Documentation

User's Guide
Custom Templates
Feature Roadmap
FAQ

Source Forge Pages

Project Page
Download

Project Documentation

Project Information
Project Reports

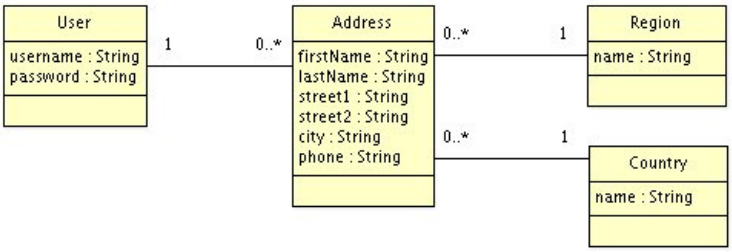
built by: maven

Last Published: Sep 22 2007

Using Groovy MDA

Start by creating a UML model in a tool that supports XMI. I am going to use ArgoUML for all of the examples.

For this example, we will create a simple address book application where a users can store addresses.



Running the groovymda.jar from the command line

```
java -jar groovymda-1.0.jar 'jar:file:./addressbook.zargo!/addressbook.xmi'
```

After running the command you will notice Java source files are created. If you want to change the output directory add a second argument for the output directory when running the command.

For the command above, there will be a generated directory structure and java source files show below:

```
-- com
   `-- acme
       `-- domain
           |-- Address.java
           |-- Country.java
           |-- Region.java
           `-- User.java
```

Example of the User entity that was generated:

```
package com.acme.domain;

import java.io.Serializable;
import java.util.Date;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.Lob;
import javax.persistence.ManyToMany;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
* User Domain Object
*/
@Entity
@Table(name = "t_user")
@NamedQueries({
@NamedQuery(name = "User.findAll", query = "from User")
})
public class User implements Serializable {
```

```
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
@Column(name = "pk_id")
private Long id;

@Column(name = "c_username")
private java.lang.String username;

@Column(name = "c_password")
private java.lang.String password;

@OneToMany(
    mappedBy = "user",
    cascade = {CascadeType.PERSIST, CascadeType.MERGE}
)
private java.util.Set<Address> addresss;

public java.lang.String getUsername() {
    return username;
}

public void setUsername(java.lang.String username) {
    this.username = username;
}

public java.lang.String getPassword() {
    return password;
}

public void setPassword(java.lang.String password) {
    this.password = password;
}

public java.util.Set<Address> getAddresss() {
    return addresss;
}

public void setAddresss(java.util.Set<Address> addresss) {
    Address[] tempAddresss = (Address[]) getAddresss().toArray(new Address[getAdd
    for (Address address : tempAddresss) {
        removeFromAddresss(address);
    }
    if (addresss != null) {
        for (Address address : addresss) {
            addToAddresss(address);
        }
    }
}

public void addToAddresss(Address address) {
    if (address != null) {
        address.setUser(this);
        getAddresss().add(address);
    }
}

public void removeFromAddresss(Address address) {
    if (address != null) {
        address.setUser(null);
        getAddresss().remove(address);
    }
}

public int hashCode() {
    int hashCode;
    if (id != null) {
        hashCode = 29 * id.hashCode();
    } else {
        hashCode = super.hashCode();
    }
    return hashCode;
}

public boolean equals(Object o) {
    if (this == o) {
        return true;
    }
    if (!(o instanceof User)) {
        return false;
    }
    final User otherUser = (User) o;
    if (hashCode() != otherUser.hashCode()) {
        return false;
    }
```

```
            return true;
        }

    }
```