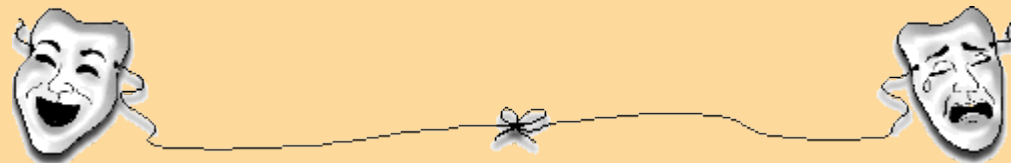


Introduzione a Catalyst

IPW 2009

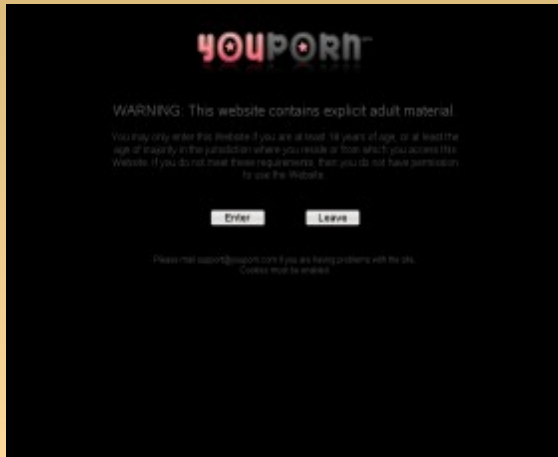
Pisa, 23 e 24 ottobre 2009



Emanuele Zeppieri & Stefano Negri

Siti di successo che usano Catalyst

Come youtube ma monotematico



youporn.com [51]

Come dovrebbe essere fatto Raiclik



bbc.co.uk/iplayer/ [46]

Abbonamenti a periodici on-line



magazines.com [13.008]

Blog community



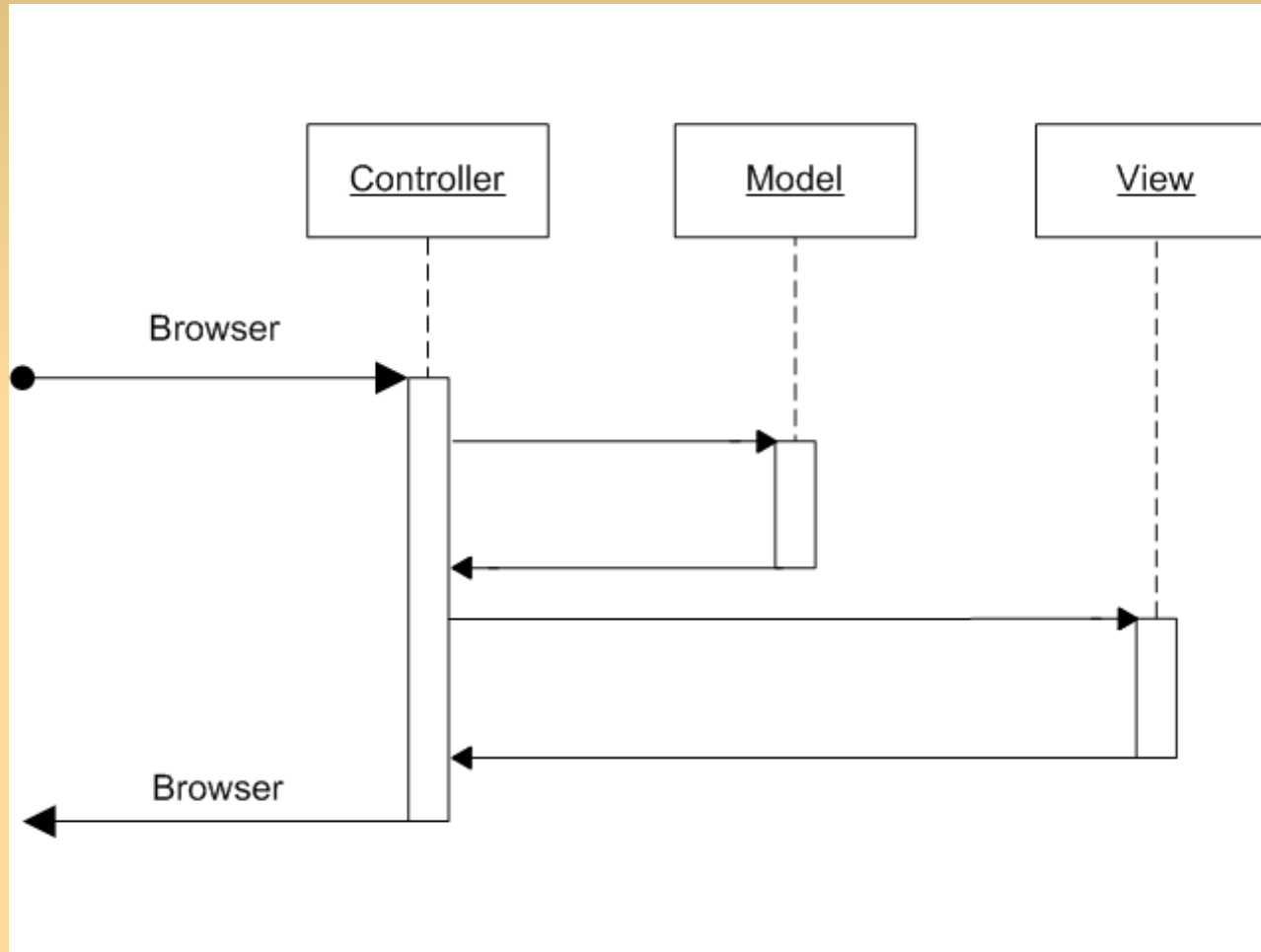
vox.com [1.263]

Introduzione a Catalyst

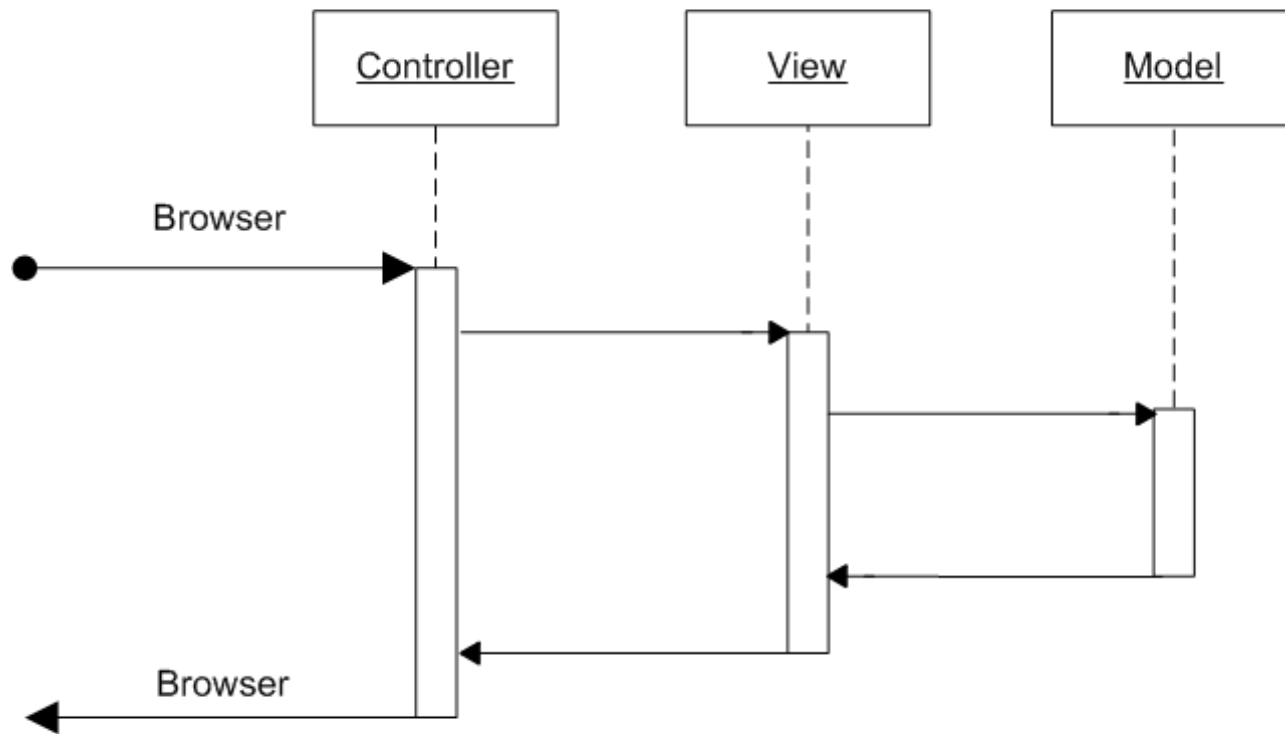
MVC

Model View Controller

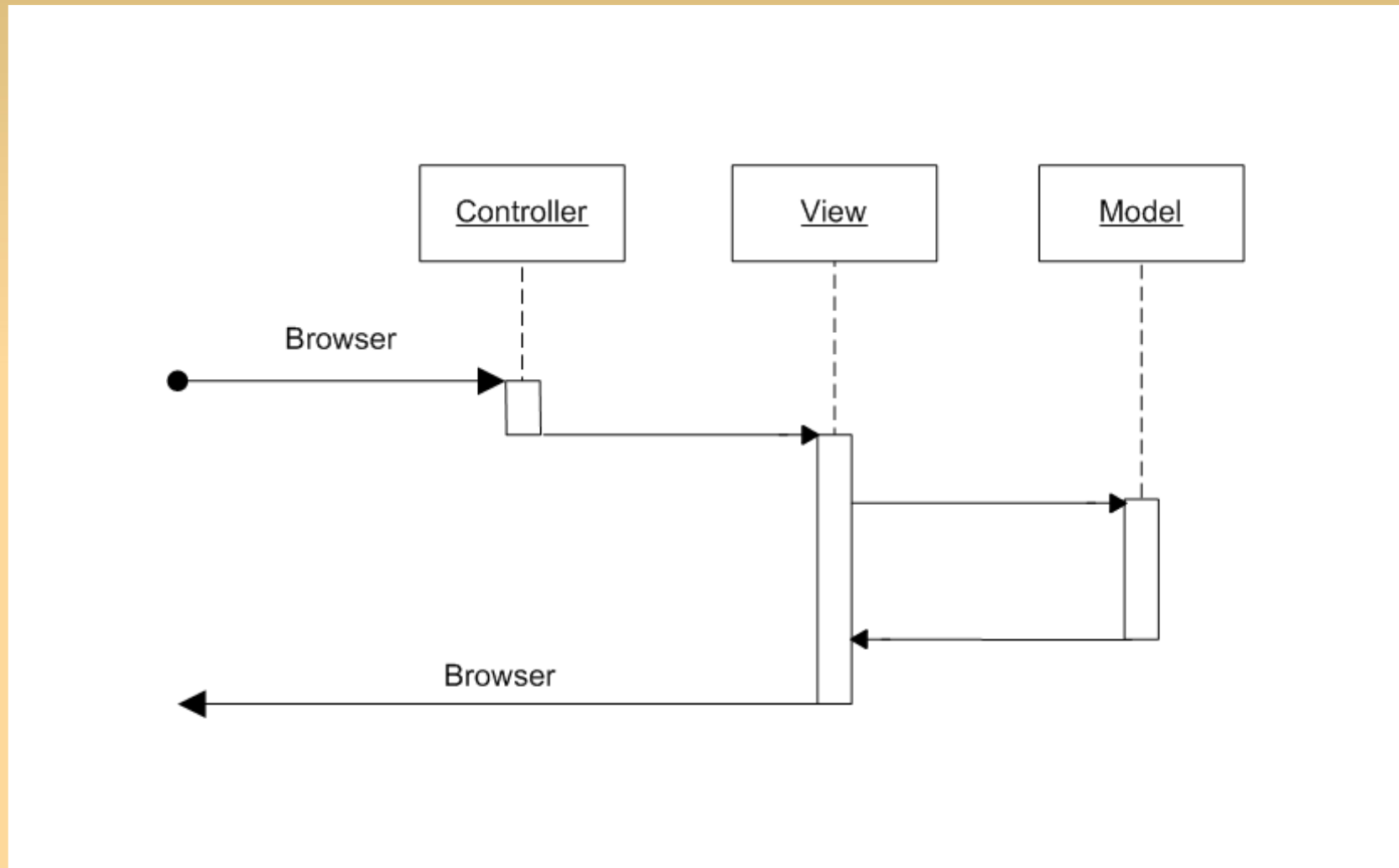
MVC Passivo



MVC Attivo



MVC Attivissimo



Action

Le Action, in prima approssimazione, determinano il modo in cui Catalyst effettua il dispatching delle URL richieste al server HTTP.

Tipi di Action

Namespace prefissato (:Local)

```
Package MyApp::Controller::My::Controller;  
sub foo : Local { }
```

Corrispondono ad essa tutte le url che cominciano con:

<http://localhost:3000/my/controller/foo>.

Il path risulta quindi composto dal namespace e dal nome della subroutine con gli “ / ” al posto dei “ :: ” e tutto in minuscolo.

Tipi di Action ...continua

Matching a livello di Namespace (:Global)

```
package MyApp::Controller::Foo;  
  sub foo : Global { }
```

Corrispondono ad essa tutte le url dei tipo

<http://localhost:3000/foo>

Modificatori della Action

(:Args)

Restrizioni sul numero di argomenti che possono essere presenti nel path

(Args:) esempio

```
MyApp::Controller::Foo  
  sub bar :Local
```

Se si vuole che ad essa corrispondano url che iniziano con `/foo/bar` ma che abbiano un solo elemento nel path dopo “`bar`” è sufficiente:

```
sub bar :Local :Args(1)
```

Al contrario con `Args(0)` non viene presa in considerazione nessuna url avente argomenti dopo “`bar`”

Notare che omettendo `:Args` vengono considerate tutte le url che metchano con il path del controller indipendentemente dal numero di argomenti in esse presenti.

Tipi di Action ...continua

Matching letterale (:Path)

Ad essa corrispondono solo le url aventi uno specifico path.

Se nella Action non viene specificato lo slash iniziale il confronto avviene considerando il path relativo al namespace corrente.

Al contrario, la presenza dello slash iniziale, indica che deve essere usato come path quello di root ignorando il namespace corrente.

(:Path) esempi

```
package MyApp::Controller::My::Controller  
  sub bar : Path('foo/bar') { }
```

La corrispondente url è: <http://localhost:3000/my/controller/foo/bar>

```
package MyApp::Controller::My::Controller  
  sub bar : Path('foo/bar') { }
```

In questo caso la corrispondente url è: <http://localhost:3000/foo/bar>

Tipi di Actions ...continua

Matching mediante Pattern (:Regex e :LocalRegex)

```
package MyApp::Controller::My::Controller;  
  sub bar : Regex('^item(\d+)/order(\d+)$'){ }
```

Possono essere usate delle regex per il confronto delle url. `:LocalRegex` effettua il confronto a partire dalla url del namespace corrente, mentre `:Regex` non tiene conto di quest'ultimo e parte dal namespace principale.

Tipi di Actions ...continua

(:Chained)

Catalyst fornisce metodi per il dispatching di azioni concatenate

Tipi di Action (:Chained) ...continua

```
sub catalog : Chained : CaptureArgs(1) {  
    my ( $self, $c, $arg ) = @_;  
    ...  
}
```

```
sub item : Chained('catalog') : Args(1) {  
    my ( $self, $c, $arg ) = @_;  
    ...  
}
```

Action Private

:Private

Le action private possono essere richiamate solo all'interno di Catalyst usando opportuni metodi (`forward`, `detach`, `visit`, `go`) ma non dalla richiesta di una determinata url.

```
sub foo : Private { }  
  
$c->forward('foo')
```

Inoltre l'attributo `:Private` è esclusivo, non può essere cioè usato in combinazione con altri attributi.

Action speciali

default : Path

È richiamata quando alla url non corrisponde nessuna action.

index : Path : Args (0)

Simile alla precedente ma non accetta argomenti. Inoltre viene valutata prima di :Default e :Path.

auto

Eseguite a catena.

Action speciali ...continua

begin : Private

Viene richiamata all'inizio di una richiesta al livello di controller ma prima della Action corrispondente alla url richiesta.

end : Private

Viene richiamata quando tutte le action corrispondenti alla url richiesta sono state eseguite.

Processamento delle URL

Catalyst determina la Action da richiamare a fronte della richiesta di una determinata url eseguendo il confronto per ordine di specificità crescente. Viene cioè scelta la Action che ha il maggior numero di componenti presenti nella url esaminata.

Processamento delle URL ...continua

Consideriamo di voler processare le seguenti url:

`/foo/$bar/$baz` con \$bar, \$baz che possono variare

`/foo/boo`

`/foo/boo/hoo`

Avendo definito le seguenti tre Action

```
sub foo : Regex('^foo$') { my ($self, $context, $bar, $baz) = @_; }
```

```
sub boo : Path('foo/boo') { .. }
```

```
sub hoo : Path('foo/boo/hoo') { .. }
```

Metodo `uri_for`

```
$c->uri_for( $path, @args?, \%query_values? )
```

```
$c->uri_for( $action, \@captures?, @args?, \%query_values? )
```

Costruisce una url assoluta (oggetto URI) partendo dalla root dell'applicazione, dal path fornito e da parametri query. Utilizzata in contesto scalare fornisce la stringa contenente la url.

Catalyst helper

Creazione dell'applicazione

```
$ catalyst.pl MyApp
```

Catalyst helper

Aggiunta di un nuovo Controller

```
$ script/myapp_create.pl controller StorageTest
```

```
exists "lib/PreFer/Controller"
```

```
exists "t"
```

```
created "lib/PreFer/Controller/StorageTest.pm"
```

```
created "t/controller_StorageTest.t"
```

Catalyst helper

Aggiunta di una nuova View (TT)

```
$ script/myapp_create.pl view TT TT
```

```
exists "lib/MyApp/View/"
```

```
exists "t"
```

```
created "lib/MyApp/View/TT.pm"
```

```
created "t/view_TT.t"
```