

# Spring Roo





Speaker

Software Engineer and Architect

ProNetics

Founder

Spring Italian User Group

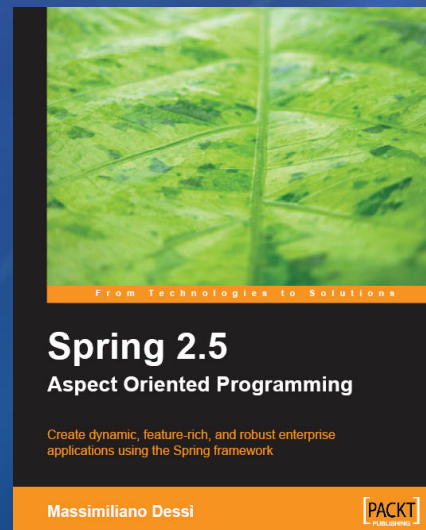
Presidente

JugSardegna Onlus

Committer/Contributor

OpenNMS – MongoDB

Autore Spring 2.5 Aspect Oriented programming



## Kanga Roo In Action

30 minutes

(10 to build 20 to describe)

less than one SBK manche  
in Phillip Island Circuit.

Troy Bayliss





ROO migliora la produttività  
dello Sviluppo Java senza nessun compromesso  
sulla qualità e flessibilità.

Per l'utente finale è un piccolo genio  
che lavora in silenzio sulle cose di cui non  
vorremmo preoccuparci

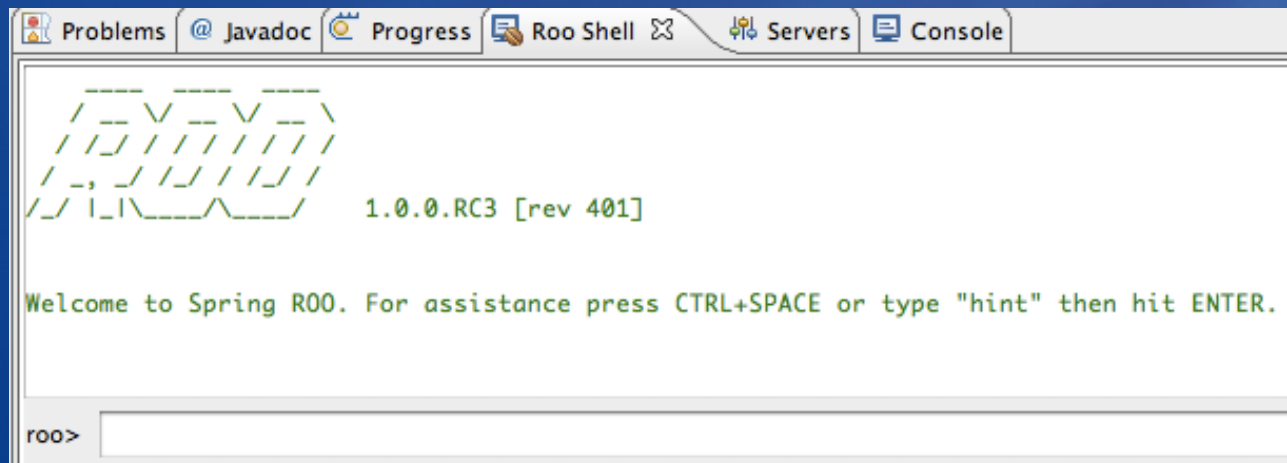
10 minuti con Roo valgono  
più di alcuni giorni di lavoro

Roo is an extensible, text-based RAD tool for Java

Roo is development-time only (no Roo runtime)

```

  _ _ _ _ _
 / _ \ / _ \ / _ \
/_ _ \/_ _ \/_ _ \
/_ _ \/_ _ \/_ _ \
/_ _ \/_ _ \/_ _ \
1.0.0.RC3 [rev 401]
Welcome to Spring Roo. For assistance press TAB or type "hint" then hit ENTER.
roo>
```



Roo semplifica la creazione di un progetto  
assistendo l'utente

- `hint` e `help` aiutano nei vari passi richiesti
  - TAB completamento ovunque
    - Comandi contestuali
- In caso di errore ROO permette il rollback.
- Comandi “scriptabili” per poterli riprodurre
- Funzionamento conservativo e predittivo

Roo è costruito su:

Java

Spring Framework

Java Persistence API (Hibernate)

Java Server Pages

Spring Security

Spring Web Flow

Log4J, Maven, AspectJ, Eclipse/STS



Roo è un generatore di codice ibrido

- utilizza in meglio del modello di generazione passivo e attivo

## Generazione passiva

- Utilizzo della Roo shell per generare qualcosa (solitamente files .xml e .java)

## Generazione attiva

- Costruzione di metadati con le annotazioni @Roo\*
- Aggiornamento incrementale di file .aj, .jsp

## Out of the Box

- Gestione progetto
- Oggetti generici
- Entity
- Gestione campi
- Persistenza
- JUnit test
- Finder dinamici
- Spring MVC, Spring Web Flow, Selenium
- JMS, SMTP
- Build system integration (Maven)
- Spring Security, Bundlor, Logging

# Project management

project

dependency add

dependency remove

backup

Tutte le dipendenze usano Enterprise Bundle Repo  
Versionati, OSGi-compatible

- @RooJavaBean – getters e setters
- @RooConfigurable – aggiunge  
@Configurable
- @RooToString



`entity`

- `@RooEntity`

- JPA entity feature-rich

- JPA `@Entity` con id, version, EntityManager

- Automaticamente aggiunge count, findAll, findById, findEntries, persiste, rimuove

- Comando con molte direttive opzionali

`field`

Scelta tra i tipi “comuni”

La shell ricorda l'ultima entity su cui si è lavorato

Opzioni per:

- JPA annotations
- JavaBean validation annotations
- Java modifiers

Hibernate, OpenJPA, EclipseLink

database più comuni

`persistence exception translation`

`database properties list`

`database properties set`

`database properties remove`

# JUnit test

test integration

test mock



Roo evita la scrittura di JPA Query language  
finders Aggiunti come metodi statici alle entity

```
finder list
```

```
finder add
```

controller scaffold

- Automatic REST controller per una entity
- Automatic JSP creazione e manutenzione
- modifica dei metodi permessi (CRUD subset)

controller all

Scaffold di tutte le entity

controller class

Solo il Controller nudo e crudo

## Spring JS, Tiles e URL Rewrite

selenium test

webflow

# Smtplib e JMS

```
email sender setup
```

```
field email
```

```
jms setup
```

```
field jms setup
```

```
jms listener class
```



# Build System Integration

perform clean -> mvn clean eclipse:clean"

perform eclipse -> mvn eclipse:eclipse

perform tests -> mvn test

perform package -> mvn package

perform command -> custom mvn command

security setup imposta spring security

bundlor setup imposta Bundlor per OSGi

logging setup imposta log4j

script esegue uno script

perform command -> custom mvn command

Si possono creare degli addon, i comandi

predefiniti sono:

`addon list`

`addon install`

`addon uninstall`

`addon cleanup`

Quando si aggiorna roo, possono cambiare i file .aj,  
che sono gli unici che l'utente non deve toccare  
perchè sono i responsabili della generazione attiva  
del codice



- 1) Nell' STS utilizzare `push-in refactoring`, questa trasferisce il contenuto dei file `.aj` nei file `.java`
- 2) Rimuovere I jar di ROO Annotation dal `pom.xml`
- 3) Rimuovere le annotazioni `@RoO` dai file `.java`

## Q & A ?



Grazie per l'attenzione !

Massimiliano Dessì

desmax74 at yahoo.it

massimiliano.dessi at pronetics.it

<http://twitter.com/desmax74>

<http://jroller.com/desmax>

<http://www.linkedin.com/in/desmax74>

<http://wiki.java.net/bin/view/People/MassimilianoDessi>

<http://www.jugsardegna.org/vqwiki/jsp/Wiki?MassimilianoDessi>

Spring Framework Italian User Group

<http://it.groups.yahoo.com/group/SpringFramework-it>