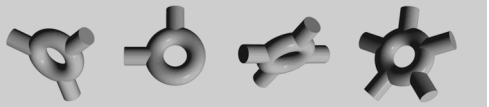
elements



User Package

TARMS Inc.

September 07, 2000

Typeset in LaTeX.

Contents

List of Figures

List of Tables

Package Description

This package models a user. A user represents a person who is using the system. The user model contains information about the person ranging from contact information through to the permissions that a user has to perform operations and trade deals with the system.

1 Interfaces

1.1 User

The user interface represents an individual person using the system. This interface contains methods that will uniquely identify the represented person.

1.1.1 Relationships

	Class	Description	Notes
⇑	Identifiable		
⇑	Responsible		
⇑	PermissionOwner		
⇑	Validatable		
↓	UserModel §2.1		
↓	UserReferenceDataModel §2.2		
↔	UserReferenceDataModel §2.2	model	

⇑:Inherits ↓:Realized by ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

1.1.2 Operations

String identifier() identifier

The "user name" of this user. The identifier enables this user to be referenced by a short name. No two users are allowed to have the same identifier.

String title() title

The title of the user's real name. i.e: Mr/Mrs/Miss/Ms.

String givenName() givenName

The given name of the user's real name.

String surname() surname

The surname of the user's real name.

String preferences() preferences

Contains a dictionary of strings to object pairs, where the string is an identifier of the type of user preference and the object is the object which contains the user preference information. The strings will be colon separated words as described by the following regular expression:

`([A-Za-z_][A-Za-z0-9_␣]*)(:[A-Za-z_][A-Za-z0-9_␣]*)*`

The objects in this structure should be reconstructed from the database from an XML (Extended Mark-up Language) string in order to provide more robustness between versions of the object model.

Collection<ContactInformation> contactInformation() contactInformation

Returns the collection of contact information for this user.

Reportable validate() validate

The validation of users is considered to be a business problem that should be specified at implementation time. An example of a validation procedure for User is given below.

This method returns an object of type Reportable which will contain all the errors and warnings generated from this method. Below is a list of the axioms that must hold for an instant of a class that realizes this interface to be valid. Each statement is followed by the error or warning message that will be issued if the axiom is violated.

- The User must have an identifier. If not, then a "User with null identifier" error is issued.
- The User is expected to have a title. If not, then a "User has no title" warning is issued.
- The User must have a given name. If not, then a "User has no given name" error is issued.
- The User is expected to have a surname. If not, then a "User has no surname" warning is issued.
- The User is expected to have contact information. If not, then a "User has no contact information" warning is issued.

This method will also call the validate method on the contact information, provided that the contact information is non null. This method also calls validate method inherited from the PermissionOwner and combines the Reportable with the one generated from this method.

Domain domain() domain

This method returns the domain in which this user is situated. A domain represents a geographical section of a business. Access rights can be assoicated with a domain.

String toString() toString

This method returns a string containing the identifier, title, givenName and surname of this user.

2 Classes

2.1 UserModel

This class is a concrete realization of the User interface.

2.1.1 Relationships

	Class	Description	Notes
⇑	PermissionOwnerModel		
↑	User §1.1		
↔	ContactInformation	contactInformation 0..n	→
↔	Domain	domain	→

⇑:Inherits ↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

2.1.2 Attributes

title: String

givenName: String

surname: String

preferences: Dictionary

2.1.3 Operations

String identifier() identifier

This method returns the result of calling the superclass's identifier method.

2.2 UserReferenceDataModel

Implements its interface by delegating to the associated model.

2.2.1 Relationships

	Class	Description	Notes
⇑	ReferenceDataModel		
↑	User §1.1		
↔	User §1.1	model	→

⇑:Inherits ↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

3 Associations

Table 1: User— Associations

Association Role	Class	Card.	Notes
model			
	User §1.1		→
	UserReferenceDataModel §2.2		
contactInformation			
	ContactInformation	0..n	→
	UserModel §2.1		
domain			
	Domain		→
	UserModel §2.1		

→:Navigable ◊:Aggregate ♦:Composite

3.1 model

Role: *Navigable* User.
Role: UserReferenceDataModel.

3.2 contactInformation

Role: *Navigable* ContactInformation, 0..n.
Role: UserModel.

3.3 domain

Role: *Navigable* Domain.
Role: UserModel.

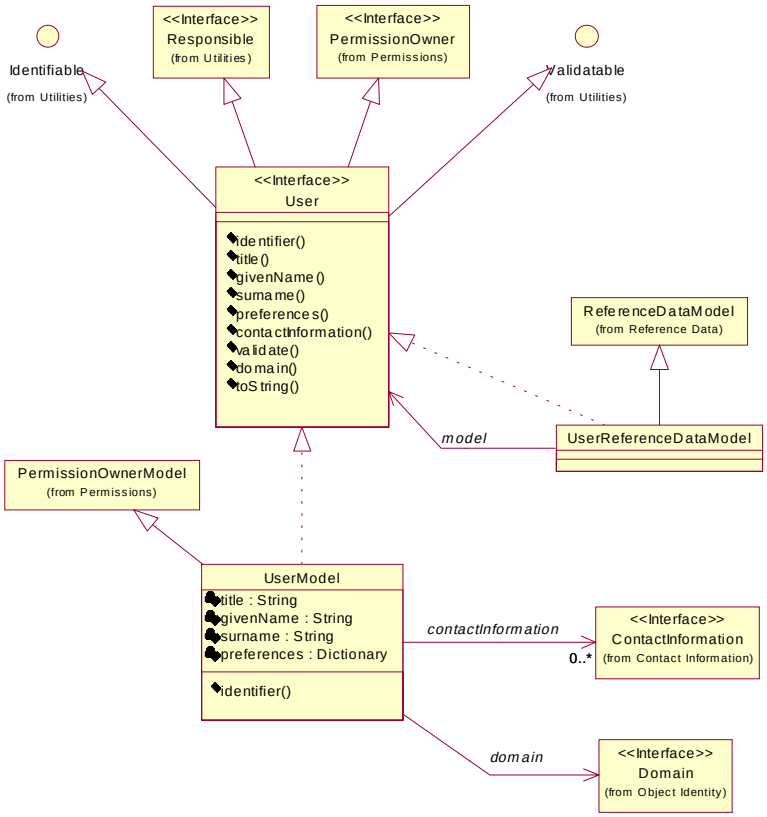


Figure 1: Class Diagram— Users

References