COMP-SCI 5551 (FS16) - Advanced Software Engineering

Active Learning

Observer Design Pattern

by

Project Team 1
Joshua Neustrom (39)
Yunlong Liu (25)
Chen Wang (58)
Dayu Wang (59)

- **Introduction**

  Definition

  Relationship

  Illustration

  Code Example

  Input/Output of Code

  Code Demo

  Questions

- **Definition**

  Behavioral Design Pattern

  One ➔ Many

  Subject (One Keeper) Change ➔ Observers (Many)

  Encapsulation (e.g. MVC)

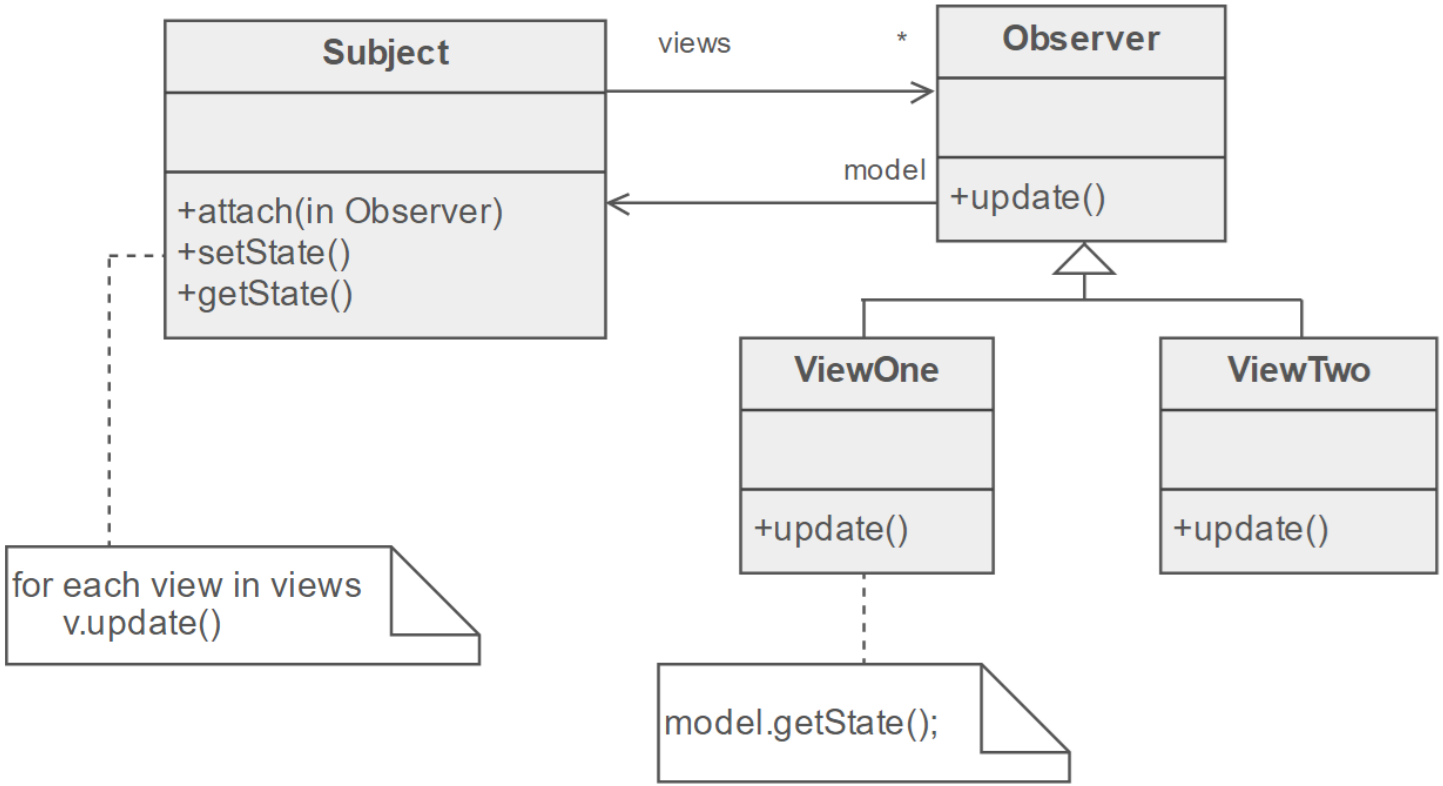
- **Comparison**

  Differences in the following send/receiver divisions

  Observer (Many at Runtime)
  Chain of Responsibility (Chain)
  Command (Subclass)
  Mediator (Indirect, Encapsulate Others' Communications)

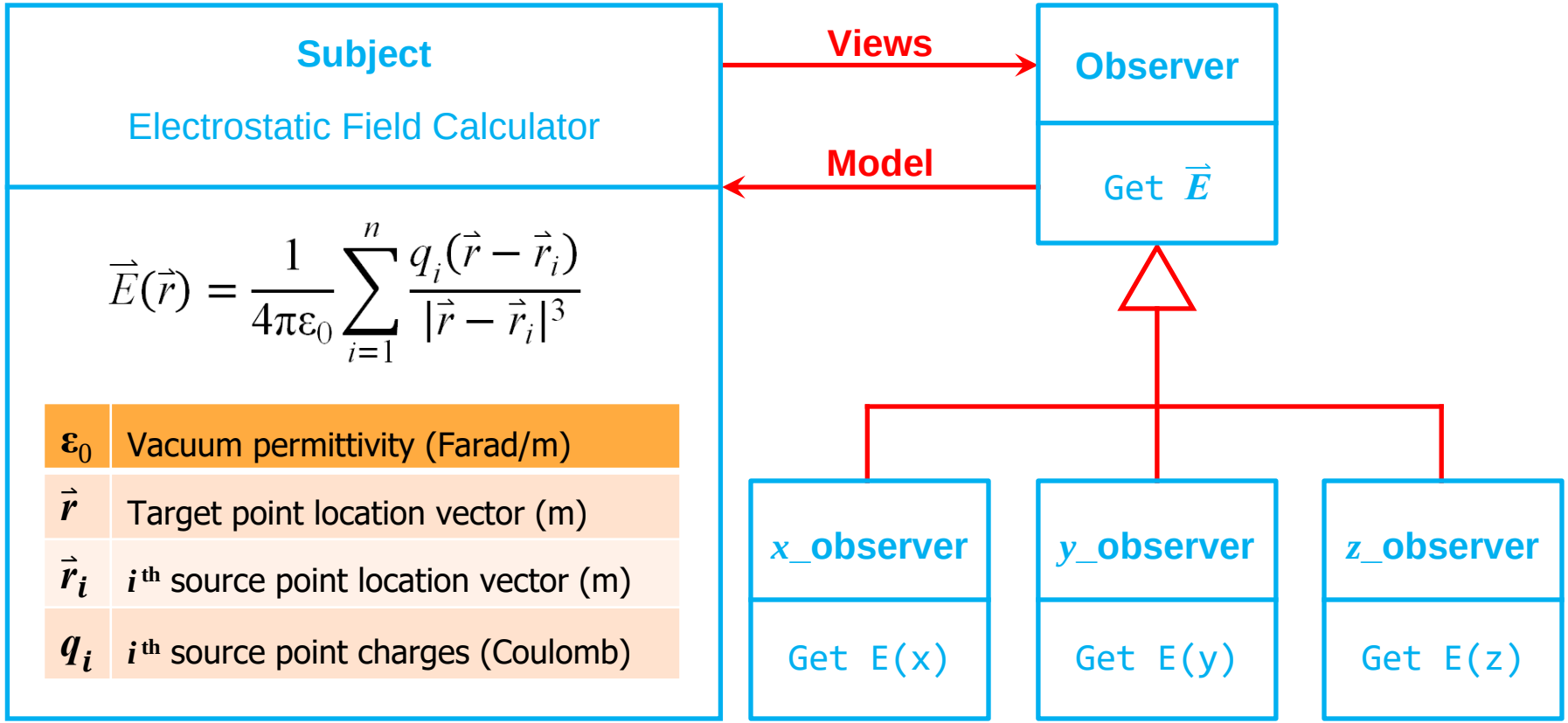
  Mediator can use Observer.

- **Conceptual UML Diagram**



Reference: https://sourcemaking.com/design_patterns/observer

• Code Example - Calculation of Electrostatic Field in Cartesian System



- **Code Example - Subject**

1. Monitor change of user input.
2. Update the calculated electrostatic field.
3. Update the observers

```
class E_Field {
    constructor(source, target) {
        this.source = source; // List of source point charges
        // Target point
        this.target = {
            x: inchToMeter(target.x),
            y: inchToMeter(target.y),
            z: inchToMeter(target.z)
        };
    }
    get target_E_field() { return this.calculate(); }
    calculate() {
        var addVector = { x: 0, y: 0, z: 0 };
        for (var node = this.source.head; node !== null; node = node.next) {
            var sourceVector = {
                x: inchToMeter(node.data.x),
                y: inchToMeter(node.data.y),
                z: inchToMeter(node.data.z)
            };
            addVector = vectorPlus(
                addVector,
                vectorScalarMultiplication(
                    vectorMinus(this.target, sourceVector),
                    eToCoulomb(parseFloat(node.data.c)) / Math.pow(
                        magnitude(vectorMinus(this.target, sourceVector)),
                        3
                    )
                )
            );
        }
        var resultVector = vectorScalarMultiplication(
            addVector,
            (1 / (4 * PI * e0))
        );
        return resultVector;
    }
}
```

- **Code Example - Observer**

```
class Observer {
    constructor(electricField) {
        this.E = electricField;
    }
}
```

```
class xObserver extends Observer {
    constructor(electricField) {
        super(electricField);
        this.x = electricField.x;
    }

    get Value() {
        return this.x;
    }
}
```

```
class yObserver extends Observer {
    constructor(electricField) {
        super(electricField);
        this.y = electricField.y;
    }

    get Value() {
        return this.y;
    }
}
```

```
class zObserver extends Observer {
    constructor(electricField) {
        super(electricField);
        this.z = electricField.z;
    }

    get Value() {
        return this.z;
    }
}
```

- **Code Example**
  Input / Output

COMP-SCI 5551 (FS16) - Advanced Software Engineering

Active Learning - **Observer** Design Pattern

Team 1: *Joshua Neustrom (39), Yunlong Liu (25), Chen Wang (58), Dayu Wang (59)*

Code Example - Calculation of Electric Field in Cartesian Coordinate System

Add Point Charge | Set Target Point

x = inches y = inches z = inches ***Charge*** = unit: e Add Cancel

x = inches y = inches z = inches Set Cancel

Point Charges

(1, 1, 1) = 1 e Delete

(2, 2, 2) = 2 e Delete

(-3, 8, 0) = 27 e Delete

Electric Fields (Result in V/m)

E_x(2, 10, 10) = 2.0670640729767747e-7
E_y(2, 10, 10) = 1.1656295743703984e-7
E_z(2, 10, 10) = 4.4557733954111475e-7

- **Conclusion**

  Definition

  Relationship

  Illustration

  Code Example

  Input/Output of Code

  Code Demo

  Questions

- **Questions?**