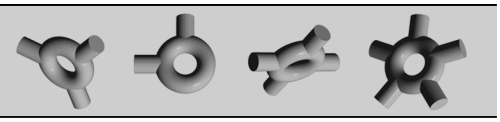
elements 

Collections Package

TARMS Inc.

September 07, 2000

Copyright ©2000 TARMS Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this model and associated documentation files (the “Model”), to deal in the Model without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Model, and to permit persons to whom the Model is furnished to do so, subject to the following conditions:

1. The origin of this model must not be misrepresented; you must not claim that you wrote the original model. If you use this Model in a product, an acknowledgment in the product documentation would be appreciated but is not required. Similarly notification of this Model’s use in a product would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice, including the above copyright notice shall be included in all copies or substantial portions of the Model.

THE MODEL IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MODEL OR THE USE OR OTHER DEALINGS IN THE MODEL.

Typeset in LaTeX.

Contents

1 Interfaces **2**
- 1.1 Filter . . . 2
  - 1.1.1 Relationships . . . 2
  - 1.1.2 Operations . . . 3
- 1.2 FilteredCollection . . . 3
  - 1.2.1 Relationships . . . 3
  - 1.2.2 Operations . . . 3
- 1.3 SearchableCollection . . . 4
  - 1.3.1 Relationships . . . 4
  - 1.3.2 Operations . . . 4
- 1.4 Collection . . . 4
  - 1.4.1 Relationships . . . 4
  - 1.4.2 Operations . . . 5
- 1.5 Tree . . . 5
  - 1.5.1 Relationships . . . 5
  - 1.5.2 Operations . . . 5
- 1.6 NamedTree . . . 6
  - 1.6.1 Relationships . . . 7
  - 1.6.2 Operations . . . 7

2 Classes **7**
- 2.1 CollectionModel . . . 7
- 2.2 FilterModel . . . 7
  - 2.2.1 Relationships . . . 7
  - 2.2.2 Attributes . . . 7
- 2.3 FilteredCollectionModel . . . 8
  - 2.3.1 Relationships . . . 8
- 2.4 FilterReferenceDataModel . . . 8
  - 2.4.1 Relationships . . . 8
- 2.5 NamedTreeReferenceDataModel . . . 8
  - 2.5.1 Relationships . . . 8
- 2.6 TreeModel . . . 9
  - 2.6.1 Relationships . . . 9
  - 2.6.2 Attributes . . . 9
- 2.7 NamedTreeModel . . . 9
  - 2.7.1 Relationships . . . 9
  - 2.7.2 Attributes . . . 9

3 Associations **9**
3.1 program 10
3.2 rawCollection 10
3.3 model 10
3.4 model 10

List of Figures

1 Class Diagram— Tree 11
2 Class Diagram— Filtering 12

List of Tables

1 Collections— Associations 9
1 . . . continued 10

Package Description

This package contains utility classes who's behavior is related to the collection and filtering of objects.

1 Interfaces

1.1 Filter

This interface provides a way of taking a raw collection of objects and selecting a subset of those objects according to some set of rules. Filters can have names and descriptions.

1.1.1 Relationships

	Class	Description	Notes
⇑	Identifiable		
⇓	FilteredCollection §1.2		
↓	FilterModel §2.2		
↓	FilterReferenceDataModel §2.4		
↔	FilterReferenceDataModel §2.4	model	

⇑:Inherits ⇓:Inherited by ↓:Realized by ↔:Association →:Navigable ◊:Aggregate ♦:Composite

1.1.2 Operations

String description() description

A description of the purpose of this filter, and what collection it can be applied to.

Collection filterThisCollection(Collection rawCollection) filterThisCollection
rawCollection: Collection

This method filters the given rawCollection. It does this by passing the element to the filter method and collecting those elements that return true.

Function program() program

This method returns a function that will be used to filter the raw collection.

1.2 FilteredCollection

This interface represents a (filtered) collection that is obtained by filtering an associated (raw) collection.

1.2.1 Relationships

	Class	Description	Notes
⇑	Filter §1.1		
⇑	SearchableCollection §1.3		
↓	FilteredCollectionModel §2.3		

⇑:Inherits ↓:Realized by

1.2.2 Operations

Boolean exists(Object element) exists
element: Object The element to find in the collection.

This method returns true if the given element exists within the filteredCollection.

Collection filteredCollection() filteredCollection

Returns the value returned from executing the filterThisCollection method with the rawCollection as an argument.

Collection rawCollection() rawCollection

A collection of objects.

1.3 SearchableCollection

An object that can hold a collection of elements, and can be searched to find an element. A searchable collection does not support adding and deleting of elements from the collection.

1.3.1 Relationships

	Class	Description	Notes
⇓	FilteredCollection §1.2		
⇓	Collection §1.4		

⇓:Inherited by

1.3.2 Operations

Boolean exists(Collectable element) exists

element: Collectable The element to find in the collection.

If the given element exists within this collection then this method returns true, otherwise false.

1.4 Collection

This interface represents the abstract notion of a group of objects. It is likely that the language in which this object model is implemented will have an existing class which encompasses this behavior. It should be possible to use an existing implementation of a collection class without any modification.

1.4.1 Relationships

	Class	Description	Notes
⇑	SearchableCollection §1.3		
⇓	Tree §1.5		

⇑:Inherits ⇓:Inherited by

1.4.2 Operations

add(Object element) add

element: Object The element to be added to the set.

Adds the given element to the collection.

remove(Object element) remove

element: Object The element to be removed from the set.

Finds and removes the given element from the collection. If the element is not found the null is returned. (Alternative implementations may raise an appropriate exception.)

1.5 Tree

This class represents the "tree" data structure. This tree structure can have any number of (unique) sub-trees and any number of (unique) elements at each node of the tree. This interface is considered to represent a tree, however it can also be thought of as representing a particular node of a tree.

1.5.1 Relationships

	Class	Description	Notes
⇑	Collection §1.4		
⇓	NamedTree §1.6		
↓	FilterModel §2.2		
↓	TreeModel §2.6		

⇑:Inherits ⇓:Inherited by ↓:Realized by

1.5.2 Operations

Set<Objects> elements() elements

This method returns a set containing all the elements at this “node” of the tree.

Set<Tree> subTrees() subTrees

This method returns a set of all the sub-trees beneath this “node” of the tree.

Boolean exists(Object element) exists

element: Object The element which is being searched for within this tree.

If any element of this tree equals the given element then return true, otherwise false. A given element can be said to exist in this tree if it exists in any of the elements at this node, or if it exists in any of the sub-trees at this node.

remove(Object element) remove

element: Object The element which is to be removed from this tree.

This method removes all traces of the given element from this tree. This can be accomplished by removing the element from the set of elements at this node, and by removing the element from the sub-trees at this node.

Note that this method may effect the sub-trees of this tree. This may cause side effects to other objects that hold onto a sub-tree of this tree.

shallowRemove(Object element) shallowRemove

element: Object The element to be removed from this tree.

This method removes the given element from the elements at this node of the tree. This method does NOT search and remove the given element from the sub-trees at this node.

removeSubTree(Tree subTree) removeSubTree

subTree: Tree The subTree to be removed from this node.

This method removes the given subTree from the subTrees at this node.

1.6 NamedTree

This interface represents a tree that has both an identifier and a description of the tree. This enables information about the contents of the tree to be stored. This interface needs to be reference data because entire trees of data may need to be referred to. For example a tree containing a hierarchy of all the organizations may be kept as reference data.

1.6.1 Relationships

	Class	Description	Notes
⇑	Tree §1.5		
⇑	Identifiable		
↓	NamedTreeModel §2.7		
↓	NamedTreeReferenceDataModel §2.5		
↔	NamedTreeReferenceDataModel §2.5	model	

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

1.6.2 Operations

String identifier() identifier

The name or identifier of the tree.

String description() description

A description of what data this tree holds.

2 Classes

2.1 CollectionModel

2.2 FilterModel

This class is a concrete realization of the Filter interface.

2.2.1 Relationships

	Class	Description	Notes
↑	Filter §1.1		
↑	Tree §1.5		
⇓	FilteredCollectionModel §2.3		
↔	Function	program	→

⇓:Inherited by ↑:Realizes ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

2.2.2 Attributes

identifier: String The name of the filter.

description: String

2.3 FilteredCollectionModel

This class is a concrete realization of the FilteredSet interface. This class holds onto a dynamic rawCollection of objects that will be used to generate a filtered subset for that rawCollection. An implementation of this class would benefit from internal caching, because it is likely that a filtered collection may be requested a number of times while the raw collection remains the same.

2.3.1 Relationships

	Class	Description	Notes
⇑	FilterModel §2.2		
↑	FilteredCollection §1.2		
↔	Object	rawCollection 0..n	→

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

2.4 FilterReferenceDataModel

This class realizes its Filter interface by delagated to the associated model.

2.4.1 Relationships

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

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

2.5 NamedTreeReferenceDataModel

This class realizes the NamedTree interface by delegating to the associated model.

2.5.1 Relationships

	Class	Description	Notes
⇑	ReferenceDataModel		
↑	NamedTree §1.6		
↔	NamedTree §1.6	model	→

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

2.6 TreeModel

elementType

This class is a concrete realization of the Tree interface.

2.6.1 Relationships

	Class	Description	Notes
↑	Tree §1.5		
⇓	NamedTreeModel §2.7		

⇓:Inherited by ↑:Realizes

2.6.2 Attributes

elements: Set<Object>

subTrees: Set<Tree>

2.7 NamedTreeModel

This class is a concrete realization of the NamedTree interface.

2.7.1 Relationships

	Class	Description	Notes
⇑	TreeModel §2.6		
↑	NamedTree §1.6		

⇑:Inherits ↑:Realizes

2.7.2 Attributes

identifier: String

description: String

3 Associations

Table 1: Collections— Associations

Association Role	Class	Card.	Notes
program			

Table 1: . . . continued

Association Role	Class	Card.	Notes
	Function		→
	FilterModel §2.2		
rawCollection			
	Object	0..n	→
	FilteredCollectionModel §2.3		
model			
	Filter §1.1		→
	FilterReferenceDataModel §2.4		
model			
	NamedTree §1.6		→
	NamedTreeReferenceDataModel §2.5		

→:Navigable ◊:Aggregate ♦:Composite

3.1 program

Role: *Navigable* Function.
Role: FilterModel.

3.2 rawCollection

Role: *Navigable* Object, 0..n.
Role: FilteredCollectionModel.

3.3 model

Role: *Navigable* Filter.
Role: FilterReferenceDataModel.

3.4 model

Role: *Navigable* NamedTree.
Role: NamedTreeReferenceDataModel.

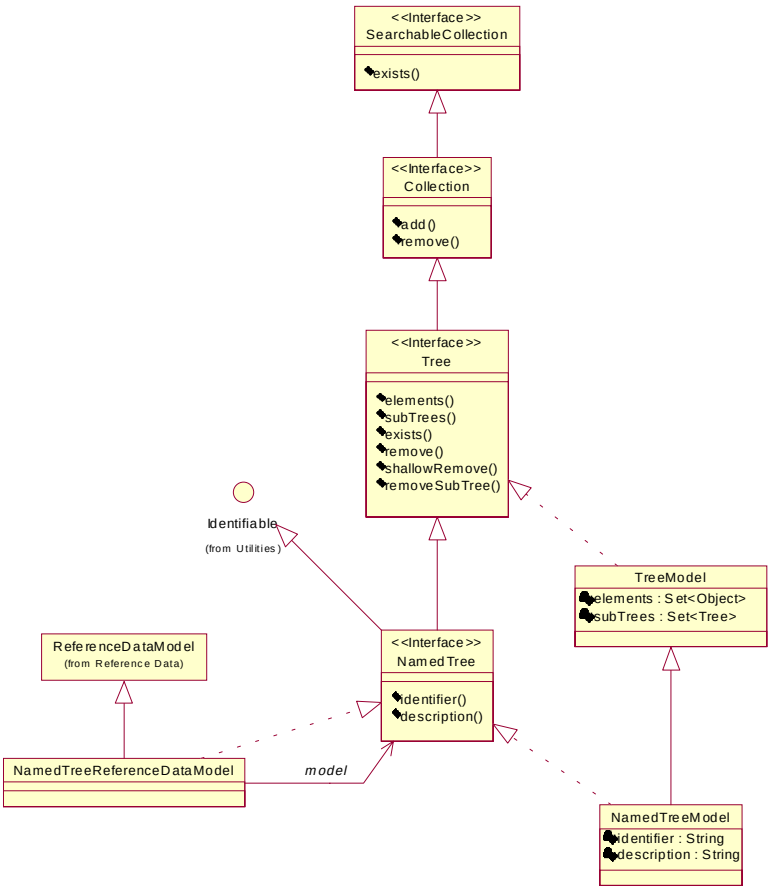


Figure 1: Class Diagram— Tree

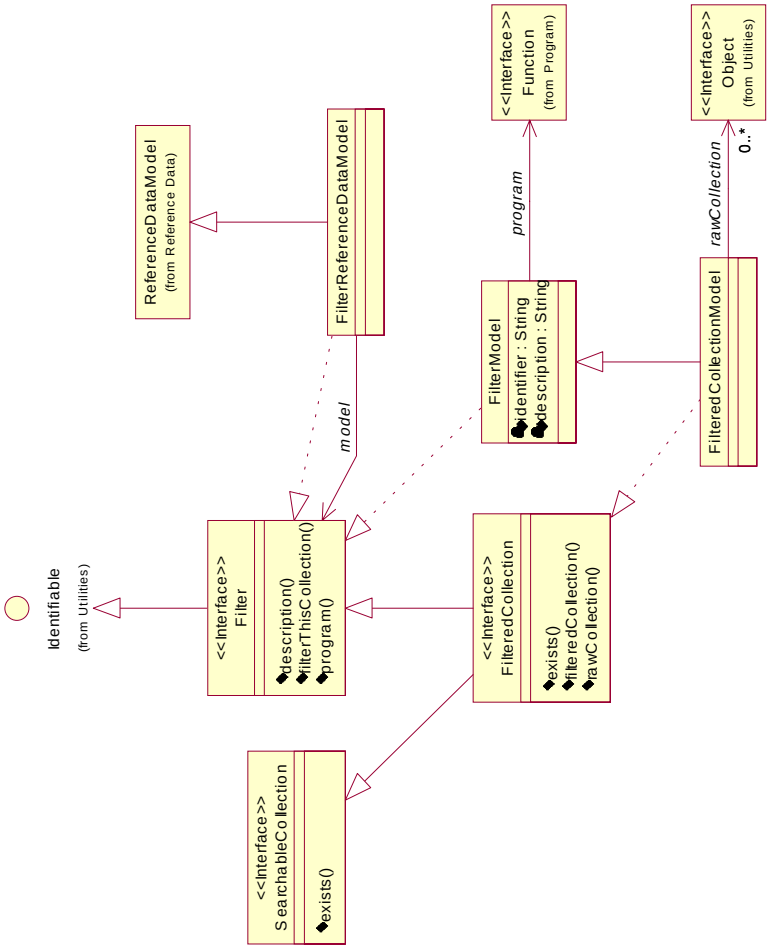


Figure 2: Class Diagram— Filtering

References