UI 아키텍처 사

~ MVC에서 MVVM으로 ~

http://yohshiy.blog.fc2.com/blog-entry-215.html 를 번역

최흥배
https://github.com/jacking75/choiHeungbae

목차

MVC이전

UI 아키텍처의 대 전제

UI 아키텍처의 대 전제

MVC 이후는 애플리케이션의 **코어 부분과 UI 부분을 분리 하는** 것이 목적

멋지게 말하면、
비지니스 로직과 프리젠테이션 로직의 분리

MVC 이전에서는 이것들을 하나의 오브젝트로 모으는 경향이 있었다.

UI 아키텍처 대 전제 2

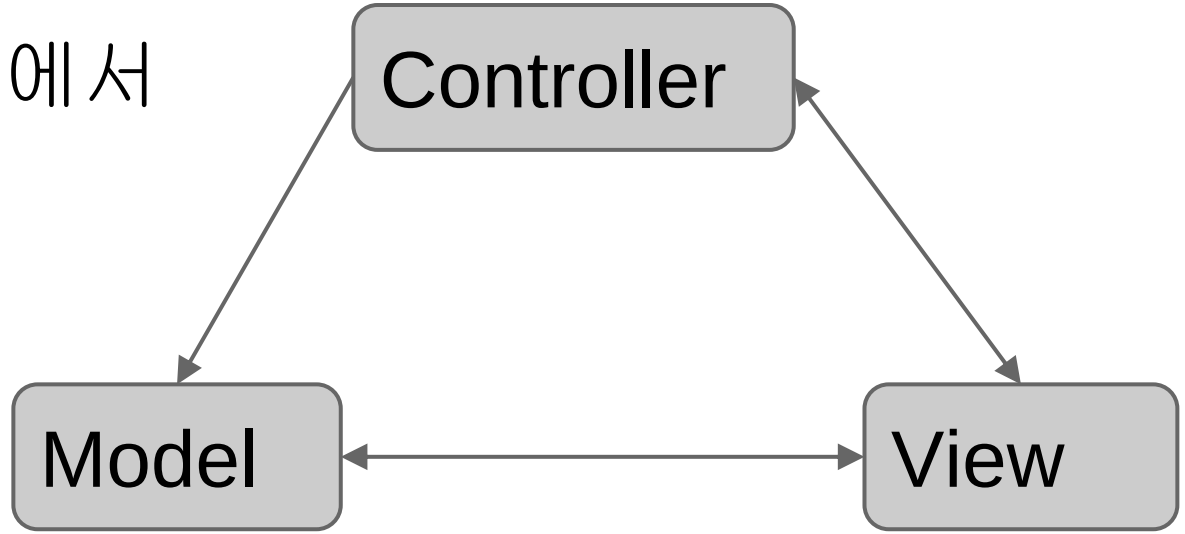
『은 탄환은 없다』

완벽한 UI 아키텍처도 없다
그래도 보다 좋은 것을 찾는 것을 멈추면 안된다

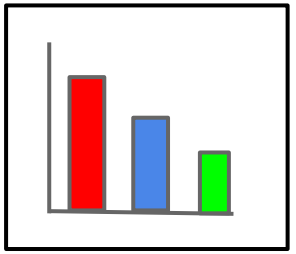
MVC와 확장MVC

MVC

GoF 책에서



a = 50%
b = 30%
c = 20%



Model	애플리케이션 오브젝트
View	화면 표현
Controller	유저 입력에 관한 인터페이스

Controller 라는 것은？

Model、View 은 이해하기 쉽다.

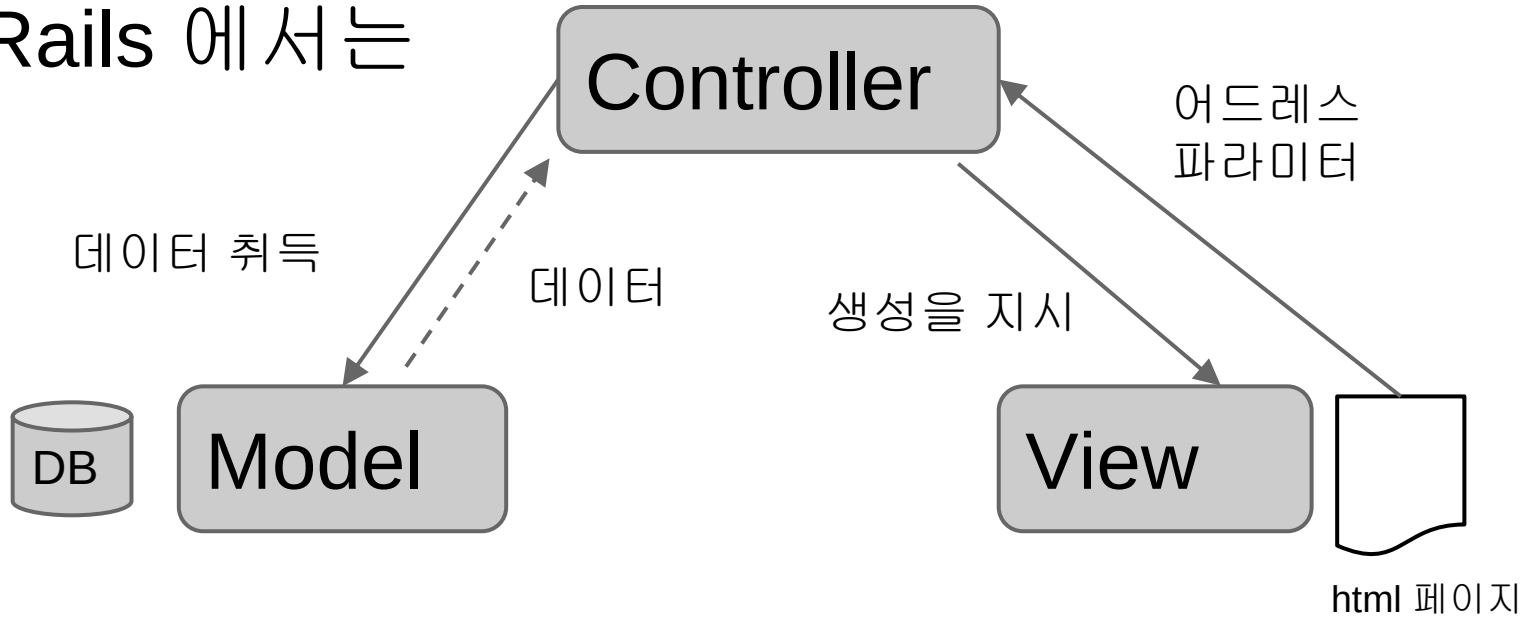
그러나 Controller 은 이해하기 어렵다.

이해하더라도 좀 더 좋은 방법이 있지 않을까?
라고 생각한다.

MVC는 사람에 따라서 해석이 다양한다

MVC (Web)

Rails 에서는

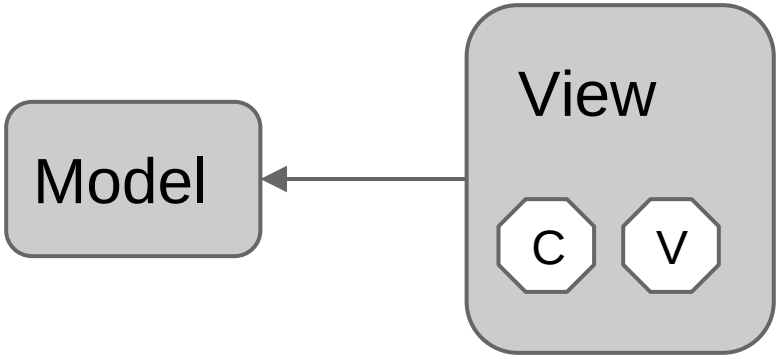


Model	DB에 액세스
View	html 페이지와 html 페이지 작성
Controler	입력을 받아들인다. 다른 모듈 조작

확장 MVC (ＰＣ애플리케이션)

GUI 부품(Widget, Control)이 고성능화

→ View 와 Controller의 분리가 불가능



그래서..　확장 MVC (줄여 지는 것이지만..)

코어, UI 분리의 기본 원칙만을 남김

- MFC : Document - View
- Qt　　: Model - View

문제점

PC : 확장 ＭＶＣ、 Web : MVC

잠시 안정되지만...

확장 MVC의 문제점

View가 너무 커짐

원래 크게 나누지 않음 → 어떻게 하든 나누어야 함

이를테면 Web도 View의 JavaScript 가 대규모화

UI 아키텍처 난립

문제점 해결을 위해서 여러가지 UI 아키텍처가 제안 되고 있음

- MVP - Model, View, Presenter
- PM - 프리젠테이션 모델
- 애플리케이션 모델

MVC의 확대 해석도 있고, 알기 어렵다

프레임워크가 없으면 정착할 수 없음

요망

디자인의 중요성이 커져간다

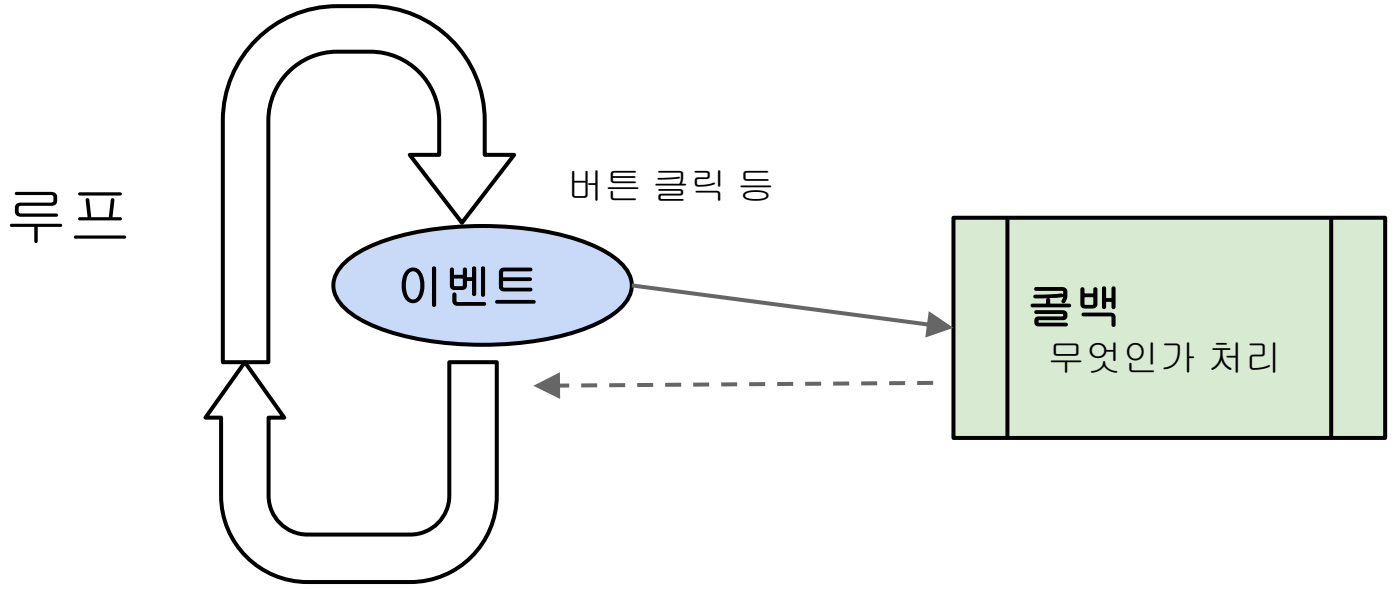
Web 페이지는 Web 디자이너에게
→ 화면 디자인도 디자이너에게

화면 레이아웃과 코드를 분리分離 하지 않으면
안 된다

화면 레이아웃, 코드 분리의 난점

이벤트 드리븐 프로그래밍

시그널(이벤트) → 콜백



버튼 등의 부품과 콜백 함수를 연결하지 않으면 안 된다

MVVM (.NET)

Model, View, ViewModel

.NET의 등장

1. Win32API과 MFC

   베이스는 C의 Win32API

   VC++에서는 MFC、 VB에서는 Form

1. Java의 대두

   크로스 플랫폼. Linux도 사용하기 쉬움.

   유저를 몽땅 가져감.

1. C#、 .NET으로 대항

   Form : 종래의 부품을 .NET용으로 랩핑

   아직 확장 MVC

WPF의 등장

.NET용으로 만들어진 GUI 툴킷

아직은 잘 사용되지 않고 있다

- Form 쪽이 부품 기능이 많고 풍부함
- 사용 방법이 어려움
- MS 제품에서도 사용하고 있지 않음

MVVM의 등장

WPF은 잘 사용되지 않음.

그러나 Form에서는 문제점과 요구를 해결할 수 없음.

- View의 거대화
- 화면 레이아웃과 코드의 분리

확장 MVC에 대한 WPF용의 MVVM 등장...

과 그 전에 채용되고 있는 기술을 소개

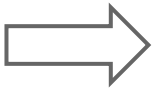
- 데이타 바인딩
- 코맨드 패턴

데이터 바인딩

DB 관련해서 발달한 기술(Model에서 사용)

DB랑 XML을 프로그램에서 사용하기 쉬운 구조체 (클래스)로 변환

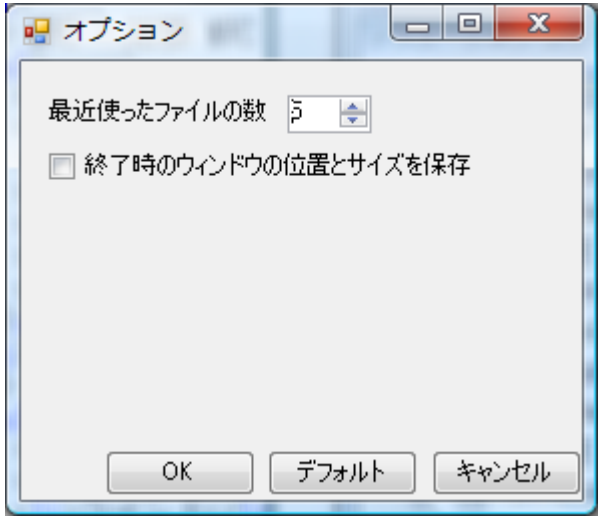
名前	年齢	身長	体重
山田太郎	28	170	58



```
Data {
    Name = "山田太郎"
    Age = 28
    Height = 170
    Weight = 58
}
```

설정 화면에서 자주 있는 패턴

화면의 내용과 구조체 (클래스)와 상호호환
기본 값이 있고, 모던한 다이얼로그



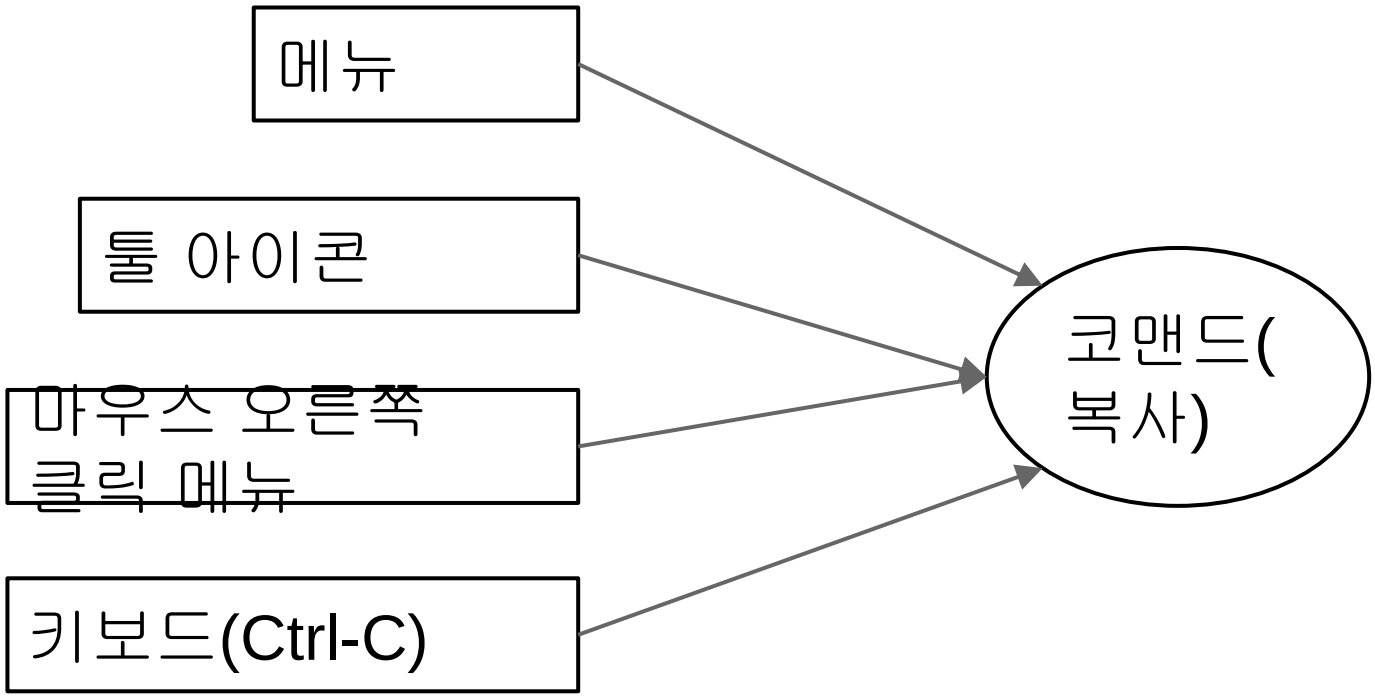
```
Options  {
    RecentFileNum = 5
    SavedSection = false
}
```

View용의 Model → ViewModel

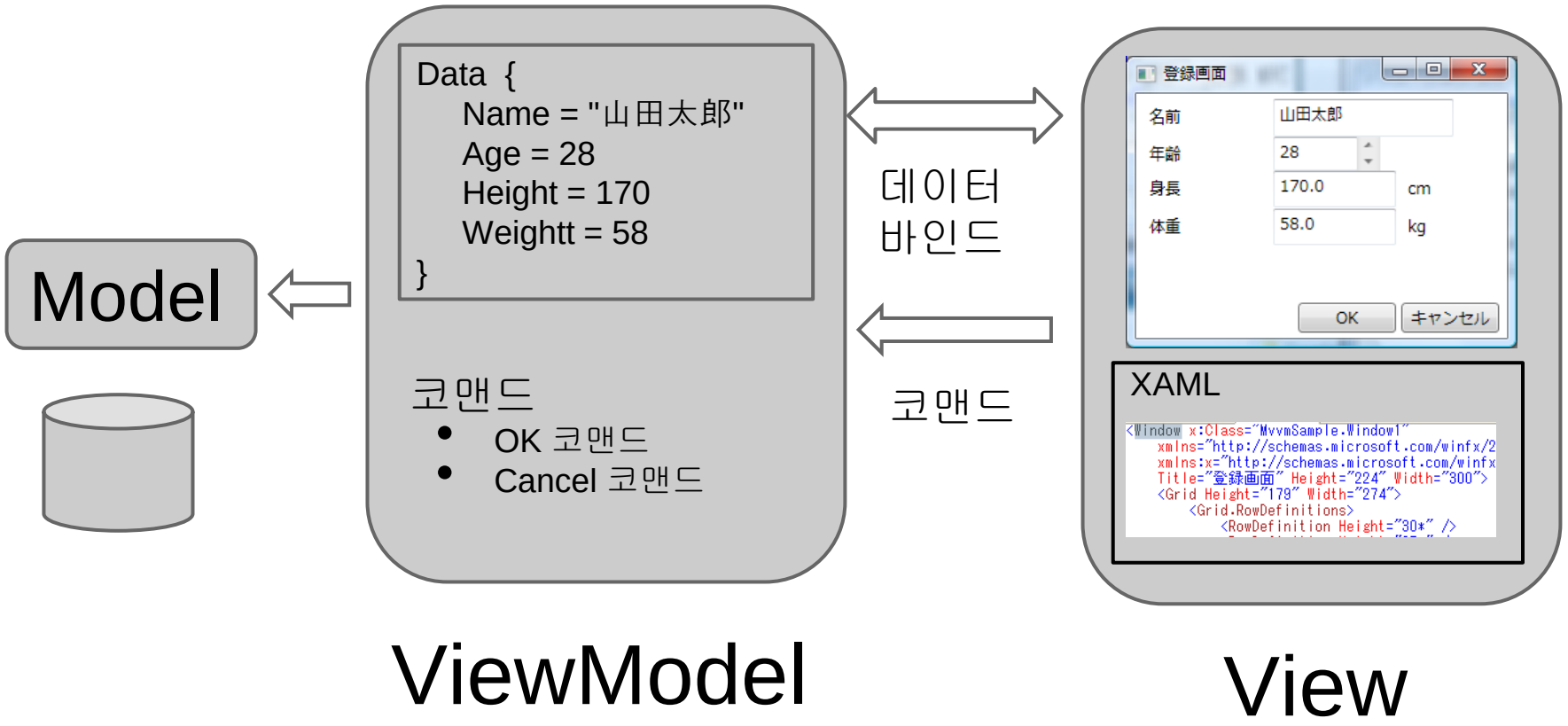
코맨드 패턴

요구를 오브젝트로 캡슐화

다양한 곳에서 호출하는 것에 대응하기 쉽다い



MVVM


Model
Data {
Name = "山田太郎"
Age = 28
Height = 170
Weightt = 58
}
코맨드
OK 코맨드
Cancel 코맨드
데이터
바인드
코맨드
登録画面
名前
山田太郎
年齢
28
身長
170.0
cm
体重
58.0
kg
OK
キャンセル
XAML
ViewModel
View

화면 레이아웃과 코드의 분리

View

XAML : XML으로 레이아웃 정의

ViewModel

데이터 : 쌍방향 바인딩으로 동기

대부분 이런식으로 한다

코맨드 : 필요 최소한의 이벤트 드리븐

View의 분리

View와 ViewModel 에 의한 분리로 거대화를 경감

코맨드 정도라면 MVC 로 들어간다.

데이터 바인드가 MVVM의 특징.

역으로 데이터 바인드 기구가 없으면 구현이 어려움.

WPF용의 UI 아키텍처.

MVVM 구현

단 ＷＰＦ는 MVVM 프레임워크가 아니다

- WPF 로 만들면 MVVM 로 되는 것은 아니다
- MVVM을 구현할 때는 매회 같은 코드를 쓸 필요가 있다

그래서 라이브러리를 사용

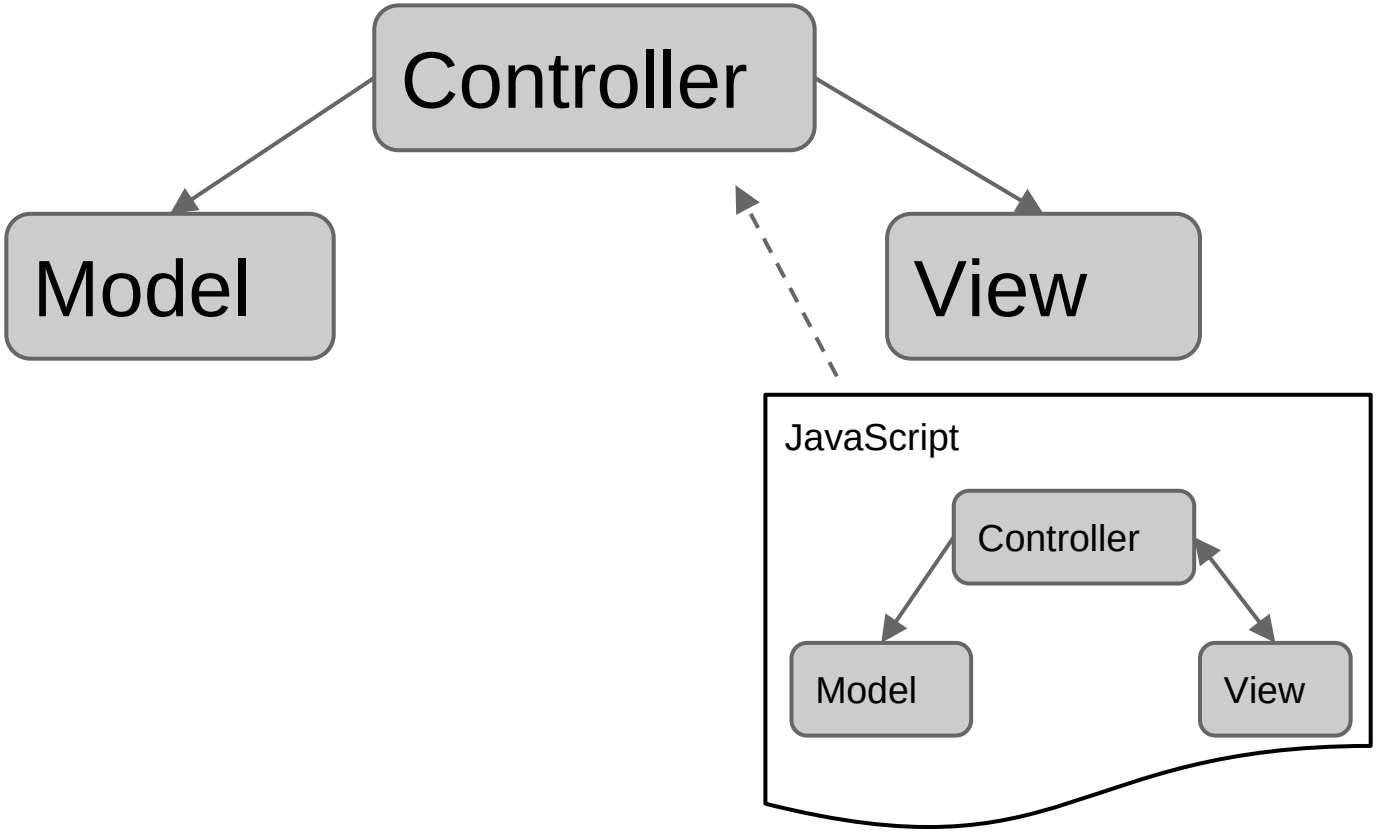
- Prism http://compositewpf.codeplex.com/
- MVVM Light Toolkit

.NET 이외의 경향

Web、모바일、 .NET 이외의 PC

Web 애플리케이션

JavaScript 에서도 MVC 프레임워크를 사용



모바일

스마트폰, 타블렛에 의한 제3 섹터 출현

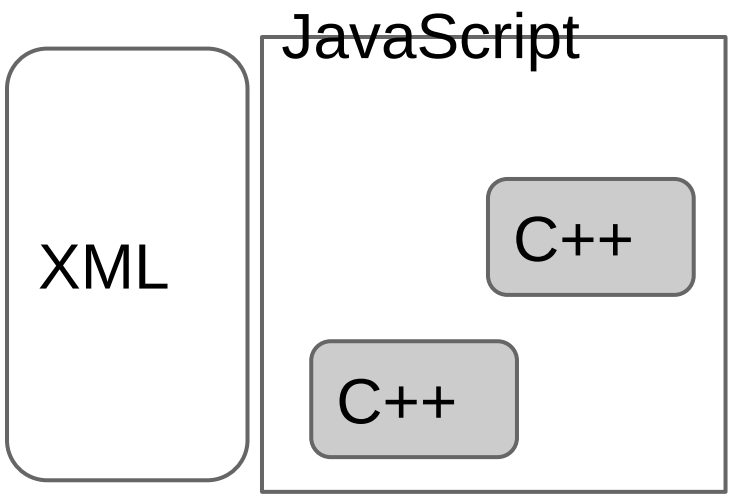
보통 만드는 경우

iOS	Objective-C
Android	Java

당연 같은 코드로 만들고 싶다

HTML5 + JavaScript

.NET 이외의 PC 애플리케이션



화면 레이아웃	XML (XUL나 QML등) 복잡해서 HTML으로는 어려움
GUI 부품(Widget)	GUI 툴킷을 제공(C++)
애플리케이션 전체	JavaScript
코어 부분	C++

GNOME에서는 GUI 애플리케이션에서 JavaScript 사용

今後どうなる？

ＨＴＭＬ5 + JavaScript

　Web 、モバイルとの統一化の傾向

Windows Phone は

　SilverLight(PC用とは別) → MVVM

Windows ではPC、モバイルとの統一化の傾向

PC、Web、モバイルで統一化なるか？