Semantic action

System Design Metamodel (SDM)

V 1.2

March 2011

Table of Contents

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 concepts and their associations.

When we use term "model" in this document, it belongs to a complete system design model based on SDM. The terminology used in this document (i.e., concept, association, source concept, target concept, etc.) is based on Tara Ontology Language (http://www.semantion.com/documentation/SBP/metamodeling/TaraOntologyLanguage_V1.2.pdf). The SDM can also be fully defined and created in Semantion Metamodeler (http://www.semantion.com/documentation/SBP/metamodeling/TaraOntologyLanguage_V1.2.pdf).

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 concepts and their attributes.

SDM concepts are classified under three types:

- Requirement
- Model
- Supportive

Requirement type includes legal requirement related concepts. Model type belongs to Model concept and concepts directly associated with Model concept. Concepts of supportive type provide additional metamodeling support. The following table classifies SDM concepts by types. Their detailed descriptions are in the sections below.

Type	**Entity**
Requirement	• Contract • ContractRequirement
Model	• Model • Component • Function • Interface • InputOutput • Objective • Requirement
Supportive	• Comment • InformationalReference • Project • Package

2.1.1 Comment

Comment models a comment that can be related to a concept in a model. A Comment belongs to one and only one concept in a model.

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 concept in the model using the "IsCommentOf" association where the Comment is a source concept. Comment can be created by any user that has at least read access on the model. When created, an association with the target concept 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: a concept in the model

2.1.2 Component

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

Associated with

- A Model where the Model is a target concept and association type is "IsComponent In".
- A Project where the Project is a target concept and association type is "IsUsedOn"
- An Interface where the Interface is a target concept and association type is "PollBy"
- An Interface where the Interface is a target concept and association type is "RespondBy"
- A Package where the Package is a target concept and association type is "IsMemberOf". If Component's Model belongs to a package, the Component 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 concept and association type is "CompliesTo".

Parent: Project, Model

2.1.3 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

2.1.4 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 comply with the ContractRequirement? (Yes/No)
version	String256	Version of the Requirement

Associated with

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

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

When created in PSD and displayed, ContractRequirements will be listed in the same order as in the original contract.

Parent: Contract, a concept in the model

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

Parent: Component

2.1.6 InformationalReference

A reference to a document associated with a concept 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
time	DateTime	Time when InformationalReference is confirmed

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

Parent: a concept in a model

2.1.7 InputOutput

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

Parent: Function, Interface, Requirement

2.1.8 Interface

This entity represents an interface between Components.

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

Parent: Component

2.1.9 Model

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

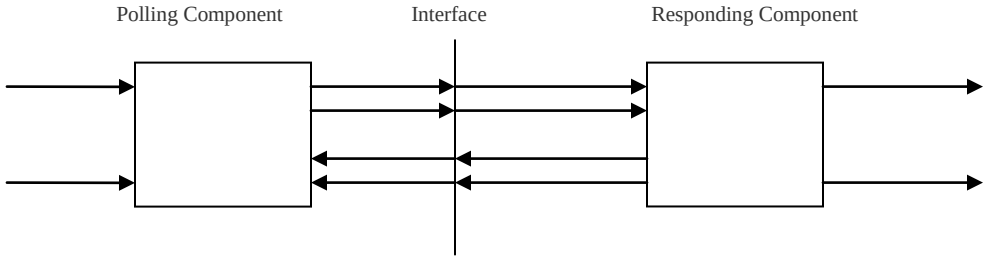
Associated with

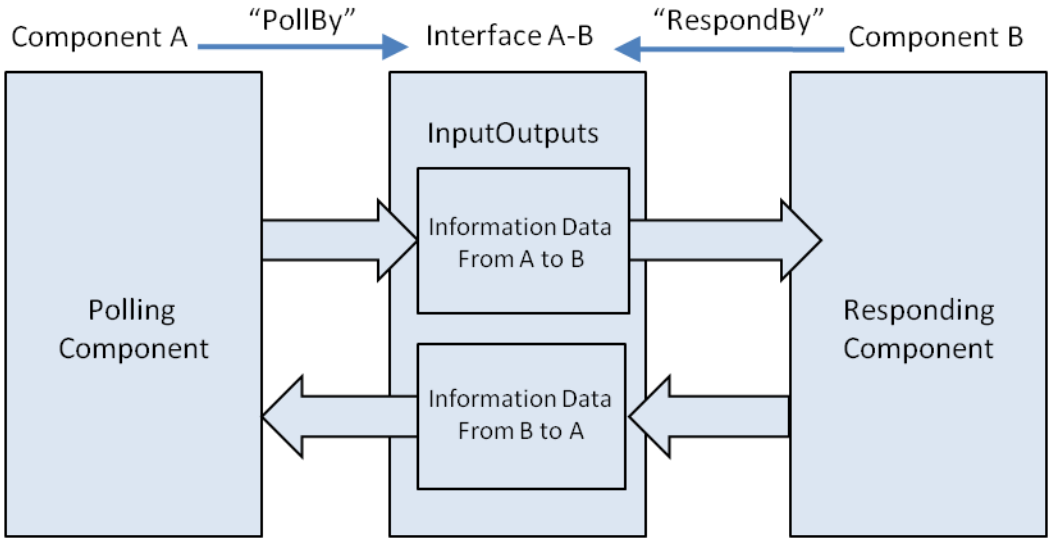
- A Project where the Project is a target concept and association type is "IsUsedOn"
- A Package where the Package is a target concept and association type is "IsMemberOf"

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

Parent: Project, Package

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 Components.

First set of Interface’s InputOutputs belongs to the communication flow from the Polling Component which is in the “PollBy” association with the Interface to the Responding Component 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 concept and association type is “IsOutputOf”. When InputOutput from this set is associated with any Function or Requirement from Polling component, association type is “IsOutputOf”. When InputOutput from this set is

associated with any Function or Requirement from Responding Component, association type is "IsInputOf".

Second set of InputOutputs belongs to the communication flow from the Responding Component which is in the "RespondBy" association with the Interface to the Polling Component 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 concept and association type is "IsInputOf". When an InputOutput from this set is associated with any Function or Requirement from Polling Component, association type is "IsInputOf". When InputOutput from this set is associated with any Function or Requirement from Responding Component, 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 concept and association type is "IsObjectiveOf"
- A Function where the Function is a target concept and association type is "IsObjectiveOf"
- An Interface where the Interface is a target concept and association type is "IsObjectiveOf"
- A Component where the Component is a target object and association type is "IsObjectiveOf"
- A Model where the Model is a target concept and association type is "IsObjectiveOf"
- A Requirement where the Requirement is a target concept and association type is "IsObjectiveOf"
- An InputOutput where the InputOutput is a target concept and association type is "IsObjectiveOf"
- A Package where the Package is a target concept and association type is "IsMemberOf"

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

2.1.11 Package

Package is used to pack together concepts and perform single operation on them.

Attribute	***Type***	***Description***
Id	String256	Unique ID
Name	String256	Package's name
description	String4000	Detailed description
version	String256	Version of the Package

2.1.12 Project

Project is a root concept of the model that "collects" all concepts' 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 concept 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 concept and association type is "IsUsedOn"
- A Package where the Package is a target concept and association type is “IsMemberOf”
- A Function where the Function is a target concept and association type is “IsRequirementFor”
- An Interface where the Interface is a target concept and association type is “IsRequirementFor”

Parent: Project, Package, Function, Interface