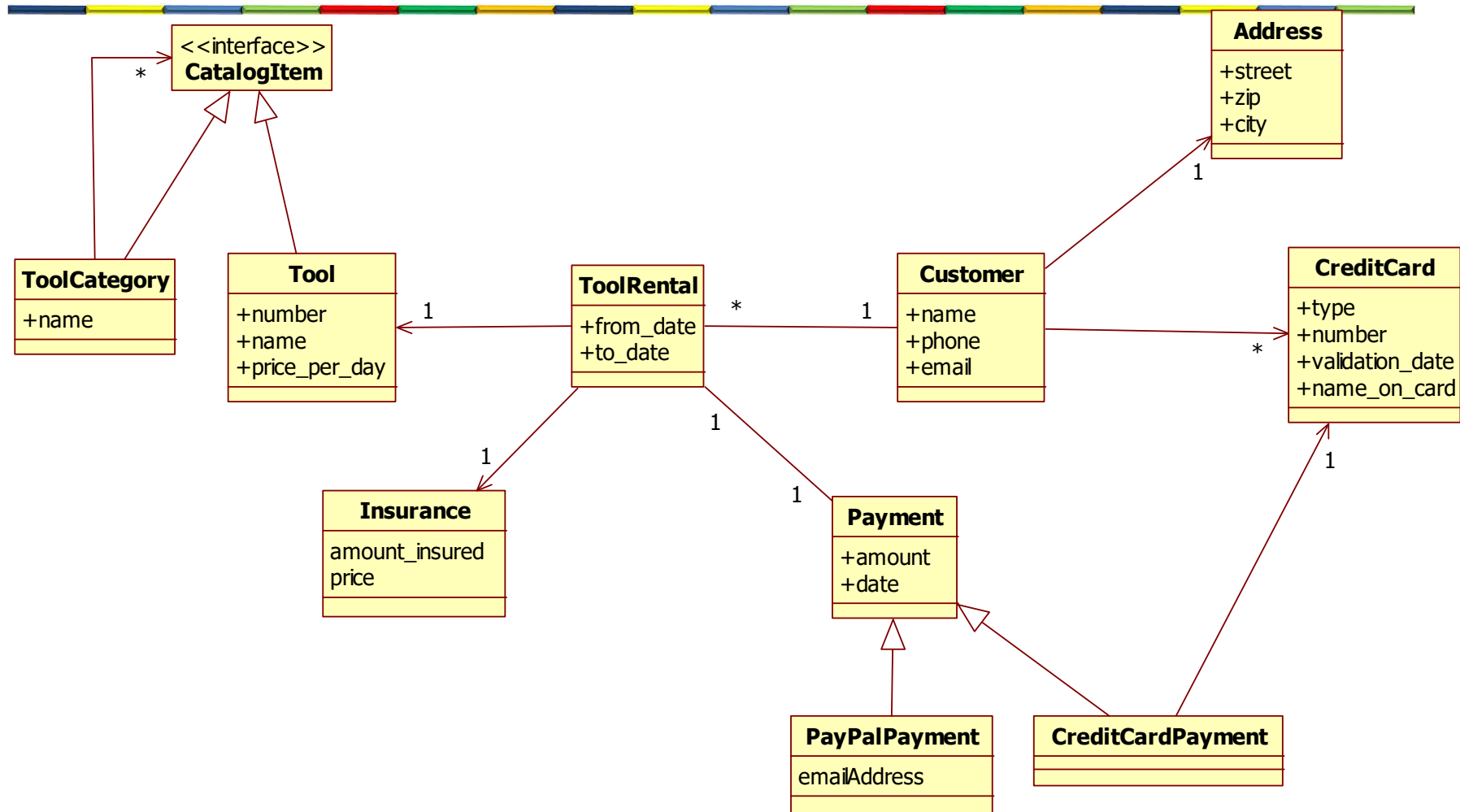


# **RENTAL FRAMEWORK**

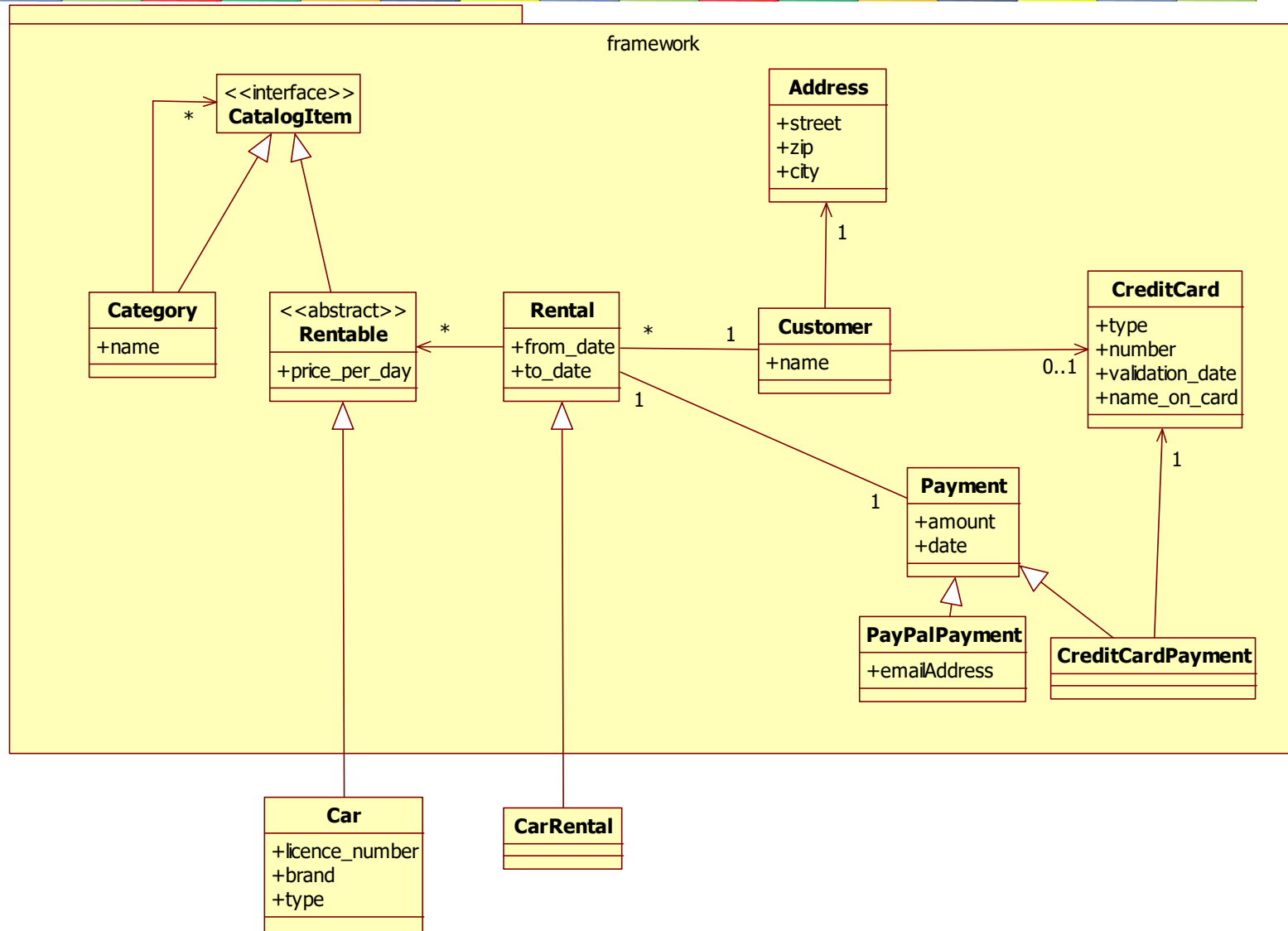
# Car rental application



# Tool rental application

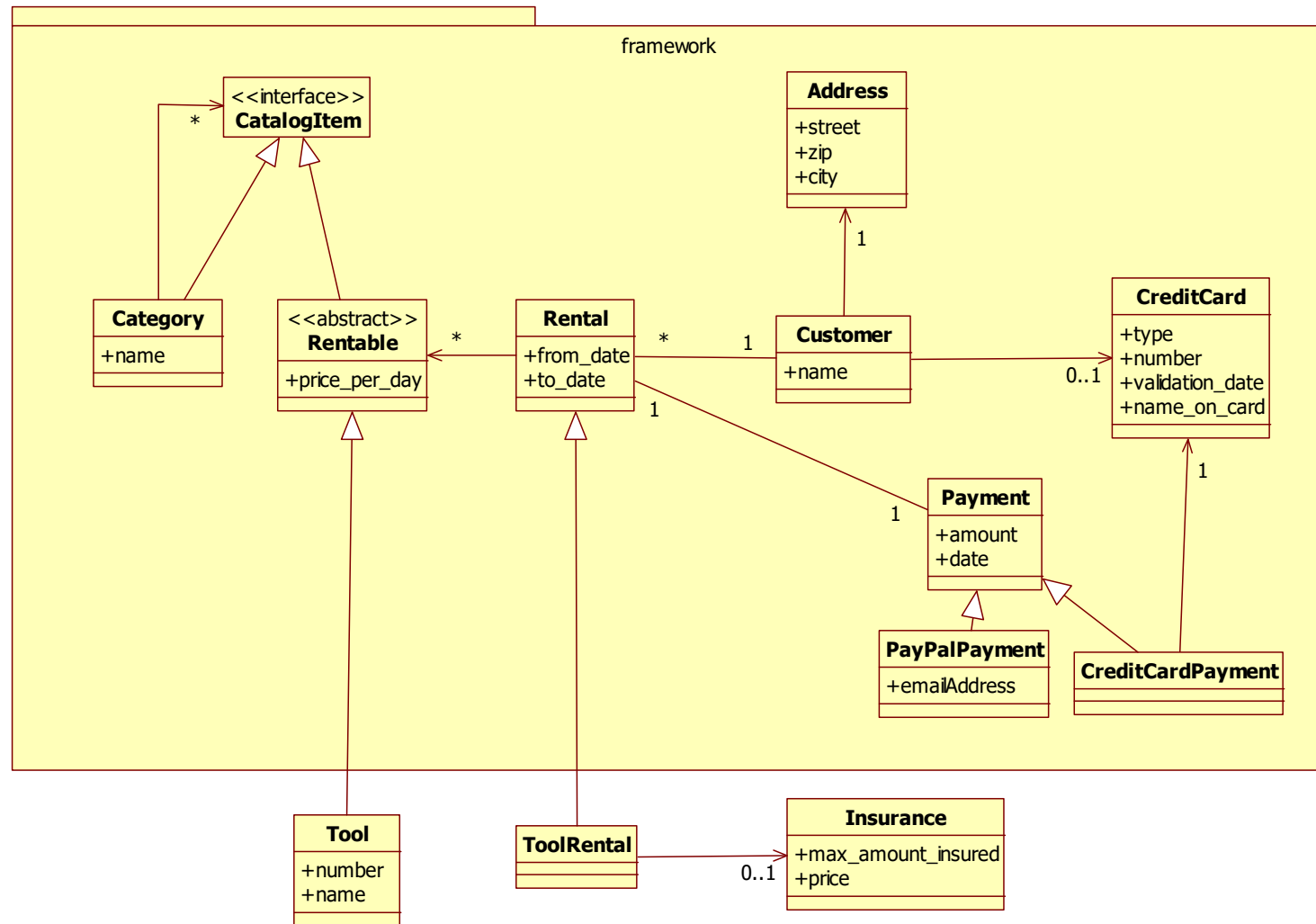


# Framework + Car rental application

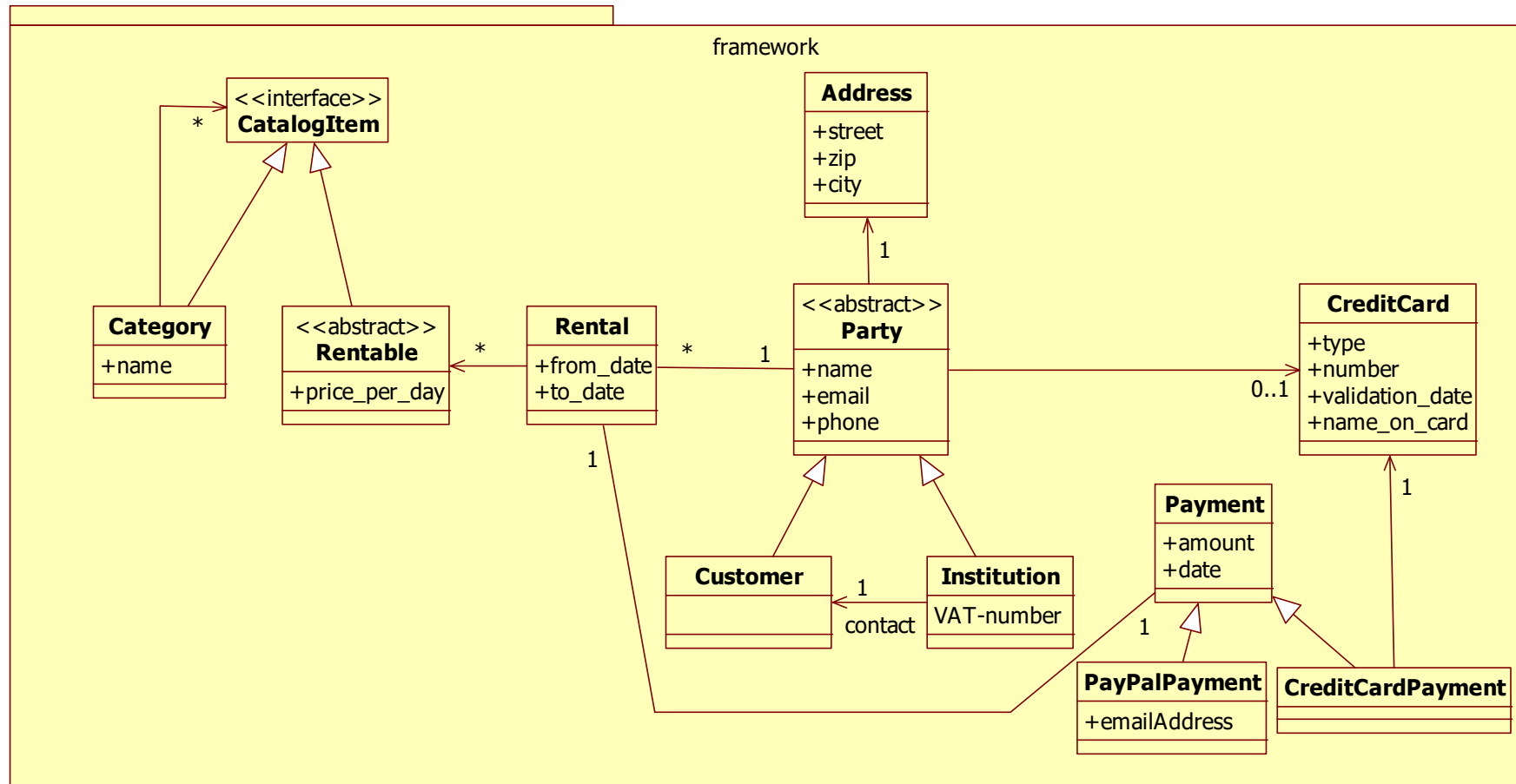




# Framework + Tool rental application

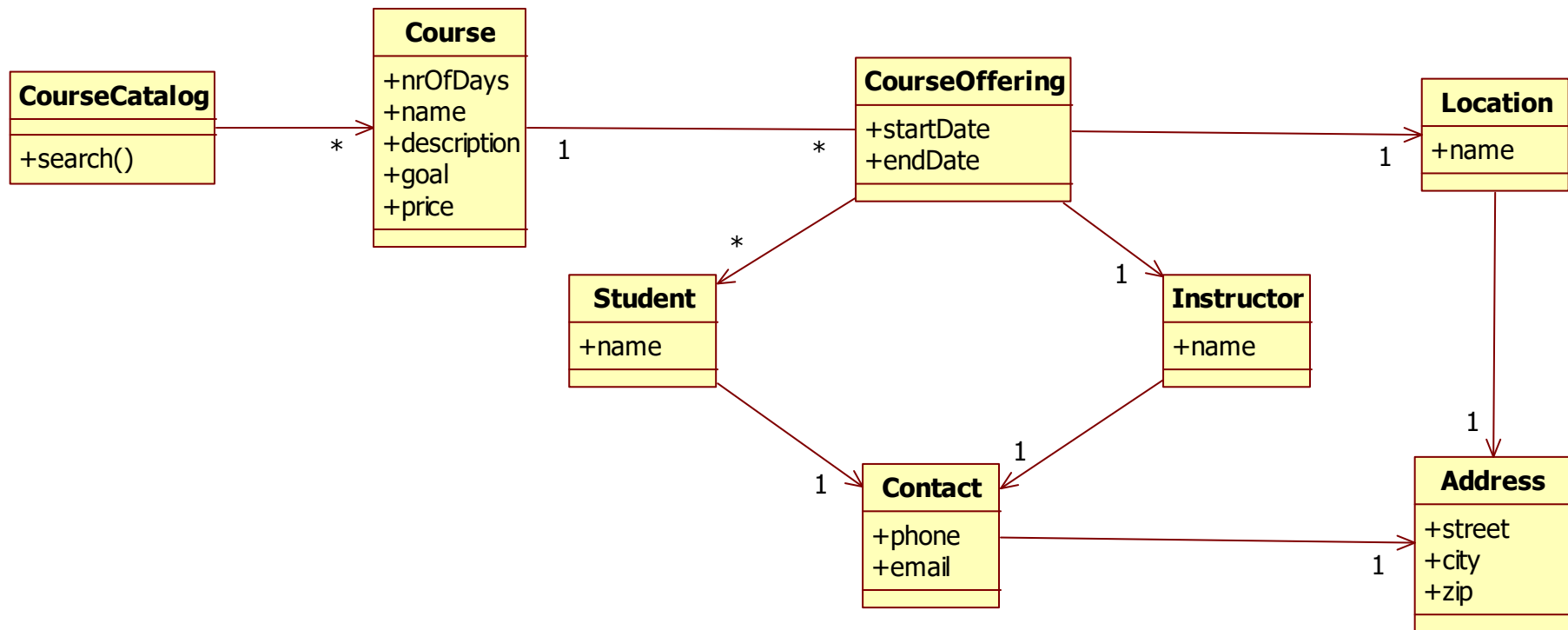


# Party pattern

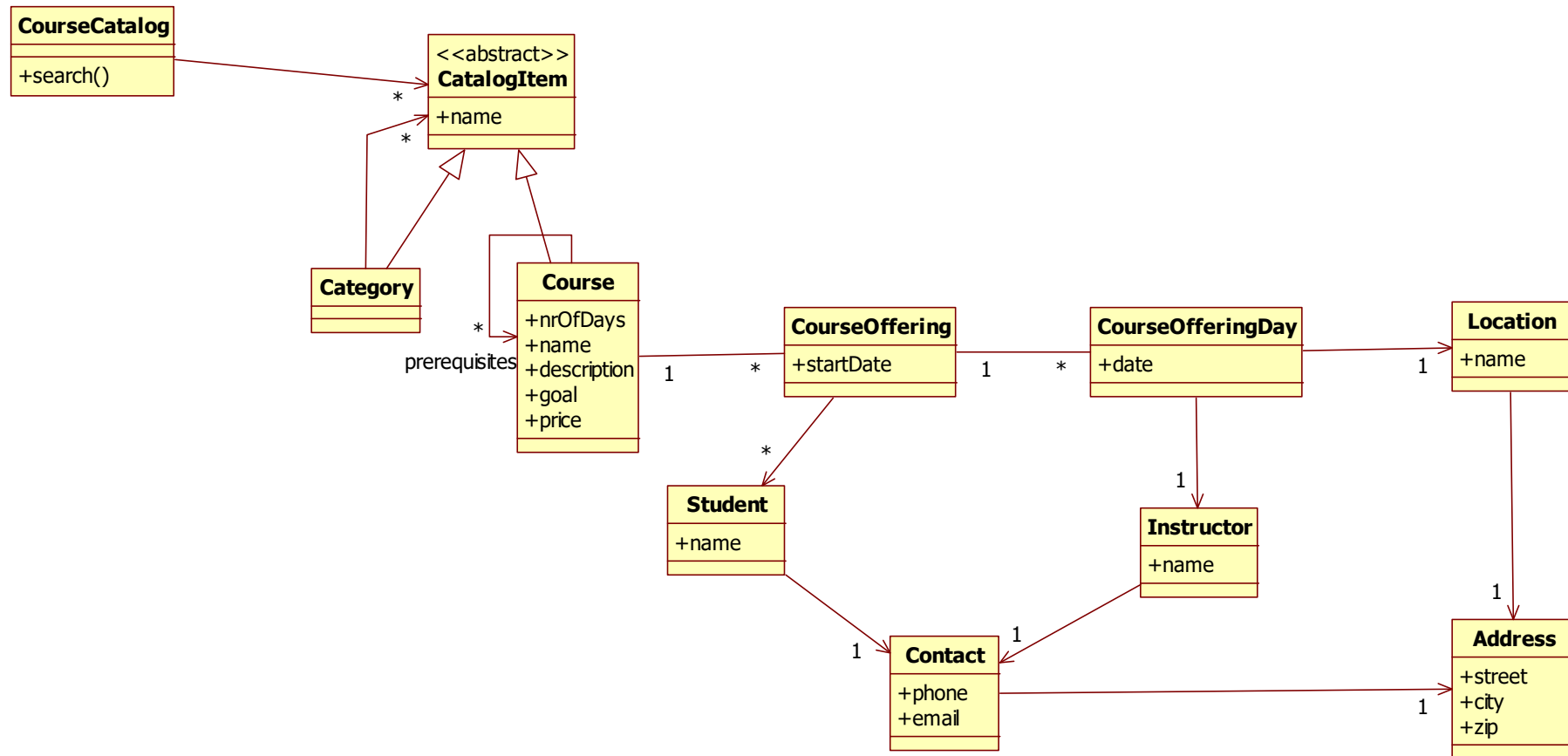


# **COURSE REGISTRATION FRAMEWORK**

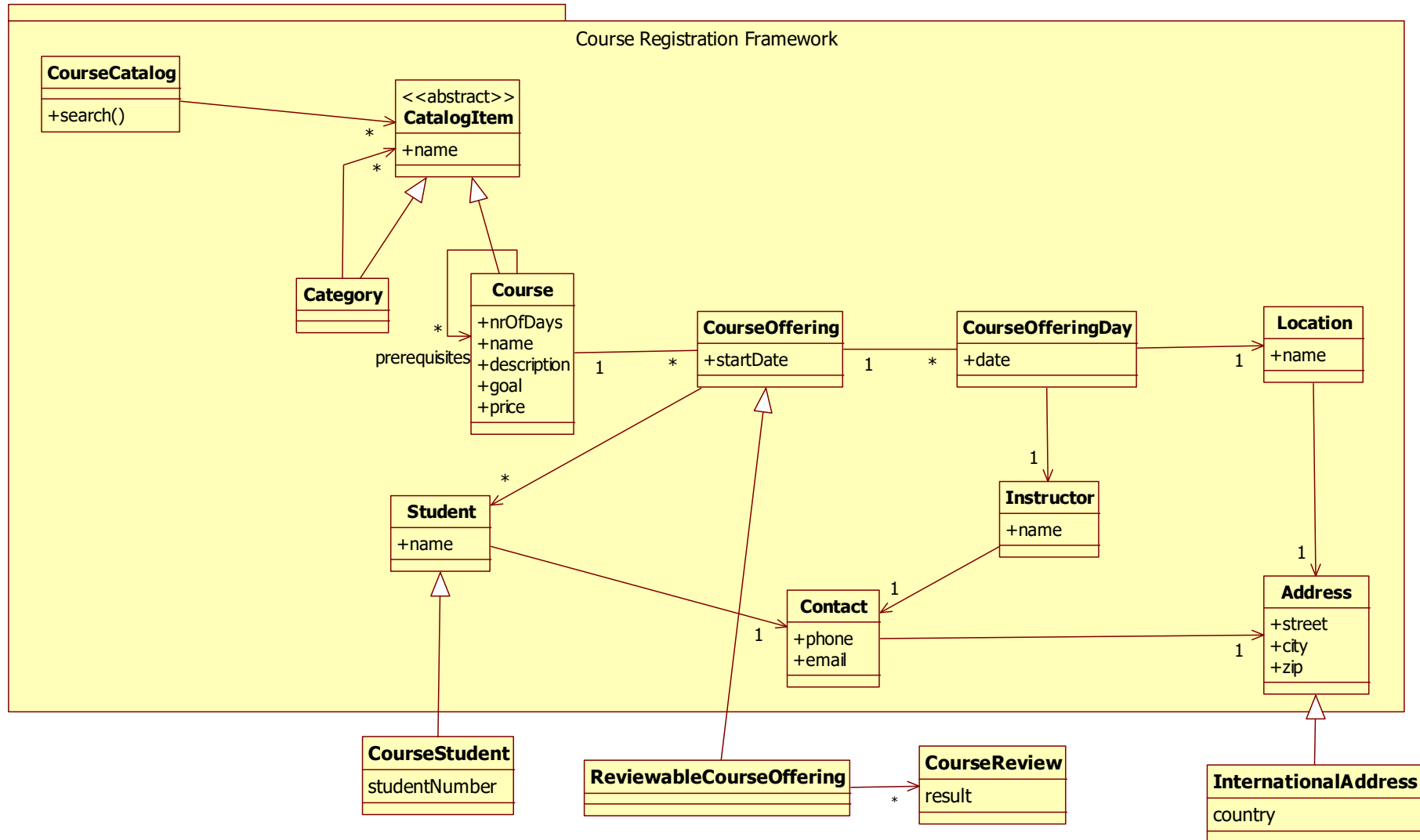
# Simple course registration system



# Advanced course registration system

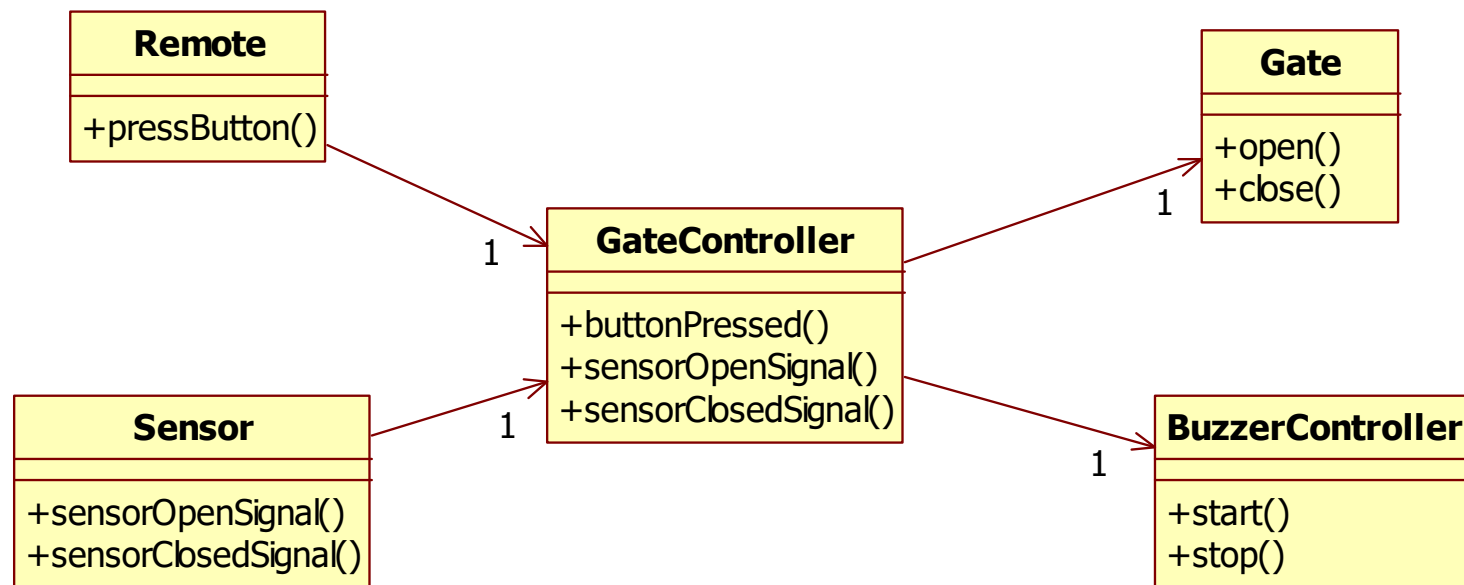


# Course registration framework



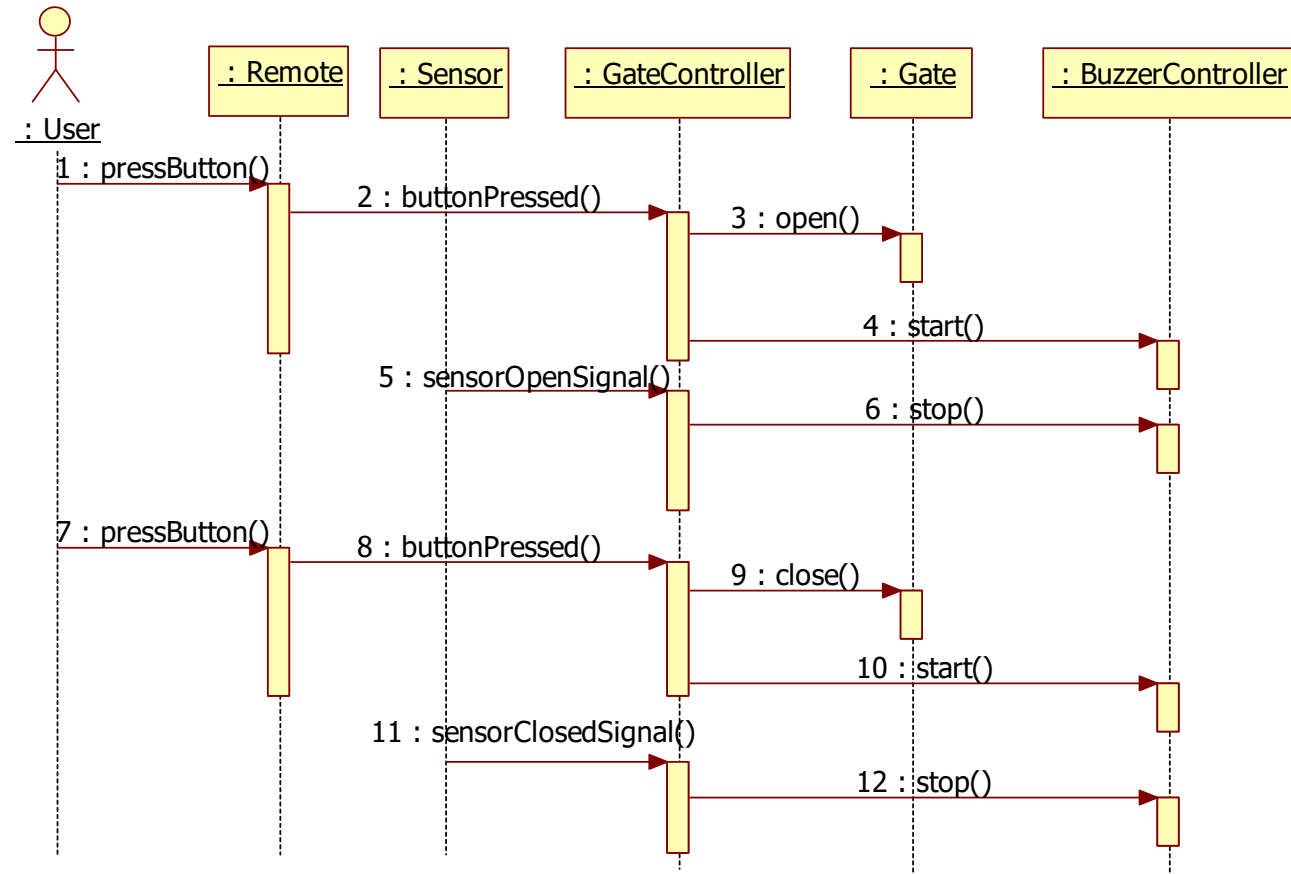
# **GATE CONTROLLER FRAMEWORK**

# Gate controller application





# Gate controller application

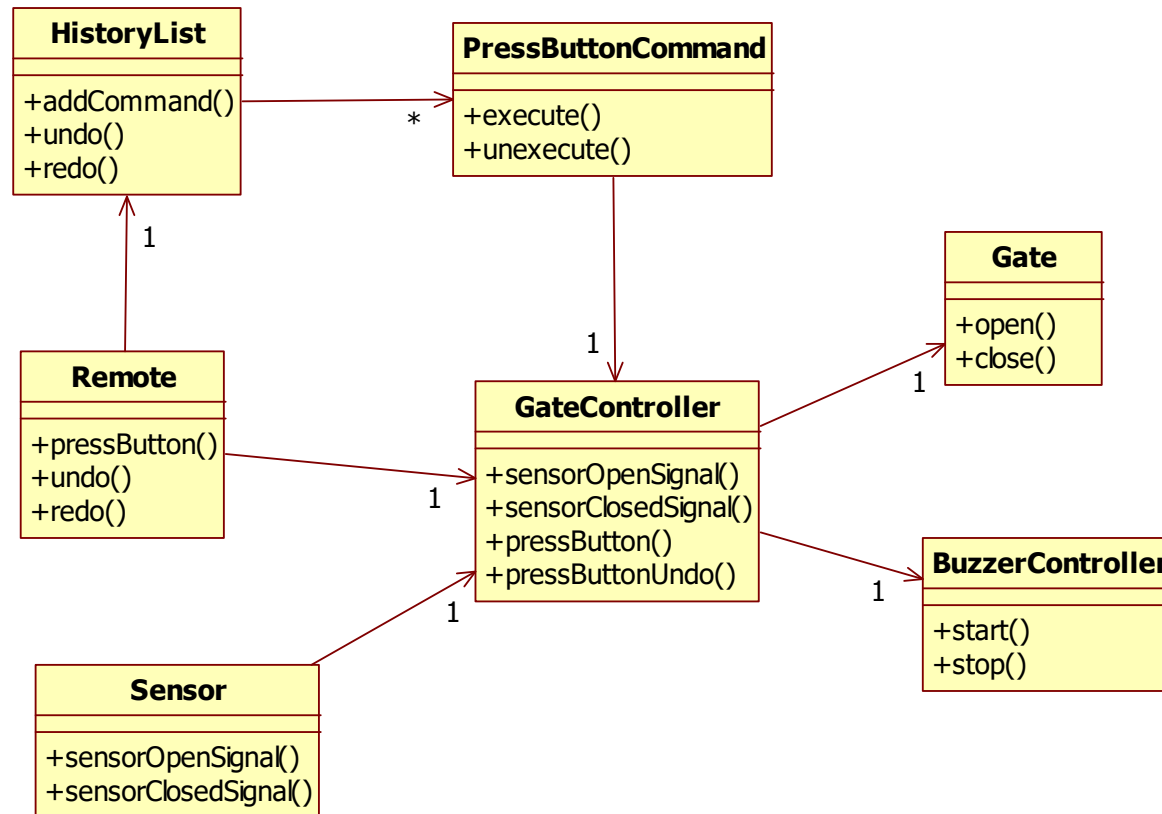


# GateController framework

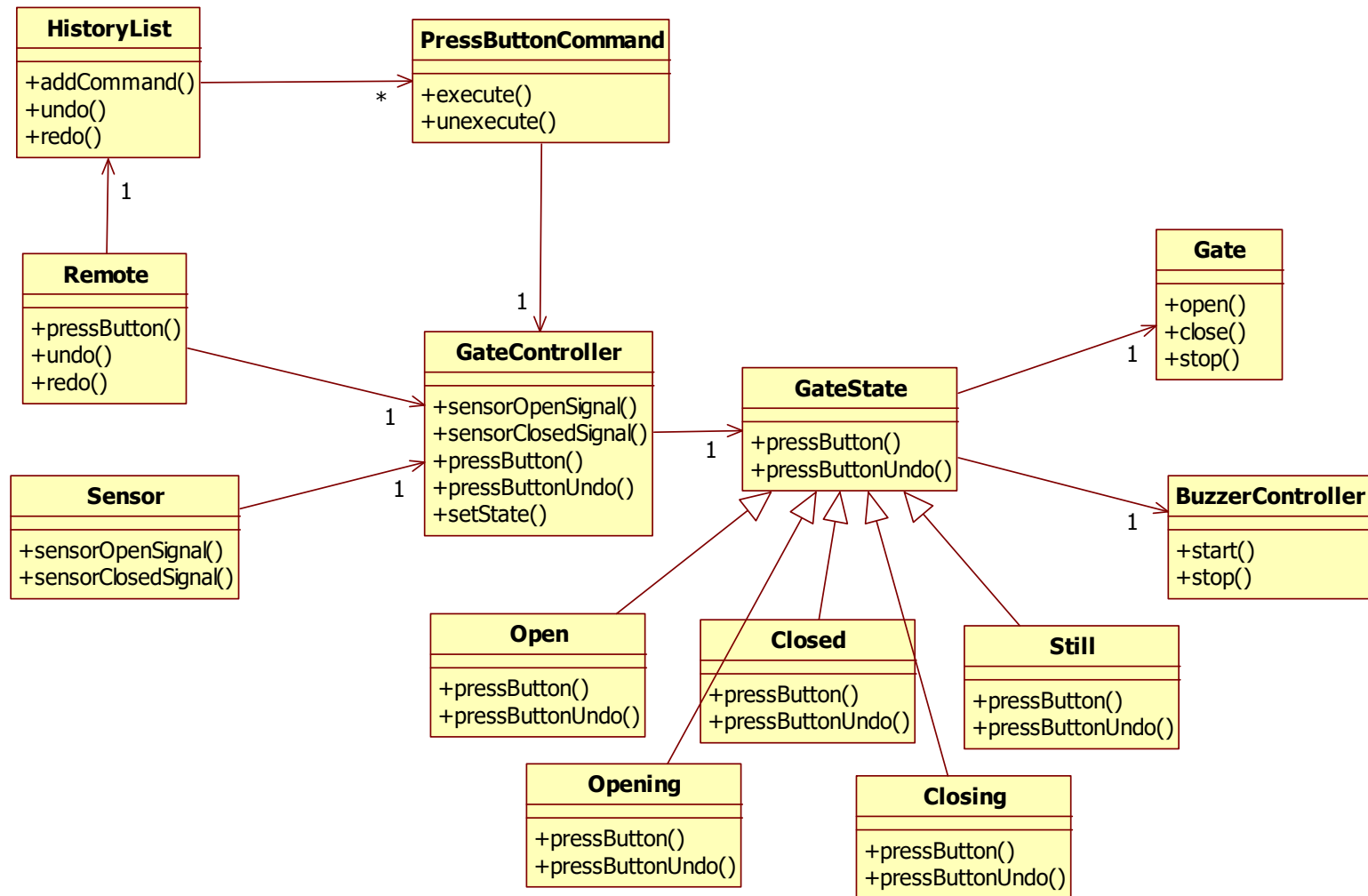
---

- Add undo/redo button
- Support different gate states (still, 75% open)
- Support multiple signaling devices (buzzers, lights, etc.)
- Support different gates

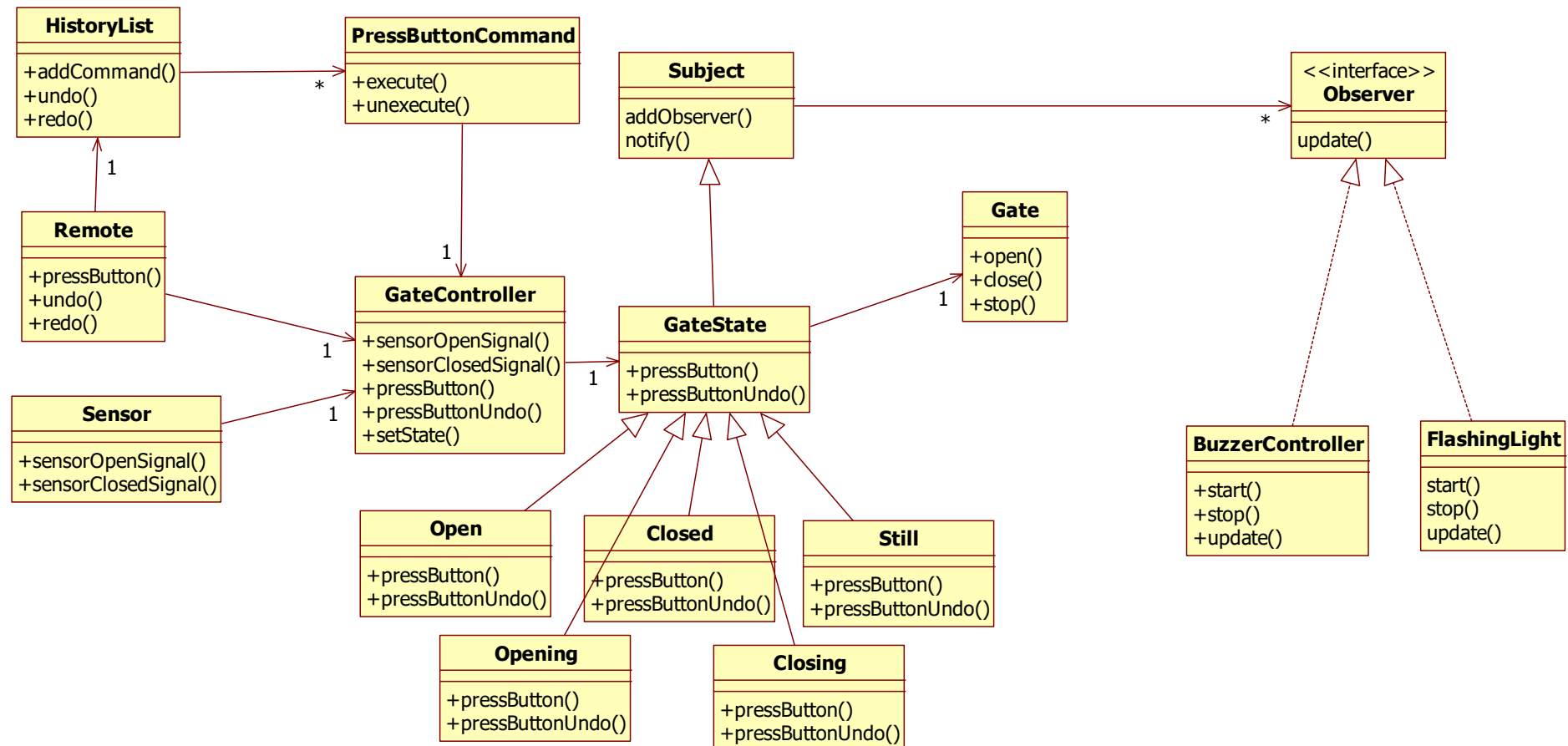
# Support undo/redo button



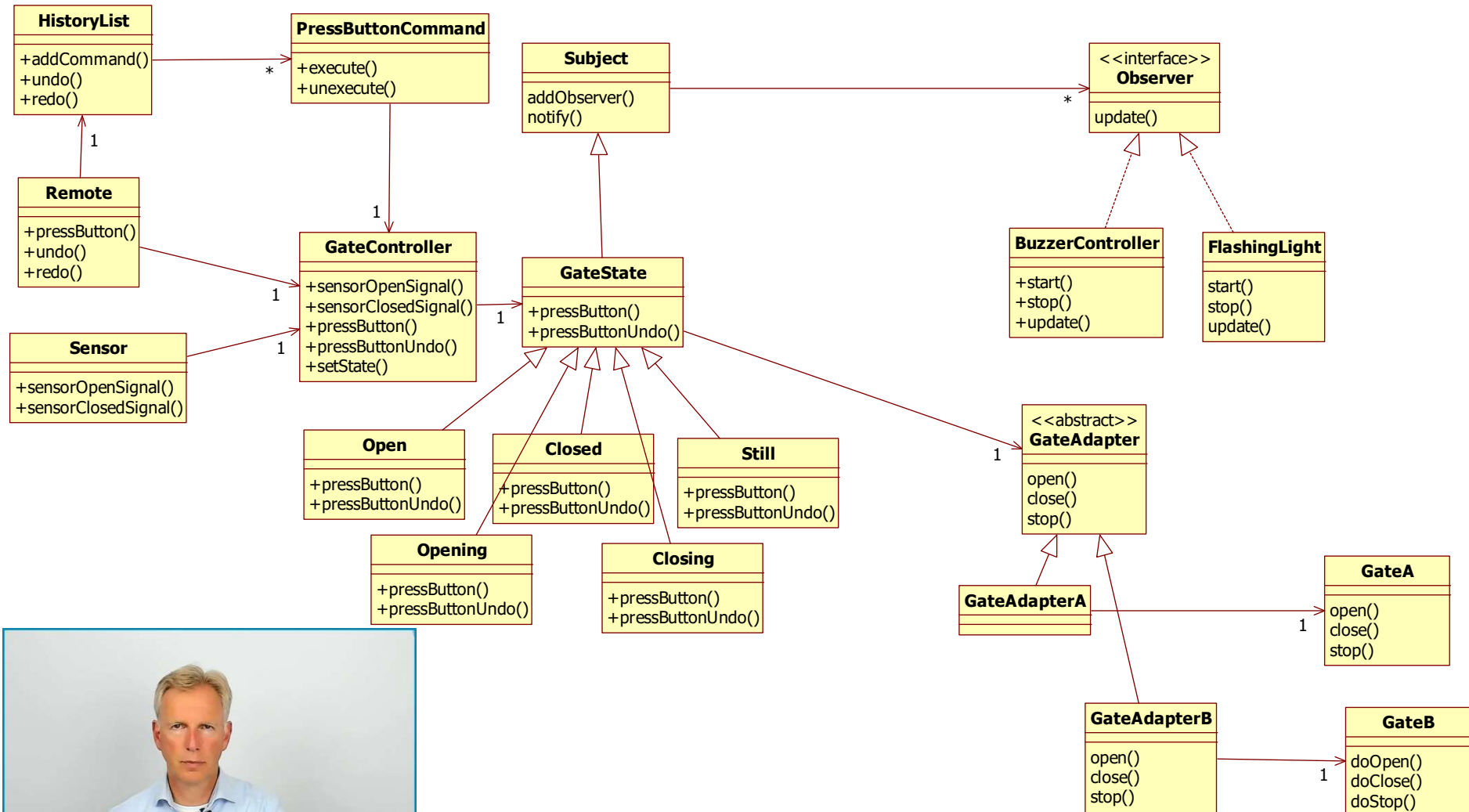
# Support different gate states (still, 75% open, half open)



# Support multiple signaling devices (buzzers, lights, etc.)



# Support different gates



# Main point

---

- A Framework captures domain specific expertise in abstract and concrete classes.
- The Unified Field which is the home of all the laws of nature, captures the intelligence of the whole universe.

# Connecting the parts of knowledge with the wholeness of knowledge

---

1. Frameworks embody expertise: this frees developers who are not necessarily experts in a certain area from the complexity of the underlying details.
  2. Frameworks are based on patterns. These patterns create the plugin points for the framework.
- 
3. **Transcendental consciousness** is the home of all the Laws of Nature which govern the entire universe.
  4. **Wholeness moving within itself:** In unity consciousness one spontaneously perceives the eternally silent, fully awake field of Pure Consciousness in the midst of all diversity.

