Tara Application Builder

V1.1

Author:
Goran Zugic
goran.zugic@semantion.com

April 2011

Copyright © 2000-2011 Semantion Inc.

Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from Semantion (http://www.semantion.com).

1.0 Introduction

Tara Ontology Language (Tara) (http://www.semantion.com/documentation/SBP/metamodeling/TaraOntologyLanguage_V1.3.pdf) provides a generic ontological foundation for modeling business, social, and technology related processes and systems. Ontologies created in Tara can be used to develop web applications for management of information based on the ontologies.

Tara Application Builder (TAB) is a tool for automatic creation of web applications from ontologies defined in Tara Ontology Language.

2.0 How to Create Applications from Tara

The following steps need to be taken in order to build a web application from a Tara ontology:

1. Create an ontology in Tara (this ontology models a domain of knowledge)
2. Generate the web application from the ontology created in Step 1 using Tara Application Builder (TAB)

2.1 Tara Ontology Creation in SM2

To create an ontology in Tara we use the Semantion Metamodeler (SM2) (http://www.semantion.com/documentation/SBP/metamodeling/SemantionMetamodeler_V1.4.pdf) tool.

We will use a simple Project Management ontology to show you how to create an ontology in SM2.

The structure of the ontology has concepts representing entities that will be managed by the application (i.e., Project, Activity, Phase, etc.). Concepts are related to each other via associations. Each association also has a rule defining conditions under which concepts are associated.

The Project Management template ontology has the following concepts:

- Project that models a project
- Stakeholder that models project stakeholders
- Phase that models project phases
- Activity that models activities included in the project phases
- Group that models groups of activities
- Role that models roles responsible for project activities
- Member that models people involved in project activities (one or more roles are assigned to each member of the project)
- EmailAddress that models a project member's email address
- PostalAddress that models a project member's postal address
- TelephoneNumber that models a project member's telephone number

Each concept has name and description as default attributes. Additional attributes and their optional values (if needed) are defined in Tara ontologies as well.

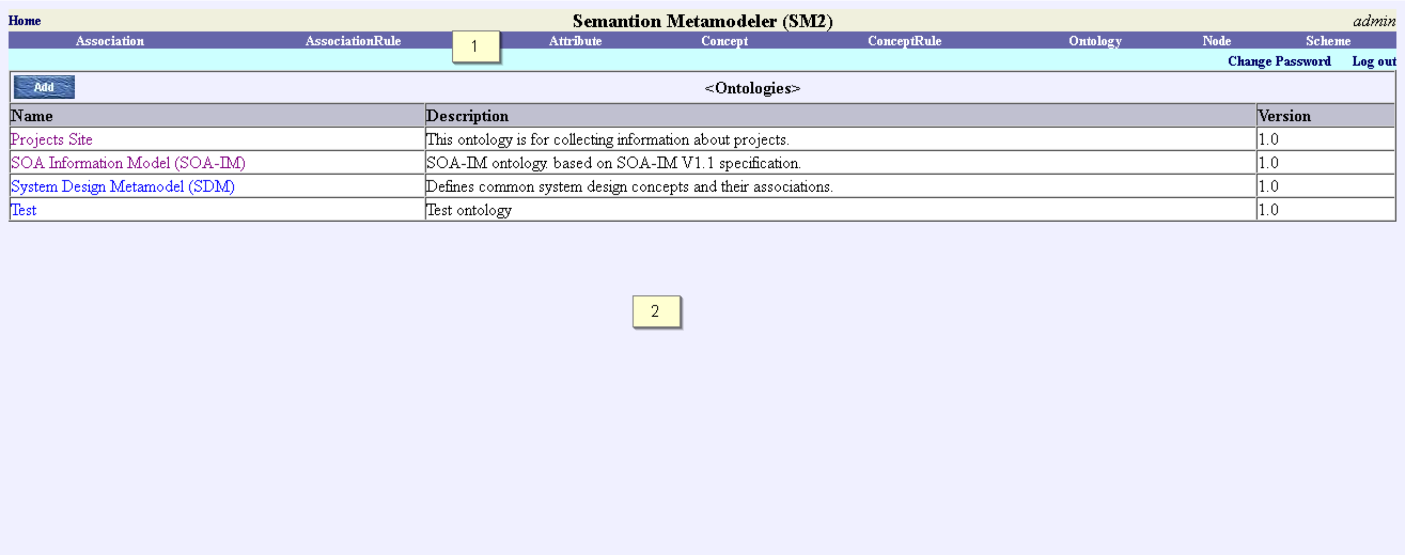
The table below contains a list of attributes for all concepts. Their optional values are listed as well.

Concept	**Attribute**	**Description of the attribute**	**Optional values**
Project	name description	Project name Project description	
Stakeholder	name description type	Stakeholder name Stakeholder description Stakeholder type	Group Organization Person

Phase	name	Phase name	
	description	Phase description	
	startDate	Start date	
	endDate	End date	
	percentageOfCompletion	Percentage of the total phase work that has been already completed	
Activity	name	Activity name	
	description	Activity description	
	startDate	Start date	
	endDate	End date	
	percentageOfCompletion	Percentage of the total activity work that has been already completed	
Group	name	Group name	
	description	Group description	
	startDate	Start date for the group of activities	
	endDate	End date for the group of activities	
	percentageOfCompletion	Percentage of the total group work that has been already completed	
Role	name	Role name	
	description	Role description	
Member	name	Member full name	
	description	Member description	
EmailAddress	name	Email address name	
	description	Email address description	
	address	Full email address	
	type	Email address type	Office Home
PostalAddress	name	PostalAddress name	
	description	PostalAddress description	
	streetNumber	PostalAddress street number	
	streetName	PotalAddress street name	
	city	PostalAddress city	
	stateOrProvince	PostalAddress state or province	
	postalCode	PostalAddress postal code	
	country	PostalAddress country	
TelephoneNumber	name	TelephoneNumber name	
	decsription	TelephoneNumber description	
	countryCode		
	areaCode	TelephoneNumber area code	
	number	TelephoneNumber main number	
	extension	TelephoneNumber extension	
	type	TelephoneNumber type	Office Cell Fax

Table 1: Project Ontology Concepts

First, you will access the Semantion Metamodeler tool. For example, if the web domain of the Semantion Business Platform (SBP) is *http://localhost:8080*, the SM2 URL will be *http://localhost:8080/sm2*. *Otherwise, use the appropriate URL accordingly.* If you have not created any additional user in SBP you can use the SBP *admin* user with a default password of *manager*. When you log in you will see the SM2 home page:



Name	Description	Version
Projects Site	This ontology is for collecting information about projects.	1.0
SOA Information Model (SOA-IM)	SOA-IM ontology. based on SOA-IM V1.1 specification.	1.0
System Design Metamodel (SDM)	Defines common system design concepts and their associations.	1.0
Test	Test ontology	1.0

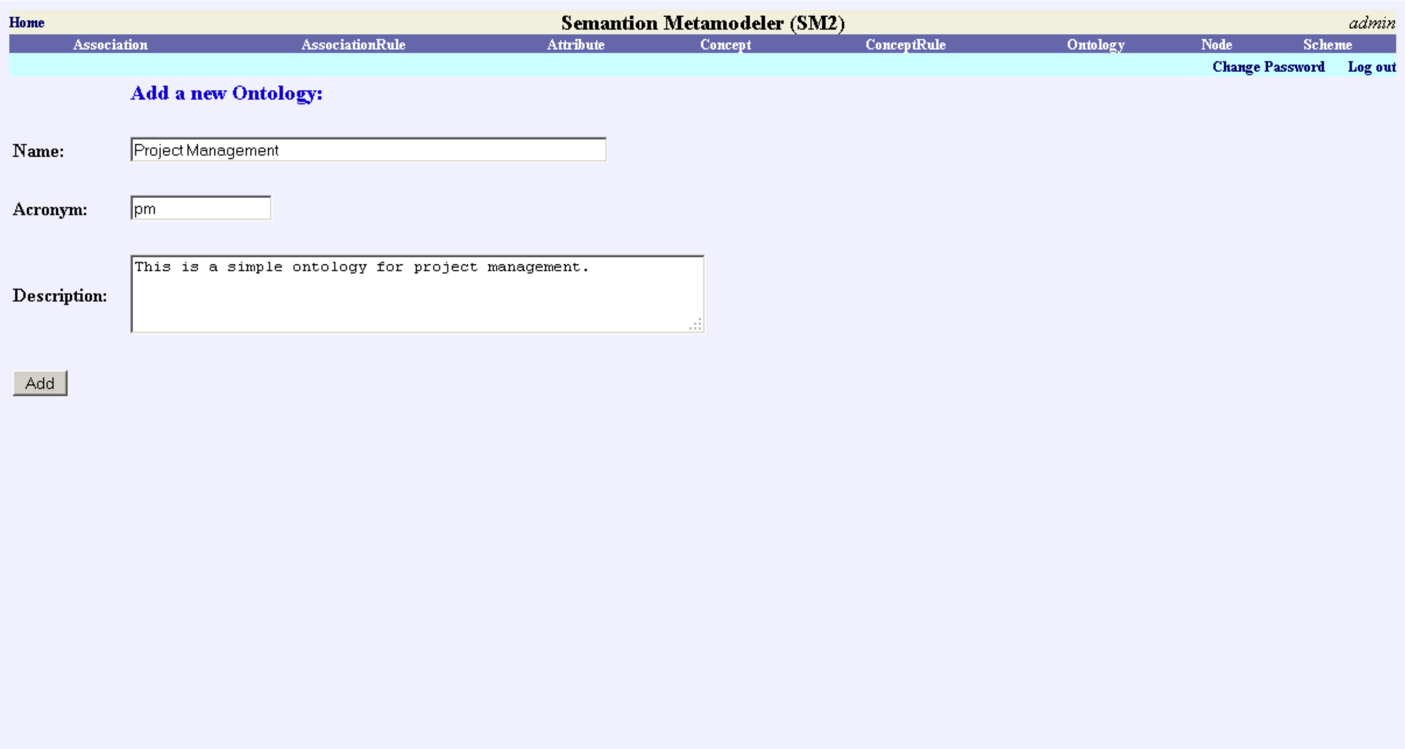
Figure 1: SM2 home page

SM2 has two panes:

(1) *Navigation* pane containing links for ontology elements (i.e., Association, AssociationRule, Attribute, Concept, etc.)
(2) *View* pane containing ontology elements and their detailed views

2.1.1 Create Project Management Ontology

The *Add* button in the View pane is used to create a new ontology. Click the *Add* button to start the creation of the *Project* Management ontology. *Add a new Ontology* page will be loaded. Enter the name and description for the new ontology (Fig. 2):



Add a new Ontology:

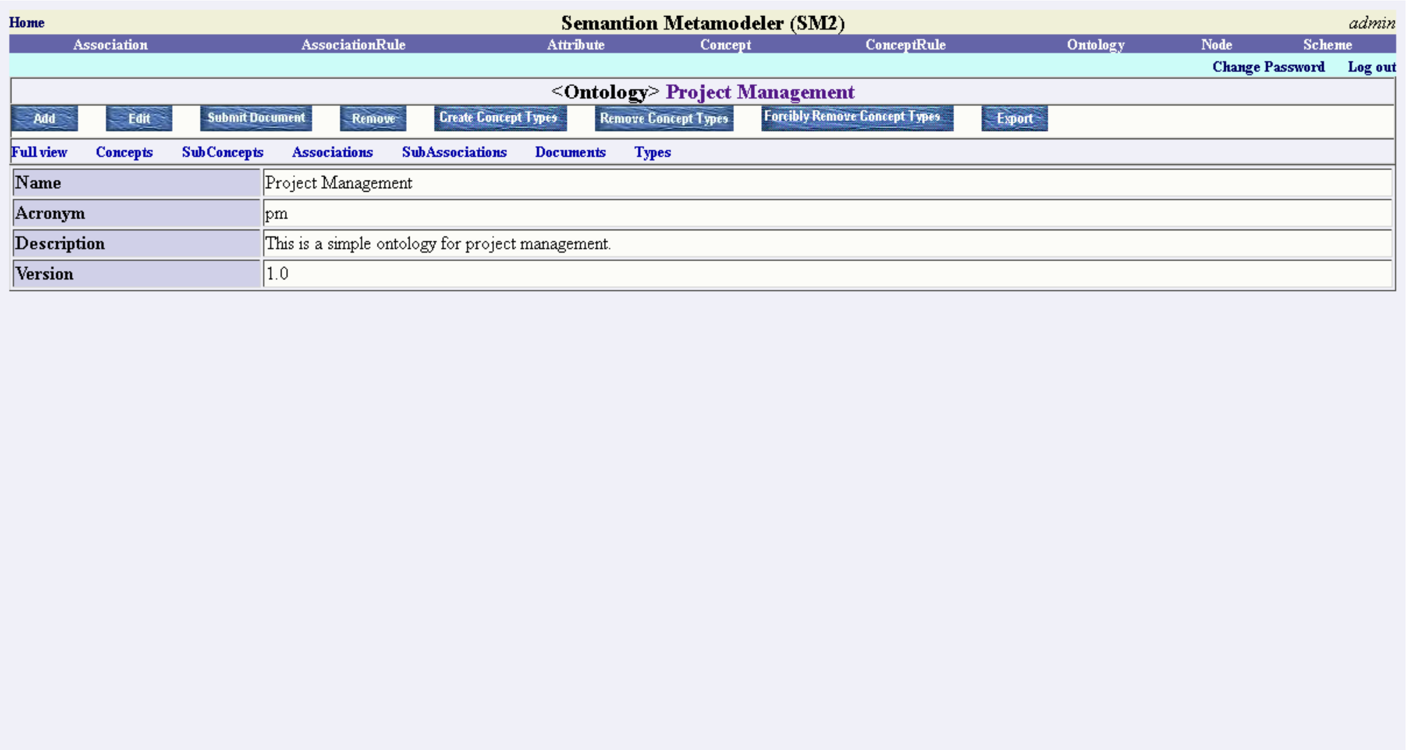
Name: Project Management

Acronym: pm

Description: This is a simple ontology for project management.

Figure 2: Project Management ontology creation

Besides standard name and description properties, the ontology has also acronym property that is used to specify the acronym of the ontology. The acronym property is mandatory. Click the *Add* button to create the *Project Management* ontology. When the *Project Management* ontology is created its detailed view page will be presented:



Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

Figure 3: Project Management ontology view

The ontology view pane has eight buttons and seven links.

Add, *Edit, Submit Documen*t, *Remove, Create Concept Types, Remove Concept Types, Forcibly Remove Concept Types, and Export* buttons are used for:

- *Add* to create new ontology.
- *Edit* to edit the ontology.
- *Submit Document* to submit a document to the repository and associate it with the ontology.
- Remove to remove the ontology.
- When all ontology's concepts are created, *Create Concept Types* button is used to create all concept types and their related folders in repository.
- *Remove Concept Types* removes all concept types and their related empty folders in the repository. Folders that contain files are not removed. They are renamed to their original name followed by a dash sign and the ID of the corresponding object type folder that is being removed.
- *Forcibly Remove Concept Types* removes all concept types and their folders.
- *Export* exports the Tara ontology in either ebXML RS/RIM or OWL 2 to a document that can be directly submitted to create the same ontology in another SM2 or another tool supporting OWL 2.

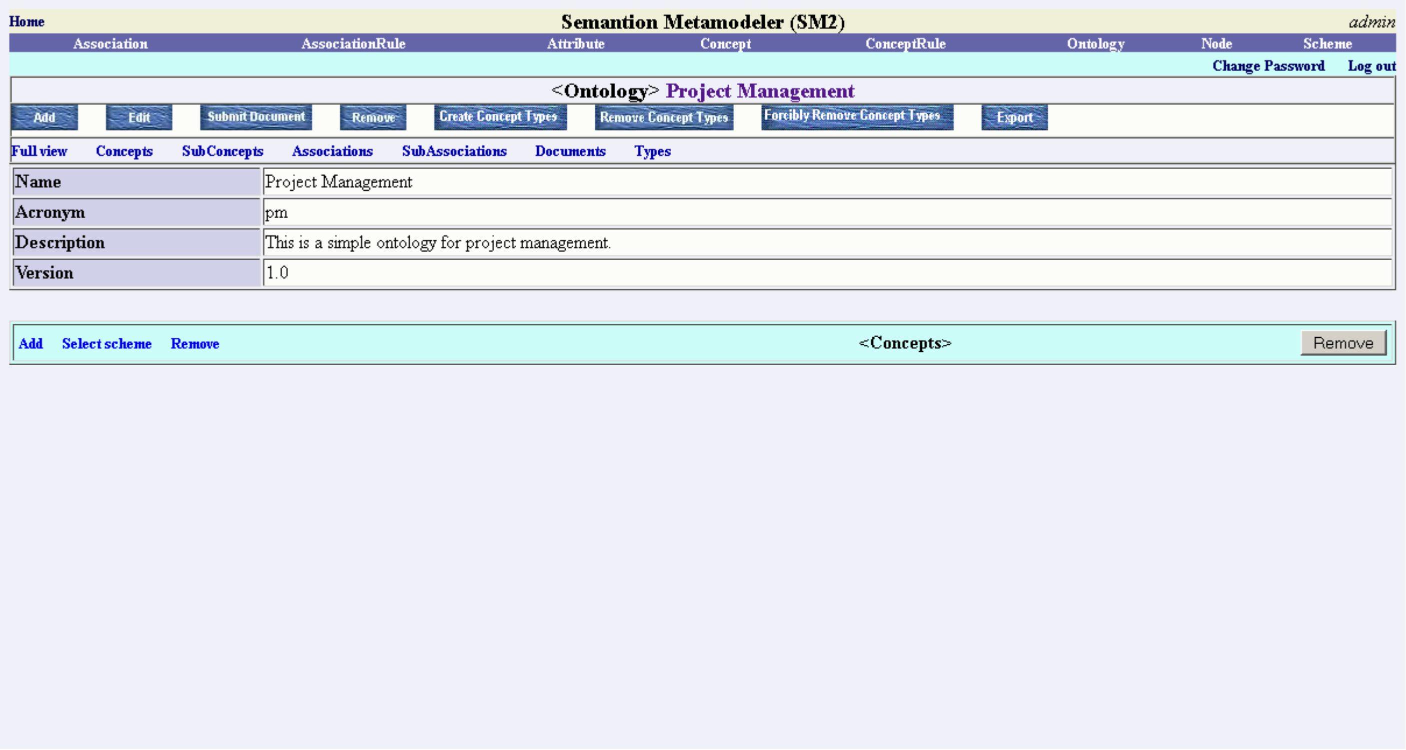
Full view, *Concepts*, SubConcepts, *Associations*, SubAssociations, *Documents*, and *Types* links are used for:

- *Full view* shows ontology properties and all its concepts, sub-concepts, associations, sub-associations, document types, and types for all attributes used in the ontology.
- *Concepts* lists all concepts defined in the ontology.
- *SubConcepts lists all sub-concepts in the ontology.*
- *Associations* lists all associations defined in the ontology.
- SubAssociations lists all sub-associations defined in the ontology.
- *Documents* lists all document types defined in the ontology
- *Types* lists all attribute types.

2.1.2 Create Concepts

Now that we have created an initial definition of the ontology, we will proceed with the creation of ontology concepts.

Click the *Concepts* link to load the following page:



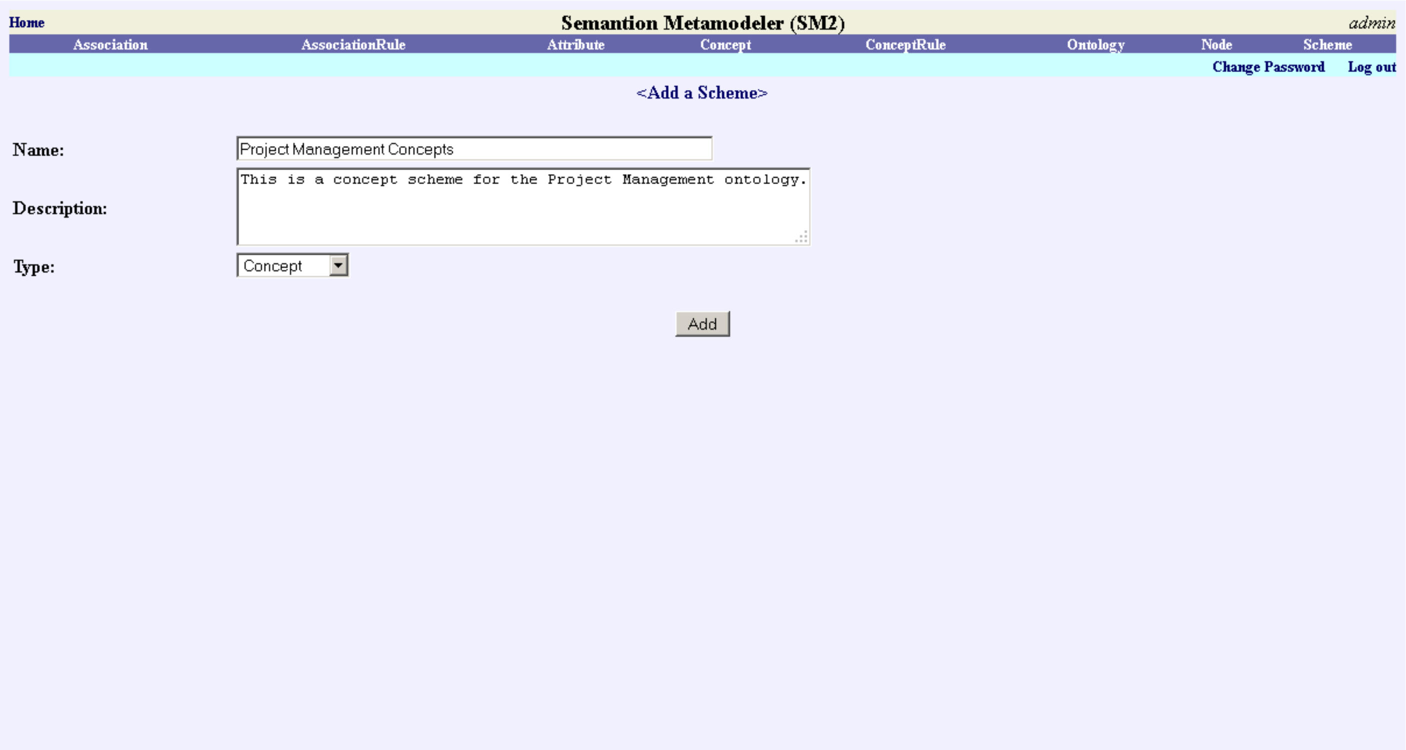
<Ontology> Project Management

Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

Figure 4: Concept scheme page

The Concepts section has three links and *Remove* button. The three links are: *Add*, *Select scheme*, and *Remove*. *Add* link loads the page that is used to create a Concept scheme. When the Concept scheme is created every next use of the *Add* link will load the page to create new Concept node that will be associated with the Concept scheme. *Select scheme* link enables selection of another already defined scheme. If we already have a scheme associated with the ontology, the selected scheme will replace the scheme that is already in use. *Remove* link will remove the *Concept* scheme from the ontology. The removed scheme will still be available for selection in this or any other ontology. *Remove* button removes selected node from the scheme.

Click the *Add* link (Fig. 5) to create the Project Management concept scheme:



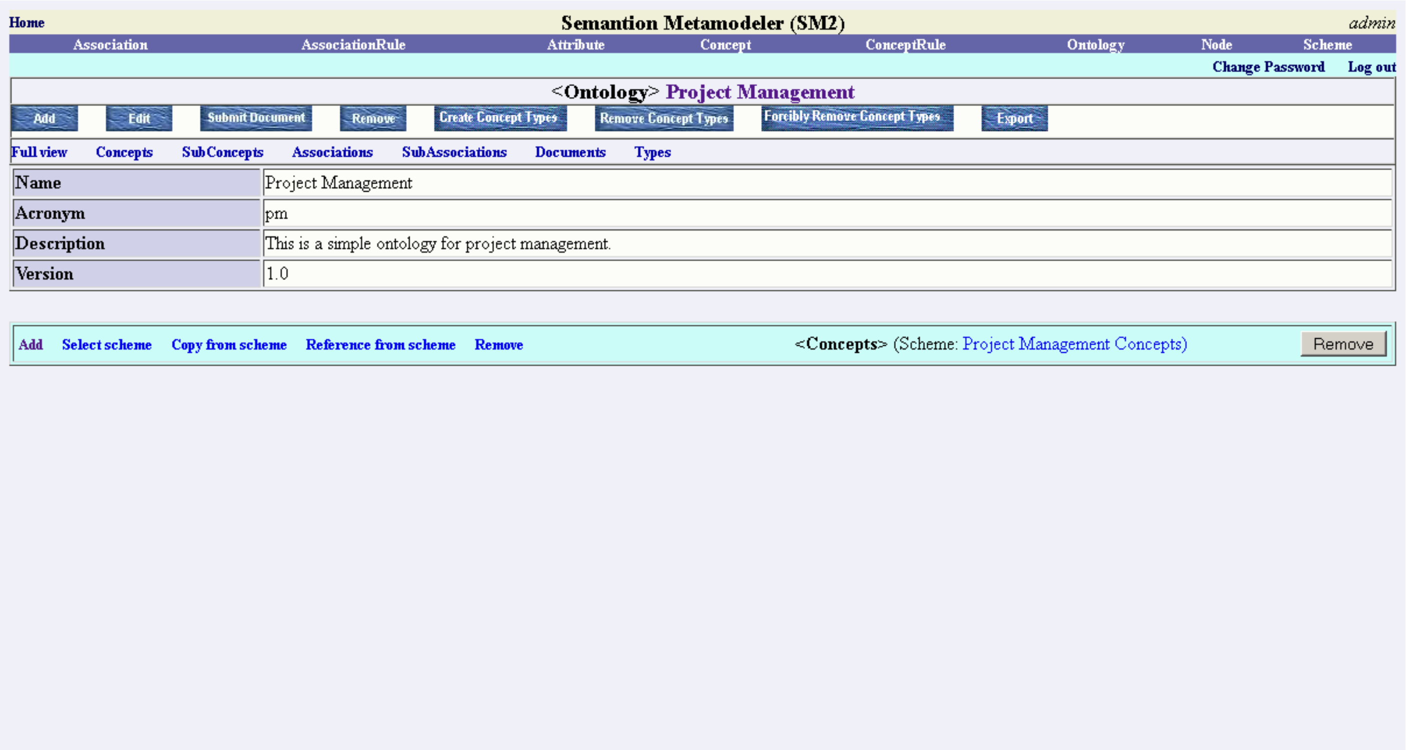
Name: Project Management Concepts

Description: This is a concept scheme for the Project Management ontology.

Type: Concept

Figure 5: Add concept scheme

It is important to make sure that the *type* property has the value *Concept* since this is the concept scheme. The following page will be loaded when the scheme is created:



<Ontology> Project Management

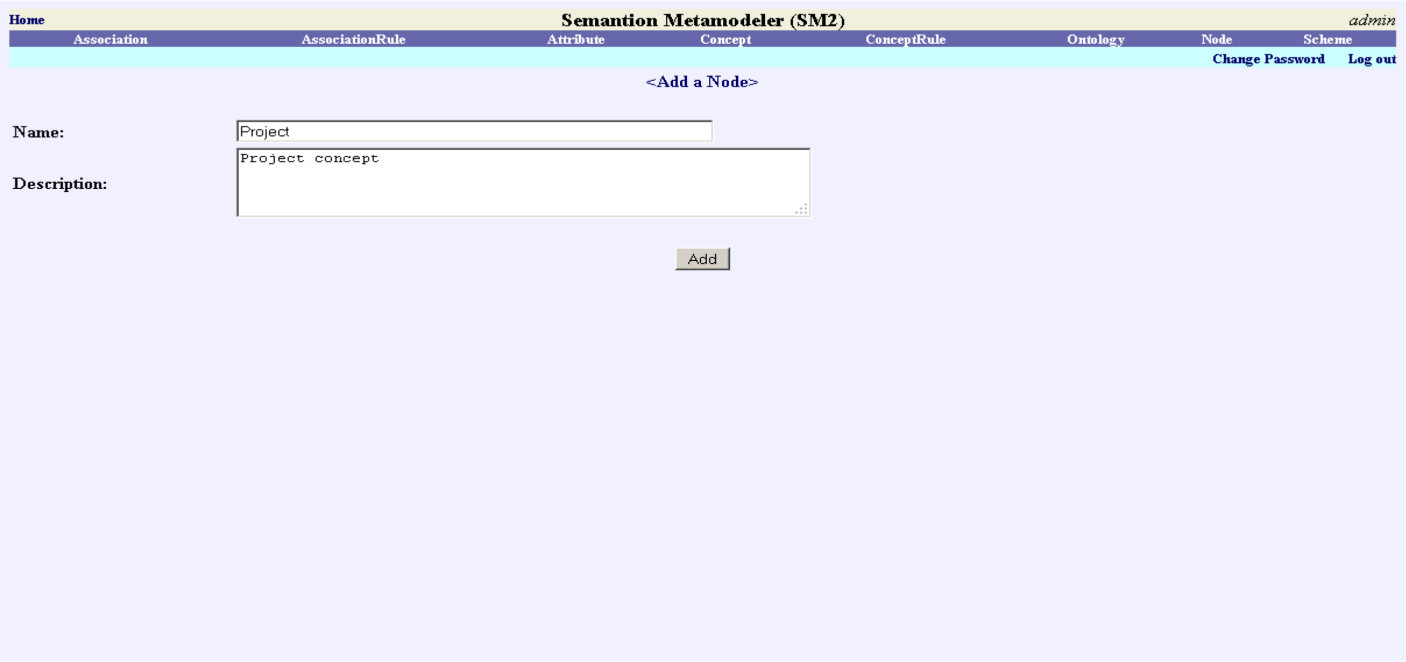
Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

<Concepts> (Scheme: Project Management Concepts)

Figure 6: Detailed ontology view with empty concept scheme

Now when concept scheme is created you can see additional links *Copy from scheme* and *Reference from scheme. Copy from* *scheme* enables addition of concepts from another concept schemes. The selected concept is transparently copied to the current ontology. Instead of creating a concept that already exists in another ontology we can just select it from any other scheme and reference it in the ontology via *Reference from scheme* link.

When you click the Add link, the concept node page will be loaded. Enter *Project* for the name property, *Project concept* for the description property, and click the *Add* button to create the Project concept node (Fig. 7).

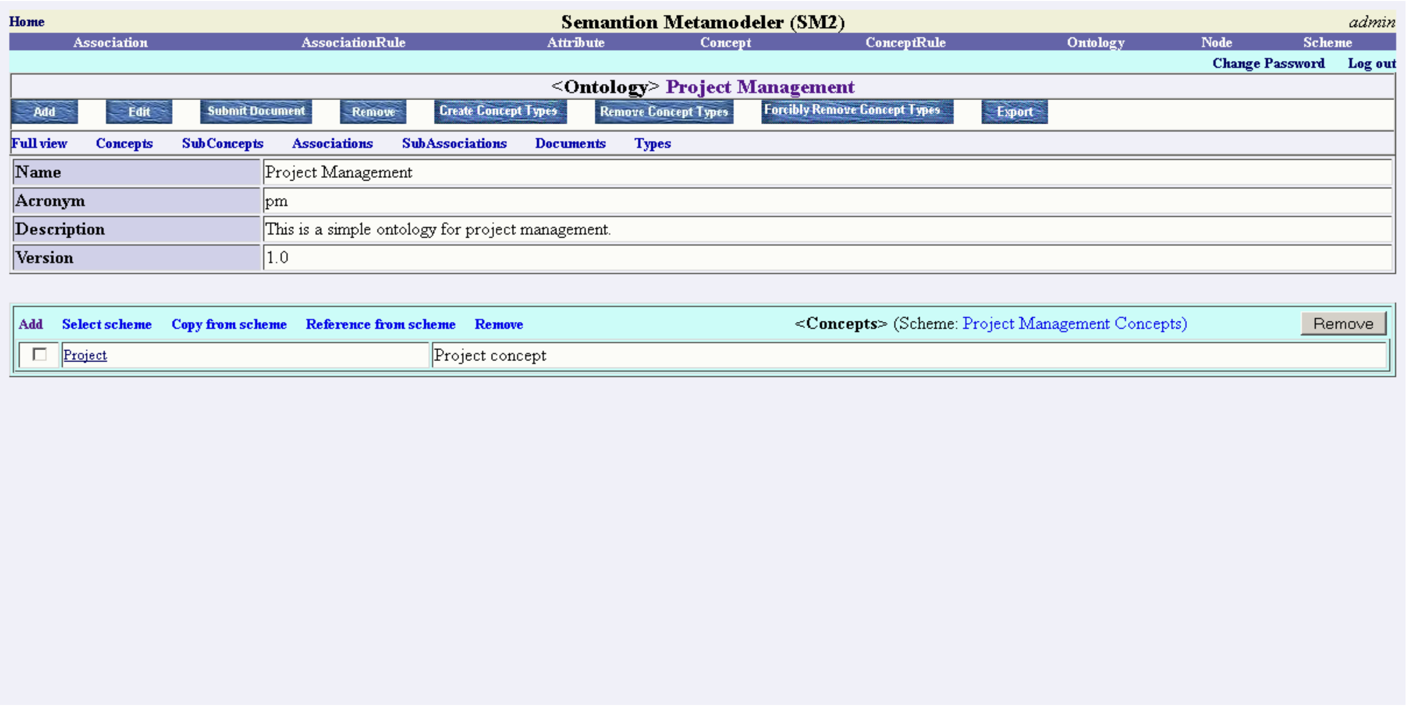


Name: Project

Description: Project concept

Figure 7: Add Project concept node to the Concept scheme

This is how the ontology view with concept details will look like when the *Project* concept is added:



Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

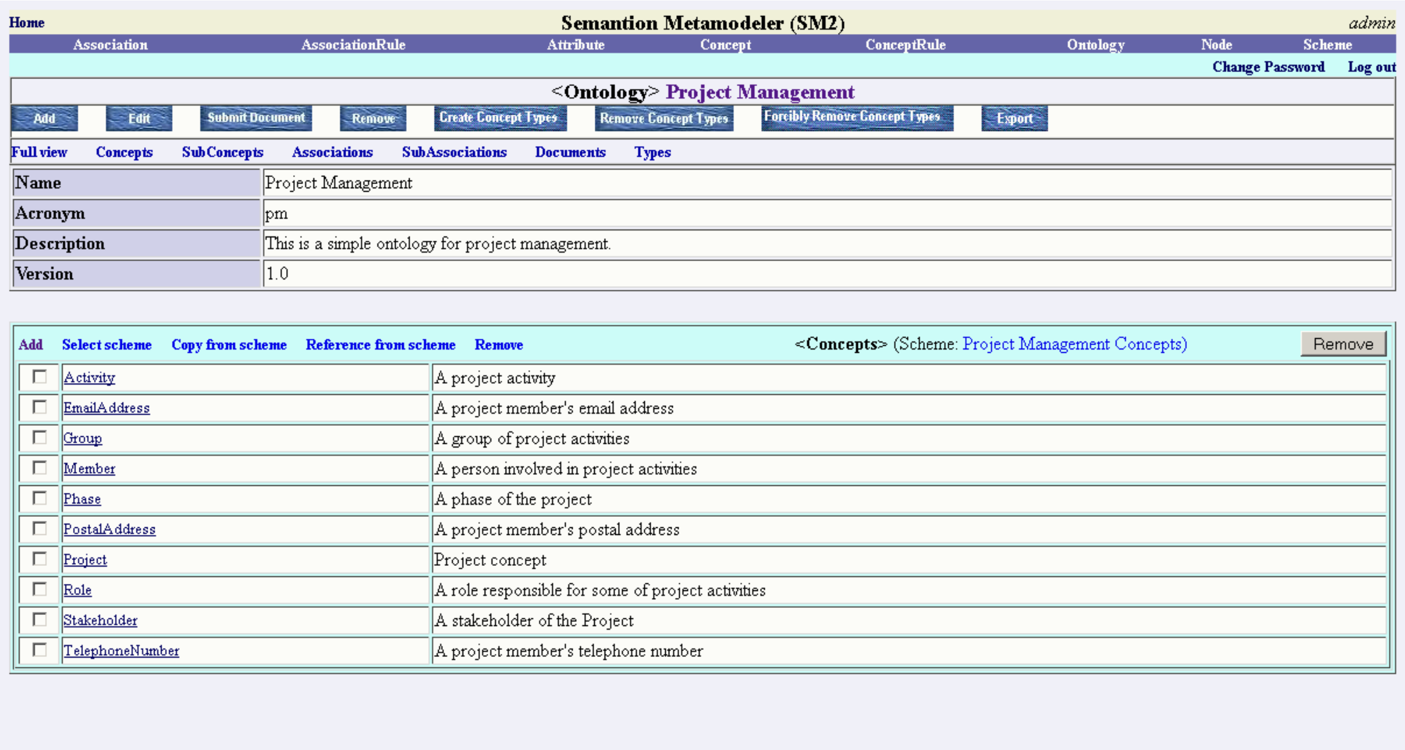
<Concepts> (Scheme: Project Management Concepts)

□	Project	Project concept

Figure 8: Ontology view with concept scheme

Other concepts (Stakeholder, *Phase*, *Activity*, and others) can be created the same way.

This is how the Project Management ontology concepts view will look like when all concepts are added to the Concept scheme:



<Ontology> Project Management

Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

<Concepts> (Scheme: Project Management Concepts)

□	Activity	A project activity
□	EmailAddress	A project member's email address
□	Group	A group of project activities
□	Member	A person involved in project activities
□	Phase	A phase of the project
□	PostalAddress	A project member's postal address
□	Project	Project concept
□	Role	A role responsible for some of project activities
□	Stakeholder	A stakeholder of the Project
□	TelephoneNumber	A project member's telephone number

Figure 9: Concepts

Before we continue with the creation of the Association scheme we will first define all associations that will be created in the ontology:

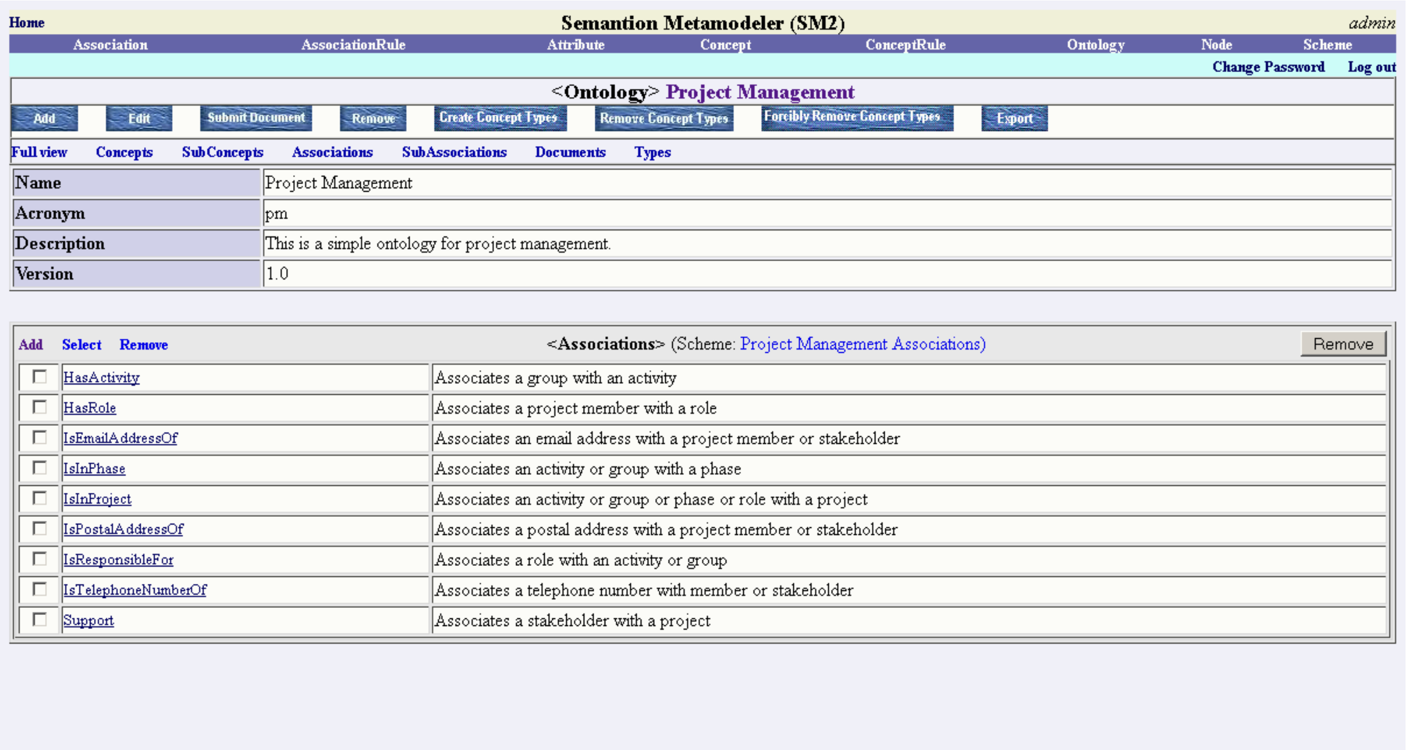
Type of the source concept	Association	Type of the target concept
Member	HasRole	Role
Activity/Group/Phase/Role	IsInProject	Project
Activity/Group	IsInPhase	Phase
Group	HasActivity	Activity
Role	IsResponsibleFor	Activity/Group
PostalAddress	IsPostalAddressOf	Member/Stakeholder
TelephoneNumber	IsTelephoneNumberOf	Member/Stakeholder
EmailAddress	IsEmailAddressOf	Member/Stakeholder
Stakeholder	Support	Project

Table 2: Project Ontology Associations

The Association scheme and Association nodes are created the same way as the Concept scheme and Concept nodes except that the type of the Association scheme is *Association*.

For example, according to *Table 2*, Group, Activity, Phase and Role are associated to Project using the same Association *IsInProject*. This is a case when more than one association rule needs to be defined to handle different pairs of concepts that will be associated by the same association.

This is how the Project Management ontology association view will look like when all associations are added to the Association scheme:



<Ontology> Project Management

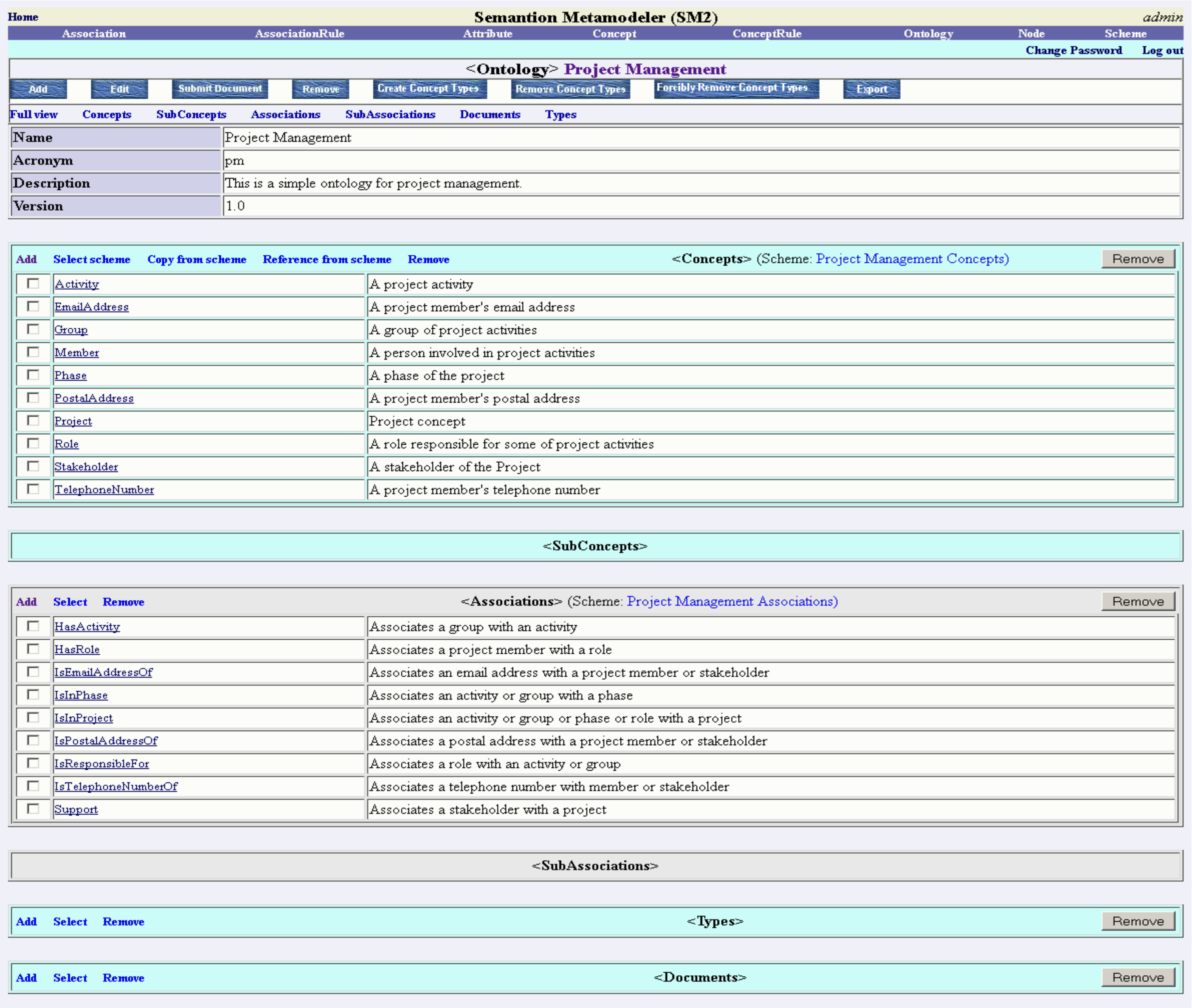
Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

<Associations> (Scheme: Project Management Associations)

□	HasActivity	Associates a group with an activity
□	HasRole	Associates a project member with a role
□	IsEmailAddressOf	Associates an email address with a project member or stakeholder
□	IsInPhase	Associates an activity or group with a phase
□	IsInProject	Associates an activity or group or phase or role with a project
□	IsPostalAddressOf	Associates a postal address with a project member or stakeholder
□	IsResponsibleFor	Associates a role with an activity or group
□	IsTelephoneNumberOf	Associates a telephone number with member or stakeholder
□	Support	Associates a stakeholder with a project

Figure 10: Associations

This is how the full view of the Project Management ontology will look like when Concept scheme and Association scheme are created:


Home
Semantion Metamodeler (SM2)
admin
Association AssociationRule Attribute Concept ConceptRule Ontology Node Scheme
Change Password Log out
Add Edit Submit Document Remove Create Concept Types Remove Concept Types Forcibly Remove Concept Types Export
Full view Concepts SubConcepts Associations SubAssociations Documents Types
Add Select scheme Copy from scheme Reference from scheme Remove
Remove
Add Select Remove
Remove
Add Select Remove
Remove
Add Select Remove
Remove

<Ontology> Project Management

Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

<Concepts> (Scheme: Project Management Concepts)

□	Activity	A project activity
□	EmailAddress	A project member's email address
□	Group	A group of project activities
□	Member	A person involved in project activities
□	Phase	A phase of the project
□	PostalAddress	A project member's postal address
□	Project	Project concept
□	Role	A role responsible for some of project activities
□	Stakeholder	A stakeholder of the Project
□	TelephoneNumber	A project member's telephone number

<SubConcepts>

<Associations> (Scheme: Project Management Associations)

□	HasActivity	Associates a group with an activity
□	HasRole	Associates a project member with a role
□	IsEmailAddressOf	Associates an email address with a project member or stakeholder
□	IsInPhase	Associates an activity or group with a phase
□	IsInProject	Associates an activity or group or phase or role with a project
□	IsPostalAddressOf	Associates a postal address with a project member or stakeholder
□	IsResponsibleFor	Associates a role with an activity or group
□	IsTelephoneNumberOf	Associates a telephone number with member or stakeholder
□	Support	Associates a stakeholder with a project

<SubAssociations>

<Types>

<Documents>

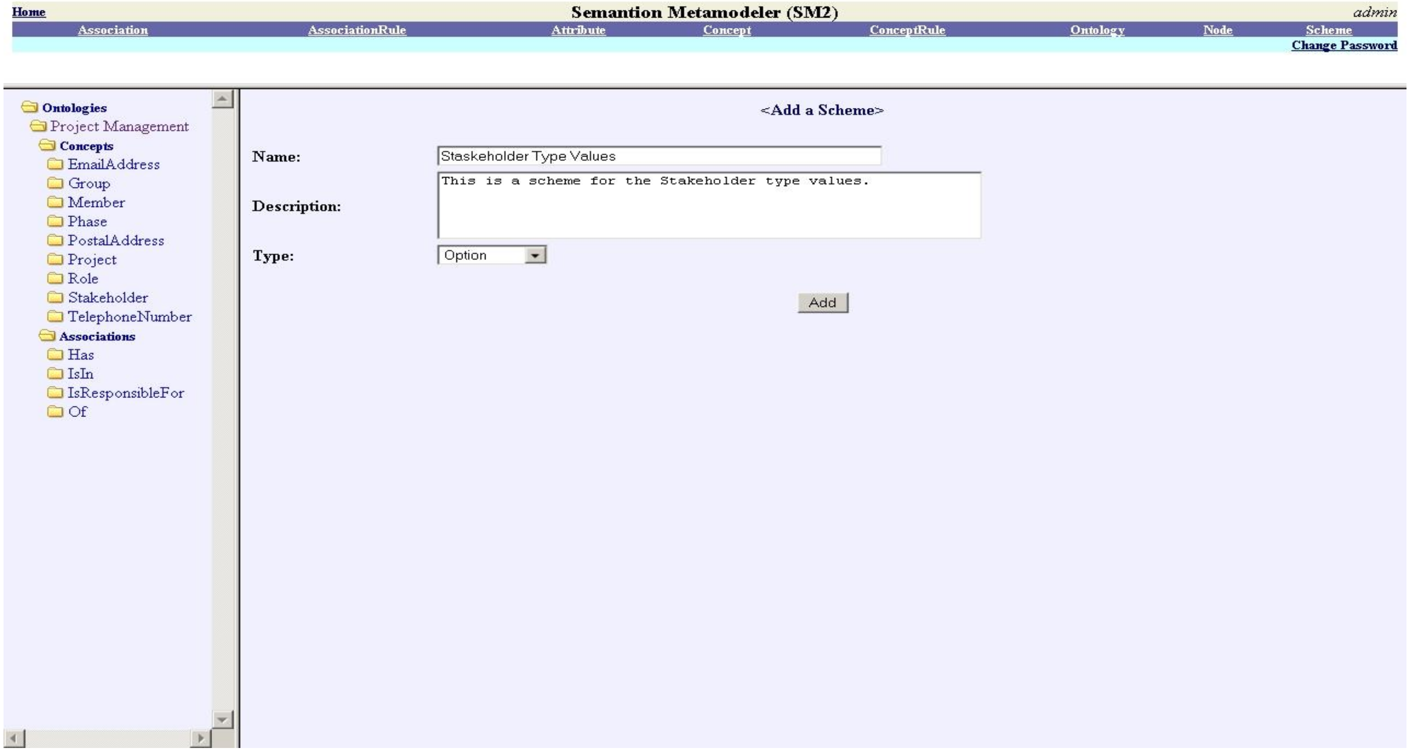
Figure 11: Ontology view with all schemes created

2.1.3 Create Options

The *Option* scheme is used to add pre-defined optional values for attributes. Each attribute with options will have its own *Option* scheme and that is why more than one *Option* scheme can be used in an ontology. We do not list the *Option* schemes in the detailed ontology view.

You can create an *Option* scheme either via the *Scheme* link in the *Navigation* pane or via the *Add* link in the *Attribute* view. This section explains how to create an *Option* scheme via the *Scheme* link in the *Navigation* pane.

When you add an *Option* scheme via the *Scheme* link in the *Navigation* pane, a page that lists all schemes will be loaded. Click the *Add* button above the list and enter information about the *Stakeholder Type Values* scheme that contains nodes (options): *Group*, *Organization*, and *Person*.



<Add a Scheme>

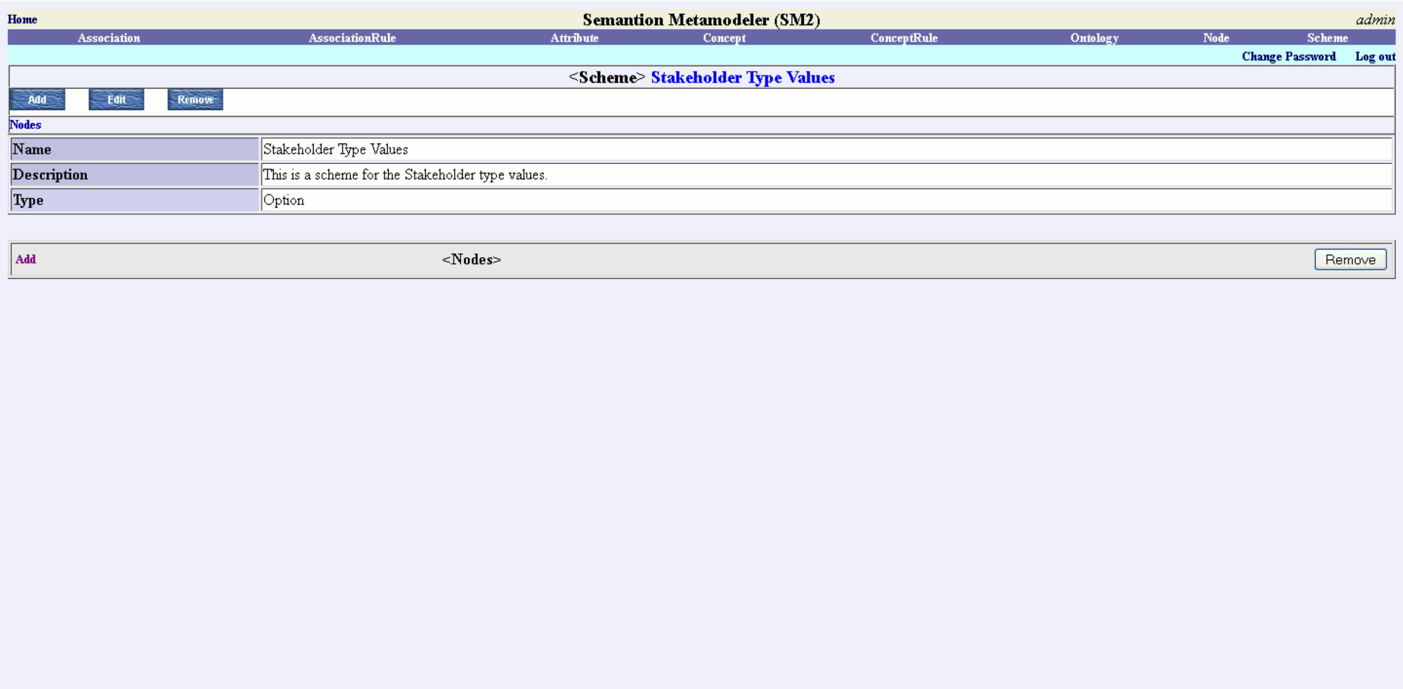
Name: Staskeholder Type Values

Description: This is a scheme for the Stakeholder type values.

Type: Option

Figure 12: Create Stakeholder Type Values option scheme

Enter the scheme’s name and description and make sure that the scheme’s type is *Option*. Click the *Add* button to create the scheme. When the scheme is created it will be listed on the next loaded page. Click the link for the *Stakeholder Type Values* scheme and then click the *Nodes* link to start adding nodes that will represent values related to this option. Click the *Add* link to add a node:

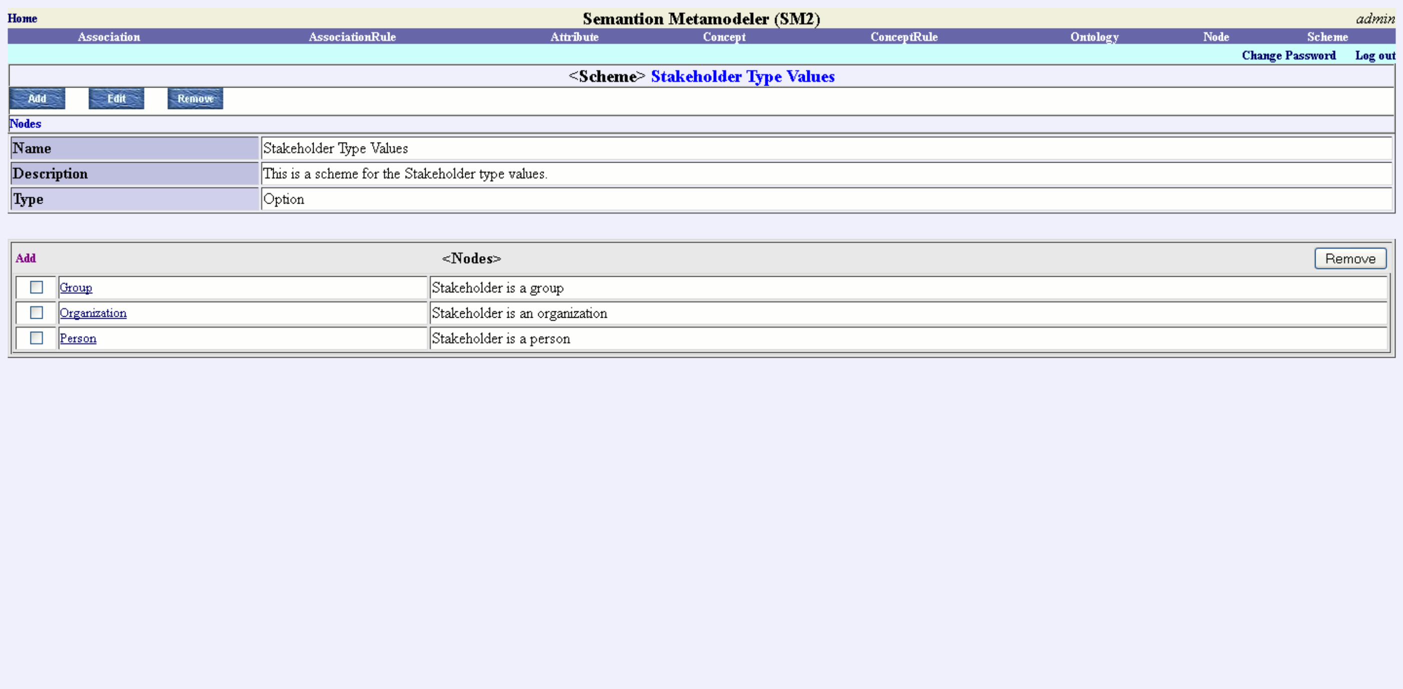


<Scheme> Stakeholder Type Values

Name	Stakeholder Type Values
Description	This is a scheme for the Stakeholder type values.
Type	Option

Figure 13: Add nodes to StakeholderTypeValues option scheme

When all nodes are added, the detailed view of the *Stakeholder Type Values* scheme will look like this:



<Scheme> Stakeholder Type Values

Name	Stakeholder Type Values
Description	This is a scheme for the Stakeholder type values.
Type	Option

<Nodes>

□	Group	Stakeholder is a group
□	Organization	Stakeholder is an organization
□	Person	Stakeholder is a person

Figure 14: Detailed view of the Stakeholder Type Values option scheme

Two other option schemes are *EmailAddress Type Values and TelephoneNumber Type Values.* They are created the same way as *Stakeholder TypeValues.* Optional values for *EmailAddress* and *TelephoneNumber* concept types are defined in Table 1.

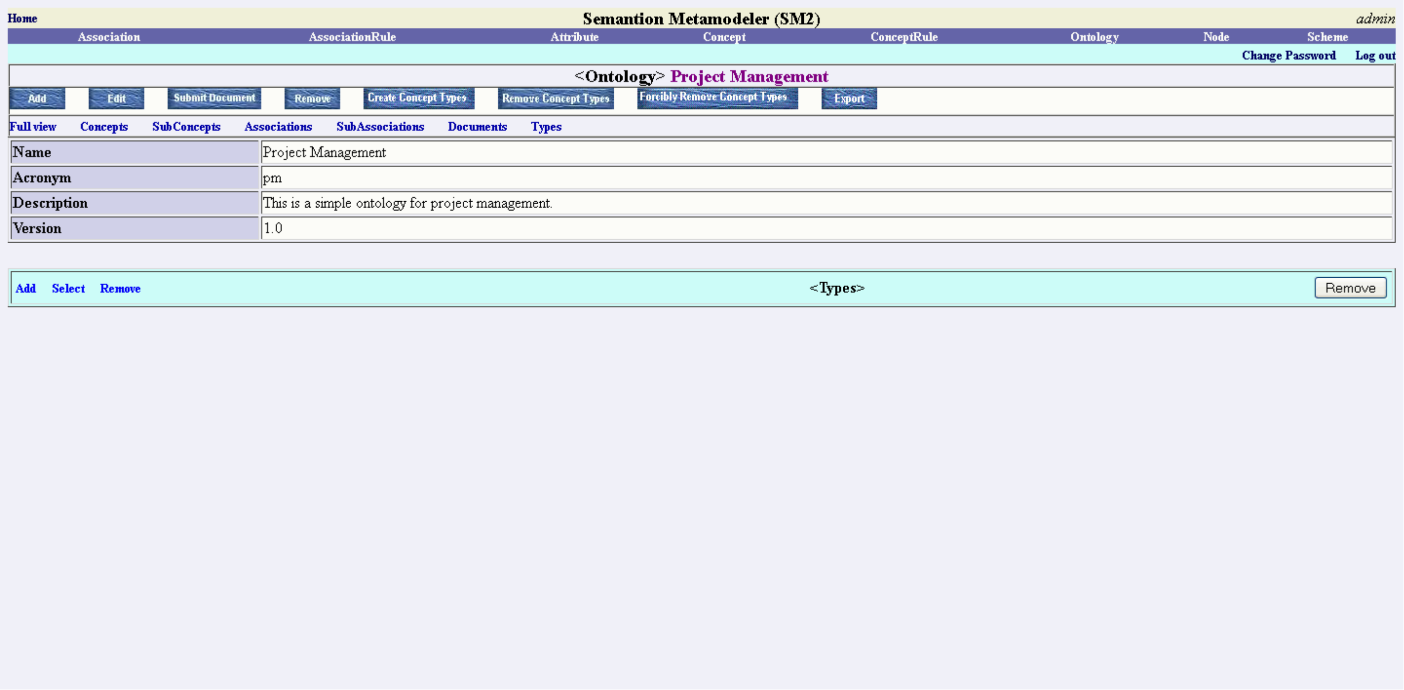
As mentioned above, the new *Option* scheme can also be added via the *Add* link in the *Attribute* view. For example, if you already created an attribute, you can load its view and under the *Options* section (view) you will see three links: *Add*, *Select*, and *Remove*. The *Add* link either creates a new *Option* scheme (and associates it with the attribute) if it does not exist or adds an option node to the scheme if the scheme already exists. The *Select* link selects another *Option* scheme for the attribute. The *Remove* link removes the association between the *Option* scheme and the attribute.

2.1.4 Attribute Types

All concepts' attributes have type *String* except the *startDate* and *endDate* attributes of the *Activity* and *Group* concepts that have type *Date*.

Semantion provides built-in types scheme *Semantion-SM2-Types that* defines all data types for the Project Management ontology attributes.

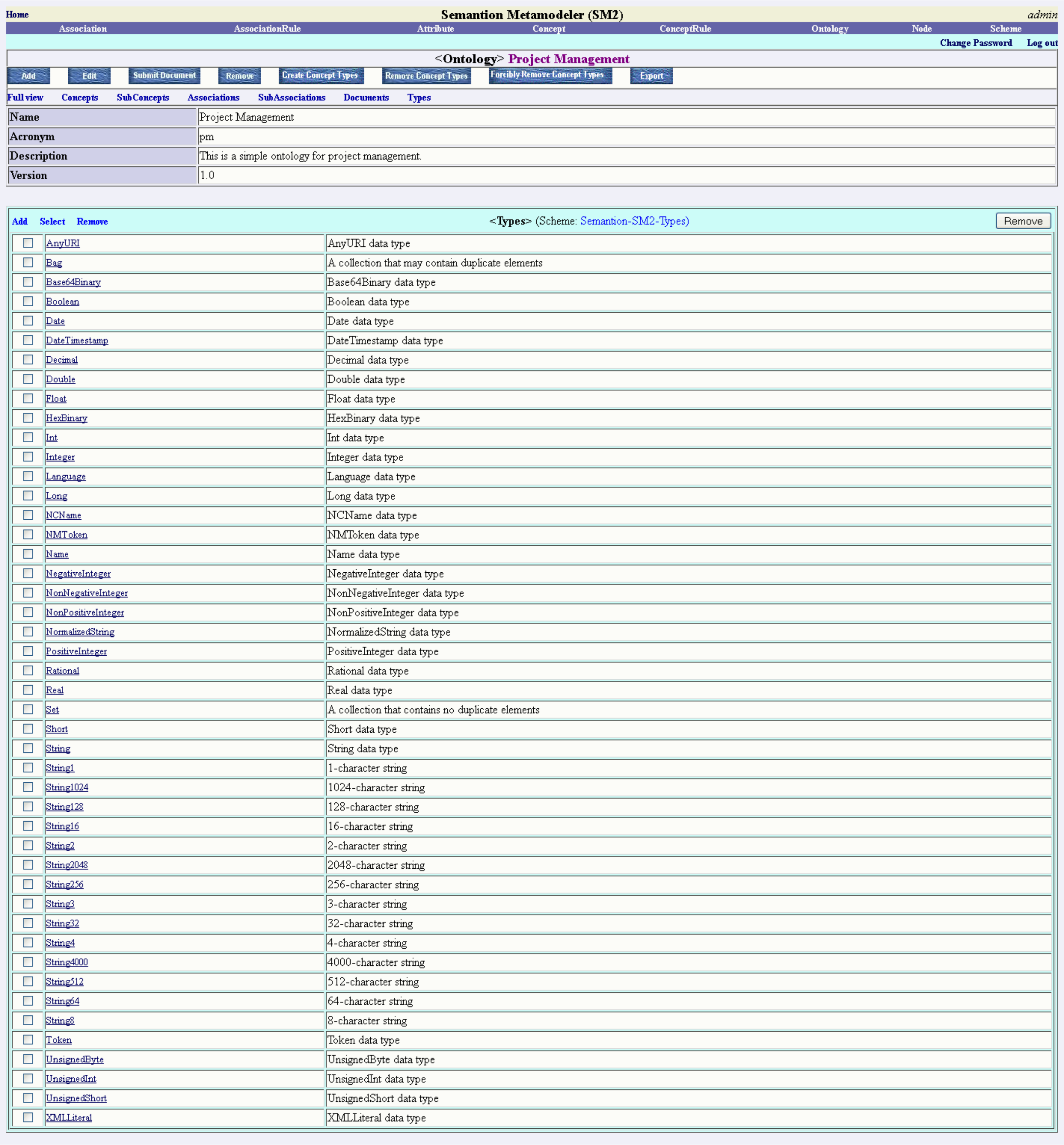
While you have an option to add your own types scheme we recommend the use of the built-in *Semantion-SM2-Types* scheme*.* To select the scheme, click the Types link in the ontology view and then click the Select link to select the *Semantion-SM2-Types* scheme*:*



Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

Figure 15: Create attribute types scheme

When you click the Select button, the *"Select a Scheme"* page will be loaded. Instead of providing scheme's name in the name field you can just click the *Submit* button and the list of types schemes will be loaded. Select the *Semantion-SM2-Types* and click the *Submit* button. The *Project Management* ontology view page with a list of all nodes under the types scheme will be loaded:



Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

□	AnyURI	AnyURI data type
□	Bag	A collection that may contain duplicate elements
□	Base64Binary	Base64Binary data type
□	Boolean	Boolean data type
□	Date	Date data type
□	DateTimestamp	DateTimestamp data type
□	Decimal	Decimal data type
□	Double	Double data type
□	Float	Float data type
□	HexBinary	HexBinary data type
□	Int	Int data type
□	Integer	Integer data type
□	Language	Language data type
□	Long	Long data type
□	NCName	NCName data type
□	NMToken	NMToken data type
□	Name	Name data type
□	NegativeInteger	NegativeInteger data type
□	NonNegativeInteger	NonNegativeInteger data type
□	NonPositiveInteger	NonPositiveInteger data type
□	NormalizedString	NormalizedString data type
□	PositiveInteger	PositiveInteger data type
□	Rational	Rational data type
□	Real	Real data type
□	Set	A collection that contains no duplicate elements
□	Short	Short data type
□	String	String data type
□	String1	1-character string
□	String1024	1024-character string
□	String128	128-character string
□	String16	16-character string
□	String2	2-character string
□	String2048	2048-character string
□	String256	256-character string
□	String3	3-character string
□	String32	32-character string
□	String4	4-character string
□	String4000	4000-character string
□	String512	512-character string
□	String64	64-character string
□	String8	8-character string
□	Token	Token data type
□	UnsignedByte	UnsignedByte data type
□	UnsignedInt	UnsignedInt data type
□	UnsignedShort	UnsignedShort data type
□	XMLLiteral	XMLLiteral data type

Figure 16: Data types

A data type must be specified for each attribute added (see Section 2.1.5). String data types include *String* or *String<number>* where *<number>* defines the maximum number of characters in the string. If *String* is specified as the data type, its maximum length is defined by the integer data type maximum value (2147483647) and constrained by available virtual memory.

At this point we demonstrated the first five steps in a ontology definition using SM2:

1. Create a ontology
2. Define concept types

3. Define association types
4. Define attribute types
5. Define optional property values (options) if needed

The next five steps include:

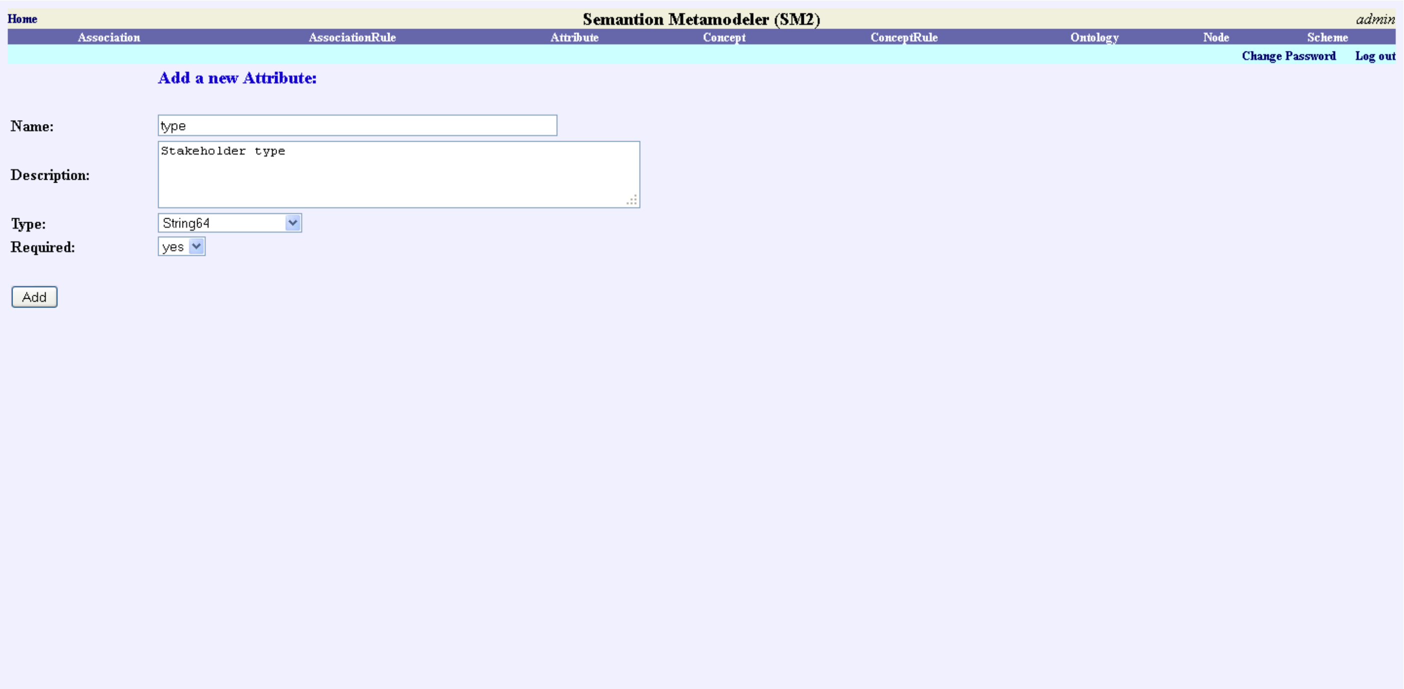
6. Define additional properties (attributes) for concepts where needed
7. Define association rules
8. Define concept rules
9. Define document types
10. Create concept types in registry

We will proceed with Steps 6 and 7. Steps 8 and 9 are not used in this example. They address the creation of concept rules and definition of all document types used in the application.

Step 10 belongs to creation of all concept types defined in the ontology. These types are created in the registry and are referenced in the application.

2.1.5 Create Attributes

Step 6 defines additional properties used in concepts. We will demonstrate how *type* can be added to the *Stakeholder* concept via the *Concept* SM2 page. First click the *Concept* link in the Navigation pane, then click the *Stakeholder* concept in the concepts list and when the *Stakeholder* view is presented click the *Attributes* link that will open the attributes related view with the *Add* link to create new attributes representing additional properties for the *Stakeholder* concept. Fig. 17 below shows data for the *Stakeholder's type* property:



Add a new Attribute:

Name: type

Description: Stakeholder type

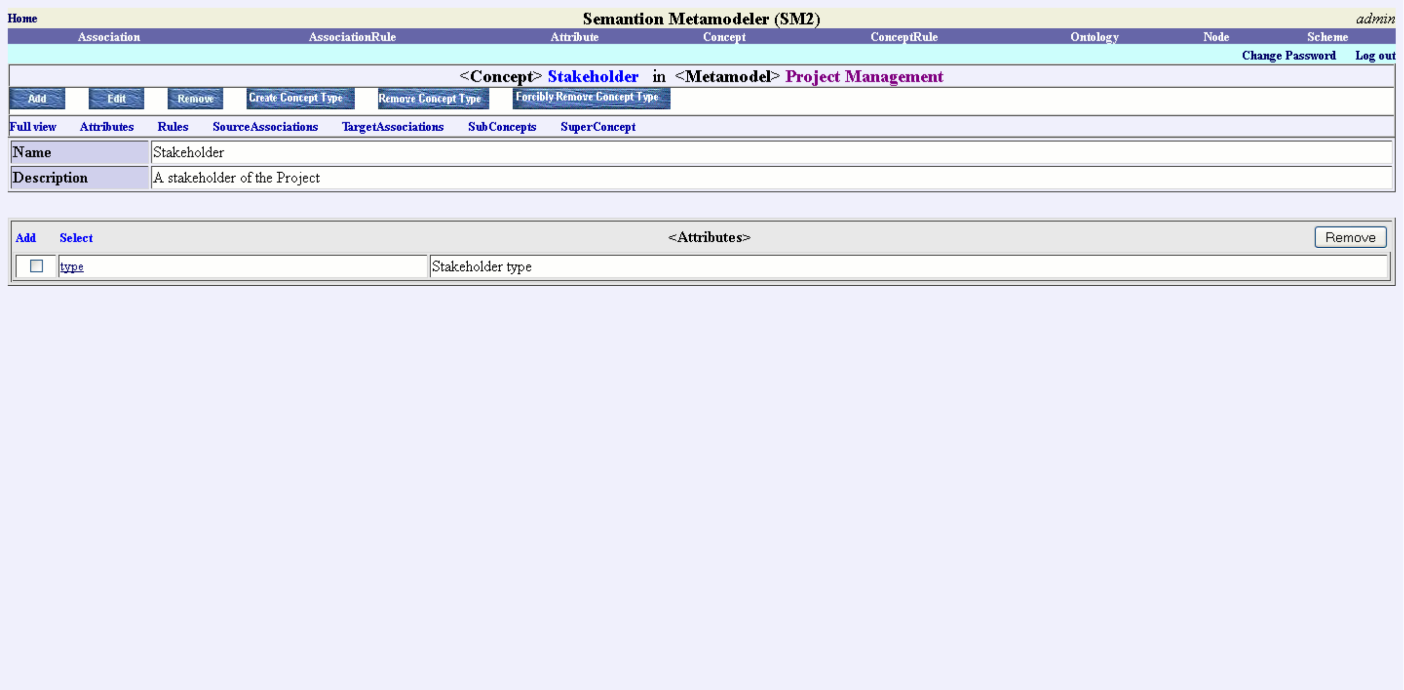
Type: String64

Required: yes

Figure 17: Add a new attribute page

Besides the *name* and *description* property for the attribute, two other properties are also presented: *type* and *required*. The *type* property specifies the data type of the property represented by the attribute while the *required* property specifies whether the value of the property represented by the attribute is mandatory. Click the Add button to create the attribute.

This is how the *Stakeholder* attributes-related view will look like when the type attribute is added:



<Concept> Stakeholder in <Metamodel> Project Management

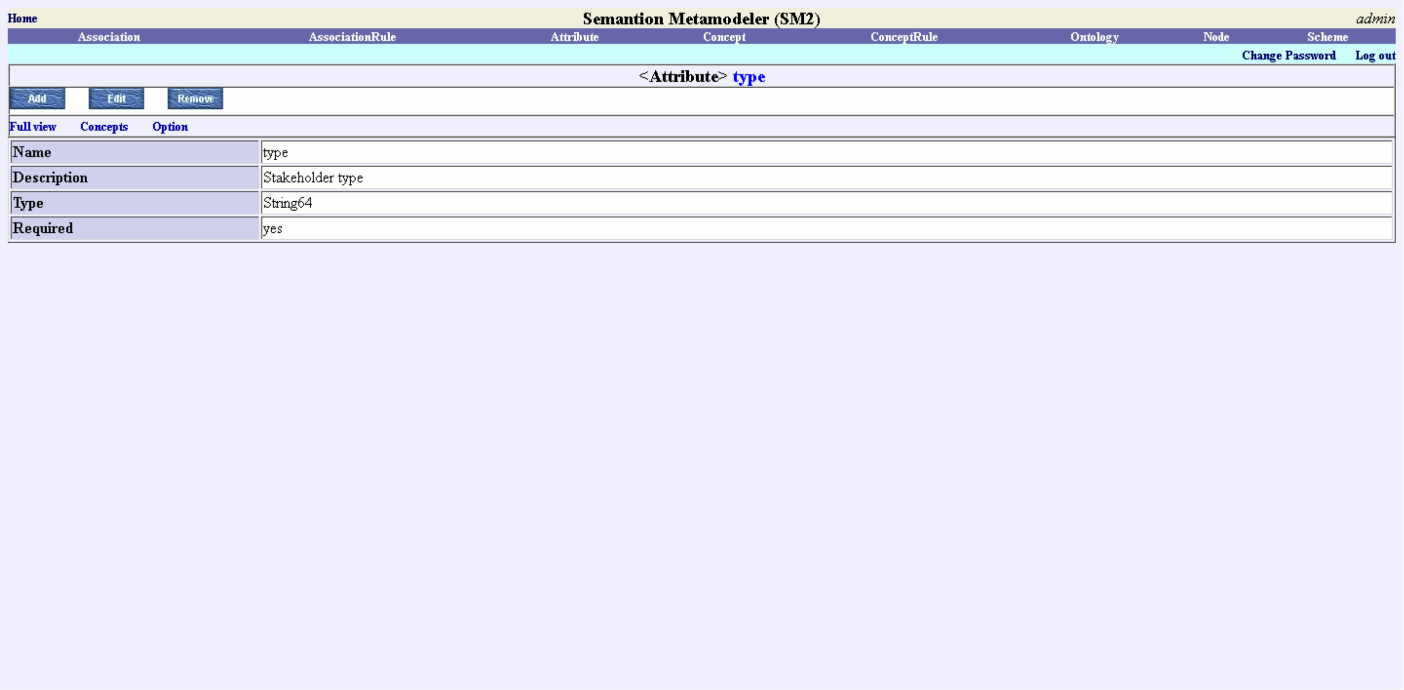
Name	Stakeholder
Description	A stakeholder of the Project

<Attributes>

□	type	Stakeholder type

Figure 18: Detailed attributes view of the Stakeholder concept

Now click the *type* link representing the attribute to open the attribute view page:

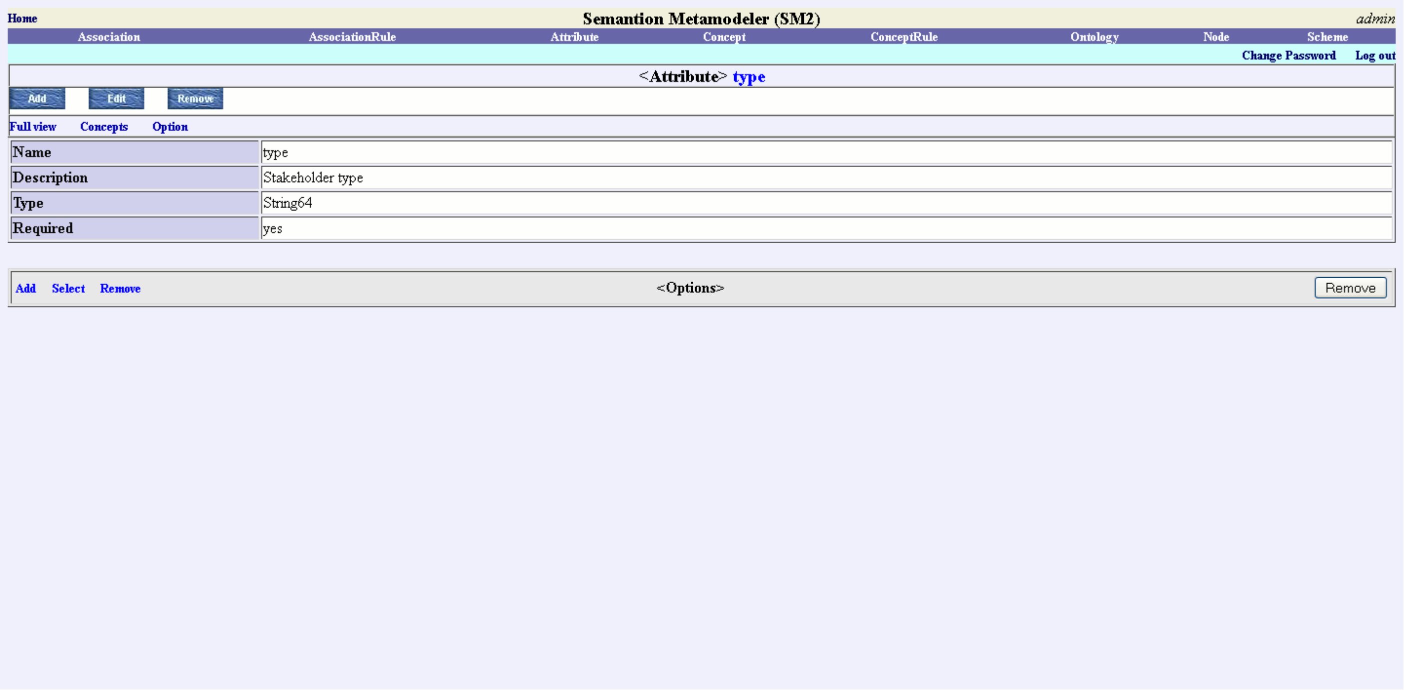


<Attribute> type

Name	type
Description	Stakeholder type
Type	String64
Required	yes

Figure 19: Attribute page

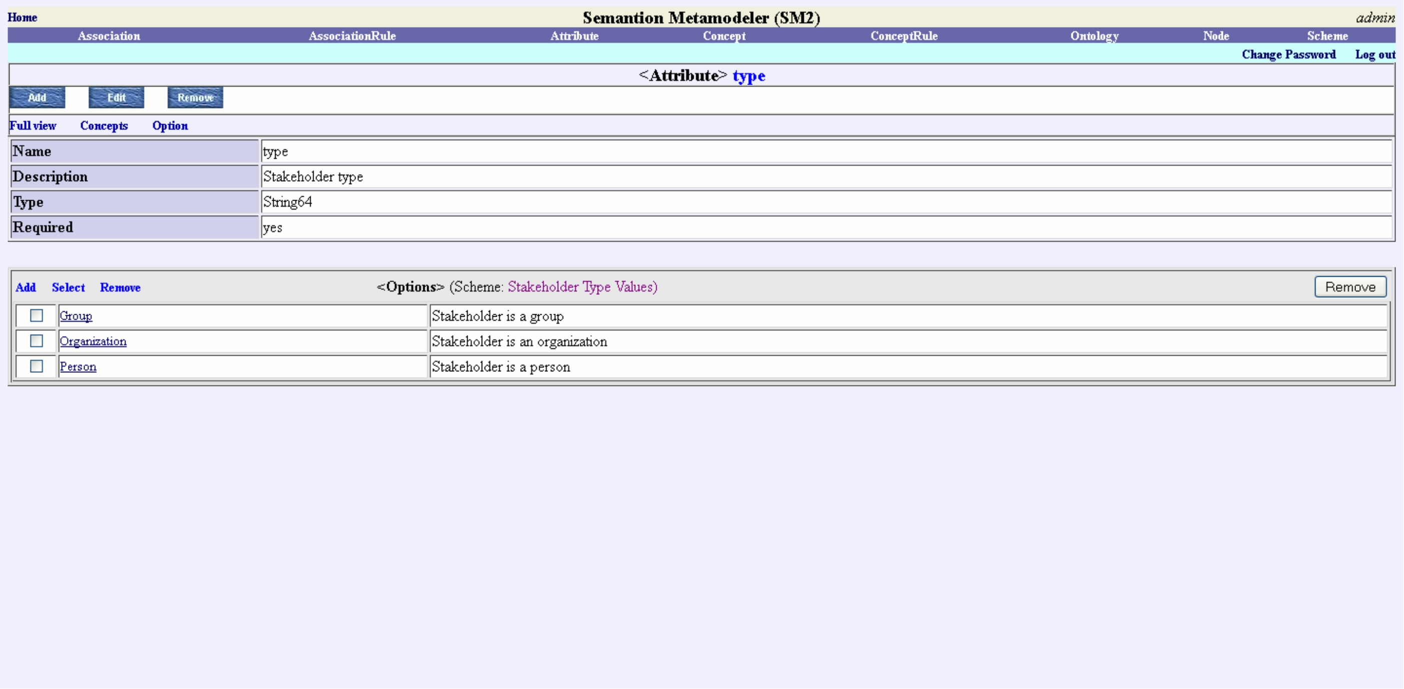
Click the *Option* link to load the *Option* view of the attribute page:



Name	type
Description	Stakeholder type
Type	String64
Required	yes

Figure 20: Option view of the attribute

Click the *Select* link to select the option scheme. Select the *Stakeholder Type Values* scheme from the list of the schemes. SM2 will associate all optional values from the *Stakeholder Type Values* scheme with the type attribute:



Name	type
Description	Stakeholder type
Type	String64
Required	yes

<Options> (Scheme: Stakeholder Type Values)

□	Group	Stakeholder is a group
□	Organization	Stakeholder is an organization
□	Person	Stakeholder is a person

Figure 21: Option view of the attribute with the selected option scheme

2.1.6 Different Ways of Creating Concepts and Associations

There are six buttons and seven links on the concept view (Fig.18). Three of these six buttons belong to concept creation, edit, and removal. *Add* button is used to add new concept, *Edit* button is used to edit the concept, and *Remove* button is used to remove the concept. The *SubConcepts* link list all sub-concepts of the current concept. There is the *Add* link in the *SubConcepts* section that can be used to add a sub-concept. The sub-concept is the new concept that will inherit all properties of the parent concept. The *SuperConcept* link is used to show the parent of the current concept if it exists.

While the functionalities of *Edit* and *Remove* button are straight forward, the *Add* button requires an additional explanation. When you click the *Add* button the following page will be loaded:

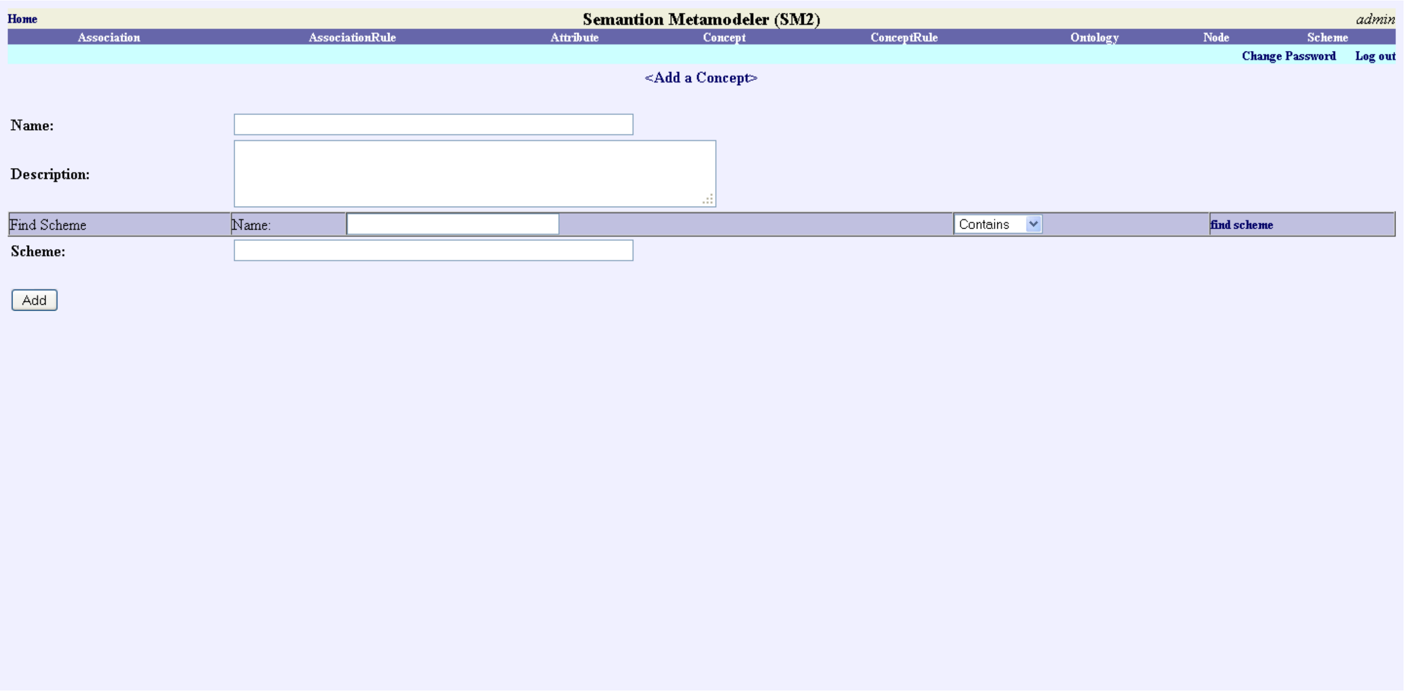


Figure 22: Add a concept page

SM2 provides four ways of creating concepts in an ontology. First one is through the ontology view previously explained. With this approach we first created the concepts scheme and then added nodes that represent concepts under this scheme. The second way of creating concepts is presented in Fig. 16. With this approach we assume that the concepts scheme is already created. You enter name and description for the concept and then select a concept scheme that this concept should belong to. The third way of creating concepts is via *SubConcepts* link on the concept view page via the *Add* link in the *SubConcepts* section. Finally, the fourth way of creating concepts is via the *Scheme* link in the *Navigation* pane. If the scheme does not exist it can be created by clicking the *Add* button. Otherwise the Scheme view page should be loaded and *Nodes* link should be used to load the page that enables creation of new concepts via the *Add* link.

Associations can be created the same way as concepts by using three approaches explained in the previous paragraph.

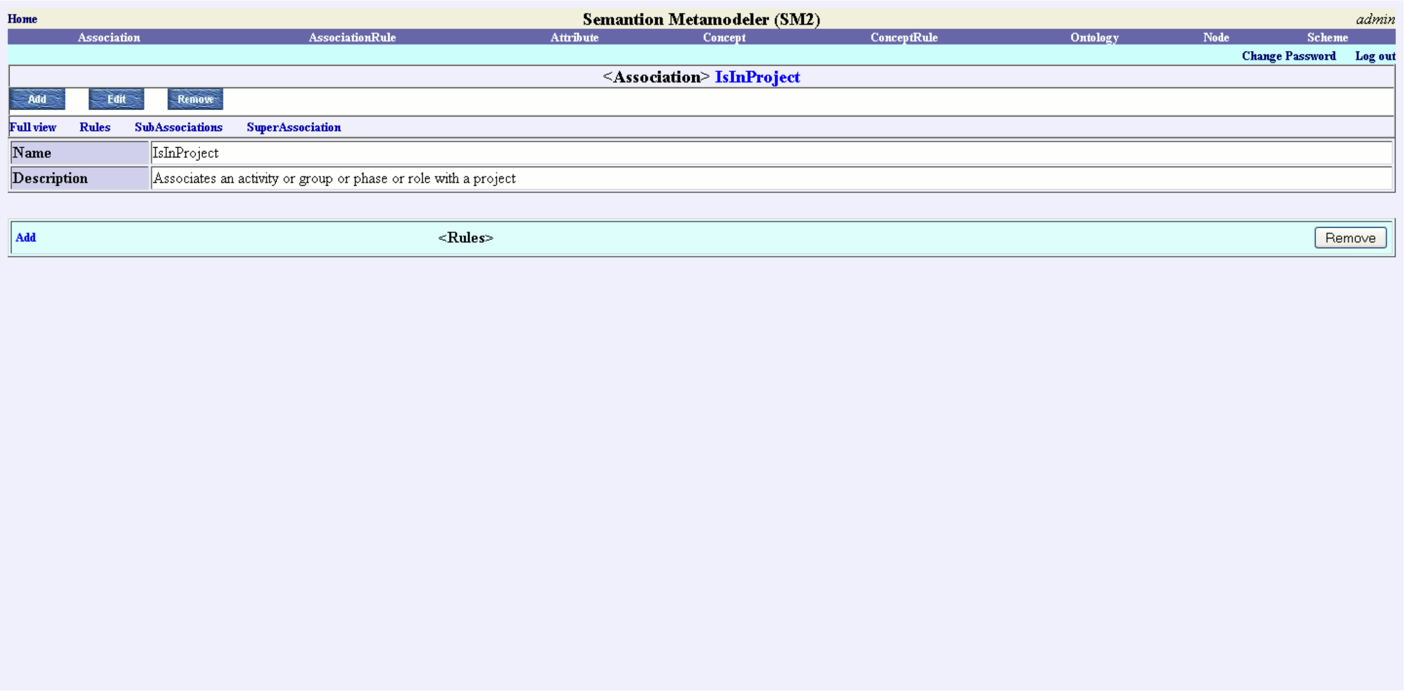
If you go back to Fig. 18 you will see seven links on the detailed *Stakeholder* concept view:

- *Full view* that loads a full detailed concept view with all attributes, rules, and associations.
- *Attributes* that loads a detailed concept view with all attributes that belong to the concept.
- Rules that loads a detailed concept view with all rules associated with the concept.
- *Source associations* that loads all associations with the concept as a source object.
- *Target associations* that loads all associations with the concept as a target object.
- *SubConcepts* loads all concepts that are sub-concepts of the current concept.
- *SuperConcept* loads the parent concept of the current concept if it exists.

By assuming that all other attributes from the *Project Managemet* ontology are defined we can say that the Step 6 is finished.

2.1.7 Create AssociationRules

Now that the associations have been created, we are ready for Step 7 that includes creation of the association rules for those associations. The *AssociationRule* link in the *Navigation* pane only lists currently created association rules. To create an association rule, access the association page via the *Association* link in the *Navigation* pane. We will add an association rule for the *IsInProject* association. When you are in the *IsInProject* association view click the *Rules* link to load the association rule related view:



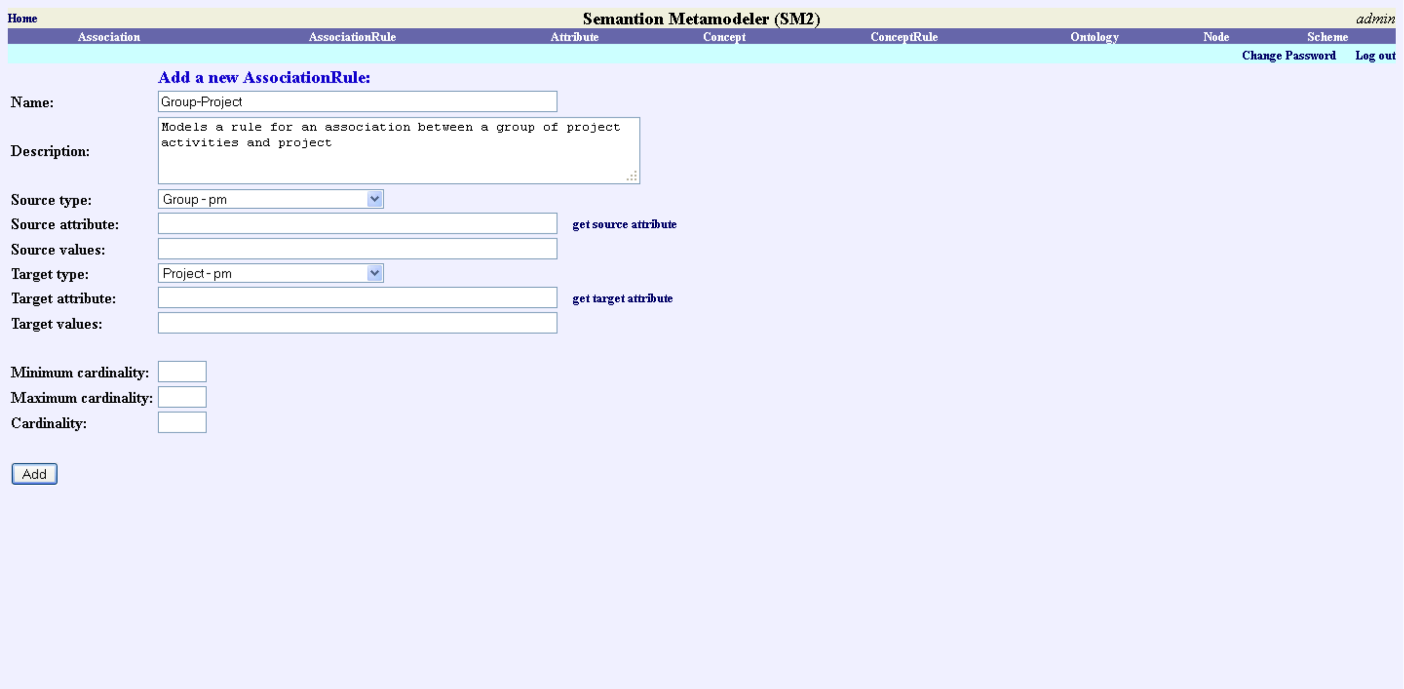
<Association> IsInProject

Name	IsInProject
Description	Associates an activity or group or phase or role with a project

<Rules>

Figure 23: Association rule-based association view

Click the *Add* link to continue with the rule creation:



Add a new AssociationRule:

Name: Group-Project

Description: Models a rule for an association between a group of project activities and project

Source type: Group - pm

Source attribute:

Source values:

Target type: Project - pm

Target attribute:

Target values:

Minimum cardinality:

Maximum cardinality:

Cardinality:

Figure 24: Add Group-Project association rule

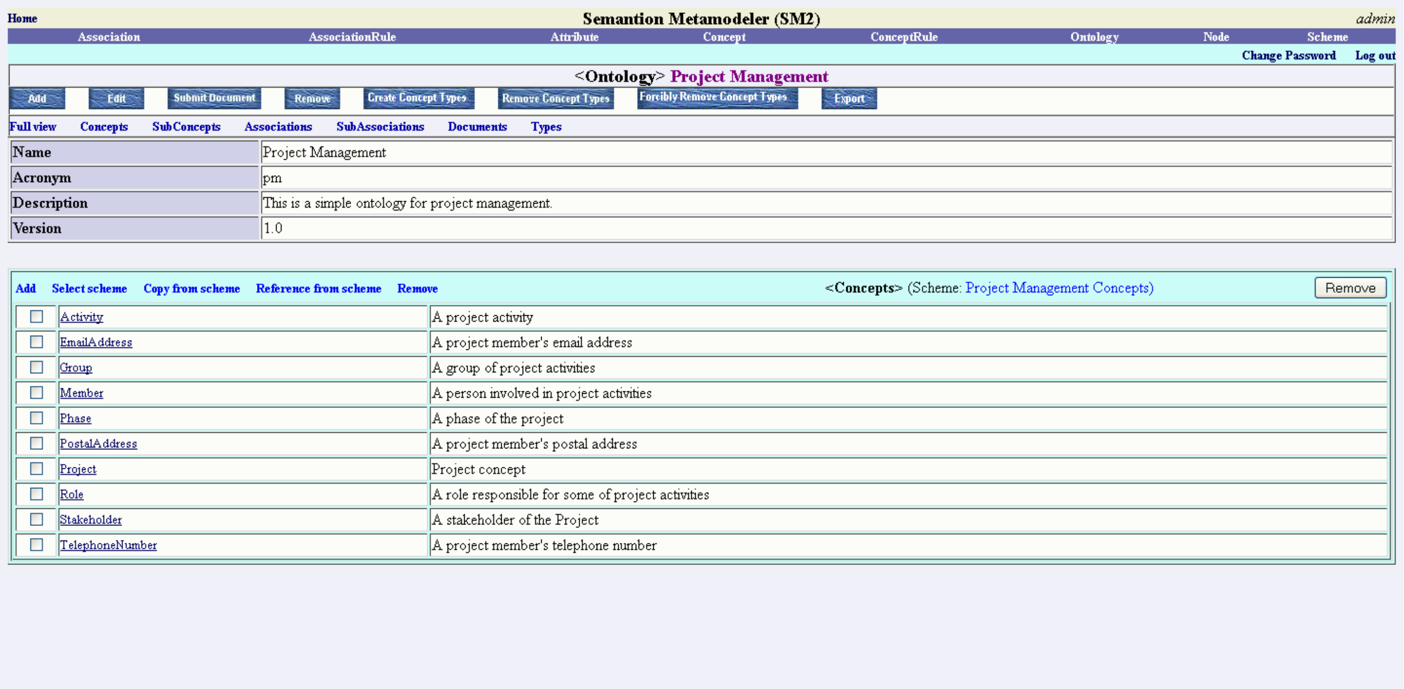
Besides the *name* and *description* properties the association rule also contains the following properties:

- *sourceType* that specifies the source concept of the association.
- *sourceAttribute* that specifies an attribute of the source concept.

- *sourceValue* that specifies a list of the source attribute values.
- *targetType* that specifies the target concept of the association.
- *targetAttribute* that specifies an attribute of the target concept.
- *targetValue* that specifies a list of the target attribute values.
- *minimumCardinality* specifies the minimum number of target concept instances the source concept instance has to be associated with.
- *maximumCardinality* specifies the maximum number of target concept instances the source concept instance can be associated with.
- *cardinality* specifies the exact number of target concept instances the source concept instance has to be associated with.

In this example we specify *sourceType* (*Group*) and *targetType* (*Project*) property values only. Please see the AssociationRule section in Tara Ontology Language document (http://www.semantion.com/documentation/SBP/metamodeling/TaraOntologyLanguage_V1.3.pdf) for more information about other association rule properties.

It is also important to mention that the Association Rules can also specify concepts from other ontologies for its *sourceType* and *targetType* properties. In this case these concepts must be included in the ontology concepts scheme via the *Reference from scheme* link on the ontology concept view page:



<Ontology> Project Management

Name	Project Management
Acronym	pm
Description	This is a simple ontology for project management.
Version	1.0

<Concepts> (Scheme: Project Management Concepts)

□	Activity	A project activity
□	EmailAddress	A project member's email address
□	Group	A group of project activities
□	Member	A person involved in project activities
□	Phase	A phase of the project
□	PostalAddress	A project member's postal address
□	Project	Project concept
□	Role	A role responsible for some of project activities
□	Stakeholder	A stakeholder of the Project
□	TelephoneNumber	A project member's telephone number

Figure 25: Ontology concept view

Since we do not use concept rules and document types in this example we will not go over Steps 8 and 9. Concept rules enable definitions of rules associated with concepts. Step 9 provides a definition of all document types that will be used in an ontology.

2.1.8 Creation of Concept Types in the Registry

The last step, creation of concept types and their folders in the repository, is done using the *Create Concept Types* button on the ontology view page.

Names of concept types include ontology name and concept name. For example, if an ontology name is "*ABC System*" and concept name is "*Activity*" the concept type will be "*ABCSystem_Activity*".

3.0 Web Application

When an ontology is created in SM2 (TAB Step 1), we have all the elements needed to create a web application (TAB Step 2).

3.1 Create Web Application with Tara Application Builder (TAB)

3.1.1 TAB Properties Setup

Before you run Tara Application Builder (TAB) to generate a web application from an ontology created in SM2, you must add a few properties to Semantion Business Platform (SBP) client configuration XML file *gateway.xml* located in your user home folder. These properties are described in the table below.

Property	**Description**
RootDir	The root directory (folder) under which the web application will be created (i.e., "*C:\*" or "*C:\MyApp*")
AppName	The Web application name that will be used in the web application J2EE XML configuration files generated by TAB (this name has to be unique and it will be also used as a part of the web application URL. For example, if the specified application name is "*prm*" the application could be accessed via the URL *http://localhost:8080/prm* by assuming that the SBP Web Server URL is *http://localhost:8080)*
OntologyName	The name of the created ontology (i.e., "*Project Management*")
PackageName	The name of the application package generated by TAB (i.e., "*com.semantion.prm*")
AppBanner	The application banner text that will be displayed on the web application pages (i.e., "*Project Management*")
RootConcept	The name of the concept that will be listed on web application home page (i.e., "*Project*") (this is the main (kind of parent/root) concept for the application)
Skin	One of four skins supported in the generated web applications (i.e., "*ExtraVanilla*") (supported skins are: *Blue*, *ExtraVanilla*, *OrderedList*, and *PlainVanilla*)

3.1.2 Run TAB to Generate Web Application

After setting up the TAB properties in the *gateway.xml* file you are ready to run TAB to generate the web application. The *tab* folder containing the TAB tool is located in the *SBP_HOME* folder. Before running TAB you also have to make sure that the JBoss Application Server folder path (*aps.base*) and Semantion Business Platform (SBP) folder path (*reg.base*) in *build.properties* are correctly set up to match your environment. The JBoss Application Server with deployed SBP must be up and running as well.

Now you are ready to run TAB to generate the *Project Management* web application based on the *Project Management* ontology created in the first step (Section 2).

```
cd SBP_HOME\tab
ant tgen
```

You can choose to rebuild TAB by running *ant* only at the command line above without any argument. If you changed the source code of TAB, you have to rebuild TAB first before running it to generate a web application from an ontology.

3.1.3 Web Application Structure

When the application is generated you can find it at <*RootDir*>/<*AppName*> folder where <*RootDir*> and <AppName> are the values of the same properties specified in *gateway.xml* file.

The generated web applications contain two folders and an Apache Ant build and build properties files:

- *etc*
    - Web application configuration files
- *src*
    - Web application source code
- *build.xml*
    - Ant build file
- *build.properties*

    - Ant build properties file

Later when the application is built (see Section 3.1.4 below), three additional folders are created: *build*, *compile*, and *lib*. These are temporary folders that contain the compiled files and libraries needed to build the application.

etc folder includes:

- *application.xml*
    - Deployment descriptor for Enterprise Application Archives
- *ejbtags.tld*
    - Tag library descriptor
- *jboss-web.xml*
    - Web container configuration
- *pager-taglib.tld*
    - JSP paging tag library descriptor
- *web.xml*
    - Deployment descriptor

src folder includes:

- *com*
    - JSP tags source code
- *jsp*
    - JSP files
- *web*
    - JSP Wiki templates

com folder contains JSP tags source code that implements web application logic on top of the Semantion Registry and Repository.

There are four generic concept related tags:

- *AddModelEntityTag*
    - Creates new concept instance
- *RemoveModelEntitiesTag*
    - Removes list of concept instances
- *SelectModelEntitiesTag*
    - Creates associations between selected concepts and parent concept
- *UpdateModelEntityTag*
    - Updates a concept instance

Two other folders, *jsp* and *web*, belong mostly to the web application user interface the structure of which is described in the section below.

The user interface of the web application is based on concepts representing entities that will be managed by the application (i.e., Project, Activity, Phase, etc.). Concepts are related to each other via associations. Each association also has a rule defining conditions under which concepts are associated. All this is already defined in the ontology creation phase (see Section 2.1) with SM2.

Every concept has a web page that is used to manage the concept. Functional links on the concept page include:

- *Add* to create new concept instances
- *Edit* to update current concept instance
- *Remove* to remove current concept instance and all its associations with other concepts
- *Submit document* to submit and store documents and associate them with the concept (the default view of the concept lists all of its attributes and associated documents)
- *Full* link shows detailed concept's view with its attributes, associated documents, and associated concepts
- Separate links exist for each associated concept as well

Each concept has the following set of JSP pages:

- *add_<concept_name>.jsp*
    - Concept is created (added) via this page
- *edit__<concept_name>.jsp*
    - This page implements concept update
- *<concept_name>_viewer.jsp*
    - This page provides a complete concept view

Based on the view selected on the concept viewer page, the viewer page presents the concept view:

- *Full*
    - The parent concept and all associated concepts are presented. Tara association rules for the concept define all its associated concepts. They are retrieved from the Registry via the QueryManager.

- *<concept_name>*
  - When a <concept_name> (i.e., *Activities*) is used as a view type, the concept and associated concept instances specified by the view type will be presented. They are retrieved from the Registry via the QueryManager.

3.1.4 Build and Deploy Web Application

After making sure that the paths in the application *build.properties* (*<RootDir>/<AppName>/build.properties)* match your environment, build the application:

```
cd  <RootDir>/<AppName>
ant
```

When the *Project Management* application is built it will be automatically deployed in the JBoss application server. By assuming that the JBoss Web Server URL is *http://localhost:8080* and the application name is *pm* you can access the application via this URL:

http://localhost:8080/pm

You will need an SBP username to access the application.

If you decide to change some of the TAB properties specified in *gateway.xml* after generating the application you will have to

- manually remove the application folder *<AppName>* in *<RootDir>*
- regenerate the application with TAB (3.1.2)
- rebuild and redeploy the application (3.1.4)