ASP.NET MVC: non solo «tennologgia»

Andrea Saltarello

(Solution) Architect @ Managed Designs S.r.l. – http://www.manageddesigns.it

andysal@gmail.com

http://blogs.ugidotnet.org/pape

http://twitter.com/andysal74

Innanzitutto…

Grazie,  DomusDotNet !

User Group romano per lo studio e la divulgazione delle tecnologie .NET

☺

Agenda

- #ifdef NIUBBO MVC 1-2-3
- Controller->View
- View->Controller
- AspNetMvc.More();

Model View Controller

Formulato nel 1979 da **Trygve Reenskaug**, membro del team Smalltalk @ **Xerox PARC**

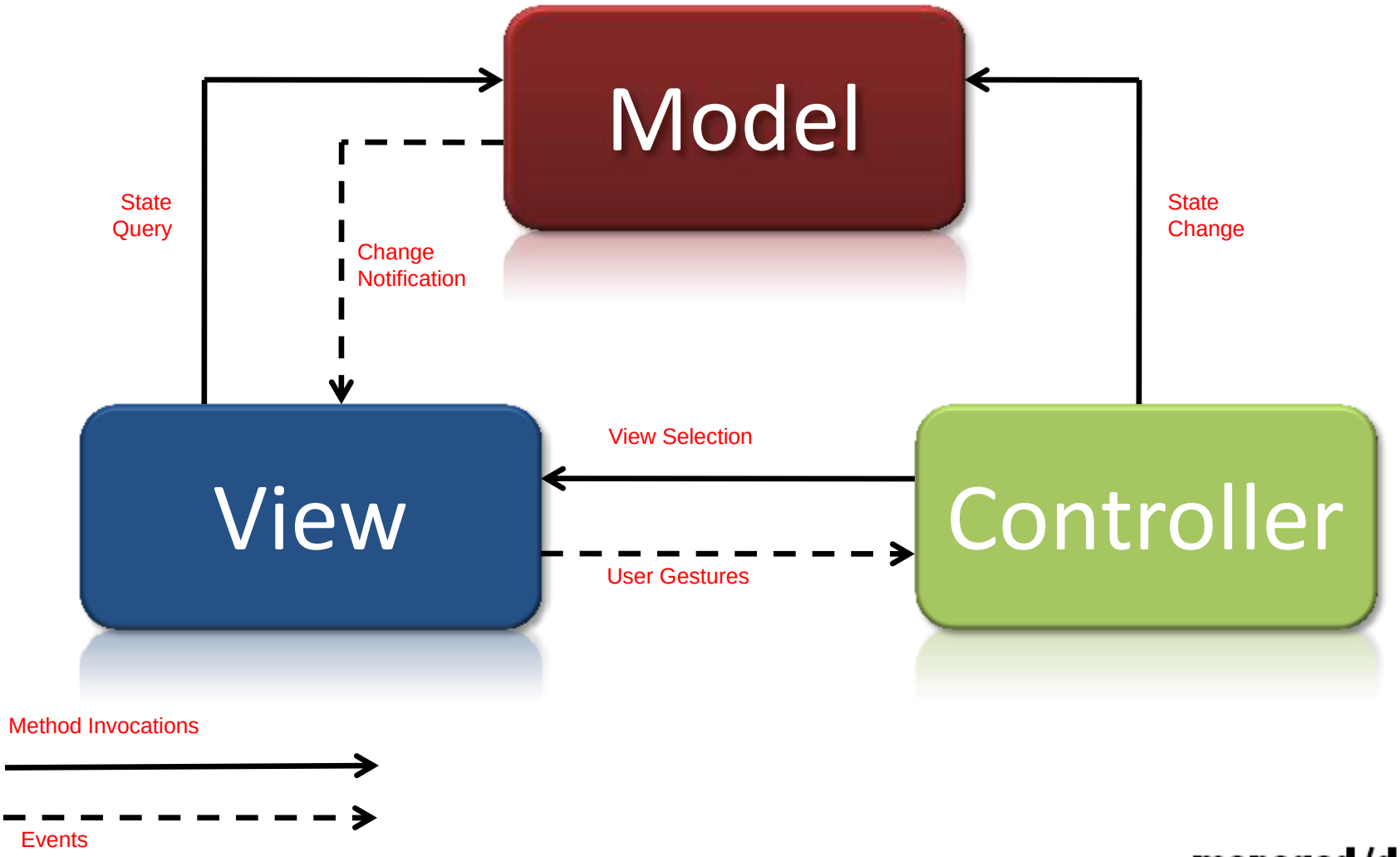
La prima motivazione di design è:

Separare gli aspetti di presentazione da quelli di dominio.

E' la “radice concettuale” di tutti gli attuali pattern:

Model View Presenter, Presenter-First, Passive View, Presentation Abstraction Control, ~~Presentation Model~~ Model View ViewModel...

Model view controller



MVC: falsi miti

- Lo scopo del **Controller** non è di separare la **View** dal **Model**.
- La responsabilità del **Controller** è di fare da mediatore tra l'utente e l'applicazione, non tra la **View** e il **Model**.
- Spesso si parla di **MVC**, ma si intende **Model 2**

Model 2

- *In a Model 2 application, requests from the client browser are passed to the **controller**, which is a **servlet**. The controller decides which view (**JSP**) it will pass the request to. The view then invokes methods in a **JavaBean** (which may access a database) and returns the **Response object** to the Web container, which is then passed on to the client browser.* [Wikipedia]
- **Legenda:**
- Servlet->HttpHandler->Front Controller [P of EAA, 344]
- JSP->Controller->Page Controller [P of EAA, 333]

Introduzione a ASP.NET MVC (1/4)

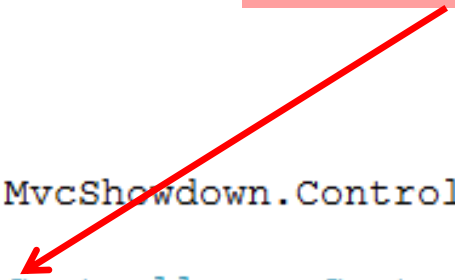
http://host/webapp/Customer/Detail/1

```
namespace ManagedDesigns.MvcShowdown.Controllers
{
    public class CustomerController : Controller
    {
        public ActionResult Index()
        {
            throw new NotImplementedException();
        }

        public ActionResult Detail(int id)
        {
            throw new NotImplementedException();
        }
    }
}
```

Introduzione a ASP.NET MVC (2/4)

http://host/webapp/Customer/Detail/1



```
namespace ManagedDesigns.MvcShowdown.Controllers
{
    public class CustomerController : Controller
    {
        public ActionResult Index()
        {
            throw new NotImplementedException();
        }

        public ActionResult Detail(int id)
        {
            throw new NotImplementedException();
        }
    }
}
```

Introduzione a ASP.NET MVC (3/4)

http://host/webapp/Customer/Detail/1

```
namespace ManagedDesigns.MvcShowdown.Controllers
{
    public class CustomerController : Controller
    {
        public ActionResult Index()
        {
            throw new NotImplementedException();
        }

        public ActionResult Detail(int id)
        {
            throw new NotImplementedException();
        }
    }
}
```

Introduzione a ASP.NET MVC (4/4)

http://host/webapp/Customer/Detail/1

```
namespace ManagedDesigns.MvcShowdown.Controllers
{
    public class CustomerController : Controller
    {
        public ActionResult Index()
        {
            throw new NotImplementedException();
        }

        public ActionResult Detail(int id)
        {
            throw new NotImplementedException();
        }
    }
}
```

Navigation Flow

Le *action* restituiscono un valore di tipo **ActionResult**. "*Pragmapolimorficamente*" parlando:

- **ViewResult**. Restituito dal metodo **View**.
- **PartialViewResult**. Restituito dal metodo **PartialView**.
- **RedirectToRouteResult**. Restituito dai metodi **RedirectToAction** e **RedirectToRoute**.
- **RedirectResult**. Restituito dal metodo **Redirect**.
- **ContentResult**. Restituito dal metodo **Content**.
- **JsonResult**. Restituito dal metodo **Json**.
- **EmptyResult**. Restituito dalle action che vogliano restituire "*null*".
- ***YourOwnPersonalResult*** (semi-cit ☺), restituito da una factory ad hoc

ASP.Net MVC

demo

Dati: Controller->View

Per trasferire dati dal controller alla view:

- ~~**ViewData**~~
- *YourView*.**Model =** ~~Presentation~~ View Model

ASP.Net MVC

demo

Dati: View->Controller

Per trasferire dati dalla view al controller:

- Parametri della action (*convention over configuration*):
  - ogni elemento in query string è mappato su un parametro omonimo
  - ogni elemento del <form /> è mappato su un parametro omonimo
- FormCollection
- **>>>Model binding<<<**

ASP.Net MVC

demo

Data Entry

Built in:

- Usare i metodi helper per definire i “campi”
- *Taggare* il ViewModel mediante **DataAnnotations**
- Ricordarsi **Html.ValidationMessageFor<>()** e/o **Html.ValidationSummary()**
- [Opzionale]Impostare **.input-validation-error** nel css
- [Opzionale]: EditorTemplates, DisplayTemplates

ASP.Net MVC

demo

Data Entry <3 JQuery

- Client side validation
  - Inclusione degli script JS
  - <% Html.EnableClientSideValidation(); %>
- JQuery plug-in, es:
  - JQuery UI (tab, datepicker, …)
  - JQGrid
- Assert.IsTrueLove(EditorTemplates, JQuery)

ASP.Net MVC

demo

AspNetMvc.More();

Esempi:

- [FX4 only]<%: … %> (equivalente a <% =Html.Encode(…) %>)
- IoC
- Custom ActionResult:
  - Generare Thumbnail
  - Generare Atom/RSS (http://www.ugidotnet.org/Article/Detail/280)
- “Componentization”: RenderPartial+RenderAction
- Test
- Aree

ASP.Net MVC

demo

MVC @ Managed Designs

In azienda usiamo il toolkit MVC dalle prime CTP della v1, ed abbiamo raggiunto una struttura «standardizzata» dei progetti:

- Model 3
- Layered Expression Trees

MVC goes Model 3

- Model 2 separa il Controller in:
  - Front Controller
  - Page Controller
- Model 3 separa il Model in:
  - View Model: rappresenta i dati che la view si impegna a presentare all'utente
  - Worker Service: è la façade che il page controller utilizza per produrre il View Model

E' il Single Responsibility Principle, baby!

Never mind the bollocks, here's the Model 3

demo

Layered Expression Trees (LET idiom)

Facciamo un gioco: invece di definire un «botto» di DTO, facciamo che layer e servizi si scambino degli **IQueryable<*YourFavouriteDomainEntity*>**, facendo «emergere» la query e specificando la proiezione solo all'ultimo momento?

L'espressione «Capra e cavoli» vi dice niente? ☺

C’mon Query LET’s go party (ah-ah-ah, yeah!)

demo

Slide e demo

- Slide sul mio blog: http://blogs.ugidotnet.org/pape
- Demo su CodePlex: http://nsk.codeplex.com