

調停者模式 / 2020.01.19

DESIGN PATTERN 讀書會 / BOB CHANG

---

# MEDIATOR PATTERN

## 自我介紹

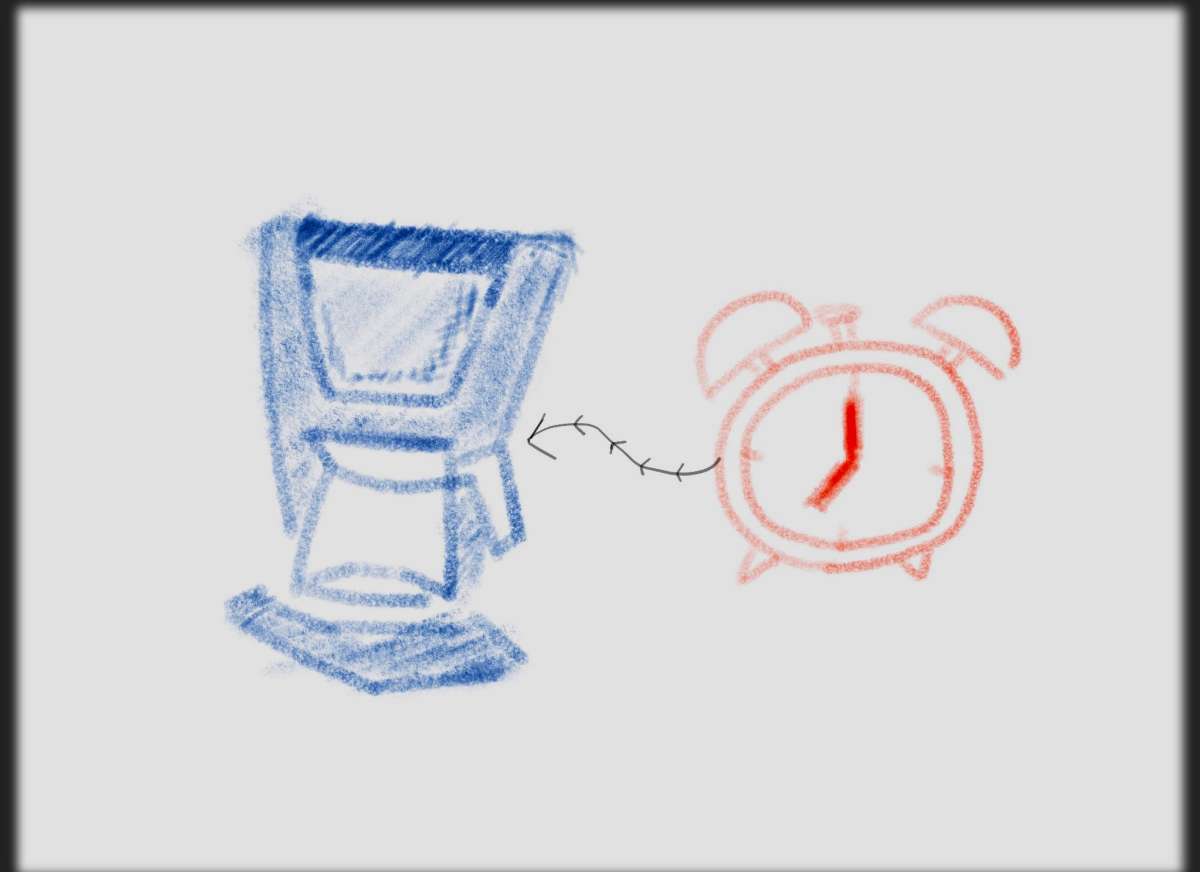
- ▶ 張家銘 Bob Chang
- ▶ 國立宜蘭大學電子系學士畢業
- ▶ iOS app 開發自學者
- ▶ 2015 July 開始學習 `Swift`
- ▶ twitter @bob910078
- ▶ bob910078@gmail.com



——一個需求小故事

## 客戶需求：全自動咖啡機

- ▶ 早上鬧鐘響的時候，咖啡機自動啟動，開始煮咖啡
- ▶ 刷牙洗臉完後，剛好就有現煮好的咖啡喝
- ▶ 我們可能會怎麼做？

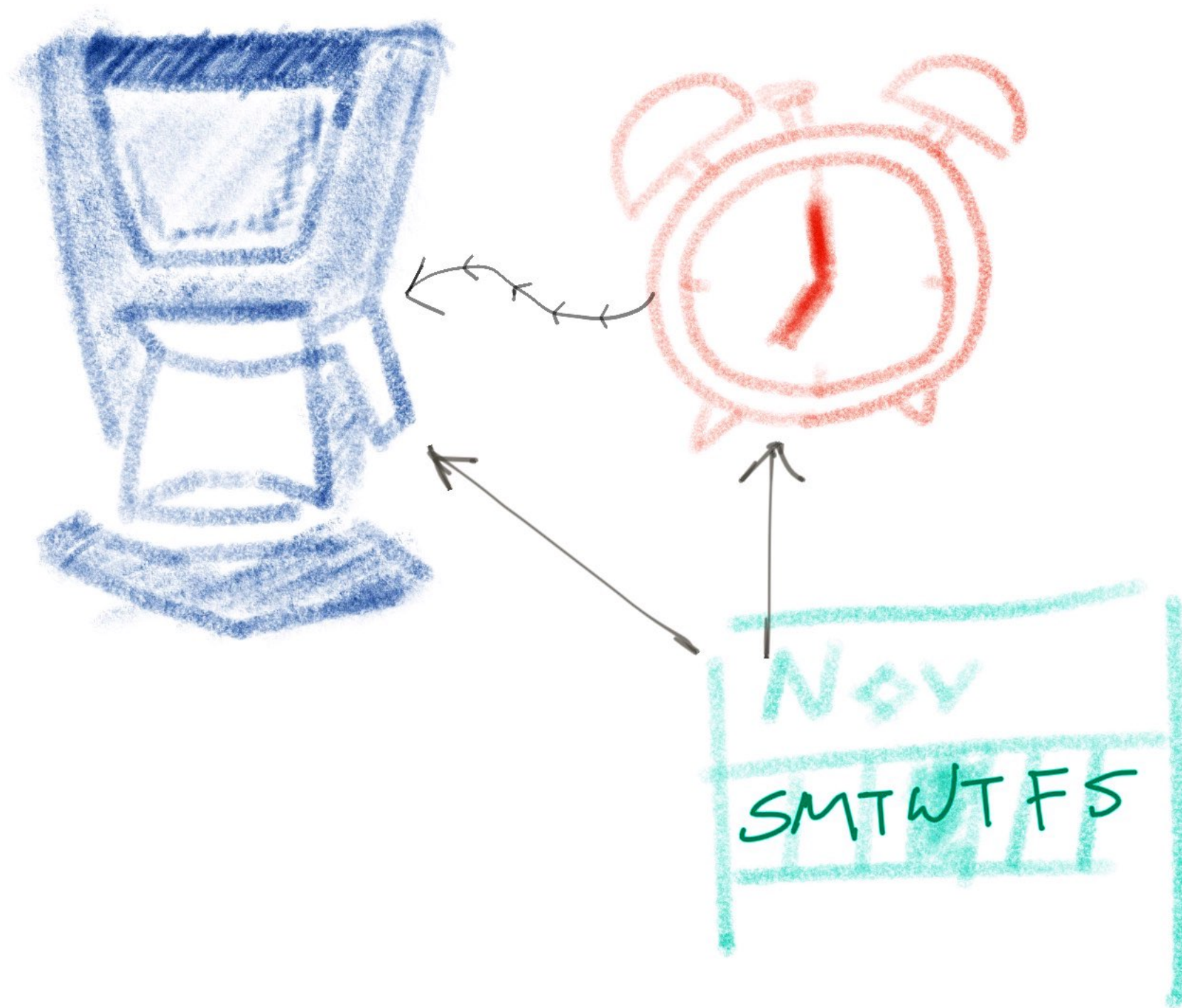


## 今天客戶心血來潮想要改變需求

- ▶ 週末要出去玩，不需要供應咖啡
- ▶ 週三要提早15分鐘出門，要提前供應咖啡
- ▶ 聖誕節和元旦要睡到自然醒，不要供應咖啡

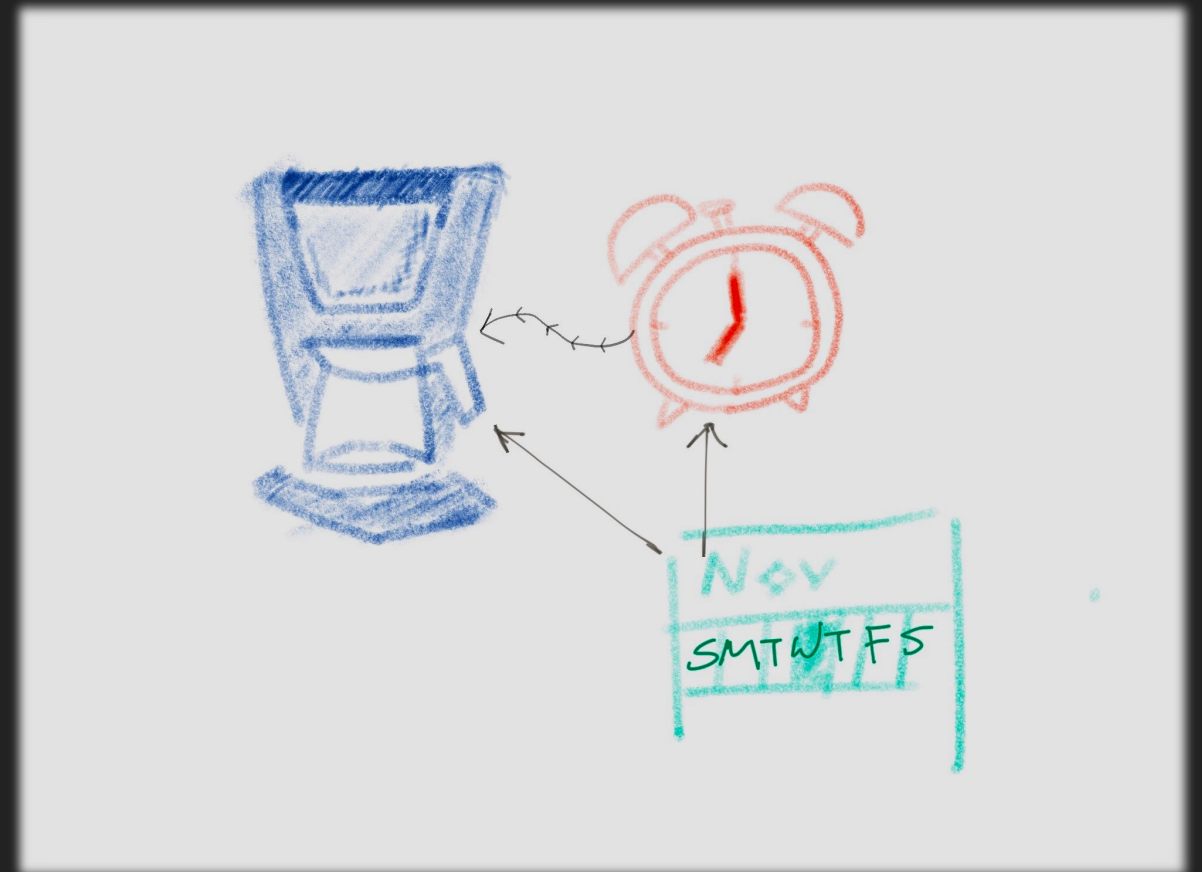


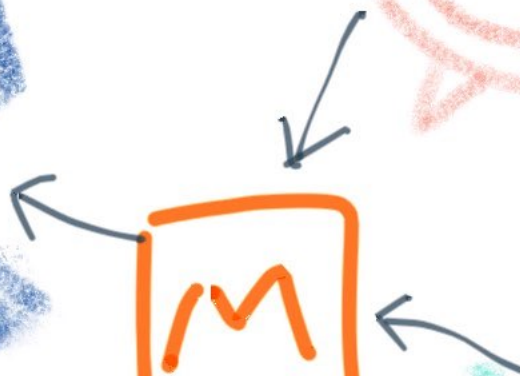




## 需求是達成了，但是...

- ▶ 咖啡機要知道週末到了，即便鬧鐘響也不用煮咖啡
- ▶ 鬧鐘要知道月曆週三的時候，需要提早15分鐘啟動咖啡機
- ▶ 月曆要通知咖啡機萬聖節和聖誕節不需要煮咖啡
- ▶ ...恩，應該沒漏掉規則？

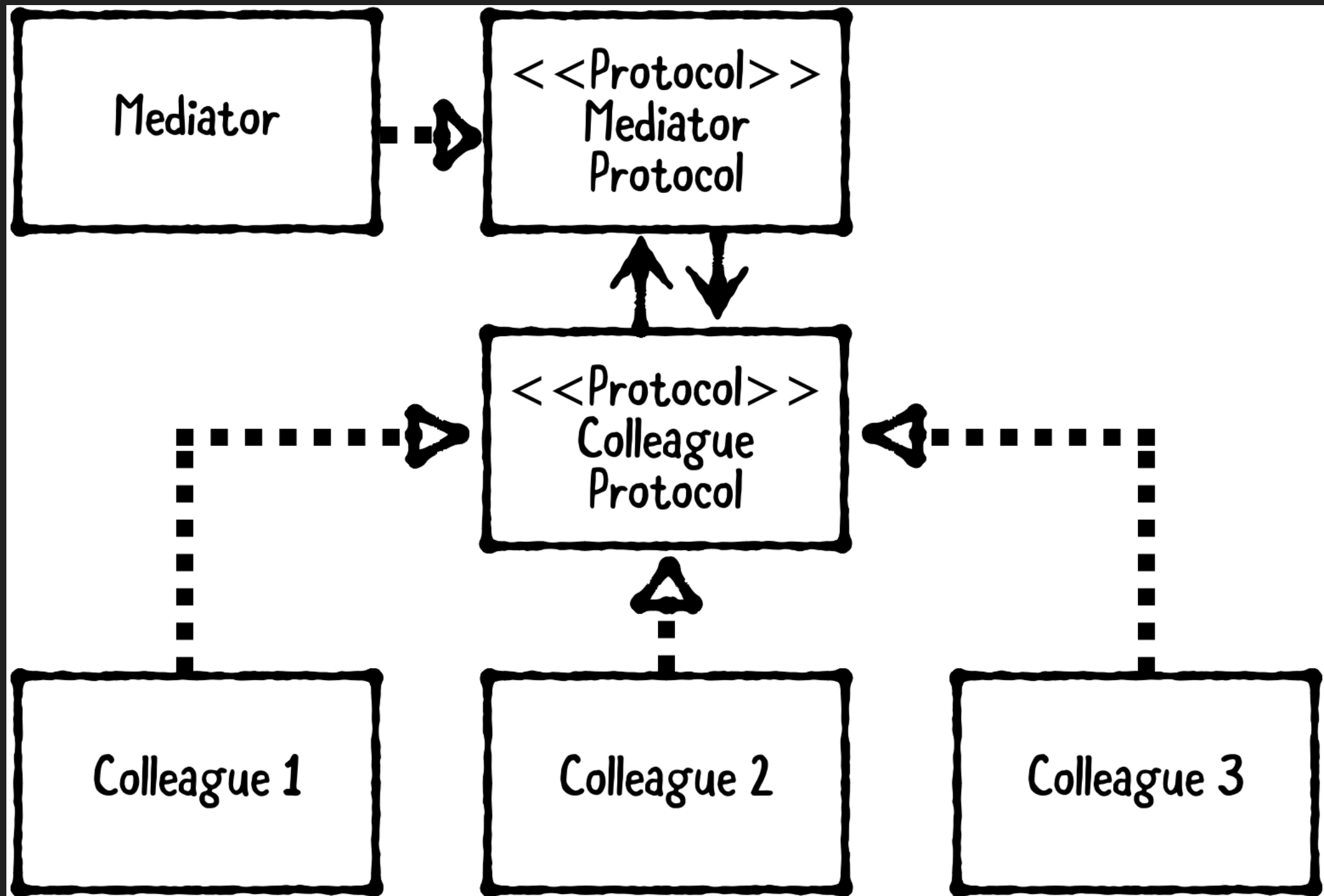






探討該解決方案  
怎麼做到的？

以圖片的方式抽象解釋，類別關係圖？



## 優缺點一眼即明

### ▶ 優點：

- ▶ 彼此從此不相往來（類別又可以各司其職了）
- ▶ 調停者是上帝（事情交給他就沒事了，他會去處理）

### ▶ 缺點：

- ▶ 當調停案件數量一大，上帝會累得半死

各單元對其他單元所知  
應當有限：只瞭解與目  
前單元最相關之單元。

最少知識原則

<Low of Demeter: Principle of Least Knowledge>



## 以此模式處理問題對於軟體後續的維護性討論

### ▶ 套用調停者模式前後

- ▶ 各元件的職責？重複利用率？可測試性？

### ▶ 套用調停者模式後

- ▶ 調停者的可讀性？

## 本集重點回顧

- ▶ 鬧鐘觸發全自動咖啡機案例
- ▶ 多種變因去觸發咖啡機作動的條件
- ▶ 咖啡機外部物件觸發的職責回歸
- ▶ 有了小幫手負責調停咖啡機啟動規則
- ▶ 未來咖啡機啟動規則的條件陡增，小幫手該怎麼辦
- ▶ 最少知識原則：只了解與目前單元最相關的單元

## REFERENCE

- ▶ [LawOfDemeter](<http://www.ccs.neu.edu/home/lieber/LoD.html>)
- ▶ [HeadFirstDesignPatterns](<https://www.oreilly.com/library/view/head-first-design/0596007124/>)
- ▶ [大話設計模式](<https://www.tenlong.com.tw/products/9789866761799>)
- ▶ [Fullmetal Alchemist OST](<https://www.youtube.com/playlist?list=PLAC7C6367B6B22EC4>)

結束