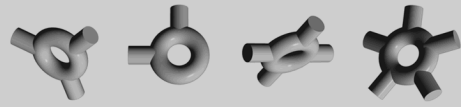


elements



## Contact Information 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 L<sup>A</sup>T<sub>E</sub>X.

## Contents

<b>1</b>	<b>Interfaces</b>	<b>2</b>
1.1	Address . . . . .	2
1.1.1	Relationships . . . . .	2
1.1.2	Operations . . . . .	2
1.2	ContactInformation . . . . .	3
1.2.1	Relationships . . . . .	3
1.2.2	Operations . . . . .	3
<b>2</b>	<b>Classes</b>	<b>4</b>
2.1	AddressModel . . . . .	4
2.1.1	Relationships . . . . .	4
2.1.2	Attributes . . . . .	4
2.2	ContactInformationModel . . . . .	5
2.2.1	Relationships . . . . .	5
2.2.2	Attributes . . . . .	5
<b>3</b>	<b>Associations</b>	<b>5</b>
3.1	location . . . . .	6
3.2	streetAddress . . . . .	6
3.3	postalAddress . . . . .	6
3.4	streetAddress . . . . .	6
3.5	postalAddress . . . . .	6
3.6	location . . . . .	6

## List of Figures

1	Class Diagram— Contact Information . . . . .	7
---	--	---

## List of Tables

1	Contact Information— Associations . . . . .	5
1	... continued . . . . .	6

## Package Description

This package models the information that can be used for contacting people or organizations.

# 1 Interfaces

## 1.1 Address

This interface describes a postal or street address.

### 1.1.1 Relationships

	Class	Description	Notes
↑	Validatable		
↓	AddressModel §2.1		
↔	ContactInformationModel §2.2	streetAddress	
↔	ContactInformationModel §2.2	postalAddress	

↑:Inherits ↓:Realized by ↔:Association →:Navigable ◇:Aggregate ◆:Composite

### 1.1.2 Operations

**String address()** address

This method returns the address of a party, for example a user or an organization. This includes the street number, street name, and suburb.

**String postCode()** postCode

This method returns a postcode. Note that throughout the world some postcodes can involve alphanumeric characters.

**Location location()** location

This method returns a country.

**Boolean validate()** validate

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 address should be non-null. If not, this method should issue a “Null address” error message.
- The postcode should be non-null. If not, this method should issue a “Null postcode” error message.

- The location should be non-null. If not, this method should issue a “Null location” error message.

## 1.2 ContactInformation

The contact information interface contains information about all of the ways that a particular person or organization can be contacted.

### 1.2.1 Relationships

Class	Description	Notes
↑ Validatable		
↓ ContactInformationModel	§2.2	

↑:Inherits ↓:Realized by

### 1.2.2 Operations

**String phone()** phone

This method returns a telephone number. The return type of the phone number needs to be a string so that zeros can occur at the front of the number and plus signs, spaces and brackets can be used.

**String fax()** fax

This method returns a facsimile number.

**String mobile()** mobile

This method returns a mobile telephone number.

**String email()** email

This method returns an e-mail address.

**Address streetAddress()** streetAddress

This method returns a street address.

**Address postalAddress()** postalAddress

This method returns a postal address.

### Reportable validate()

validate

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.

- E-mail addresses must either be null or conform to RFC:822/RFC:2156. [1, 2]. If not an “Invalid email address” warning should be raised.
- At least one of the following methods must return a non-null value: phone, fax, mobile, email, streetAddress, postalAddress. If not, then this method should issue an “Empty contact information” error message.

This method will also call validate on streetAddress and postalAddress if they are not null.

## 2 Classes

### 2.1 AddressModel

This class is a concrete realization of the Address interface.

#### 2.1.1 Relationships

Class	Description	Notes
↑ Address §1.1		
↔ Location	location	→

↑:Realizes ↔:Association →:Navigable ◇:Aggregate ◆:Composite

#### 2.1.2 Attributes

**address:** String

**postCode:** String

## 2.2 ContactInformationModel

This class is a concrete realization of the ContactInformation interface.

### 2.2.1 Relationships

	Class	Description	Notes
↑	ContactInformation §1.2		
↔	Address §1.1	streetAddress	→
↔	Address §1.1	postalAddress	→
↑:Realizes ↔:Association →:Navigable ◇:Aggregate ◆:Composite			

### 2.2.2 Attributes

**phone: String**

**fax: String**

**mobile: String**

**email: String**

## 3 Associations

Table 1: Contact Information— Associations

Association			
	Role	Class	Card. Notes
location		Location AddressModel §2.1	→
streetAddress		Address §1.1 ContactInformationModel §2.2	→
postalAddress		Address §1.1 ContactInformationModel §2.2	→
streetAddress		Address §1.1 ContactInformationModel §2.2	→
postalAddress		Address §1.1 ContactInformationModel §2.2	→

Table 1: ...continued

Association			
Role	Class	Card.	Notes
	Address §1.1		→
	ContactInformationModel §2.2		
location	Location		→
	AddressModel §2.1		

→:Navigable ◇:Aggregate ◆:Composite

### 3.1 location

**Role:** *Navigable* Location.

**Role:** AddressModel.

### 3.2 streetAddress

**Role:** *Navigable* Address.

**Role:** ContactInformationModel.

### 3.3 postalAddress

**Role:** *Navigable* Address.

**Role:** ContactInformationModel.

### 3.4 streetAddress

**Role:** *Navigable* Address.

**Role:** ContactInformationModel.

### 3.5 postalAddress

**Role:** *Navigable* Address.

**Role:** ContactInformationModel.

### 3.6 location

**Role:** *Navigable* Location.

**Role:** AddressModel.



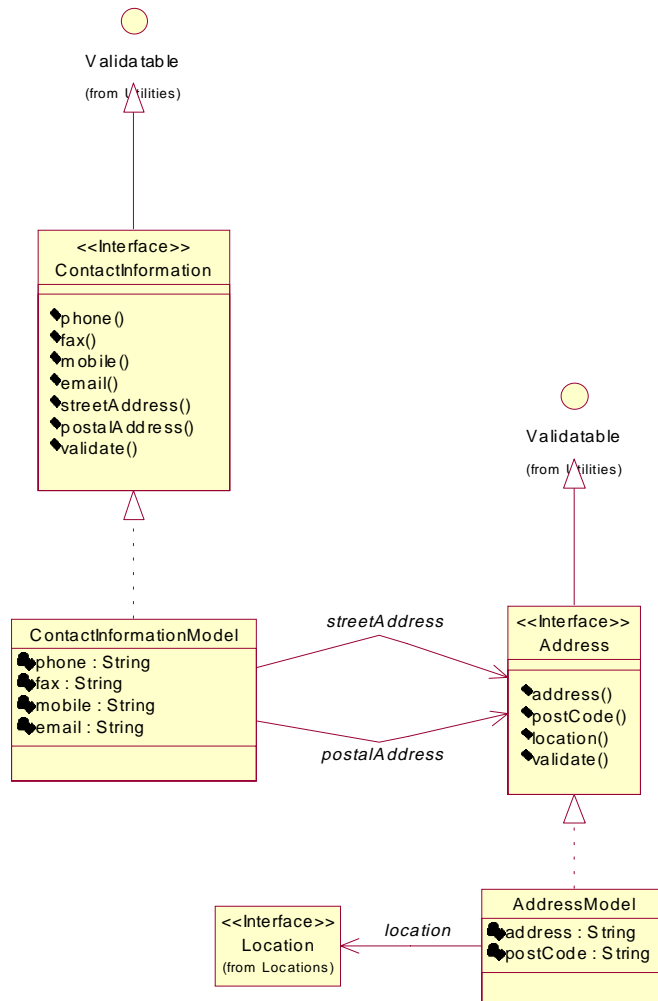


Figure 1: Class Diagram— Contact Information

## References

- [1] D. Crocker. *Standard for the format of ARPA Internet text messages*, number RFC 822, 1982.  
<ftp://ftp.isi.edu/in-notes/rfc822.txt>.
- [2] S. Kille. *MIXER (Mime Internet X.400 Enhanced Relay): Mapping between X.400 and RFC 822/MIME*, number RFC 2156, 1998.  
<ftp://ftp.isi.edu/in-notes/rfc2156.txt>.