

**System Design Metamodel
(SDM)**

V 1.0

June 2008

Copyright © 2005-2008 Semantion

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>).

Table of Contents

1.0 INTRODUCTION	2
2.0 SYSTEM DESIGN METAMODEL (SDM).....	2
2.1 SYSTEM DESIGN METAMODEL ENTITIES	2
2.1.1 <i>Comment</i>	3
2.1.2 <i>Contract</i>	3
2.1.3 <i>ContractRequirement</i>	4
2.1.4 <i>Function</i>	4
2.1.5 <i>InformationalReference</i>	5
2.1.6 <i>InputOutput</i>	6
2.1.7 <i>Interface</i>	7
2.1.8 <i>Model</i>	7
2.1.9 <i>Module</i>	8
2.1.10 <i>Objective</i>	10
2.1.11 <i>Package</i>	10
2.1.12 <i>Project</i>	11
2.1.13 <i>Requirement</i>	11

1.0 Introduction

Piva System Designer (PSD), a system design component of Semantion Business Platform (SBP), provides system design based on System Design Metamodel (SDM). SDM is a metamodel that defines common system design entities and their associations.

2.0 System Design Metamodel (SDM)

The goals of System Design Metamodel (SDM) are:

- Enforce formal language for system engineering design;
- No repetition – all objects are singletons and only association (link) is added when object is used;
- Improve design visibility and readability – all useful information for certain design under one view;
- Improve design analysis with intuitive model query interface;

2.1 System Design Metamodel Entities

This section covers all SDM entities and their attributes.

SDM entities are classified under three types:

- Requirement
- Model
- Supportive

Requirement type includes legal requirement related entities. Model type belongs to core model entities. Entities of supportive type provide additional modeling support. The following table classifies SDM entities by types. Their detailed descriptions are in the sections below.

Type	Entity
Requirement	<ul style="list-style-type: none">• Contract• ContractRequirement
Model	<ul style="list-style-type: none">• Model• Module

	<ul style="list-style-type: none"> • Function • Interface • InputOutput • Objective • Requirement
Supportive	<ul style="list-style-type: none"> • Comment • InformationalReference • Project • Package

2.1.1 Comment

Comment models a comment related to any other entity in a model. A Comment belongs to one and only one model entity.

Attribute	Type	Description
id	String256	Unique ID
name	String256	Comment's name
description	String4000	Comment's description
status	String256	The status of the Comment (Open/Close)
version	String256	Version of the Comment

Comment can be associated with any entity in the model using the "IsCommentOf" association where the Comment is a source object. Comment can be created by any user that has at least read access to the model. When created, an association to the target object will be created automatically and status attribute will be set to "Open". Only users with privileged access and user who created Comment will be able to edit Comment, including changing status attribute to "Close".

Parent: any entity in a model

2.1.2 Contract

Contract models a contract with a client.

Attribute	Type	Description
id	String256	Unique ID

name	String256	Contract's name
description	String4000	Contract's description
version	String256	Version of the Contract

2.1.3 ContractRequirement

ContractRequirement models a requirement from a contract with a client.

Attribute	Type	Description
id	String256	Unique ID
name	String256	Requirement's name
description	String4000	Detailed description
reference	String 256	References a requirement from a Contract which this requirement belongs to.
fullCompliance	String3	Does a Model complies with the ContractRequirement? (Yes/No)
version	String256	Version of the Requirement

Associated with

- A Contract where the contract is a target object and association type is "IsContractRequirementFrom"
- Any entity in a model where the entity is a target object and association type is "IsContractRequirementFor"

Each ContractRequirement can be associated with more than one entity in a model. Each entity in the model can also be associated with more than one ContractRequirement. A Model complies with a ContractRequirement if all entities in the model comply with the ContractRequirement. The ContractRequirement has the "fullCompliance" attribute to specify if all entities in the Model comply with it.

When imported in PSD and displayed they will be listed in the same order as in the original contract.

Parent: Contract, any entity in a model

2.1.4 Function

Attribute	Type	Description
id	String256	Unique ID
name	String256	Function's name
description	String4000	Detailed description
type	String256	The type of the Function (Internal/External)
version	String256	Version of the Function

Associated with

- A Project where the Project is a target object and association type is "IsUsedOn"
- A Module where the Module is a target object and association type is "IsFunctionOf"
- A Package where the Package is a target object and association type is "IsMemberOf". If function's Module is a member of a package, the function will be automatically associated with the package. This association belongs to a package other than the Module-based package.
- A ContractRequirement where the ContractRequirement is a target object and association type is "CompliesTo".

Parent: Module

2.1.5 InformationalReference

A reference to a document associated with any entity in a model.

Attribute	Type	Description
id	String256	Unique ID
name	String256	InformationalReference's name
description	String4000	Detailed description
documentId	String256	ID of a document that relates to the InformationalReference
type	String256	Type of the referenced document (any document type)
value	String256	Document's reference (URI)
version	String16	Version of the document represented by this InformationalReference

An InformationalReference can be associated with any entity in a model using the "IsReferenceFor" association where the InformationalReference is a source object.

Semantion, Inc.

Parent: any entity in a model

2.1.6 InputOutput

InputOutput belongs to Module, Interface, Function or Requirement.

Attribute	Type	Description
id	String256	Unique ID
name	String256	InputOutput's name
description	String4000	Detailed description
ioType	String256	I/O type (Input/Output/Both)
type	String256	(Internal/External)
version	String256	Version

Associated with

- A Project where the Project is a target object and association type is "IsUsedOn"
- A Function where the Function is a target object and association type is "IsInputOf"
- A Function where the Function is a target object and association type is "IsOutputOf"
- An Interface where the Interface is a target object and association type is "IsInputOf"
- A Interface where the Interface is a target object and association type is "IsOutputOf"
- A Requirement where the Requirement is a target object and association type is "IsInputOf"
- A Requirement where the Requirement is a target object and association type is "IsOutputOf"
- A Package where the Package is a target object and association type is "IsMemberOf". If InputOutput's Function/Interface/Requirement is associated with a module that is a member of a Package or if InputOutput's Module is associated with a Package, the InputOutput will be automatically associated with the package. This association belongs to a package other than the Module-based package.
- A ContractRequirement where the ContractRequirement is a target object and association type is "CompliesTo".

Parent: Function, Interface, Requirement

2.1.7 Interface

This entity represents an interface between Modules.

Attribute	Type	Description
id	String256	Unique ID
name	String256	Interface's name
description	String4000	Detailed description
type	String256	The type of the Interface (Internal/External)
version	String256	Version of the Interface

Associated with

- A Project where the Project is a target object and association type is "IsUsedOn"
- A Package where the Package is a target object and association type is "IsMemberOf". If Interface's Module belongs to a package, the Interface will be automatically associated with the package. This association belongs to a package other than the Module-based package.
- A ContractRequirement where the ContractRequirement is a target object and association type is "CompliesTo".

Parent: Module

2.1.8 Model

Attribute	Type	Description
id	String256	Unique ID
name	String256	Model's name
description	String4000	Detailed description
version	String256	Version of the Model

Associated with

- A Project where the Project is a target object and association type is "IsUsedOn"
- A Package where the Package is a target object and association type is "IsMemberOf"

Semantion, Inc.

- A ContractRequirement where the ContractRequirement is a target object and association type is "CompliesTo".

Parent: Project, Package

2.1.9 Module

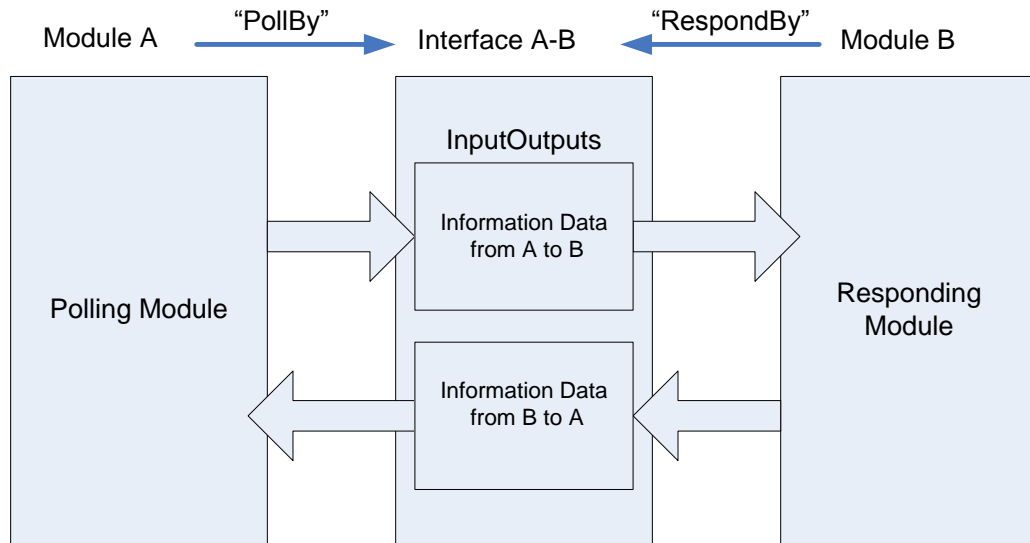
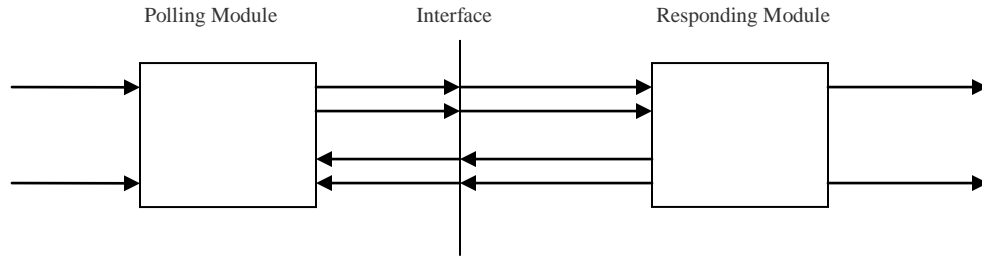
Attribute	Type	Description
id	String256	Unique ID
name	String256	Module's name
description	String4000	Detailed description
type	String256	Module's type (Internal/External)
version	String256	Version of the Module

Associated with

- A Model where the Model is a target object and association type is "IsModuleIn".
- A Project where the Project is a target object and association type is "IsUsedOn"
- An Interface where the Interface is a target object and association type is "PollBy"
- An Interface where the Interface is a target object and association type is "RespondBy"
- A Package where the Package is a target object and association type is "IsMemberOf". If Module's Model belongs to a package, the Module will be automatically associated with the package. This association belongs to a package other than the Model-based package.
- A ContractRequirement where the ContractRequirement is a target object and association type is "CompliesTo".

Parent: Project, Model

This is a semantic description of "PollBy" and "RespondBy" associations:



Interface has two sets of inputs and outputs. Each set is defined by the direction of the communication flow between the modules.

First set of Interface's InputOutputs belongs to the communication flow from the Polling Module which is in the "PollBy" association with the Interface to the Responding Module which is in the "RespondBy" association with the Interface. This set of InputOutputs ("Information Data from A to B" in Figure above) has association with Interface where the Interface is a target object and association type is "IsOutputOf". When InputOutput from this set is associated with any Function or Requirement from Polling module, association type is "IsOutputOf". When InputOutput from this set is associated with any Function or Requirement from Responding module, association type is "IsInputOf".

Second set of InputOutputs belongs to the communication flow from the Responding Module which is in the "RespondBy" association with the Interface to the Polling Module which is in the "PollBy" association with the Interface. This set of InputOutputs ("Information Data from B to A" in Figure above) has association with Interface where the Interface is a target object and association type is "IsInputOf". When an InputOutput from this set is associated with any Function or Requirement from Polling module, association type is "IsInputOf". When InputOutput from this set is associated

Semantion, Inc.

with any Function or Requirement from Responding module, association type is "IsOutputOf".

2.1.10 Objective

Attribute	Type	Description
id	String256	Unique ID
name	String256	Objective's name
description	String4000	Detailed description
type	String256	Objective's type (Internal/External)
version	String256	Version of the Objective

Associated with

- A Project where the Project is a target object and association type is "IsObjectiveOf"
- A Function where the Function is a target object and association type is "IsObjectiveOf"
- An Interface where the Interface is a target object and association type is "IsObjectiveOf"
- A Module where the Module is a target object and association type is "IsObjectiveOf"
- A Model where the Model is a target object and association type is "IsObjectiveOf"
- A Requirement where the Requirement is a target object and association type is "IsObjectiveOf"
- An InputOutput where the InputOutput is a target object and association type is "IsObjectiveOf"
- A Package where the Package is a target object and association type is "IsMemberOf"

Parent: Project, Package, Function, Interface, Module, Model, Interface, Requirement, InputOutput

2.1.11 Package

Package is used to pack together some SDM entities' instances and perform single operation on them.

Attribute	Type	Description
Id	String256	Unique ID

Semantion, Inc.

Name	String256	Package's name
description	String4000	Detailed description
version	String256	Version of the Package

2.1.12 Project

Project is a root SDM entity that "collects" all SDM entities' instances belonging to a specific project.

Attribute	Type	Description
id	String256	Unique ID
name	String256	Project's name
description	String4000	Detailed description
contract	String256	An ID of a Contract that the Project belongs to.
version	String256	Version of the Project

Associated with

- A Contract where the Contract is a target object and association type is "HasToComplyTo"

Parent: Contract

2.1.13 Requirement

Attribute	Type	Description
id	String256	Unique ID
name	String256	Requirement's name
description	String4000	Detailed description
type	String256	An ownership type (Internal/External)
statementType	String256	A statement type of the Requirement (Technical/Program)
version	String256	Version of the Requirement

Associated with

- A Project where the Project is a target object and association type is "IsUsedOn"

Semantion, Inc.

- A Package where the Package is a target object and association type is "IsMemberOf"
- A Function where the Function is a target object and association type is "IsRequirementFor"
- An Interface where the Interface is a target object and association type is "IsRequirementFor"

Parent: Project, Package, Function, Interface