Semantic action

Semantion Metamodeler

V1.4

Author:
Goran Zugic
goran.zugic@semantion.com

April 2011

1.0 Introduction

Semantion Metamodeler (SM2) is a tool for creation and management of ontologies in Tara Ontology Language (http://www.semantion.com/documentation/SBP/metamodeling/TaraOntologyLanguage_V1.3.pdf). SM2 also provides full exports of Tara ontologies in OWL 2.

SM2 provides creation and management of different types of ontologies. Some of them include ontologies for business processes, system design, product life cycle management, project management, collaborative product development, and others.

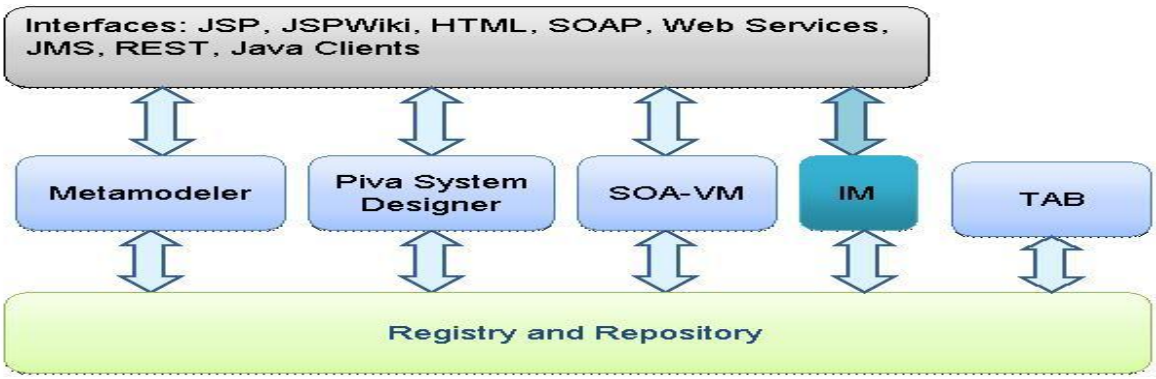


Figure 1: Semantion Business Platform (SBP)

Ontologies are foundations of information and process management automatizations. They are one of the key elements of any business and social platform openness, flexibility and efficiency.

SM2 runs on Semantion Business Platform (SBP) that also includes (Fig. 1) Tara Application Builder (TAB) for generation of web applications (directly executed in SBP) from ontologies, Semantion Registry and Repository for metadata and document management, Piva System Designer for system engineering and modeling, and Semantion SOA Virtual Machine (SOA-VM) for business process execution and analysis.

SM2 is a JEE tool you can access using your web browser. For example, if the web domain of SBP is *http://localhost:8080*, the SM2 URL will be *http://localhost:8080/sm2.*

2.0 Ontology Creation in SM2

We will use the Customer ontology example from Section 4.1.1 in Tara Ontology Language document (http://www.semantion.com/documentation/SBP/metamodeling/TaraOntologyLanguage_V1.3.pdf) to show you how to create an ontology in SM2.

First, you will access the tool via *http://localhost:8080/sm2* URL. If you have not created any additional user in SBP you can use the SBP *admin* user which default password is *manager*. When you log in you will see SM2 home page:

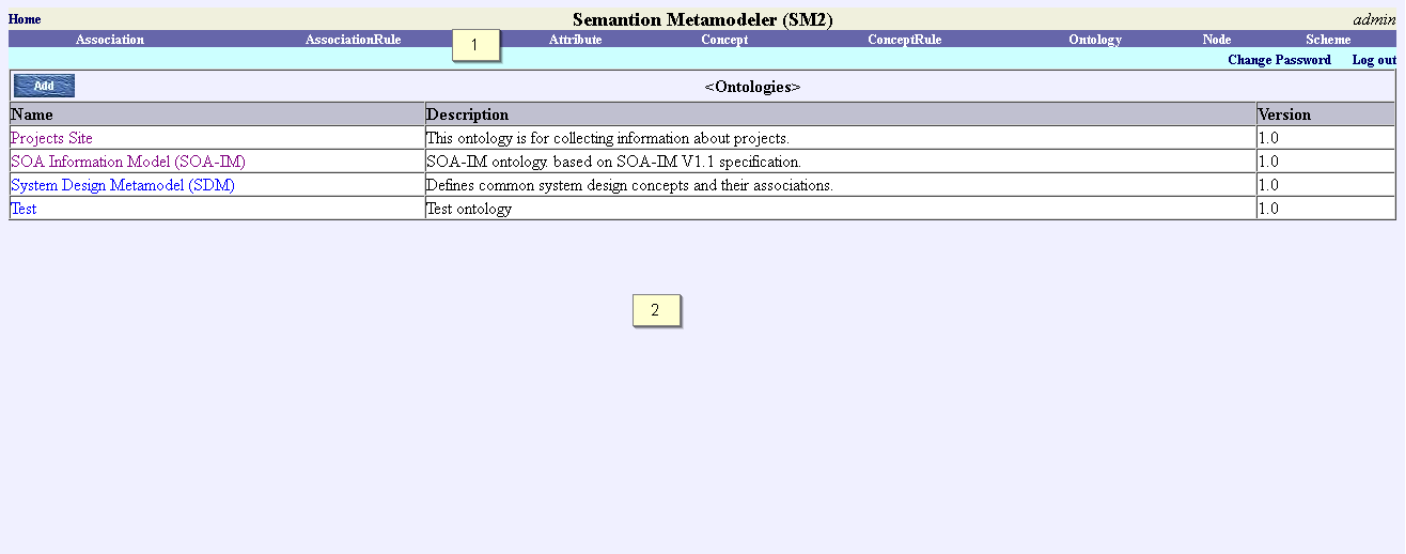


Figure 2: SM2 home page

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 Create Customer Ontology

The *Add* button in the View pane is used to create new ontology. Click the *Add* button to start the creation of the *Customer* ontology. *Add a new Ontology* page will be loaded. Enter the name and description for the new ontology (Fig. 3):

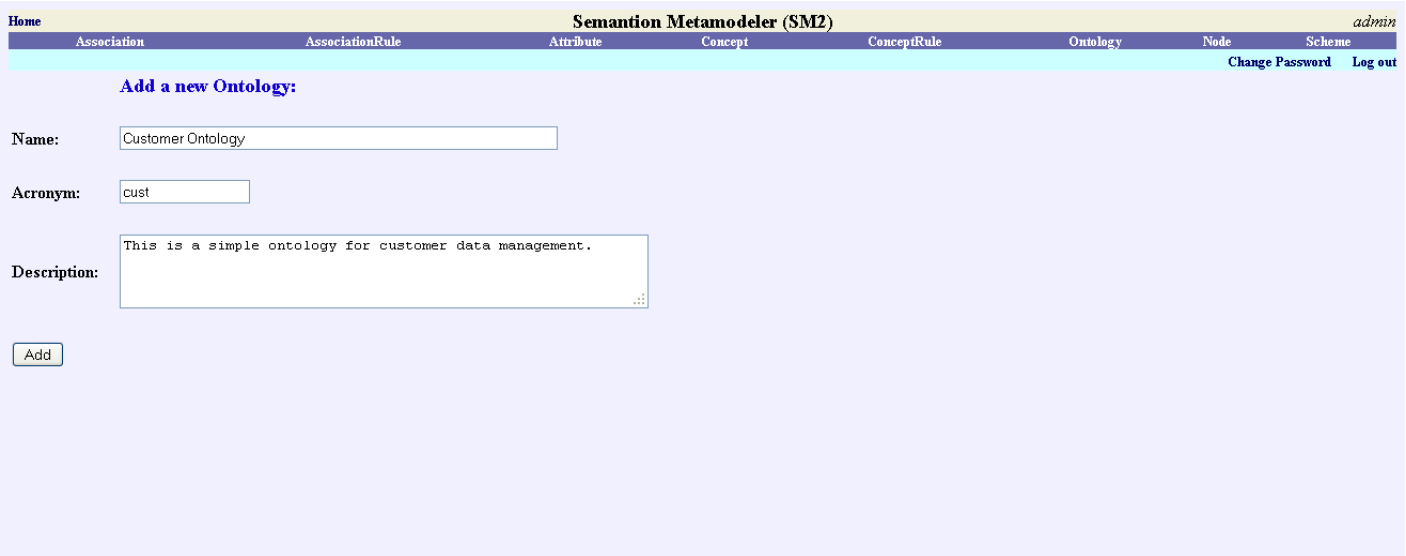


Figure 3: New ontology creation

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 Customer ontology. When the Customer ontology is created the Customer ontology detailed view page will be presented:

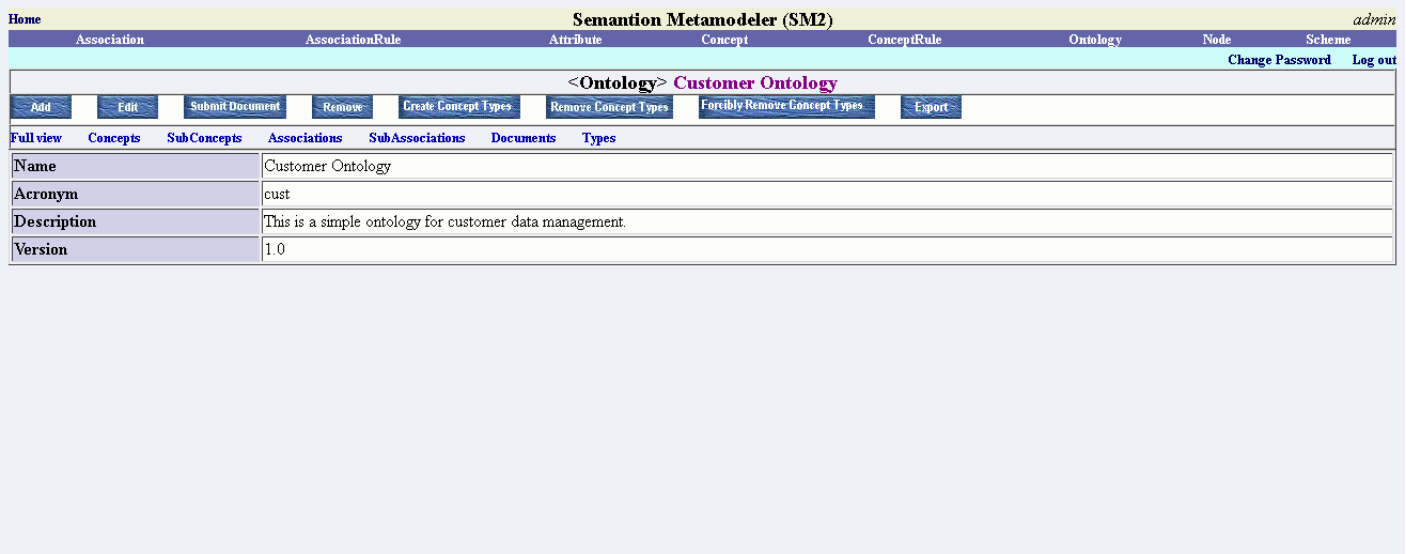


Figure 4: Customer ontology view

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.

2.2 Create Concepts

Now when we created an initial definition of the ontology, the concepts have to be created next. When you click the *Concepts* link, the following page will be loaded:

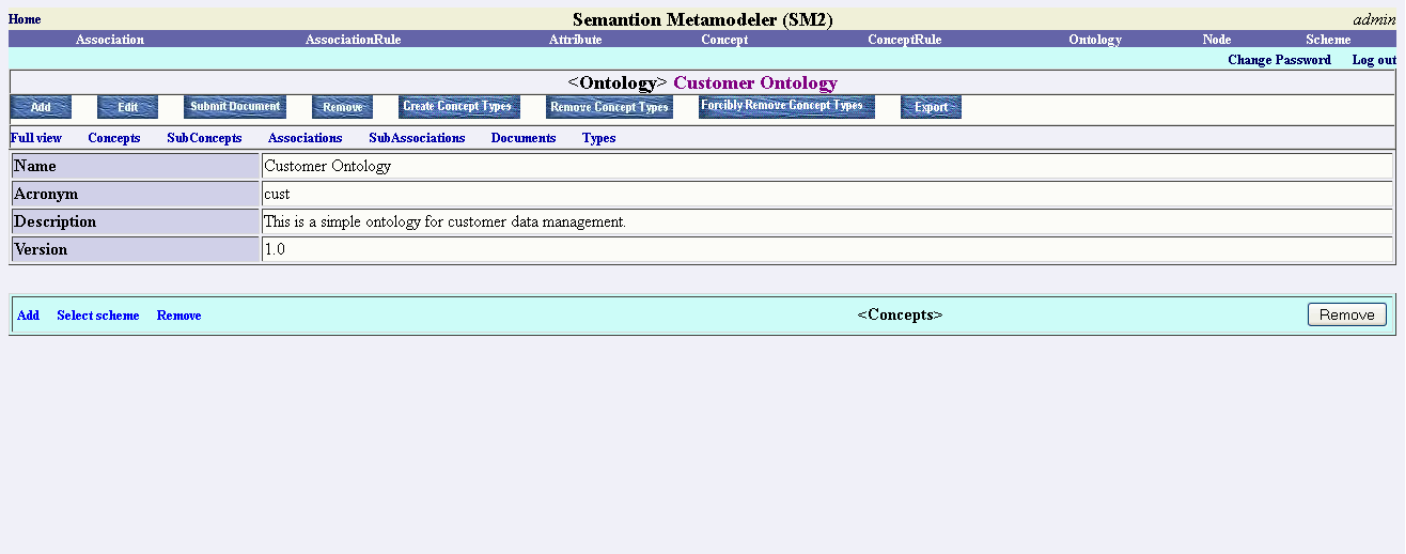


Figure 5: Concept scheme page

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. 6) to create the Customer concept scheme:

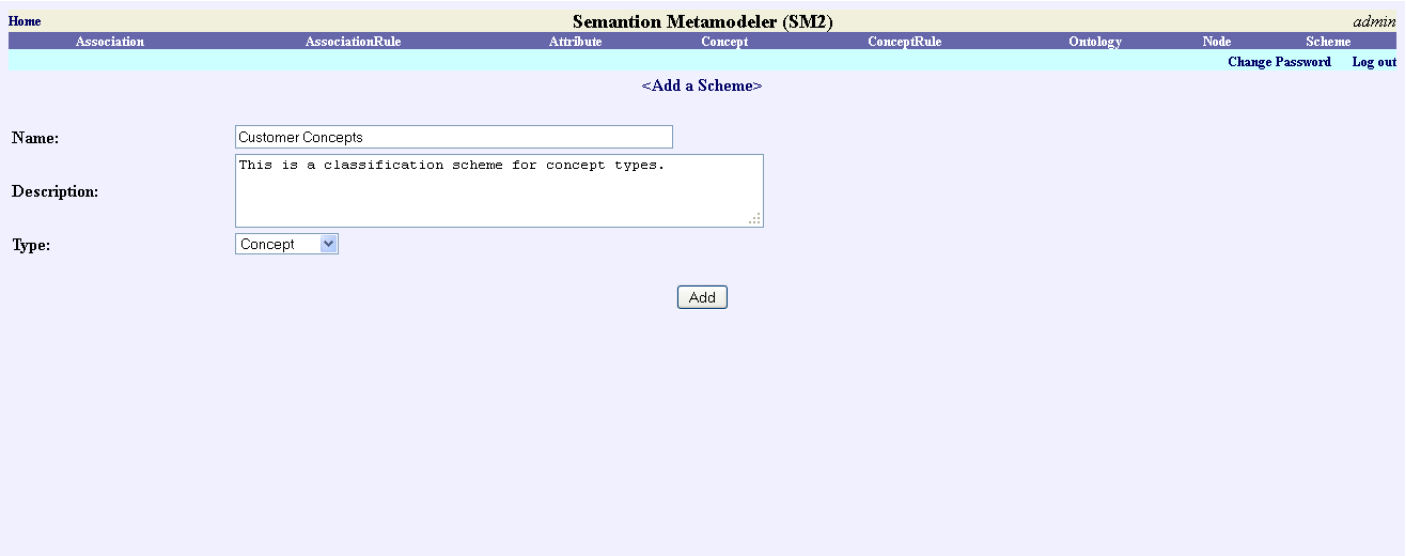


Figure 6: Add concept scheme

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:

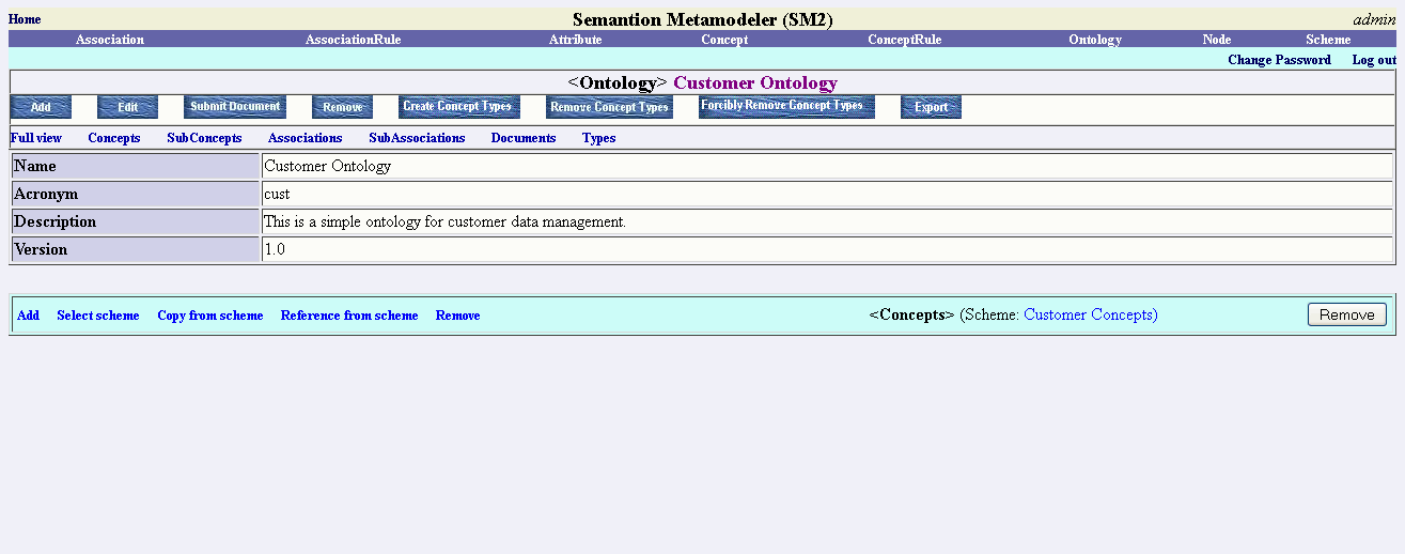


Figure 7: Detail ontology view with empty concept scheme

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 *Customer* for the name property, *Customer type* for the description property and click *Add* button to create the Customer concept node.

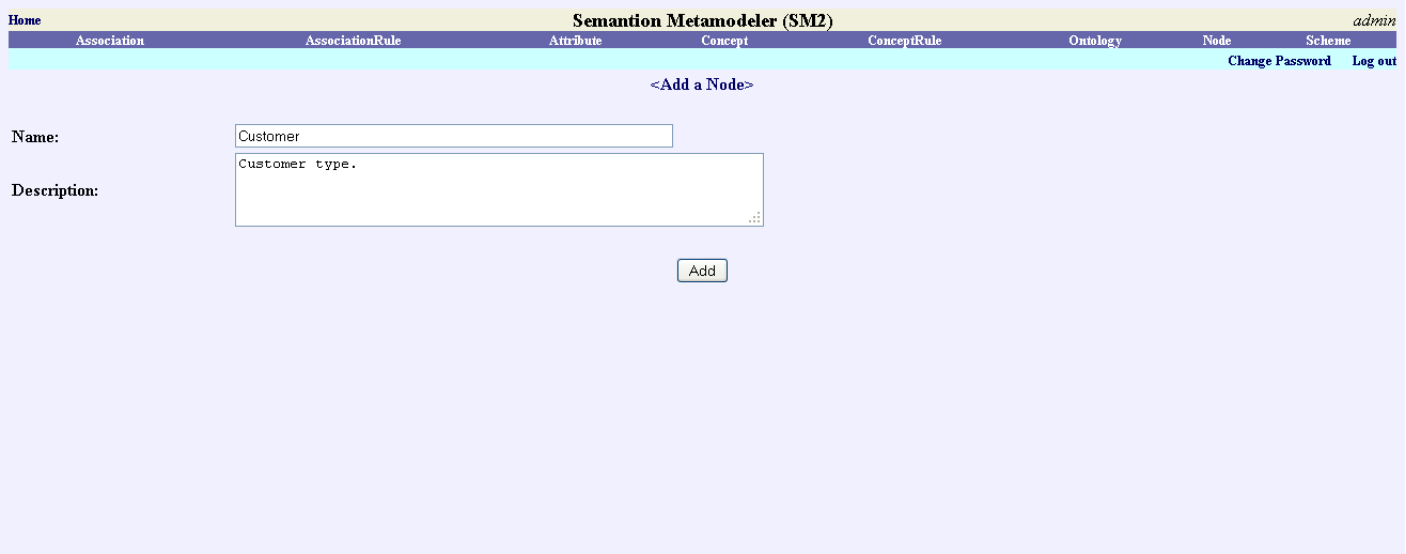


Figure 8: Add concept node to concept scheme

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

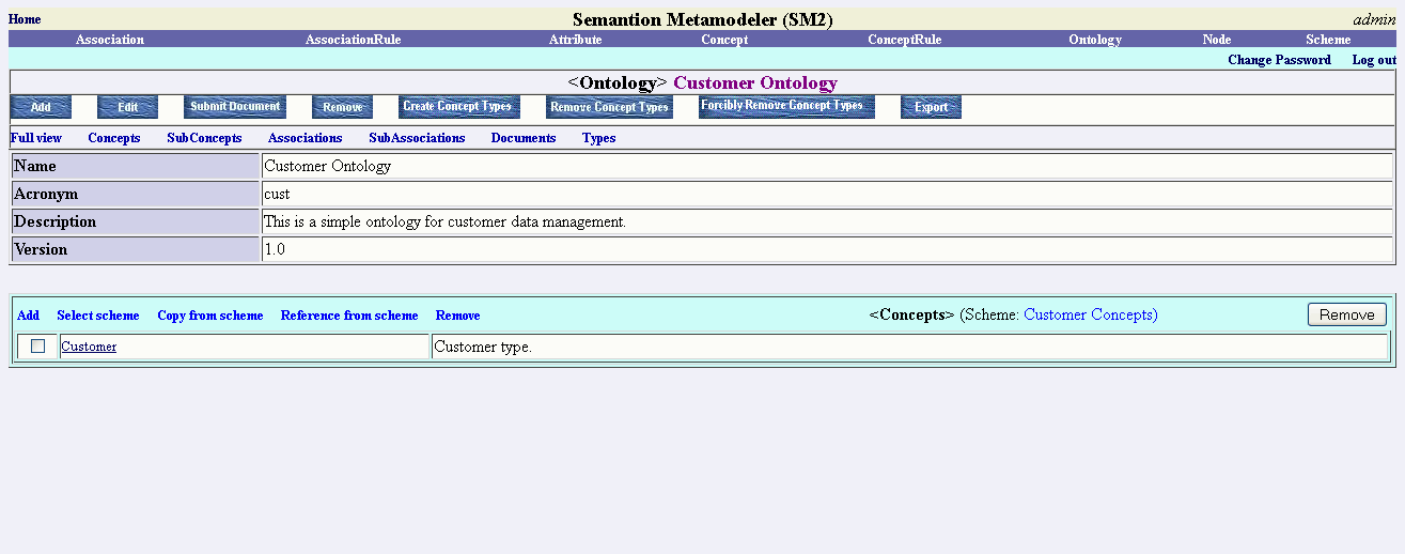


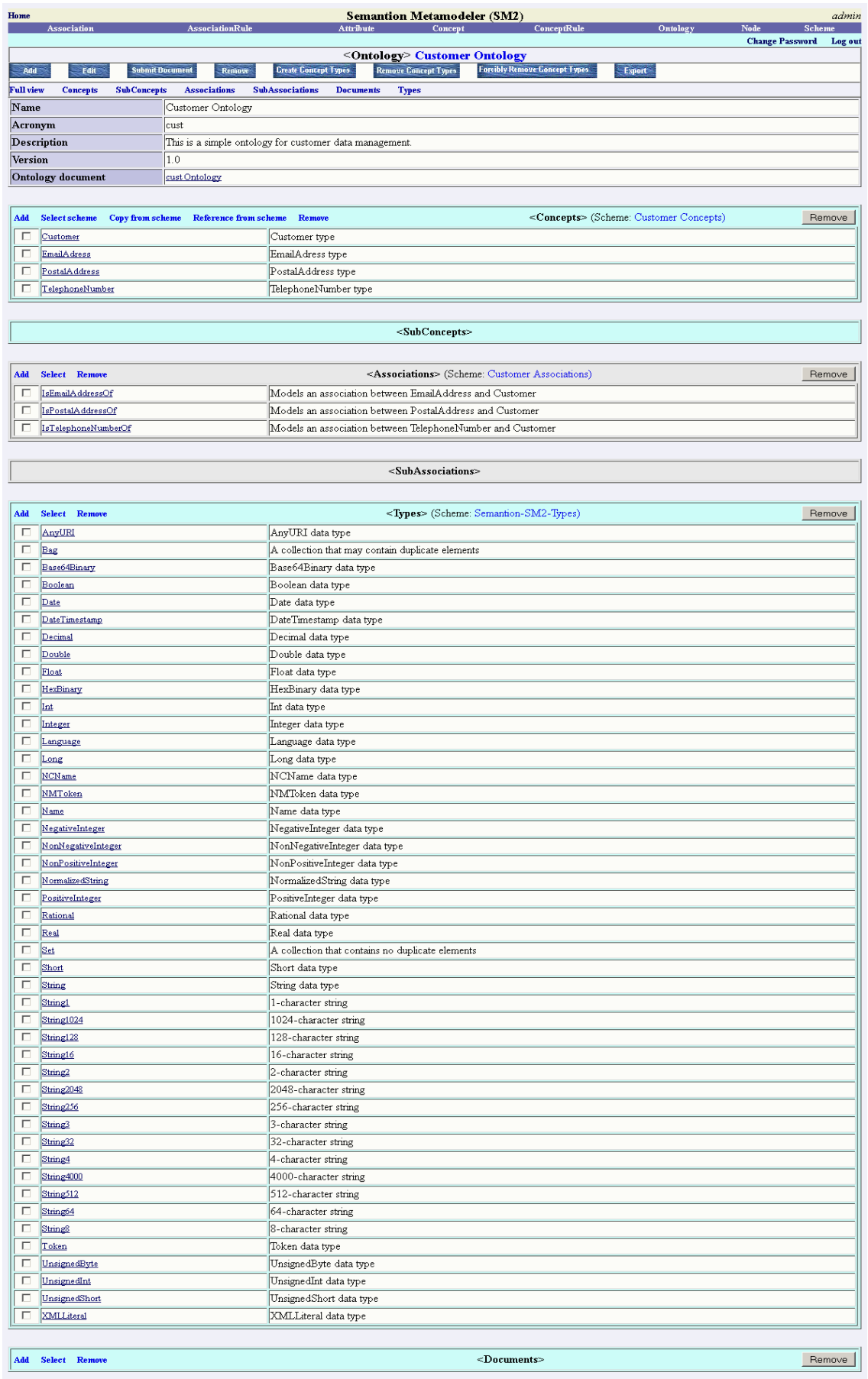
Figure 9: Ontology view with concept scheme

Other concepts (*PostalAddress*, *EmailAddress*, and *TelephoneNumber*) can be created the same way.

Other schemes (Association, Document, and Type) can be created using the same steps explained above for the Concept scheme.

The data types scheme Semantion-SM2-Types is a built-in type scheme that can be selected and used for all ontologies. To select the type scheme you click the *Types* link on the ontology detailed view page (Fig. 9) and then the *Select* link under the *Types* section.

This is how the full view of the Customer ontology will look like when Concept scheme, Association scheme and Type scheme are created:



Home | Semantion Metamodeler (SM2) | admin

Association | AssociationRule | Attribute | Concept | ConceptRule | Ontology | Node | Scheme

Change Password | Log out

<Ontology> Customer Ontology

Add | Edit | Submit Document | Remove | Create Concept Types | Remove Concept Types | Forcibly Remove Concept Types | Export

Full view | Concepts | SubConcepts | Associations | SubAssociations | Documents | Types

Name	Customer Ontology
Acronym	cust
Description	This is a simple ontology for customer data management.
Version	1.0
Ontology document	cust Ontology

Add | Select scheme | Copy from scheme | Reference from scheme | Remove — <Concepts> (Scheme: Customer Concepts) — Remove

Customer	Customer type
EmailAdress	EmailAdress type
PostalAddress	PostalAddress type
TelephoneNumber	TelephoneNumber type

<SubConcepts>

Add | Select | Remove — <Associations> (Scheme: Customer Associations) — Remove

IsEmailAddressOf	Models an association between EmailAddress and Customer
IsPostalAddressOf	Models an association between PostalAddress and Customer
IsTelephoneNumberOf	Models an association between TelephoneNumber and Customer

<SubAssociations>

Add | Select | Remove — <Types> (Scheme: Semantion-SM2-Types) — Remove

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

Add | Select | Remove — <Documents> — Remove

Figure 10: Ontology view with all schemes created

2.3 Create Options

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 *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 *TelephoneNumberTypeValues* scheme:

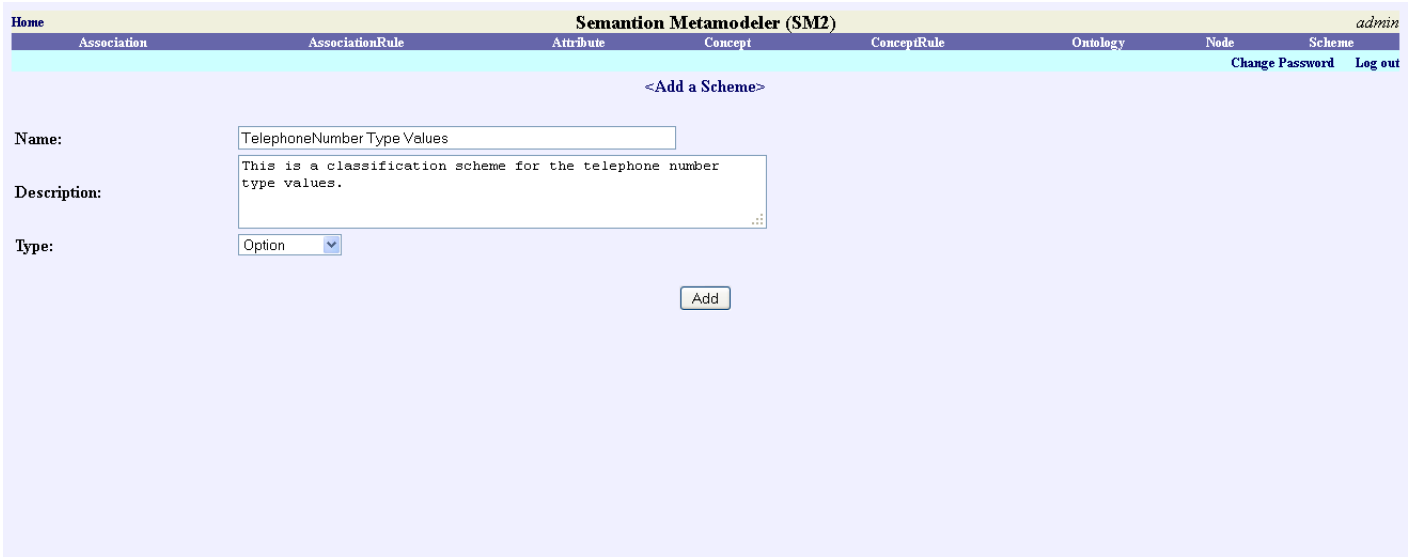


Figure 11: Create TelephoneNumberTypeValues option scheme

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 *TelephoneNumberTypeValues* 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:

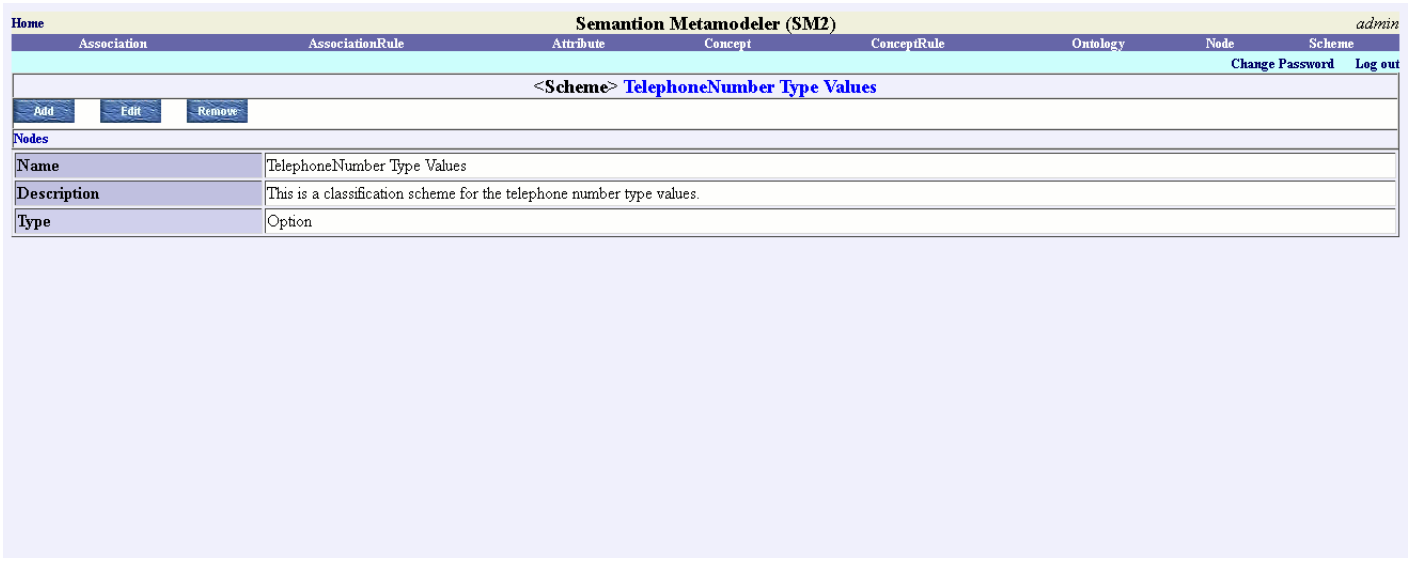


Figure 12: Add nodes to TelephoneNumberTypeValues option scheme

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

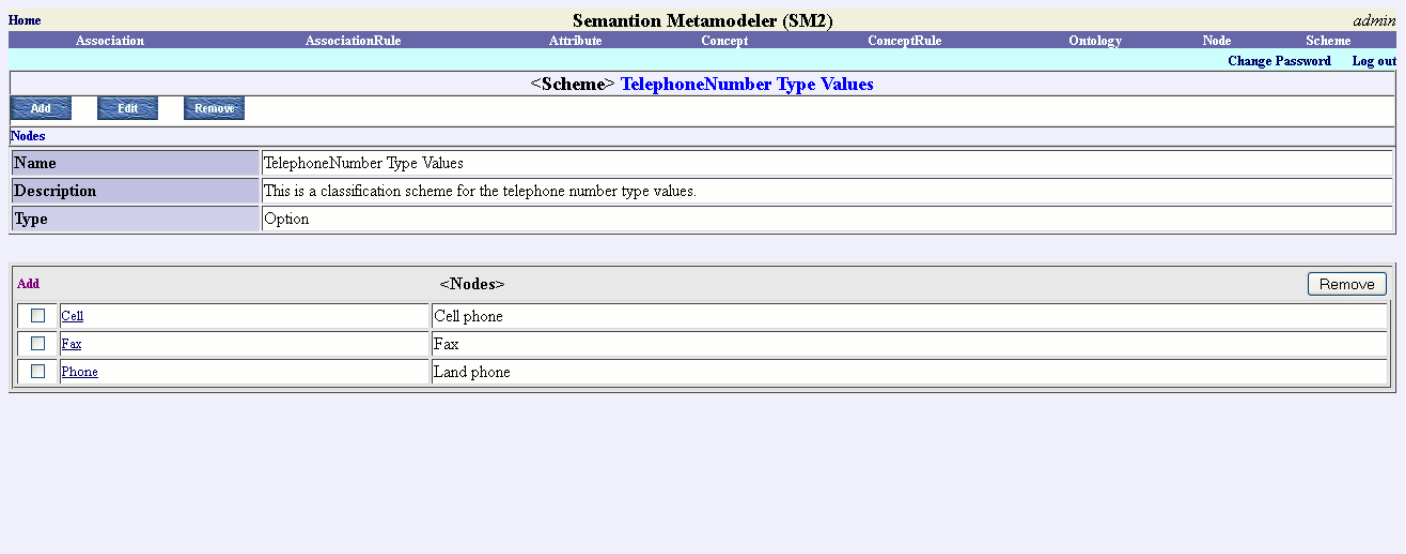


Figure 13: Detailed view of the TelephoneNumberTypeValues option scheme

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*. *Add* link either creates 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. *Select* link selects another *Option* scheme for the attribute. *Remove* link removes the association between the *Option* scheme and the attribute.

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

1. Create an ontology
2. Define concept types
3. Define association types
4. Define attribute types
5. Define optional property values (options) if needed

Other five steps include:

6. Define additional properties 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 belong to 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 they are referenced in the application.

2.4 Create Attributes

Step 6 is to define additional properties used in concepts. We will just demonstrate how *firstName*, *middleName*, and *lastName* properties can be added to *Customer* concept via the *Concept* SM2 page. First click the *Concept* link in Navigation pane, then click the *Customer* concept in the concepts list and when *Customer* view is presented click the *Attributes* link that will open attributes related view with *Add* link to create new attributes representing additional properties for the *Customer* concept. Fig. 14 below shows data for the *Customer's firstName* property:

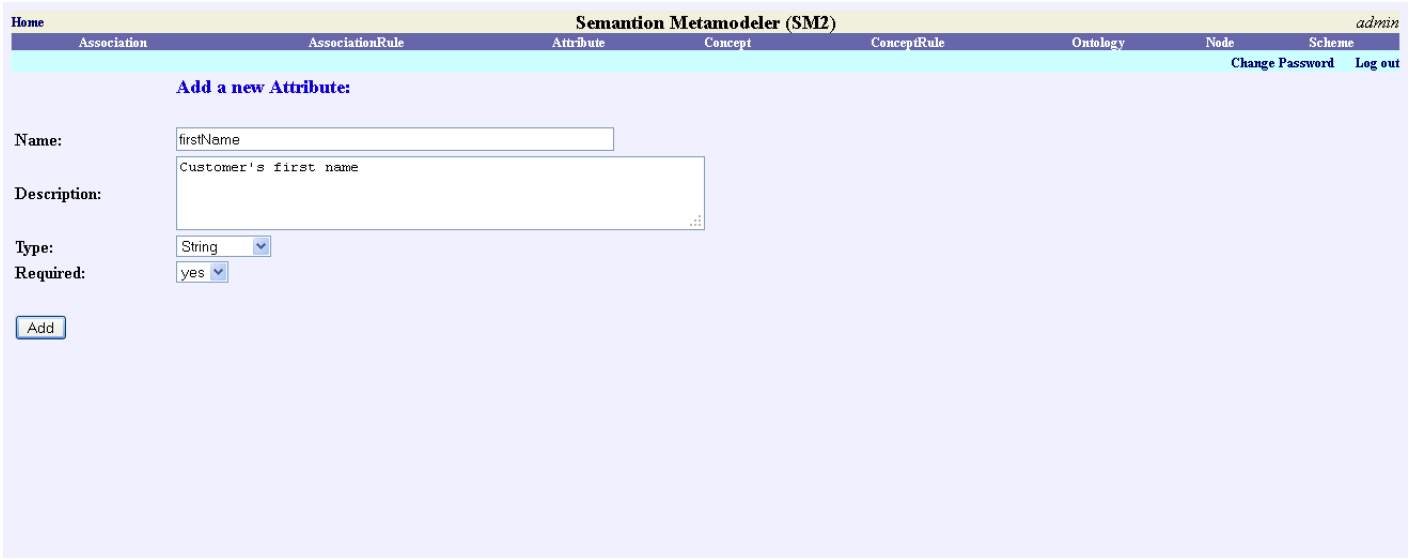


Figure 14: Add a new attribute page

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 if the value of the property represented by the attribute is mandatory. Click the Add button to create the attribute.

This is how the Customer attributes related view will look like when two other attributes (*middleName* and *lastName*) are added:

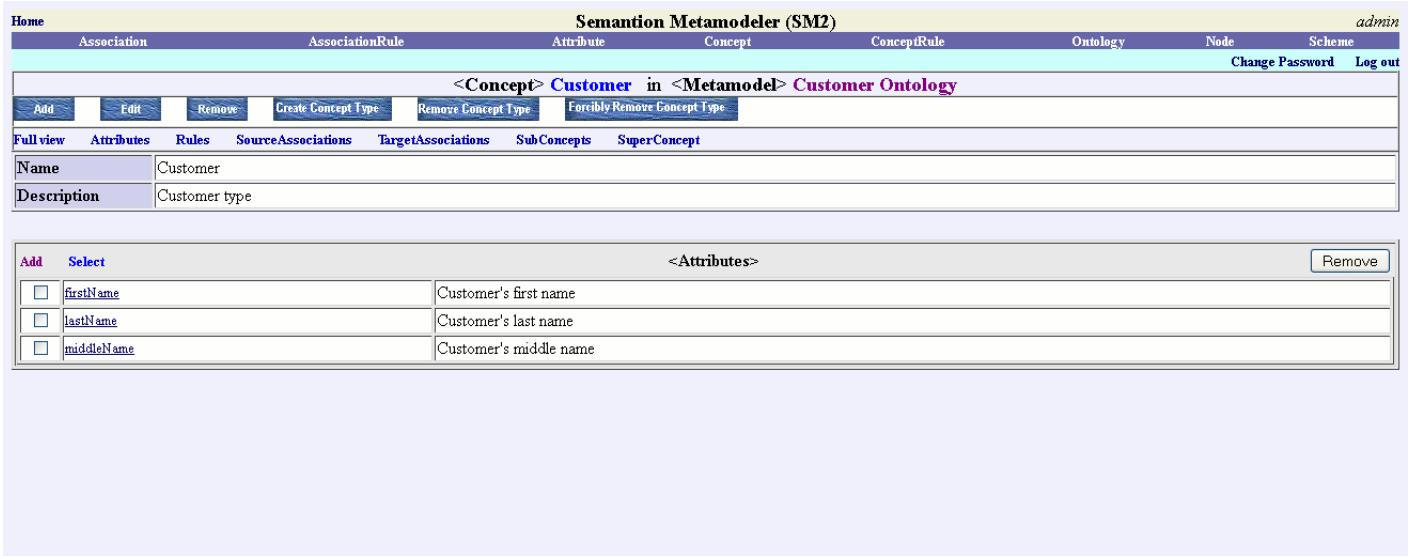


Figure 15: Detailed attributes view of the Customer concept

Remove Concept Type and *Forcibly Remove Concept Type* are used to remove a single concept type. *Remove Concept Type* removes the concept type and its related empty folder in the repository. If the folder contains files it is not removed. It is renamed to its original name followed by a dash sign and the ID of the concept's object type folder that is being removed. *Forcibly Remove Concept Type* removes the concept type and its folder.

2.5 Different Ways of Creating Concepts and Associations

There are six buttons and seven links on the concept view (Fig.15). 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:

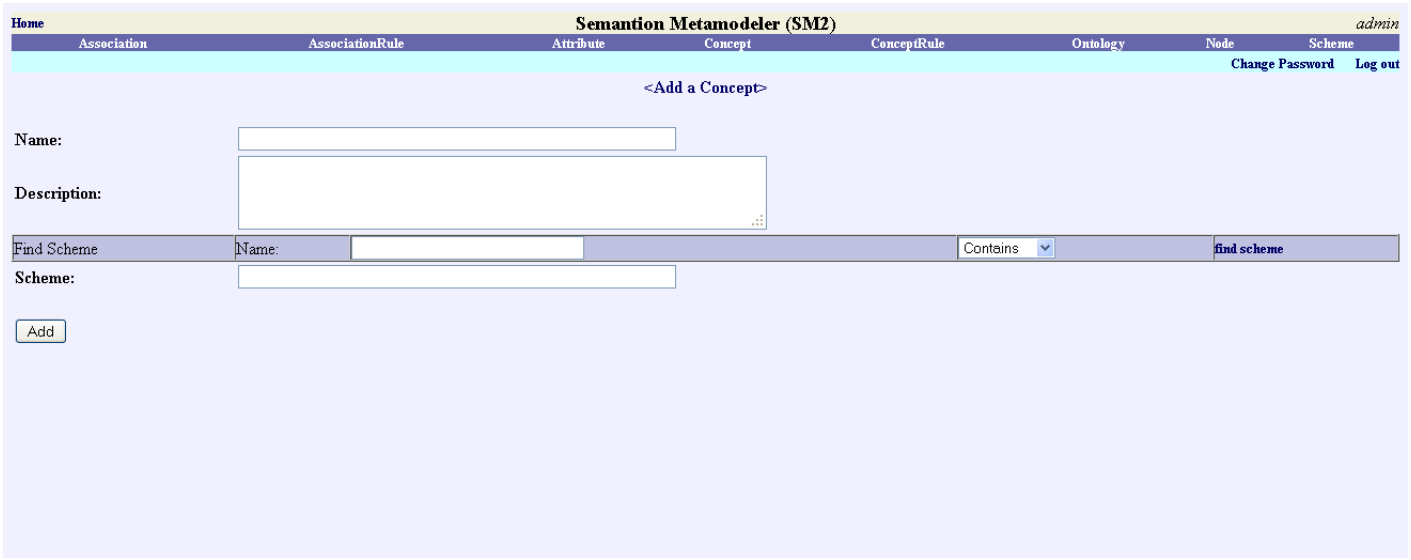


Figure 16: Add a concept page

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 *SubConcept*s 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. 15 you will see seven links on the detailed Customer 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 *Customer* ontology are defined we can say that the Step 6 is finished.

2.6 Create AssociationRules

Now we are ready for Step 7 that includes creation of association rules. The definition of the association rule without the

association does not make sense. *AssociationRule* link in the *Navigation* pane only lists currently created association rules. To create an association rule you have to access the association page via the *Association* link in the *Navigation* pane. We will add an association rule for the IsEmailAddressOf association. When you are in the *IsEmailAddressOf* association view click the *Rules* link to load the association rule related view:

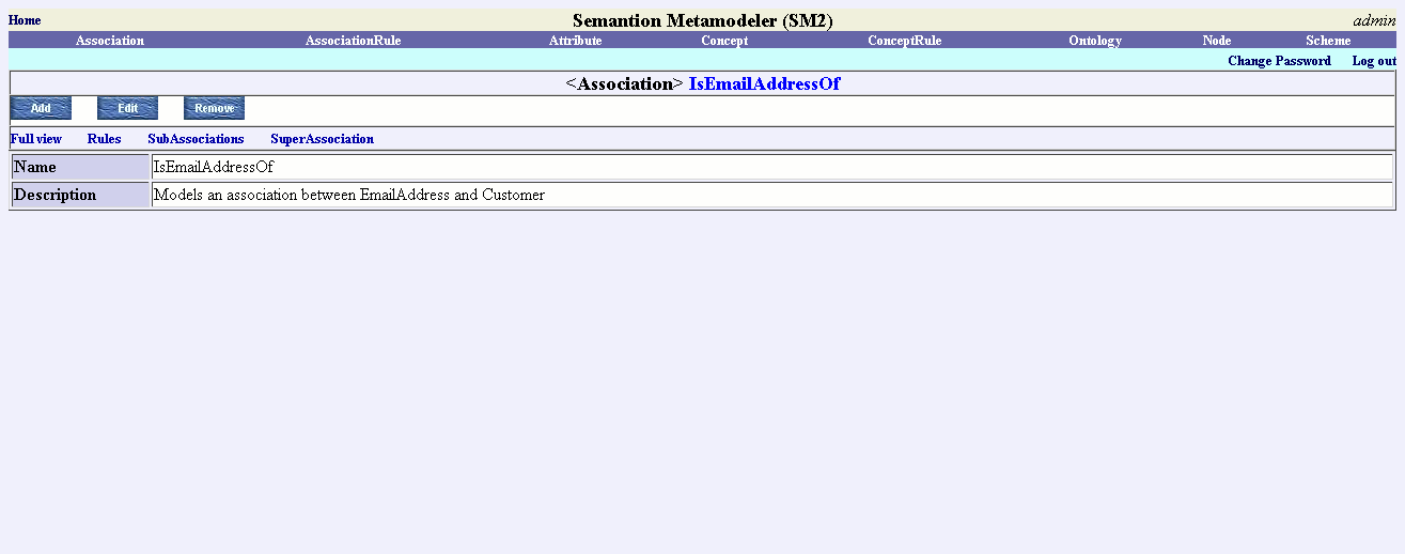


Figure 17: Association rule-based association view

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

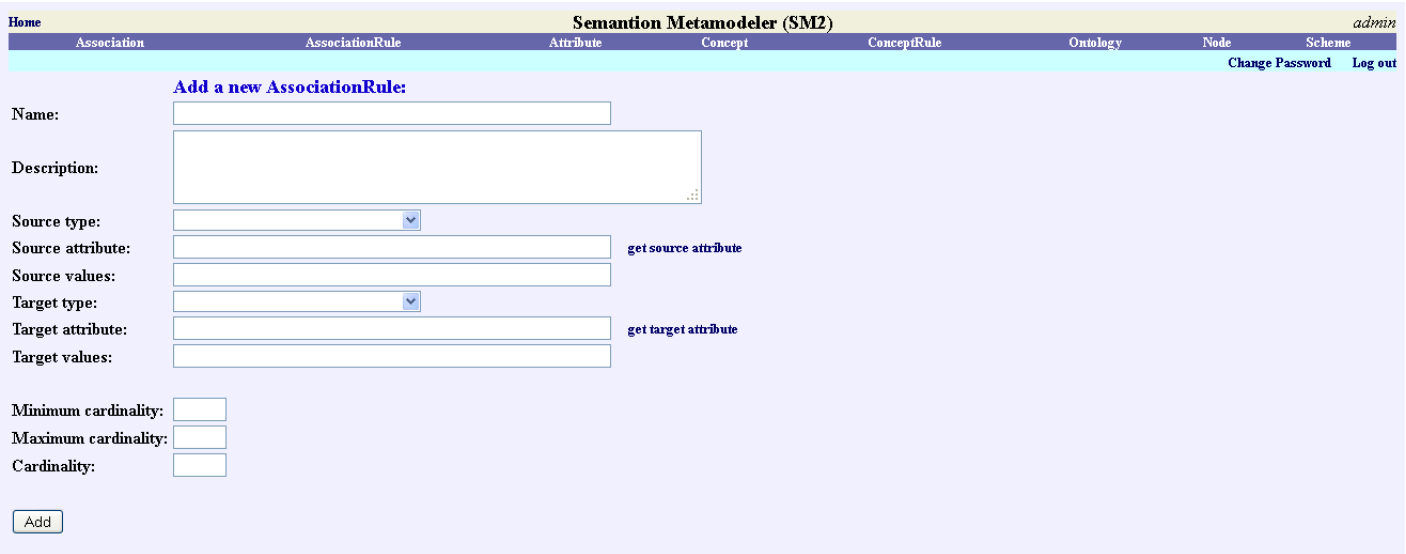


Figure 18: Add EmailAddress-Customer association rule

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* (*EmailAddress*) and *targetType* (*Customer*) property values only. Please see 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.

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

The last step, creation of concept types and their folders in repository, is done using *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 "*Customer Ontology*" and concept name is "*Customer*" the concept type will be "*CustomerOntology _Customer*".

2.7 Navigation Pane

The navigation pane contains links for Associations, AssociationRules, Attributes, Concepts, ConceptRules, Ontologies, Nodes, and Schemes. The *Ontology* and *Node* links list all ontologies and nodes respectively ordered by their names. All other links list corresponding elements and ontologies they belong to. The listed elements are ordered either by their names or by the ontologies they belong to. The *Name* and *Ontology* headers are used as links for these two types of orderings.

For example, if you click the Concept link,

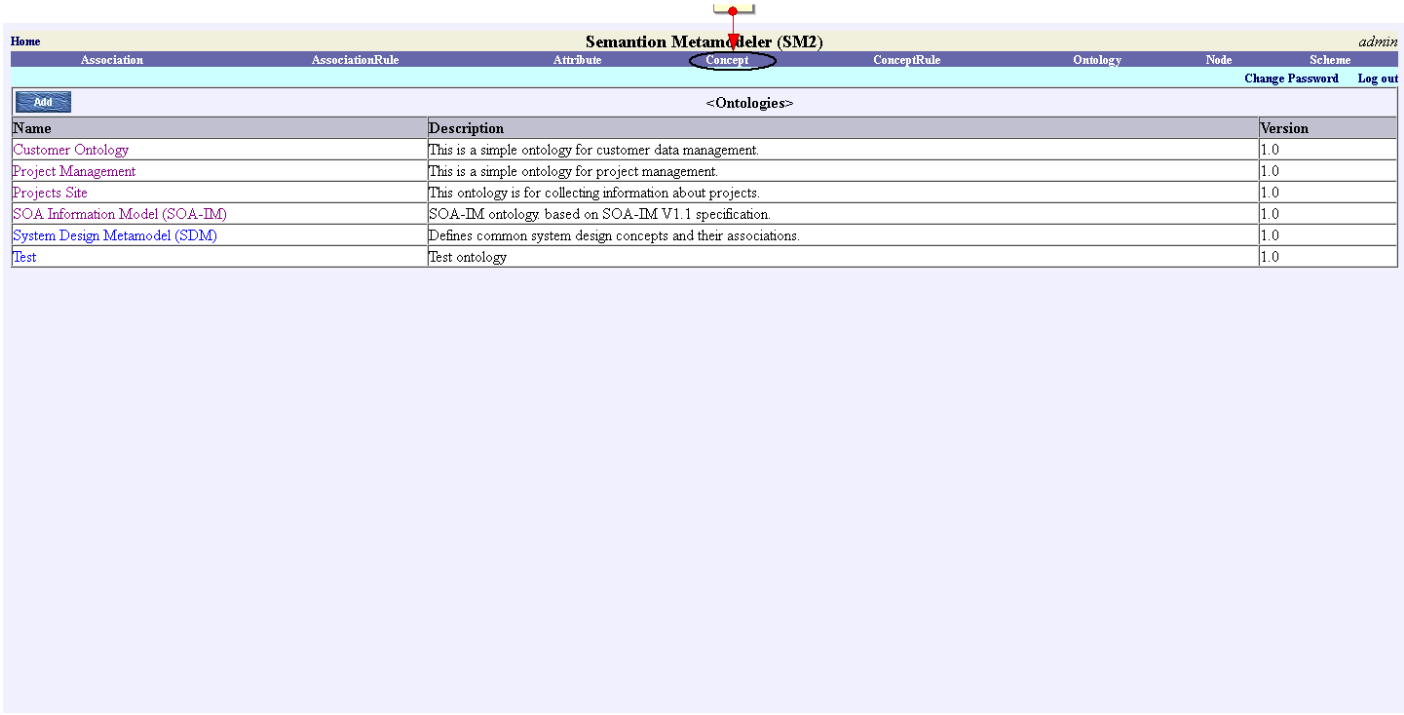
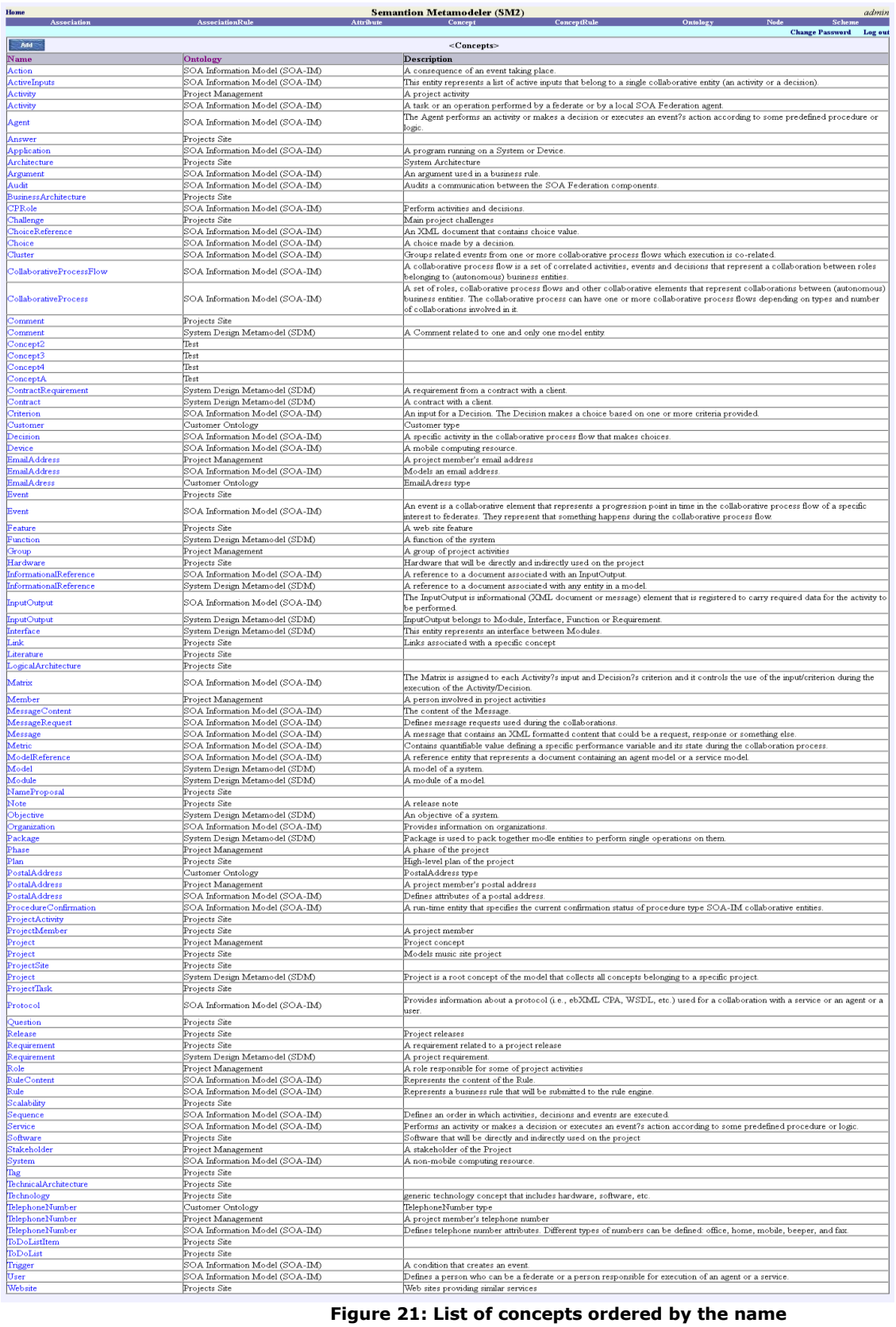


Figure 20: Navigation Pane

the following page with all concepts will be loaded:



Home — Semantion Metamodeler (SM2) — admin

Association | AssociationRule | Attribute | Concept | ConceptRule | Ontology | Node | Scheme

Change Password | Log out

Add

<Concepts>

Name	Ontology	Description
Action	SOA Information Model (SOA-IM)	A consequence of an event taking place.
ActiveInputs	SOA Information Model (SOA-IM)	This entity represents a list of active inputs that belong to a single collaborative entity (an activity or a decision).
Activity	Project Management	A project activity
Activity	SOA Information Model (SOA-IM)	A task or an operation performed by a federate or by a local SOA Federation agent.
Agent	SOA Information Model (SOA-IM)	The Agent performs an activity or makes a decision or executes an event?s action according to some predefined procedure or logic.
Answer	Projects Site	
Application	SOA Information Model (SOA-IM)	A program running on a System or Device.
Architecture	Projects Site	System Architecture
Argument	SOA Information Model (SOA-IM)	An argument used in a business rule.
Audit	SOA Information Model (SOA-IM)	Audits a communication between the SOA Federation components.
BusinessArchitecture	Projects Site	
CPRole	SOA Information Model (SOA-IM)	Perform activities and decisions.
Challenge	Projects Site	Main project challenges
ChoiceReference	SOA Information Model (SOA-IM)	An XML document that contains choice value.
Choice	SOA Information Model (SOA-IM)	A choice made by a decision.
Cluster	SOA Information Model (SOA-IM)	Groups related events from one or more collaborative process flows which execution is co-related.
CollaborativeProcessFlow	SOA Information Model (SOA-IM)	A collaborative process flow is a set of correlated activities, events and decisions that represent a collaboration between roles belonging to (autonomous) business entities.
CollaborativeProcess	SOA Information Model (SOA-IM)	A set of roles, collaborative process flows and other collaborative elements that represent collaborations between (autonomous) business entities. The collaborative process can have one or more collaborative process flows depending on types and number of collaborations involved in it.
Comment	Projects Site	
Comment	System Design Metamodel (SDM)	A Comment related to one and only one model entity.
Concept2	Test	
Concept3	Test	
Concept4	Test	
ConceptA	Test	
ContractRequirement	System Design Metamodel (SDM)	A requirement from a contract with a client.
Contract	System Design Metamodel (SDM)	A contract with a client.
Criterion	SOA Information Model (SOA-IM)	An input for a Decision. The Decision makes a choice based on one or more criteria provided.
Customer	Customer Ontology	Customer type
Decision	SOA Information Model (SOA-IM)	A specific activity in the collaborative process flow that makes choices.
Device	SOA Information Model (SOA-IM)	A mobile computing resource.
EmailAddress	Project Management	A project member's email address
EmailAddress	SOA Information Model (SOA-IM)	Models an email address.
EmailAdress	Customer Ontology	EmailAdress type
Event	Projects Site	
Event	SOA Information Model (SOA-IM)	An event is a collaborative element that represents a progression point in time in the collaborative process flow of a specific interest to federates. They represent that something happens during the collaborative process flow.
Feature	Projects Site	A web site feature
Function	System Design Metamodel (SDM)	A function of the system
Group	Project Management	A group of project activities
Hardware	Projects Site	Hardware that will be directly and indirectly used on the project
InformationalReference	SOA Information Model (SOA-IM)	A reference to a document associated with an InputOutput.
InformationalReference	System Design Metamodel (SDM)	A reference to a document associated with any entity in a model.
InputOutput	SOA Information Model (SOA-IM)	The InputOutput is informational (XML document or message) element that is registered to carry required data for the activity to be performed.
InputOutput	System Design Metamodel (SDM)	InputOutput belongs to Module, Interface, Function or Requirement.
Interface	System Design Metamodel (SDM)	This entity represents an interface between Modules.
Link	Projects Site	Links associated with a specific concept
Literature	Projects Site	
LogicalArchitecture	Projects Site	
Matrix	SOA Information Model (SOA-IM)	The Matrix is assigned to each Activity?s input and Decision?s criterion and it controls the use of the input/criterion during the execution of the Activity/Decision.
Member	Project Management	A person involved in project activities
MessageContent	SOA Information Model (SOA-IM)	The content of the Message.
MessageRequest	SOA Information Model (SOA-IM)	Defines message requests used during the collaborations.
Message	SOA Information Model (SOA-IM)	A message that contains an XML formatted content that could be a request, response or something else.
Metric	SOA Information Model (SOA-IM)	Contains quantifiable value defining a specific performance variable and its state during the collaboration process.
ModelReference	SOA Information Model (SOA-IM)	A reference entity that represents a document containing an agent model or a service model.
Model	System Design Metamodel (SDM)	A model of a system.
Module	System Design Metamodel (SDM)	A module of a model.
NameProposal	Projects Site	
Note	Projects Site	A release note
Objective	System Design Metamodel (SDM)	An objective of a system.
Organization	SOA Information Model (SOA-IM)	Provides information on organizations.
Package	System Design Metamodel (SDM)	Package is used to pack together modle entities to perform single operations on them.
Phase	Project Management	A phase of the project
Plan	Projects Site	High-level plan of the project
PostalAddress	Customer Ontology	PostalAddress type
PostalAddress	Project Management	A project member's postal address
PostalAddress	SOA Information Model (SOA-IM)	Defines attributes of a postal address.
ProcedureConfirmation	SOA Information Model (SOA-IM)	A run-time entity that specifies the current confirmation status of procedure type SOA-IM collaborative entities.
ProjectActivity	Projects Site	
ProjectMember	Projects Site	A project member
Project	Project Management	Project concept
Project	Projects Site	Models music site project
ProjectSite	Projects Site	
Project	System Design Metamodel (SDM)	Project is a root concept of the model that collects all concepts belonging to a specific project.
ProjectTask	Projects Site	
Protocol	SOA Information Model (SOA-IM)	Provides information about a protocol (i.e., ebXML CPA, WSDL, etc.) used for a collaboration with a service or an agent or a user.
Question	Projects Site	
Release	Projects Site	Project releases
Requirement	Projects Site	A requirement related to a project release
Requirement	System Design Metamodel (SDM)	A project requirement.
Role	Project Management	A role responsible for some of project activities
RuleContent	SOA Information Model (SOA-IM)	Represents the content of the Rule.
Rule	SOA Information Model (SOA-IM)	Represents a business rule that will be submitted to the rule engine.
Scalability	Projects Site	
Sequence	SOA Information Model (SOA-IM)	Defines an order in which activities, decisions and events are executed.
Service	SOA Information Model (SOA-IM)	Performs an activity or makes a decision or executes an event?s action according to some predefined procedure or logic.
Software	Projects Site	Software that will be directly and indirectly used on the project
Stakeholder	Project Management	A stakeholder of the Project
System	SOA Information Model (SOA-IM)	A non-mobile computing resource.
Tag	Projects Site	
TechnicalArchitecture	Projects Site	
Technology	Projects Site	generic technology concept that includes hardware, software, etc.
TelephoneNumber	Customer Ontology	TelephoneNumber type
TelephoneNumber	Project Management	A project member's telephone number
TelephoneNumber	SOA Information Model (SOA-IM)	Defines telephone number attributes. Different types of numbers can be defined: office, home, mobile, beeper, and fax.
ToDoListItem	Projects Site	
ToDoList	Projects Site	
Trigger	SOA Information Model (SOA-IM)	A condition that creates an event.
User	SOA Information Model (SOA-IM)	Defines a person who can be a federate or a person responsible for execution of an agent or a service.
Website	Projects Site	Web sites providing similar services

Figure 21: List of concepts ordered by the name

They are ordered by the name and if you want to order them by ontology and name you need to click the *Ontology* header link.