Revisione di qualifica

Team QuiXoft

12 Marzo 2009



Model Controller View Helper MiddleMan Algorithm

Parte I

Definizione di prodotto

Team QuiXoft

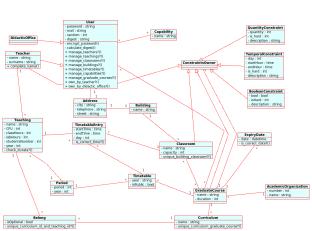
finizione di prodotto

Model

View Helper MiddleMan Algorithm

- ogni classe deriva direttamente da ActiveRecord::Base
- set di metodi ausiliari che implementano validazioni comuni a molti models
- metodi di callback (monitorano lo stato dell'oggetto)

Model



Model

MiddleMan Algorithm

View Helper MiddleMan Algorithm

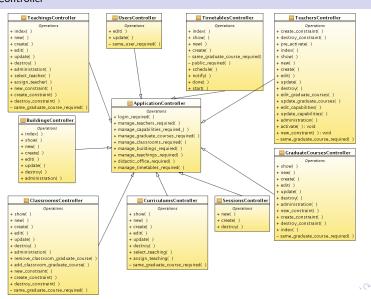
- derivano da ApplicationController
- stretto legame con i relativi Model
- variabili d'istanza per le View associate

Action comuni a più controller:

 index, show, new, create, edit, update, destroy, administration

Controller

Definizione di prodotto



Team QuiXoft

Controller View

MiddleMan Algorithm

View

zione di prodotto

Model

Controlle View

Helper MiddleMan Algorithm

Team QuiXoft

izione di prodotto

Model Controller

View

Helper MiddleMan Algorithm

War architecture

Team QuiXoft

Model

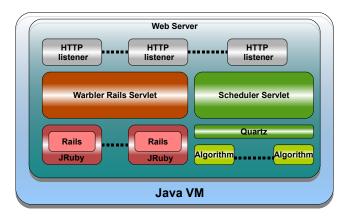
Controlle /iew

Helper MiddleMan Algorithm

WAR-Based Mode

- Web ARchive (WAR)
- standard packaging format per applicazioni su Java EE servers
- Warbler plugin per comprimere l'applicazione Ruby-on-Rails

War architecture



Team QuiXoft

View

MiddleMan Algorithm

Quartz scheduler

Model

.ontroller /iew Jalpar

MiddleMan Algorithm

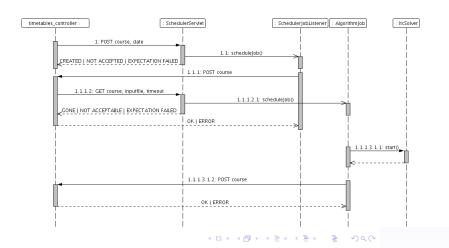
From BackgroundRB to Quartz scheduler Perchè Quartz:

- semplicità di configurazione
- persistenza dei trigger e dei job, che possono essere salvati su database in modo trasparente
- clusterizzazione, quartz è stato progettato per essere clusterizzato
- estensibilità delle funzioni tramite plugin
- possibilità di monitorare tutti gli aspetti della schedulazione e di tenere traccia dell'esito dei ogni job

Sequence diagram

Team QuiXoft

Model Controller



inizione di prodetto

Model Controller View Helper MiddleMan Algorithm

Scelta dell'algoritmo

- Riutilizzo di un algoritmo già esistente
- International Timetabling Competition 2007 (terminata in agosto 2008)
- Algoritmo di Tomáš Müller, vantaggi:
 - ▶ free sotto licenza LGPL
 - scritto in Java, interfacciamento tramite JRuby
 - possibilità di impostare un timeout
 - modulare e (secondo l'autore) facile da estendere
 - ampia e chiara documentazione
 - vincitore dell'ITC 2007

Algorithm

Team QuiXoft

Model Controller View Helper MiddleMan

Algorithm

Specializzazione

- Mancanza di due funzionalità importanti:
 - preferenze dei docenti
 - segnalazione delle preferenze non rispettate
- studio dell'algoritmo
- aggiunta delle funzionalità mancanti
- test

Revisione di qualifica

${\sf Team\ QuiXoft}$

Resoconto dei test Esempio unit test Esempio functional test

Parte II

Piano di qualifica

Resoconto dei test Esempio unit test Esempio functional test

Piano di qualifica

Resoconto dei test

C0 code coverage information Generated on Thu Mar 05 16:54:24 +0100 2009 with roov 0.8.1.2 Total lines Lines of code Total coverage Code coverage TOTAL 1994 1456 46.2% 141 112 99 83 25 22 162 145 135 113 106 13 13 11 13 36.4% 40.0% 92.3% 100.0% 100.0% 85.0% 100.0% 57 77.1% 100.0% 70.4% 50.0% 64.3% I 93.9% 70.6% dela/period.rb 13 100.0% 82.1% dela/teacher.rb 18.2% 72.4% 16 68.8% nodels/timetable.rb 70.6% modela/timetable_entry.rb 64.0% models/user.rb 113 70.8% 61.6% Generated using the roov code coverage analysis tool for Ruby version 0.8.1.2. WSC XHTHL WSC COS

Piano di qualifica

Esempio unit test

Team QuiXoft

Esempio unit test Esempio functional test

```
class AddressTest < ActiveSupport::TestCase</pre>
fixtures :addresses, :buildings
  def setup
    @a=Address.new
  end
  def test_correct_address
    @a.telephone="049-9050231"
    @a.street="Via Belzoni 6"
    @a.city="Villafranca veronese"
    assert @a.save
  end
  def test_destroy_address_in_building_address_id_must_nil
    addresses(:address_2).delete
    !assert(buildings(:building_1).address_id)
  end
```

Piano di qualifica

Esempio functional test

Team QuiXoft Esempio functional test

Esempio unit test

```
class SessionsControllerTest < ActionController::TestCase
 test "Guest usa New" do
   get :new
   assert_template "new"
    assert_response :success
 end
 test "Immissione di email e password validi" do
   user = stub(:id => :an_id, :mail => "a_mail",
                :password => "a_password")
   User.stubs(:authenticate).returns(user)
   user.stubs(:active?).returns(true)
   post :create, :mail => user.mail,
                  :password => user.password
    assert_equal session[:user_id], :an_id
    assert redirected to timetables url
 end
end
```

Revisione di qualifica

Team QuiXoft

Confronto delle ore di Confronto delle ore di Progettazione e Realizzazione

Parte III

Piano di progetto

Piano di progetto

Confronto delle ore di Analisi

Revisione di qualifica

Team QuiXoft

Confronto delle ore di Analisi Confronto delle ore di Progettazione e

Costo stimato: 5218€; costo effettivo: 5196€.

Piano di progetto

Confronto delle ore di Progettazione e Realizzazione

Team QuiXoft

Confronto delle ore di Progettazione e Realizzazione

Costo stimato totale 13793€.