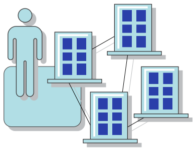
Unified Architecture Method

Business Perspective Language UML Profile

Contents

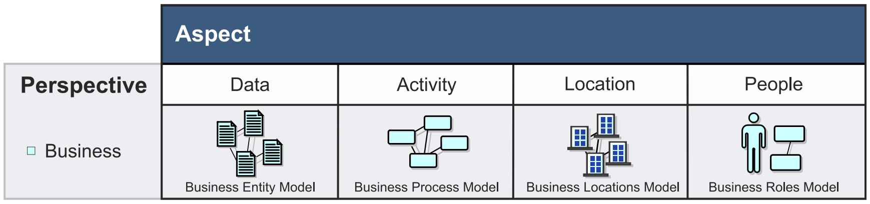
Ottawa, Ontario, Canada

Introduction



In UAM, the Business Perspective is defined as a set of viewpoints that collectively define a complete conceptual level model for the business, at the business level and using business terminology. The term *Business Perspective* is used instead of business model since this latter term brings with it many pre-conceived notions and creates confusion. *Perspective* also implies that there is no one view, but a set of views used to define the complete architecture.

The Business Perspective describes the system under study at the conceptual level. The Business Perspective language is an integrated set of four viewpoint languages. Each cell in Figure 1 defines a *viewpoint* for the architecture, with an associated *viewpoint language* that is used to document the architecture. These individual viewpoint languages are combined to define the *perspective language*, the Business Perspective language. It unites these individual viewpoint languages into a comprehensive, coherent and comprehensible language for defining the complete system architecture from that perspective.



	Aspect			
Perspective	Data	Activity	Location	People
□ Business	Business Entity Model	Business Process Model	Business Locations Model	Business Roles Model

Figure 1 – Business Perspective Viewpoints

The UML profile described here defines the modelling language for the Business Perspective and included viewpoints in UAM. They define the *structural* and *behavioural component signatures* for the viewpoint languages. The perspective language is summarized followed by detailed descriptions of each of the four viewpoints and their language elements.

Business Perspective

The Business Perspective Language is based upon a very small subset of BPMN elements, with the addition of business entities, business locations, and business roles elements. Each element is described, including a definition of the metamodel—the structure and relationship rules for the Business Perspective Language (i.e. the grammar).

> **Note**: OMG was just receiving proposals for UML Profiles for BPMN when this book was written. Therefore the profiles defined for the Business, Logical and Technical Perspectives will eventually migrate to the officially approved OMG UML profiles for BPMN.

An overview of the language components and relationships is show in Figure 2.

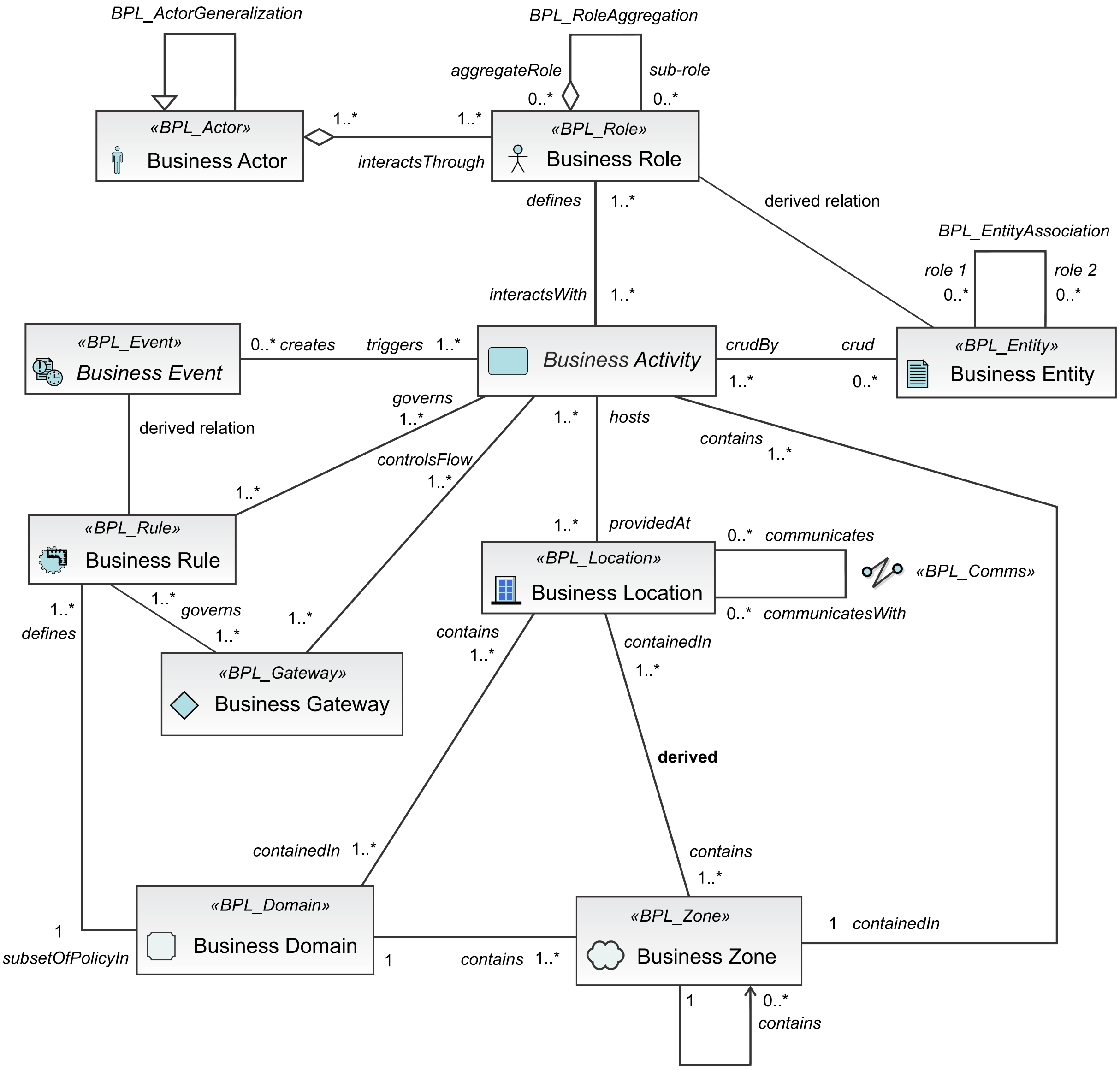


Figure 2 – Business Perspective Language Overview in UML

This metamodel defines the Business Perspective language, which in words says:

- A Business Actor aggregates (has) one or more associated Business Roles;
- A Business Actor interacts with the system through one or more Business Roles;

- A Business Actor may be generalized (or specialized);
- A Business Role interacts with one or more Business Activities;
- A Business Activity defines one or more Business Roles;
- A Business Role may be aggregated into another role;
- A Business Event triggers one or more Business Activities;
- A Business Activity may create zero or more Business Events;
- A Business Activity may contain a (hierarchical) structure of (more detailed) activities (i.e. it may be a sub-process);
- Business Activities may create, read, update, and delete (CRUD) Business Entities;
- Business Entities may have (named) associations between them (in support of the business) with roles defining the relationship;
- Business Entities have derived associations to Business Roles (and in turn to Actors and specific Users) through Business Activities;
- Business Activities are provided at one or more Business Locations;
- Business Locations may communicate with each other;
- Business Locations contain one or more Business Zones;
- A Business Zone may contain other Business Zones;
- A Business Zone is contain in one Business Domain;
- Business Rules govern one or more Business Activities;
- Business Rules govern one or more Business Gateways;
- A Business Domain defines the Business Rules (policies) that govern the Gateways, and Activities (and by extension the Locations and Zones);
- A Business Domain contains one or more Business Zones;
- Business Rules have derived associations to Business Events through a Business Activity.

Other derived associations are not shown on the diagram, for example, between Business Rule and Business Entity (e.g. security access rules) and between Business Event and Business Entity (e.g. Business Entity based events).

How this overall perspective language structure is used within each of the four viewpoints (i.e. Business Entity Model, Business Process Model, Business Locations Model, and Business Roles Model) is described in the sections that follow.

Each element of the Business Perspective Language is described below in terms of a UML profile, specifying:

- Stereotype name for the element;
- UML elements extended;
- Element semantics;
- Element properties;
- Element (graphical) notation;
- Element constraints.

Business Entity Viewpoint

The Business Entity Model defines the high-level business entities involved in the enterprise or system, as defined by the scope of the modelling effort. The objective is to clearly define the business entities involved, and the relationships between them. The definition must be valid for the defined scope, and must eliminate any ambiguity regarding what they represent.

Relationships between entities are derived directly from analysis of the business, its functions and processes. Two main aspects are important, the relationships supporting the business processes and services, and the relationships supporting the monitoring and management of the business. Note that generalizations, aggregations and other types of analysis are not done, and shouldn't be required when working at this conceptual business level of analysis. Only simple associations are modeled.

More information on the Business Entity Model, its recommended structure and content along with how-to advice:

Guidance > Guidelines > Business Entity Model

The business actors and business roles that work with these entities are also identified. This is simply a start at identifying these important relationships. The majority of the analysis takes place at the logical level where the structure of actors and roles is used, along with their relationships to business entities, to identify the access controls aspect of IT security required.

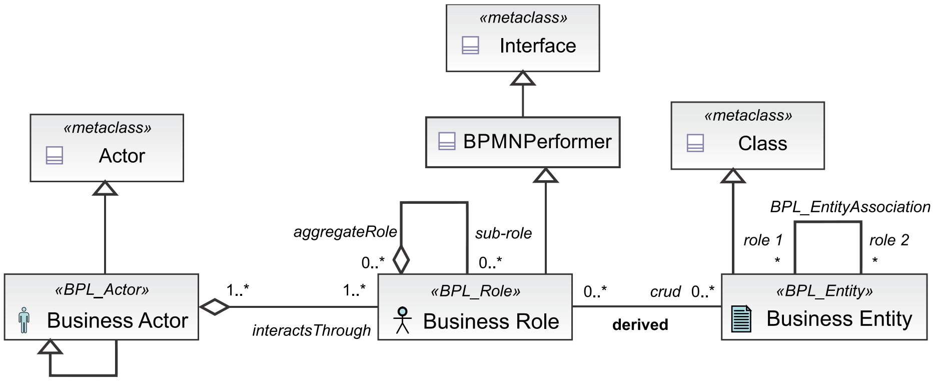


Figure 3 – Business Entity Viewpoint Language

Figure 3 shows the components of the Business Entity modelling language (i.e. the metamodel) along with the relationship allowed between them.

Note that the Business Entity element is the only element in the Business Entity Viewpoint Language; the other elements are shown for context. That is, in future iterations of the Business Entity Model (i.e. after development of the Logical Perspective) it is possible and desirable to define Business Roles and illustrate the relationships between Business Roles and Business Entities and Business Entities and Data Objects.

The elements defined for the Business Entity Viewpoint Language are:

«stereotype» BPL_Entity

Extends

«metaclass» Class

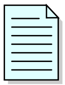
Semantics

It is a representation of the business information and data used within the system being modelled. Business entities identify significant and persistent pieces of information that are manipulated by business actors and activities. No structure is attempted at this point, the sole objective being their identification and creation of clear and precise definitions; however relationships between business entities and business roles may be identified.

Properties

Name	Description
id: string	This attribute is used to uniquely identify model elements.
name: string	A descriptive name for the entity.
owner: string	Defines the owner of the information—the organizational element that makes usage and access decisions about the information. Normally defined as a specific organizational position within the enterprise or business line (e.g., COO).
attribute: variable	The many defined attributes of the entity, including the type. As many attributes are added as necessary to properly model the entity. Optional for Business Level.

Notation



Constraints

- Shall not have any owned operations;
- Shall not have any owned behaviours;
- May have relationships with only Business Roles or with other Business Entities;
- All owned properties shall be public.

«stereotype» BPL_EntityAssociation

Extends

«metaclass» Association

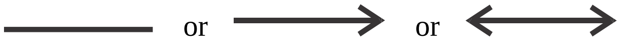
Semantics

Used to define a (*simple* Business level) Association between two Business Entities in the model. Roles are assigned if desired to indicate how the Entities are related or used.

Properties

Name	Description
id: string	This attribute is used to uniquely identify model elements.
name: string	A descriptive name for the association.
sourceRef: entity	The `Entity` that the Association is connecting from.
targetRef: entity	The `Entity` that the Association is connecting to.
sourceRole: string	The role that the source entity has with the target entity.
targetRole: string	The role that the target entity has with the source entity.
associationDirection: AssociationDirection = None {None | One | Both}	Defines whether or not the Association shows any directionality with an arrowhead. The default is `None` (no arrowhead). A value of `One` means that the arrowhead shall be at the Target Entity. A value of `Both` means that there shall be an arrowhead at both ends of the association line.

Notation



Constraints

➲ May define simple relationships between any two `BPL_Entity` elements;

Business Process Viewpoint

The Business Process viewpoint defines the high-level business activities and processes involved in the enterprise or system, as defined by the scope of the modelling effort. The objective is to clearly define the business processes involved. The definition must be valid for the defined scope, and must eliminate any ambiguity regarding what they represent.

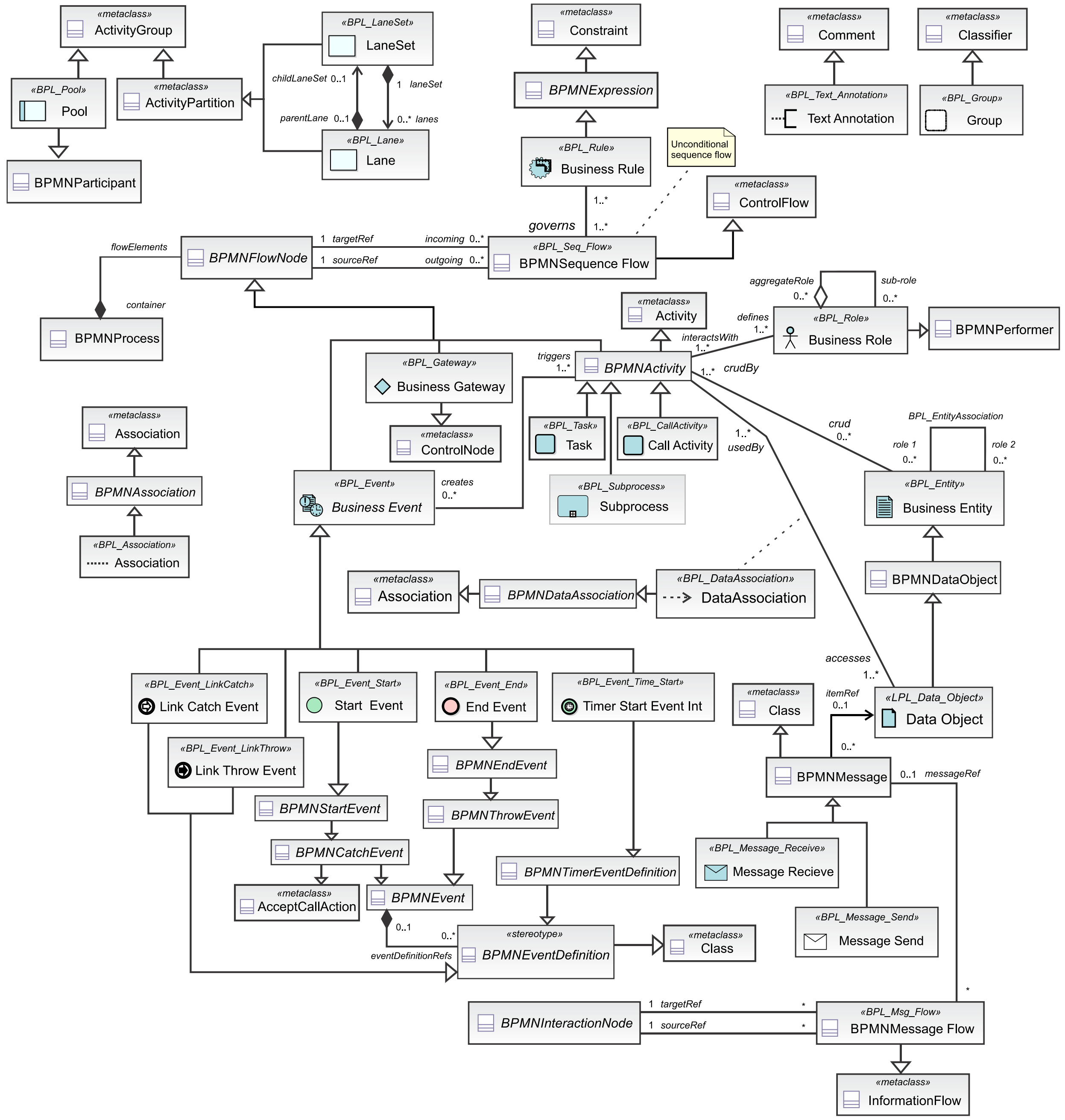


Figure 4 – Business Process Viewpoint Language

> Note that the Business Entity element is shown for context and is not part of the Business Process Viewpoint Language. However, it may be desirable to illustrate the relationships between Activities and Data Objects in later iterations of the Business Process Model

The workflow of a business activity describes what the business must do to provide the value the served business role requires. A business process consists of a sequence of activities that, together, produce something of value for the business role (and associated actor). The workflow often consists of a basic flow and one or more alternative flows. The structure of the workflow is described graphically with the help of activity diagrams. In summary, a business process is a sequence of activities that, together, produce something of value for the business role. Activities may have hierarchical structure defined in order to capture the level of detail required by the context and objective of the architecture effort.

Optionally, relationships between Business Activities and Business Entities may be defined—namely, the Business Entities created, read, updated or deleted (CRUD) by the Activity.

> More information on the Business Process Model, its recommended structure and content along with how-to advice:
>
> *Guidance > Guidelines > Business Process Model*

The elements defined for the Business Process Viewpoint Language are:

«stereotype» BPL_Association

Extends

«*metaclass*» Association

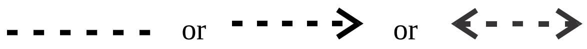
Semantics

Used to define a (Business level) Association between two elements in the model; used to associate information and Artifacts with *Flow Objects*. Text and graphical non-*Flow Objects* can be associated with the *Flow Objects* and Flow. An Association is also used to connect user-defined text (an Annotation) with a *Flow Object* (OMG 2013).

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
associationDirection: AssociationDirection = None {None | One | Both}	`associationDirection` is an attribute that defines whether or not the Association shows any directionality with an arrowhead. The default is None (no arrowhead). A value of `One` means that the arrowhead SHALL be at the Target Object. A value of `Both` means that there SHALL be an arrowhead at both ends of the Association line.
sourceRef: BaseElement	The `BaseElement` that the Association is connecting from.
targetRef: BaseElement	The `BaseElement` that the Association is connecting to.

Notation



Constraints

- May define relationships between any two `baseElements`;
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_CallActivity

Extends

«metaclass» Activity

«metaclass» StructuredActivityNode

Semantics

A Business Call Activity captures the work (business services) performed by the system, which usually consume and produce business entities. A Call Activity is either atomic (i.e. it contains no further detail) or compound (i.e. it contains further detail, it is a Subprocess) and is a wrapper for globally defined functionality that is reused within a process. They are often initially defined with no structure applied, the sole objective being their identification and clear and precise definition, with detail being added in further iterations.

Properties

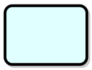
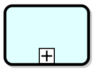
Name	Description
description: string	A description of the business activities, task, and functions captured by the activity.
owner: string	Defines the owner of the business activity or process—the organizational element that makes decisions about the process.
crud: BPL_Entity [0..*]	References to the Entities that are Created, Read, Updated or Deleted by this activity.

In addition to the above, **Call Activities** have the following attributes:

Name	Description (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
isForCompensation: boolean = false	A flag that identifies whether this **Activity** is intended for the purposes of *compensation*. If *false*, then this **Activity** executes as a result of normal execution flow. If *true*, this **Activity** is only activated when a **Compensation Event** is detected and initiated under **Compensation Event** visibility scope.
loopCharacteristics: LoopCharac-teristics [0..1]	An **Activity** MAY be performed once or MAY be repeated. If repeated, the **Activity** MUST have `loopCharacteristics` that define the repetition criteria (if the `isExecutable` attribute of the **Process** is set to *true*).
resources: ResourceRole [0..*]	Defines the resource that will perform or will be responsible for the **Activity**. The resource, e.g., a performer, can be specified in the form of a specific individual, a group, an organization role or position, or an organization.
default: SequenceFlow [0..1]	The **Sequence Flow** that will receive a *token* when none of the `conditionExpressions` on other *outgoing* **Sequence Flows** evaluate to *true*. The *default* **Sequence Flow** should not have a

Name	**Description** (OMG 2013)
	conditionExpression. Any such Expression SHALL be ignored.
ioSpecification: Input OutputSpecification [0..1]	The InputOutputSpecification defines the *inputs* and *outputs* and the InputSets and OutputSets for the **Activity**. See page 211 for more information on the InputOutputSpecification.
properties: Property [0..*]	Modeler-defined properties MAY be added to an **Activity**. These properties are contained within the **Activity**.
boundaryEventRefs: BoundaryEvent [0..*]	This references the **Intermediate Events** that are attached to the boundary of the **Activity**.
dataInputAssociations: DataInputAssociation [0..*]	An optional reference to the DataInputAssociations. A DataInputAssociation defines how the DataInput of the **Activity's** InputOutputSpecification will be populated.
dataOutputAssociations: DataOutputAssociation [0..*]	An optional reference to the DataOutputAssociations.
startQuantity: integer = 1	The default value is 1. The value MUST NOT be less than 1. This attribute defines the number of *tokens* that MUST arrive before the **Activity** can begin. Note that any value for the attribute that is greater than 1 is an advanced type of modeling and should be used with caution.
completionQuantity: integer = 1	The default value is 1. The value MUST NOT be less than 1. This attribute defines the number of *tokens* that MUST be generated from the **Activity**. This number of tokens will be sent done any *outgoing* **Sequence Flow** (assuming any **Sequence Flow** conditions are satisfied). Note that any value for the attribute that is greater than 1 is an advanced type of modeling and should be used with caution.
calledElement: CallableElement [0..1]	The element to be called, which will be either a **Process** or a GlobalTask. Other CallableElements, such as **Choreography**, GlobalChoreographyTask, **Conversation**, and GlobalCommunication MUST NOT be called by the **Call Conversation** element.

Notation

Undefined:  Collapsed: 

Constraints

- Shall not have any owned operations;
- Shall not have any owned behaviours;
- May only have relationships with Business Roles or with other Business Entities.

«stereotype» BPL_Data_Input

Extends

«*metaclass*» ParameterSet

«metaclass» ActivityParameterNode

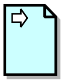
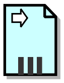
Semantics

A Data Input is a declaration that a particular kind of data will be used as input of the `InputOutputSpecification`. There may be multiple Data Inputs associated with an `InputOutputSpecification` for an Activity. (OMG 2013)

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
inputSetRefs: InputSet [1..*]	A `DataInput` is used in one or more `InputSets`. This attribute is derived from the `InputSets`.
inputSetwithOptional: InputSet [0..*]	Each `InputSet` that uses this `DataInput` can determine if the Activity can start executing with this `DataInput` state in “unavailable.” This attribute lists those `InputSets`.
inputSetWithWhileExecuting: Inputset [0..*]	Each `InputSet` that uses this `DataInput` can determine if the Activity can evaluate this `DataInput` while executing. This attribute lists those `InputSets`.
isCollection: boolean = false	Defines if the DataInput represents a collection of elements. It is needed when no `itemDefinition` is referenced. If an `itemDefinition` is referenced, then this attribute *must* have the same value as the `isCollection` attribute of the referenced `itemDefinition`. The default value for this attribute is *false*.

Notation

 or  (collection)

Constraints

- Has a relationship with an `inputSet` (`inputSetRefs`);
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Data_Object and BPL_Data_Collection

Extends

«metaclass» Class

Semantics

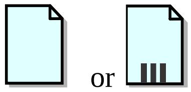
Defines a (Business) Data Object (`DataObject`) is the primary construct for modeling data within a Process flow. It has a well-defined lifecycle, with resulting access constraints (OMG 2013).

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the

	element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
isCollection: boolean = false	Defines if the Data Object represents a collection of elements. It is needed when no `itemDefinition` is referenced. If an `itemDefinition` is referenced, then this attribute MUST have the same value as the `isCollection` attribute of the referenced `itemDefinition`.

Notation



Constraints

- A kind of `FlowElement` and `ItemAwareElement` that has relationships defined through a `DatObjectReference`;
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Data_Output

Extends

«metaclass» ParameterSet

«metaclass» ActivityParameterNode

Semantics

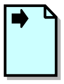
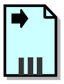
A Data Output is a declaration that a particular kind of data will be used as output of the `InputOutputSpecification`. There may be multiple Data Outputs associated with an `InputOutputSpecification` for an Activity. (OMG 2013)

Properties

Name	Description (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
outputSetRefs: OutputSet [1..*]	A `DataOutput` is used in one or more `OutputSets`. This attribute is derived from the `OutputSets`.
outputSetwithOptional: OutputSet [0..*]	Each `OutputSet` that uses this `DataOutput` can determine if the Activity can start executing with this `DataOutput` state in “unavailable.” This attribute lists those `OutputSets`.
outputSetWithWhileExecuting: OutputSet [0..*]	Each `OutputSet` that uses this `DataOutput` can determine if the Activity can evaluate this `DataOutput` while executing. This attribute lists those `OutputSets`.
isCollection: boolean = false	Defines if the DataOutput represents a collection of elements. It is needed when no `itemDefinition` is referenced. If an `itemDefinition` is referenced, then this attribute *must* have the same value as the

Name	**Description** (OMG 2013)
	`isCollection` attribute of the referenced `itemDefinition`. The default value for this attribute is *false*.

Notation

 or  (collection)

Constraints

- Has a relationship with an `outputSet` (`outputSetRefs`);
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_DataAssociation

Extends

«metaclass» Association

Semantics

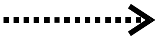
The core concepts of (Business) DataAssociations are that they have sources, a target, and an optional transformation. When a data association is "executed," data is copied to the target. What is copied depends if there is a transformation defined or not. If there is no transformation defined or referenced, then only one source *must* be defined, and the contents of this source will be copied into the target (OMG 2013).

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
transformation: Expression [0..1]	Specifies an optional transformation `Expression`. The actual scope of accessible data for that `Expression` is defined by the source and target of the specific **Data Association** types.
assignment: Assignment [0..*]	Specifies one or more data elements `Assignments`. By using an `Assignment`, single data structure elements can be assigned from the source structure to the target structure.
sourceRef: ItemAwareElement [0..*]	Identifies the source of the **Data Association**. The source *must* be an `ItemAwareElement`.
targetRef: ItemAwareElement	Identifies the target of the **Data Association**. The target *must* be an `ItemAwareElement`.

-

Notation



Constraints

- May define relationships between `ItemAwareElements`;
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Event_End

Extends

«metaclass» FinalNode

«metaclass» CallOperationAction

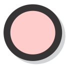
Semantics

A Business Event is something that happens internally or externally to the system that drives the activities and processes of the system being modelled. An End Event represents a significant occurrence in the activities of the business that stops a Business Activity or Process. A Terminate Business Event can be anything from a timer, a trigger in a data base, a user action of some sort, or the arrival or modification of a Business Entity or other artefact. Note that the Business Event element is abstract and is therefore not instantiated within Business Process Models.

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.

Notation



Constraints

- Shall not have any owned operations;
- May only have relationships with other Flow Elements;
- Shall not have any owned behaviours.

«stereotype» BPL_Event_Link_Catch

Extends

«metaclass» AcceptEventAction

Semantics

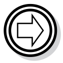
Defines a (Business) Link Catch Event. A Link Event is a mechanism for connecting two sections of a Process. Link Events can be used to create looping situations or to avoid long Sequence Flow lines. The use of Link Events is limited to a single Process level (i.e., they cannot link a *parent* Process with a Sub-Process). Paired Link Events can also be used as “Off-Page Connectors” for printing a Process across multiple pages. (OMG 2013).

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	If the *trigger* is a *Link*, then the name MUST be entered.

Name	**Description** (OMG 2013)
sources: LinkEventDefinition [1..*]	Used to reference the corresponding 'catch' or 'target' `LinkEventDefinition`, when this `LinkEventDefinition` represents a 'throw' or 'source' `LinkEventDefinition`.
target: LinkEventDefinition [1]	Used to reference the corresponding 'throw' or 'source' `LinkEventDefinition`, when this `LinkEventDefinition` represents a 'catch' or 'target' `LinkEventDefinition`.

Notation



Constraints

- May have relationships (sequence flows) with other flow elements as defined in Table 1;
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Event_Link_Throw

Extends

«metaclass» CallOperationAction

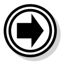
Semantics

Defines a (Business) Link Throw Event. A Link Event is a mechanism for connecting two sections of a Process. Link Events can be used to create looping situations or to avoid long Sequence Flow lines. The use of Link Events is limited to a single Process level (i.e., they cannot link a *parent* Process with a Sub-Process). Paired Link Events can also be used as "Off-Page Connectors" for printing a Process across multiple pages. (OMG 2013).

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	If the *trigger* is a *Link*, then the name MUST be entered.
sources: LinkEventDefinition [1..*]	Used to reference the corresponding 'catch' or 'target' `LinkEventDefinition`, when this `LinkEventDefinition` represents a 'throw' or 'source' `LinkEventDefinition`.
target: LinkEventDefinition [1]	Used to reference the corresponding 'throw' or 'source' `LinkEventDefinition`, when this `LinkEventDefinition` represents a 'catch' or 'target' `LinkEventDefinition`.

Notation



Constraints

- May have relationships (sequence flows) with other flow elements as defined in Table 1;
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Event_Start

Extends

«*metaclass*» CallEvent

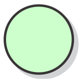
Semantics

A Business Event is something that happens internally or externally to the system that drives the activities and processes of the system being modelled. A Start Event represents a significant occurrence in the activities of the business that requires immediate action, normally triggering actions within a Business Activity. A Start Business Event can be anything from a timer, a trigger in a data base, a user action of some sort, or the arrival or modification of a Business Entity or other artefact. Note that the Business Event element is abstract and is therefore not instantiated within Business Process Models.

Properties

Name	Description (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
isInterrupting: boolean = true	This attribute only applies to **Start Events** of **Event Sub-Processes**; it is ignored for other **Start Events**. This attribute denotes whether the **Sub-Process** encompassing the **Event Sub-Process** should be canceled or not, If the encompassing **Sub- Process** is not canceled, multiple *instances* of the **Event Sub-Process** can run concurrently. This attribute cannot be applied to **Error Events** (where it's always *true*), or **Compensation Events** (where it doesn't apply).

Notation



Constraints

- Shall not have any owned operations;
- May only have relationships with other Flow Elements;
- Shall not have any owned behaviours.

«stereotype» BPL_Event_Time_Start

Extends

«*metaclass*» AcceptEventAction

«*metaclass*» TimeEvent

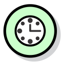
Semantics

Defines a (Business) Timer Start Event that is interrupting. A specific time-date or a specific cycle (e.g., every Monday at 9am) can be set that will trigger the start of the Process. If there is only one `EventDefinition` associated with the Start Event and that `EventDefinition` is of the subclass `TimerEventDefinition`, then the Event is a Timer Start Event (OMG 2013).

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
isInterrupting: boolean = true	This attribute only applies to **Start Events** of **Event Sub-Processes**; it is ignored for other **Start Events**. This attribute denotes whether the **Sub-Process** encompassing the **Event Sub-Process** should be canceled or not, If the encompassing **Sub- Process** is not canceled, multiple *instances* of the **Event Sub-Process** can run concurrently. This attribute cannot be applied to **Error Events** (where it's always *true*), or **Compensation Events** (where it doesn't apply).
timeDate: Expression [0..1]	If the *trigger* is a Timer, then a `timeDate` MAY be entered. Timer attributes are mutually exclusive and if any of the other Timer attributes is set, `timeDate` MUST NOT be set (if the `isExecutable` attribute of the **Process** is set to *true*). The return type of the attribute `timeDate` MUST conform to the ISO-8601 format for date and time representations.
timeCycle: Expression [0..1]	If the *trigger* is a Timer, then a `timeCycle` MAY be entered. Timer attributes are mutually exclusive and if any of the other Timer attributes is set, `timeCycle` MUST NOT be set (if the `isExecutable` attribute of the **Process** is set to *true*). The return type of the attribute `timeCycle` MUST conform to the ISO-8601 format for recurring time interval representations.
timeDuration: Expression [0..1]	If the *trigger* is a Timer, then a `timeDuration` MAY be entered. Timer attributes are mutually exclusive and if any of the other Timer attributes is set, `timeDuration` MUST NOT be set (if the `isExecutable` attribute of the **Process** is set to *true*). The return type of the attribute `timeDuration` MUST conform to the ISO-8601 format for time interval representations.

Notation



Constraints

- May have relationships (sequence flows) with other flow elements as defined in Table 1;
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Gateway

Extends

«metaclass» ControlNode

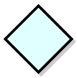
Semantics

A Business Gateway is used to control the process flows—it is a decision point but besides branching merging is also allowed. At the business level they are generic, simply document the inputs and decision rules.

Properties

Name	Description (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
gatewayDirection: GatewayDirection = Unspecified { Unspecified | Converging | Diverging | Mixed }	An attribute that adds constraints on how the **Gateway** MAY be used. ▪ **Unspecified**: There are no constraints. The **Gateway** MAY have any number of *incoming* and *outgoing* **Sequence Flows**. ▪ **Converging**: This **Gateway** MAY have multiple *incoming* **Sequence Flows** but MUST have no more than one (1) *outgoing* **Sequence Flow**. ▪ **Diverging**: This **Gateway** MAY have multiple *outgoing* **Sequence Flows** but MUST have no more than one (1) *incoming* **Sequence Flow**. ▪ **Mixed**: This **Gateway** contains multiple *outgoing* and multiple *incoming* **Sequence Flows**.
default: SequenceFlow [0..1]	The **Sequence Flow** that will receive a *token* when none of the `conditionExpressions` on other *outgoing* **Sequence Flows** evaluate to *true*. The *default* **Sequence Flow** should not have a `conditionExpression`. Any such `Expression` SHALL be ignored.

Notation



Constraints

- Shall not have any owned operations;
- May only have relationships with other Flow Elements;
- Shall not have any owned behaviours.

«stereotype» BPL_Group

Extends

«metaclass» Classifier

Semantics

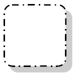
The Group object is an `Artifact` (i.e. additional information about a Process that is not directly related to the Sequence Flows or Message Flows) that provides a visual mechanism to group elements of a diagram informally. The grouping is tied to the `CategoryValue` supporting element. That is, a Group is a visual depiction of a single `CategoryValue`. The graphical elements within the Group will be assigned the `CategoryValue` of the Group. `Categories`, which have user-defined semantics, can be used for documentation or analysis purposes. For example, `FlowElements` can be categorized as being customer oriented vs. support oriented (OMG 2013).

Properties

Name	Description (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.

Name	**Description** (OMG 2013)
categoryValueRef: CategoryValue [0..1]	The `categoryValueRef` attribute specifies the `CategoryValue` that the **Group** represents. The name of the `Category` and the value of the `CategoryValue` separated by delineator "." provides the label for the **Group**. The graphical elements within the boundaries of the **Group** will be assigned the `CategoryValue`.

Notation



Constraints

- May group (categorize) any elements, even across Pools and Lanes;
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Lane

Extends

«metaclass» Activity Partition

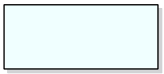
Semantics

A Lane is a sub-partition within a Process (often within a Pool) and will extend the entire length of the Process level, either vertically or horizontally. Lanes are used to organize and categorize Activities within a Pool. The meaning of the Lanes is up to the modeller. (OMG 2013)

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the Lane.
partitionElement: BaseElement [0..1]	A reference to a `BaseElement` that specifies the partition value and partition type. Using this partition element a **BPMN** compliant tool can determine the `FlowElements` that have to be partitioned in this **Lane**.
partitionElementRef: BaseElement [0..1]	A reference to a `BaseElement` that specifies the partition value and partition type. Using this partition element a **BPMN** compliant tool can determine the `FlowElements` that have to be partitioned in this **Lane**.
childLaneSet: LaneSet [0..1]	A reference to a `LaneSet` element for embedded Lanes**.**
flowNodeRefs: FlowNode [0..*]	The list of `FlowNodes` partitioned into this Lane according to the `partitionElement` defined as part of the Lane element.

Notation



Constraints

- May have relationships (message flows) with other message flow elements as defined in Table 2.
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_LaneSet

Extends

«metaclass» Activity Partition

Semantics

The `LaneSet` element defines the container for one or more `Lanes`. A Process can contain one or more `LaneSets`. Each `LaneSet` and its Lanes can partition the *Flow Nodes* in a different way. (OMG 2013)

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	The name of the `LaneSet`. A `LaneSet` is not visually displayed on a **BPMN** diagram. Consequently, the name of the `LaneSet` is not displayed as well.
process: Process	The **Process** owning the `LaneSet`
lanes: Lane [0..*]	One or more **Lane** elements, which define a specific partition in the `LaneSet`.
parentLane: Lane [0..1]	The reference to a **Lane** element which is the parent of this `LaneSet`.

Notation

Constraints

- May have relationships (message flows) with other message flow elements as defined in Table 2.
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Message_Flow

Extends

«metaclass» Association

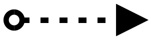
Semantics

Defines a (Business) Message Flow which is used to show the flow of Messages between two Participants that are prepared to send and receive them. A Message Flow *must* connect two separate Pools. They connect either to the Pool boundary or to Flow Objects within the Pool boundary. They *must not* connect two objects within the same Pool. (OMG 2013)

Properties

Name	Description (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	Name is a text description of the **Message Flow**.
sourceRef: InteractionNode	The `InteractionNode` that the **Message Flow** is connecting from. Of the types of `InteractionNode`, only **Pools**/*Participants*, **Activities**, and **Events** can be the *source* of a **Message Flow**.
targetRef: InteractionNode	The `InteractionNode` that the **Message Flow** is connecting to. Of the types of `InteractionNode`, only **Pools**/*Participants*, **Activities**, and **Events** can be the *target* of a **Message Flow**.
messageRef: Message [0..1]	The `messageRef` model association defines the **Message** that is passed via the **Message Flow**.

Notation



Constraints

- Specifies relationships (message flows) between message flow elements as defined in Table 2.
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Message_Receive

Extends

«metaclass» Class

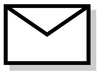
Semantics

Defines a (Business) Receive Message. A Message represents the content of a communication between two *Participants*. In BPMN 2.0, a Message is a graphical decorator (it was a supporting element in BPMN 1.2). An `ItemDefinition` is used to specify the Message structure. (OMG 2013)

Properties

Name	Description (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	Name is a text description of the **Message**
itemRef: ItemDefinition [0..1]	An `ItemDefinition` is used to define the "payload" of the **Message**.

Notation



Constraints

- Part of the message flow definitions with other message flow elements; constraints are defined in Table 2.
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Message_Send

Extends

«metaclass» Class

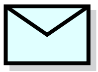
Semantics

Defines a (Business) Send Message. A Message represents the content of a communication between two *Participants*. In BPMN 2.0, a Message is a graphical decorator (it was a supporting element in BPMN 1.2). An `ItemDefinition` is used to specify the Message structure. (OMG 2013)

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	Name is a text description of the **Message**
itemRef: ItemDefinition [0..1]	An `ItemDefinition` is used to define the "payload" of the **Message**.

Notation



Constraints

- Part of the message flow definitions with other message flow elements; constraints are defined in Table 2.
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Pool

Extends

«metaclass» Activity Partition

Semantics

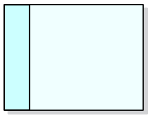
A (Business) Pool is the graphical representation of a `Participant` in a Collaboration. A *Participant* can be a specific `PartnerEntity` (e.g., a company) or can be a more general `PartnerRole` (e.g., a buyer, seller, or manufacturer). A Pool *may* or *may not* reference a Process. A Pool is *not required* to contain a Process, i.e., it can be a "black box." (OMG 2013)

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.

Name	Description (OMG 2013)
name: string [0..1]	Name is a text description of the *Participant*. The name of the *Participant* can be displayed directly or it can be substituted by the associated PartnerRole or PartnerEntity. Potentially, both the PartnerEntity name and PartnerRole name can be displayed for the *Participant*.
processRef: Process [0..1]	The processRef attribute identifies the **Process** that the Participant uses in the *Collaboration*. The **Process** will be displayed within the *Participant's* Pool.
partnerRoleRef: PartnerRole [0..*]	The partnerRoleRef attribute identifies a PartnerRole that the *Participant* plays in the Collaboration. Both a PartnerRole and a PartnerEntity may be defined for the *Participant*. This attribute is derived from the participantRefs of PartnerRole.
partnerEntityRef: PartnerEntity [0..*]	The partnerEntityRef attribute identifies a PartnerEntity that the *Participant* plays in the *Collaboration*. Both a PartnerRole and a PartnerEntity MAY be defined for the *Participant*.This attribute is derived from the participantRefs of PartnerEntity.
interfaceRef: Interface [0..*]	This association defines Interfaces that a *Participant* supports. An Interface defines a set of operations that are implemented by Services.
participantMultiplicity: participantMultiplicity [0..1]	The participantMultiplicityRef model association is used to define *Participants* that represent more than one (1) instance of the *Participant* for a given interaction. See the next section for more details on ParticipantMultiplicity.
endPointRefs: EndPoint [0..*]	This attribute is used to specify the address (or endpoint reference) of concrete services realizing the *Participant*.

Notation



Constraints

- May have relationships (message flows) with other message flow elements as defined in Table 2.
- See BPMN v2.0 (OMG 2013).

«stereotype» BPL_Rule

Extends

«metaclass» Constraint

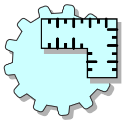
Semantics

A Business Rule defines an actionable rule that the system being modelled needs to support. It is a declaration of policy or a condition that must be satisfied. Business Rules govern the actions and functions within Business Activities, the path taken and tasks performed when following a process for example. They also specify other policy such as access control rules among many others.

Properties

Name	Description (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	Name is a short name / description of the **Business Rule.**
rule: string	The **Business Rule** definition
type: string	Defines the type of business rule (from Ronald Ross): ➢ Term ➢ Fact ➢ Constraint ➢ Derivation

Notation



Constraints

- ➲ Shall not have any owned operations;
- ➲ Shall not have any owned behaviours;
- ➲ May only have relationships with Business Events, Business Activities or Business Entities.

«stereotype» BPL_Seq_Flow

Extends

«metaclass» Association

Semantics

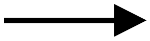
Defines a (Business) Sequence Flow which is used to show the order of Flow Elements in a Process or a Choreography. Each Sequence Flow has only one *source* and only one *target*. The *source* and *target* must be from the set of the following Flow Elements: Events (Start, Intermediate, and End), Activities (Task and Sub-Process; for Processes), Choreography Activities (Choreography Task and Sub-Choreography; for Choreographies), and Gateways (OMG 2013).

Properties

Name	Description (OMG 2013)
sourceRef: FlowNode	The `FlowNode` that the Sequence Flow is connecting from. For a Process: Of the types of `FlowNode`, only Activities, Gateways, and Events can be the *source*. However, Activities that are Event Sub-Processes are not allowed to be a *source*.
targetRef : FlowNode	The `FlowNode` that the Sequence Flow is connecting to. For a Process: Of the types of `FlowNode`, only Activities, Gateways, and Events can be the *target*. However, Activities that are Event Sub-Processes are not allowed to be a *target*.
conditionExpression: Expression [0..1]	An optional boolean `Expression` that acts as a gating condition. A *token* will only be placed on this Sequence Flow if this `conditionExpression` evaluates to true.

Name	Description (OMG 2013)
isImmediate: boolean [0..1]	An optional boolean value specifying whether Activities or Choreography Activities not in the model containing the Sequence Flow can occur between the elements connected by the Sequence Flow. If the value is true, they may not occur. If the value is false, they may occur. Also see the isClosed attribute on Process, Choreography, and Collaboration. When the attribute has no value, the default semantics depends on the kind of model containing **Sequence Flows**: ➢ For non-executable **Processes** (public **Processes** and non-executable private **Processes**) and **Choreographies** no value has the same semantics as if the value were *false*. ➢ For an executable **Process** no value has the same semantics as if the value were *true*. ➢ For executable **Processes**, the attribute MUST NOT be *false*.

Notation



Constraints

- Used to define / diagram the sequence flows between flow elements; constraints are defined in Table 1;
- See BPMN v2.0 (OMG 2013).

Sequence Flow Connections Rules (OMG 2013)

Table 1 displays the BPMN *Flow Objects* and shows how these objects can connect to one another through *Sequence Flows*. These rules apply to the connections within a *Process Diagram* and within a *Choreography Diagram*. The ⇑ symbol indicates that the object listed in the row can connect to the object listed in the column. The quantity of connections into and out of an object is subject to various configuration dependencies are not specified here. Note that if a *Sub-Process* has been expanded within a diagram, the objects within the *Sub-Process* cannot be connected to objects outside of the *Sub-Process*. Nor can *Sequence Flows* cross a *Pool* boundary.

Only those objects that can have incoming and outgoing *Sequence Flows* are shown in the table. Thus, *Pool*, *Lane*, *Data Object*, *Group*, and *Text Annotation* are not listed in the table. Also, the *Activity* shapes in the table represent *Activities* and *Sub-Processes* for *Processes*, and *Choreography Activities* and *Sub-Choreographies* for *Choreography*.

Table 1 – Sequence Flow Connection Rules

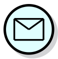
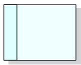
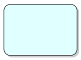
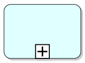
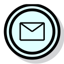
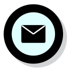
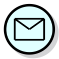
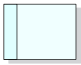
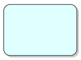
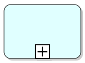
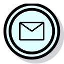
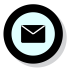
From\To	○	▭	⊞	◇	◎	●
○		⇑	⇑	⇑	⇑	⇑
▭		⇑	⇑	⇑	⇑	⇑
⊞		⇑	⇑	⇑	⇑	⇑
◇		⇑	⇑	⇑	⇑	⇑
◎		⇑	⇑	⇑	⇑	⇑

Message Flow Connection Rules (OMG 2013)

Table 2 displays the *BPMN* modeling objects and shows how these objects can connect to one another through *Message Flows*. These rules apply to the connections within a *Collaboration* diagram. The ⇑ symbol indicates that the object listed in the row can connect to the object listed in the column. The quantity of connections into and out of an object is subject to various configuration dependencies that are not specified here. *Note that Message Flows cannot connect to objects that are within the same Pool*.

Only those objects that can have incoming and outgoing *Message Flows* are shown in the table. Thus, *Lane*, *Gateway*, *Data Object*, *Group*, and *Text Annotation* are not listed in the table.

Table 2 – Message Flow Connection Rules

From\To						
						
	⇑	⇑	⇑	⇑	⇑	
	⇑	⇑	⇑	⇑	⇑	
	⇑	⇑	⇑	⇑	⇑	
	⇑	⇑	⇑	⇑	⇑	
	⇑	⇑	⇑	⇑	⇑	

«stereotype» BPL_Subprocess

Extends

«metaclass» Activity

«metaclass» StructuredActivityNode

Semantics

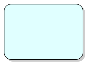
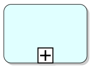
Defines a (Business) Sub-Process, either expanded or collapsed, and is a type of activity. As defined in BPML V2.0 "an Activity is work performed as part of a business process" (OMG 2013); a sub-process contains a set of activities, gateways, events, and sequence flows. It can be viewed as a subroutine that captures a possibly reusable set of activities and flows.

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by

Name	**Description** (OMG 2013)
	something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
isForCompensation: boolean = false	A flag that identifies whether this **Activity** is intended for the purposes of *compensation*. If *false*, then this **Activity** executes as a result of normal execution flow. If *true*, this **Activity** is only activated when a **Compensation Event** is detected and initiated under **Compensation Event** visibility scope
loopCharacteristics: LoopCharacteristics [0..1]	An **Activity** MAY be performed once or MAY be repeated. If repeated, the **Activity** MUST have `loopCharacteristics` that define the repetition criteria (if the `isExecutable` attribute of the **Process** is set to *true*).
resources: ResourceRole [0..*]	Defines the resource that will perform or will be responsible for the **Activity**. The resource, e.g., a performer, can be specified in the form of a specific individual, a group, an organization role or position, or an organization.
default: SequenceFlow [0..1]	The **Sequence Flow** that will receive a *token* when none of the `conditionExpressions` on other *outgoing* **Sequence Flows** evaluate to *true*. The *default* **Sequence Flow** should not have a `conditionExpression`. Any such Expression SHALL be ignored.
ioSpecification: Input OutputSpecification [0..1]	The InputOutputSpecification defines the *inputs* and *outputs* and the `InputSets` and `OutputSets` for the **Activity**.
properties: Property [0..*]	Modeler-defined `properties` MAY be added to an **Activity**. These `properties` are contained within the **Activity.**
boundaryEventRefs: BoundaryEvent [0..*]	This references the **Intermediate Events** that are attached to the boundary of the **Activity**.
dataInputAssociations: DataInputAssociation [0..*]	An optional reference to the `DataInputAssociations`. A `DataInputAssociation` defines how the `DataInput` of the **Activity's** `InputOutputSpecification` will be populated.
dataOutputAssociations: DataOutputAssociation [0..*]	An optional reference to the `DataOutputAssociations`.
startQuantity: integer = 1	The default value is 1. The value MUST NOT be less than 1. This attribute defines the number of *tokens* that MUST arrive before the **Activity** can begin. Note that any value for the attribute that is greater than 1 is an advanced type of modeling and should be used with caution.
completionQuantity: integer = 1	The default value is 1. The value MUST NOT be less than 1. This attribute defines the number of *tokens* that MUST be generated from the **Activity**. This number of tokens will be sent done any *outgoing* **Sequence Flow** (assuming any **Sequence Flow** conditions are satisfied). Note that any value for the attribute that is greater than 1 is an advanced type of modeling and should be used with caution.
state: string = None	The lifecycle of an **Activity** is described; see (OMG 2013).
triggeredByEvent: boolean = false	A flag that identifies whether this **Sub-Process** is an **Event Sub-Process**. ➢ If *false*, then this **Sub-Process** is a normal **Sub-Process**. ➢ If *true*, then this **Sub-Process** is an **Event Sub-Process** and is subject to additional constraints In UAM only *false* is supported, Event Sub-Processes are not used.
artifacts: Artifact [0..*]	This attribute provides the list of **Artifacts** that are contained within the **Sub- Process**.

Notation

Expanded:  Collapsed: 

Constraints

- Shall not have any owned operations;
- Shall not have any owned behaviours;
- May only have relationships with Business Roles or with other Business Entities.

«stereotype» BPL_Task

Extends

«metaclass» Action

Semantics

A Business Task captures the work (business service) performed by the system, which usually consume and produce business entities. A Task is atomic; they do not contain further detail.

Properties

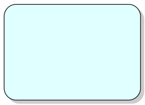
Name	Description
description: string	A description of the business activities, task, and functions captured by the activity.
owner: string	Defines the owner of the business activity or process—the organizational element that makes decisions about the process.
crud: BPL_Entity [0..*]	References to the Entities that are Created, Read, Updated or Deleted by this activity.

In addition to the above, **Tasks** have the following attributes:

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
isForCompensation: boolean = false	A flag that identifies whether this **Activity** is intended for the purposes of *compensation*. If *false*, then this **Activity** executes as a result of normal execution flow. If *true*, this **Activity** is only activated when a **Compensation Event** is detected and initiated under **Compensation Event** visibility scope
loopCharacteristics: LoopCharacteristics [0..1]	An **Activity** MAY be performed once or MAY be repeated. If repeated, the **Activity** MUST have `loopCharacteristics` that define the repetition criteria (if the `isExecutable` attribute of the **Process** is set to *true*).
resources: ResourceRole [0..*]	Defines the resource that will perform or will be responsible for the **Activity**. The resource, e.g., a performer, can be specified in the form of a specific individual, a group, an organization role or position, or an organization.

Name	Description (OMG 2013)
default: SequenceFlow [0..1]	The **Sequence Flow** that will receive a *token* when none of the `conditionExpressions` on other *outgoing* **Sequence Flows** evaluate to *true*. The *default* **Sequence Flow** should not have a `conditionExpression`. Any such Expression SHALL be ignored.
ioSpecification: Input OutputSpecification [0..1]	The InputOutputSpecification defines the *inputs* and *outputs* and the `InputSets` and `OutputSets` for the **Activity**.
properties: Property [0..*]	Modeler-defined `properties` MAY be added to an **Activity**. These `properties` are contained within the **Activity.**
boundaryEventRefs: BoundaryEvent [0..*]	This references the **Intermediate Events** that are attached to the boundary of the **Activity**.
dataInputAssociations: DataInputAssociation [0..*]	An optional reference to the `DataInputAssociations`. A `DataInputAssociation` defines how the `DataInput` of the **Activity's** `InputOutputSpecification` will be populated.
dataOutputAssociations: DataOutputAssociation [0..*]	An optional reference to the `DataOutputAssociations`.
startQuantity: integer = 1	The default value is 1. The value MUST NOT be less than 1. This attribute defines the number of *tokens* that MUST arrive before the **Activity** can begin. Note that any value for the attribute that is greater than 1 is an advanced type of modeling and should be used with caution.
completionQuantity: integer = 1	The default value is 1. The value MUST NOT be less than 1. This attribute defines the number of *tokens* that MUST be generated from the **Activity**. This number of tokens will be sent done any *outgoing* **Sequence Flow** (assuming any **Sequence Flow** conditions are satisfied). Note that any value for the attribute that is greater than 1 is an advanced type of modeling and should be used with caution.
state: string = None	The lifecycle of an **Activity** is described; see (OMG 2013).

Notation



Constraints

- Shall not have any owned operations;
- Shall not have any owned behaviours;
- May only have relationships with Business Roles or with other Business Entities.

«stereotype» BPL_TextAnnotation

Extends

«metaclass» Comment

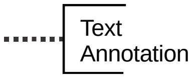
Semantics

Text Annotations are a mechanism for a modeller to provide additional information for the reader of a **BPMN** Diagram (OMG 2013).

Properties

Name	Description (OMG 2013)
text: string	Text is an attribute that is text that the modeller wishes to communicate to the reader of the Diagram.
textFormat: string	This attribute identifies the format of the text. It MUST follow the mime-type format. The default is “text/plain.”

Notation



Constraints

- The **Text Annotation** object can be connected to a specific object on the Diagram with an **Association**, but does not affect the flow of the **Process**. Text associated with the **Annotation** can be placed within the bounds of the open rectangle;
- See BPMN v2.0 (OMG 2013).

Business Locations Viewpoint

The Business Location Model defines the high-level business locations where the system has a presence, as defined by the scope of the modelling effort. The objective is to clearly define at the conceptual level the business locations involved, and the sorts of services provided within and between them. This information is important (in the long term) in deciding on the architecture for applications and services, so that they operate effectively over a slow speed WAN for example. Security implications and requirements are also eventually derived from this type of model, for example at this conceptual level some of these defined “locations” may include security *domains* and *zones*.

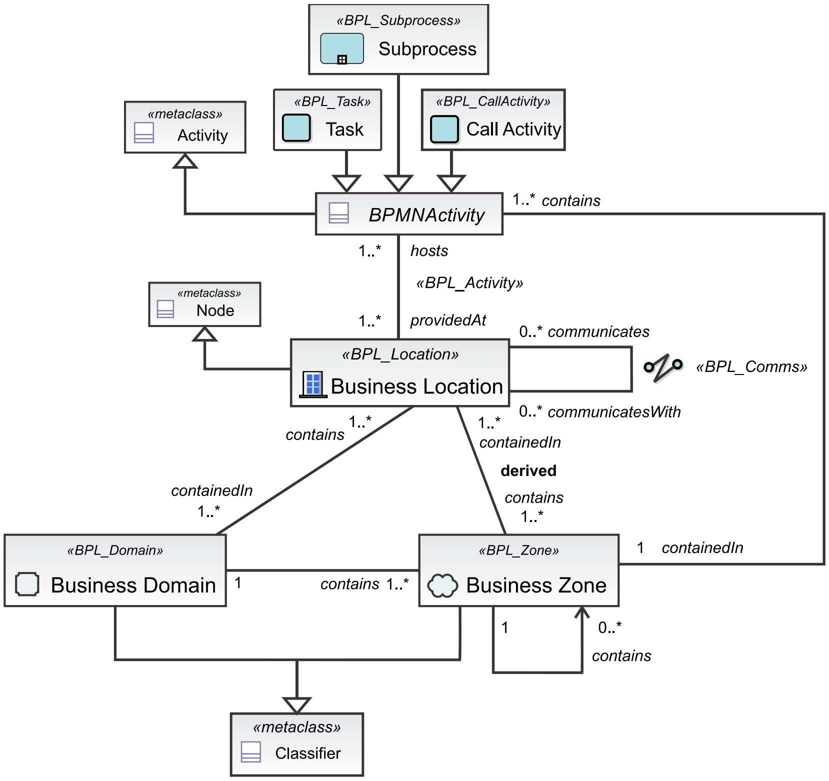


Figure 5 – Business Locations Viewpoint Language

It is important to clearly define the business locations and the relationships between them in terms of the services (activities and workflows) provided and notionally the services provided (data, voice, etc.)

More information on the Business Locations Model, its recommended structure and content along with how-to advice:

Guidance > Guidelines > Business Locations Model

The semantics of the location diagrams are similar to those of UML deployment diagrams, and locations are represented as stereotyped UML *nodes*. In the UML standard, a *node* is a classifier that “... is a physical object that represents a processing resource, generally, having at least a memory and often processing capability as well. Nodes include computing devices but also human resources or mechanical processing resources.” Therefore, UML allows us to extend the semantics of nodes, and the associations that connect them, through stereotyping and the application of tagged values. This is used to define Business Locations and the connections between them.

Activities are shown in Business Locations Models in order to illustrate where and (in the Logical and Technical Perspectives) how these activities are deployed. At eh Business Level there is a lot of granularity and abstraction, therefore high-level activities are likely the only ones appearing within the Business Locations Models. They are contained within Zones, which are contained within a Domain (but since Domains often encompass complete systems and even enterprises the Domain may be simple specified and omitted from the models), which are contained within a Location.

The elements defined for the Business Locations Viewpoint Language are:

«stereotype» BPL_Comms

Extends

«metaclass» CommunicationPath

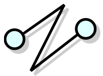
Semantics

Business Communications defines the types of communications required and provided between locations in the Business Locations Model. This communications is conceptual at this point and may therefore represent any form of communications, from the physical transport of goods to electronic communications. A Business Location is the generic or conceptual level locations at which the system (business) under study has a presence.

Properties

Name	Description
id: string	This attribute is used to uniquely identify elements.
name: string	A descriptive name for the Communications.
communicates: string	The *source* end of the communications.
communicatesWith: string	The *destical* end of the communications.
nonFunctional: string	Defines things such as business importance, survivability, and availability requirements of the communications path.

Notation



Constraints

- Shall not have any owned operations;
- Shall not have any owned behaviours;
- May have relationships with only Business Locations.

«stereotype» BPL_Domain

Extends

«metaclass» Node

«metaclass» Classifier

«metaclass» Package

Semantics

A Business (Security) Domain is an environment or context that is defined by security policies, security models, and security architecture, including a set of resources and set of system entities that are authorized to access the resources. A Business Domain is managed by a single *authority*, and may contain one or more sub-domains. Different sub-domains are created when security models or policies (and possibly architecture) are significantly different from one domain to the other, or are conflicting. Separate logical domains provide clearer separation of concerns and ease policy enforcement and system management. Synonyms: security domain or policy domain.

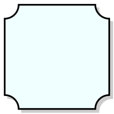
Properties

Name	Description
id: string	This attribute is used to uniquely identify elements.

Name	Description
name: string	A descriptive name for the Domain.
authority: string	The authority for the Domain, normally defined as a specific organizational position within the enterprise or business line (e.g. COO).

Notation

Normally an architecture deals with a single Business Domain and therefore it may be left off (but documented in the preamble to the architecture description), however if it is required in a viewpoint then a simple rectangular background geometric shape (of an appropriate colour if desired to illustrate the fundamental nature of the Domain) may be used to depict the Security Domain, with the *name* applied to one corner or on the boundary (e.g. "COMPANY-CONFIDENTIAL" or "TOP SECRET").



Constraints

- Contains (classifies) one or more Zones;
- Cannot contain other domains.

«stereotype» BPL_Location

Extends

«metaclass» Node

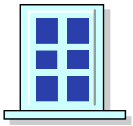
Semantics

A Business Location is the generic or conceptual level locations at which the system (business) under study has a presence. Each location is defined in terms of the Business Activities present and provided at the location along with the types of communications required and provided between these locations.

Properties

Name	Description
id: string	This attribute is used to uniquely identify elements.
name: string	A descriptive name for the location.
activities: Activities[0..*]	A list of the business activities provides at the location, which relates directly to the Business Process Model and how the business (or system under study) wants to deliver its products and services.
nonFunctional: string	Defines things such as business importance, survivability, and availability requirements of the location.
description: string	A description of the location in terms of building, address or other distinguishing characteristics.

Notation



Constraints

- Shall not have any owned operations;

- Shall not have any owned behaviours;
- May have relationships with only Business Roles or with other Business Entities;
- All owned properties shall be public.

«stereotype» BPL_Zone

Extends

«metaclass» Node

«metaclass» Classifier

«metaclass» Package

Semantics

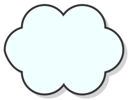
A Business Security Zone is an environment or context that is defined by security policies, security models, and security architecture, including a set of resources and set of system entities that are authorized to access the resources. A Business Zone may contain one or more sub-zones. Different sub-zones are created when security models or policies (and possibly architecture) are significantly different from one zone to the other, or are conflicting. Separate Business Zones provide clearer separation of concerns and ease policy enforcement and system management. Synonyms: security zone or policy zone.

Properties

Name	Description
id: string	This attribute is used to uniquely identify elements.
name: string	A descriptive name for the Business Zone.
owner: string	The owner of the Zone normally defined as a specific organizational position within the enterprise or business line (e.g. HR) which owns the information within the Zone. Sub-zones inherit ownership from the parent Zone. The (parent Domain) **Authority** may delegate responsibilities to Zone owners.
parentDomain: domain [0..1]	The Domain within which the Zone is contained.
parentZone: zone [0..1]	The Zone, if any, within which the Zone is contained.
location: string	The location of the zone within the system or enterprise.

Notation

A Business Security Zone is represented as a simple cloud shaped background geometric shape (of an appropriate colour if desired to illustrate the fundamental nature of the Zone) with the *name* applied to one corner or outside the perimeter (e.g. “HR Services”).



Constraints

- Zones must be wholly contained within other Zones;
- Zones must be wholly contained within one Domain;
- Wholly contains (classifies) Activities, Tasks, and Sub-Processes;
- Zones are defined only when needed.

Business Roles Viewpoint

The Business Roles Model defines the high-level Business Actors and Roles involved in the enterprise or system, as defined by the scope of the modeling effort. The objective is to clearly define the Business Actors and Roles involved, and the sorts of interactions they have with the system. This information is important in deciding on the presentation architecture activities / services, so that usability concerns, integration concerns (and the user level) and other issues are understood. Security implications are also derived from this model—for example, access to information and services may be controlled based upon a user's role (e.g. Analyst vs. reporter vs. editor roles within a newspaper for example).

It is important to clearly define the business roles and the interactions that they have with the system. This information is used by stakeholders, business-process analysts and business designers to understand and improve the way the business interacts with its environment, and by systems analysts and software architects to provide context and requirements for software development specifically the interface to activities and tasks (i.e. Graphical User Interfaces (GUI), etc.).

More information on the Business Roles Model, its recommended structure and content along with how-to advice:

Guidance > Guidelines > Business Roles Model

These elements of the Business Perspective language have already been defined, but it should be noted (and as shown in Figure 6) that there can be multiple roles associated with a Business Activity and that each of these roles may have some specialization defined. This enables the capturing of multiple way of interacting with the system for a given set of business activities and functions. The high-level conceptual perspective that one typically works at (at least initially) when developing the business views demands that the language supports this ability.

Note that in early iterations of the Business Perspective only Business Actors are defined. That is, in future iterations of the Business Roles Model (i.e. after development of the Logical Perspective) it is possible and desirable to finally define Business Roles and illustrate the relationships between Business Roles and the Business Activities (Tasks) that define them.

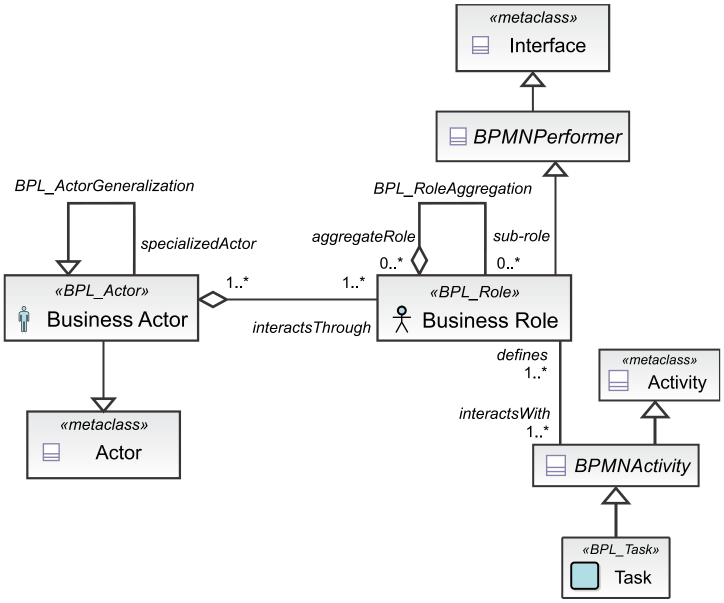


Figure 6 – Business Roles Viewpoint Language

The elements defined for the Business Roles Viewpoint Language are:

«stereotype» BPL_Actor

Extends

«metaclass» Actor

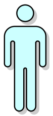
Semantics

A Business Actor can be equated to the external people or systems that interact with the system under study. Actors are not necessarily external to the business; they may also represent internal "user" interactions with the system. Business Actors are high-level conceptualizations of the Actors with the system; further analysis is required to fully understand and define required actors. Note that actors are ***not*** individuals nor are they necessarily equivalent to job titles**;** instead, they describe the behaviour in the enterprise and the responsibilities of the associated "user".

Properties

Name	Description
id: string	This attribute is used to uniquely identify elements.
name: string	A descriptive name for the actor.
characteristics: string	A business level description of the actor and what it represents. Specifics in terms of people, jobs or systems should be avoided at this point—keep it technical

Notation



Constraints

- Must be named;
- Can only have associations to Business Roles. Furthermore these associations must be binary;
- Shall aggregate at least one associated Business Role;
- Allowed to have (derived) associations with Business Entities (through associated business roles);
- Allowed to have (derived) associations with Business Activities (through associated business roles);

«stereotype» BPL_Role

Extends

«metaclass» Interface

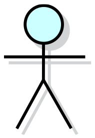
Semantics

A role played in relation to the business by someone or something in the business environment. A role is equated or defined by the behaviour (and by extension its responsibilities) that it interacts with in the system—the activities that it interacts with. A role is in essence the "interface" to the activities with which it interacts. Business Roles are high-level conceptualizations of the Roles with the system; further analysis is required to fully understand and define required roles. A role can be assumed (aggregated) by any kind of actor, either human or a system. An actor may be aggregate multiple roles, and an actor can be associated with multiple physical people (or systems). Roles also include security concepts and constructs. An employee of the enterprise is typically multiple actors with the system and therefore has many different roles.

Properties

Name	Description
id: string	This attribute is used to uniquely identify elements.
name: string	A descriptive name for the role.
description: string	Description of the Role, its interactions and other relevant information such as security aspects.

Notation



Constraints

- Must be named, usually with “er” or “or” extensions on the associated activity name (e.g. “invoicer”, “claimer”, “helper”);
- Can only have associations to Business Actors and Business Activities. Furthermore these associations must be binary;
- Shall be aggregate by at least one associated Business Actor;
- Allowed to have (derived) associations with Business Entities (through associated business roles);
- Allowed to have (derived) associations with Business Activities (through associated business roles);
- Structuring (i.e. aggregation) of Business Roles is allowed, based upon the Business Activities involved.

«stereotype» BPL_ActorGeneralization

Extends

«metaclass» Association

Semantics

This is used to define generalization associations between two Actors in the model that permit better understanding of the Actors, their relationships and capabilities. Analysis may also result in the simplification of the model.

Properties

Name	Description
id: string	This attribute is used to uniquely identify model elements.
name: string	A descriptive name for the association.
sourceRef: entity	The `Entity` that the Association is connecting from
targetRef: entity	The `Entity` that the Association is connecting to
type: AssociationType = Simple {Simple |	**Simple:** a simple association (with or without Roles and Multiplicities) with perhaps a defined meaning; **Aggregation**: represents a part-whole or part-of relationship;

Name	Description
Aggregation | Generalization}	**Generalization**: a specialized form of the other (the *super type*) and super-class is considered as '***Generalization'*** of subclass (without Roles and without Multiplicities).
associationDirection: AssociationDirection = None {None | One | Both}	Defines whether or not the Association shows any directionality with an arrowhead. The default is `None` (no arrowhead). A value of `One` means that the arrowhead shall be at the Target Entity. A value of `Both` means that there shall be an arrowhead at both ends of the association line.

Notation

Generalization association: Specialized Actor ──▷ Generalized Actor

Constraints

- May define relationships between any two `BPL_Actor` elements;

«stereotype» BPL_RoleAggregation

Extends

«metaclass» Association

Semantics

This is used to define aggregation associations between a Role and an Actor in the model that permit better understanding of the Actors and Roles, their relationships and capabilities. Analysis may also result in the simplification of the model.

Properties

Name	Description
id: string	This attribute is used to uniquely identify model elements.
name: string	A descriptive name for the association.
sourceRef: entity	The `Entity` that the Association is connecting from
targetRef: entity	The `Entity` that the Association is connecting to
sourceRole: string	The role that the source entity has with the target entity for simple and aggregation/composition associations.
targetRole: string	The role that the target entity has with the source entity for simple and aggregation/composition associations.
type: AssociationType = Simple {Simple | Aggregation | Generalization}	**Simple:** a simple association (with or without Roles and Multiplicities) with perhaps a defined meaning; **Aggregation**: represents a part-whole or part-of relationship; **Generalization**: a specialized form of the other (the *super type*) and super-class is considered as '***Generalization'*** of subclass (without Roles and without Multiplicities).
associationDirection: AssociationDirection = None {None | One | Both}	Defines whether or not the Association shows any directionality with an arrowhead. The default is `None` (no arrowhead). A value of `One` means that the arrowhead shall be at the Target Entity. A value of `Both` means that there shall be an arrowhead at both ends of the association line.

Notation

Aggregation association: Actor ◇⟶ Role

Constraints

- May define relationships between a `BPL_Role` element and a `BPL_Actor`;

«stereotype» BPL_Task (from Business Process Viewpoint)

Extends

«metaclass» Action

Semantics

A Business Task captures the work (business service) performed by the system, which usually consume and produce business entities. A Task is atomic; they do not contain further detail. Tasks in the Business Roles Model represent the support for the interactions with the system by Actors through Roles.

Properties

Name	**Description** (OMG 2013)
id: string	This attribute is used to uniquely identify BPMN elements. The `id` is REQUIRED if this element is referenced or intended to be referenced by something else. If the element is not currently referenced and is never intended to be referenced, the `id` MAY be omitted.
name: string	A descriptive name for the element.
isForCompensation: boolean = false	A flag that identifies whether this **Activity** is intended for the purposes of *compensation*. If *false*, then this **Activity** executes as a result of normal execution flow. If *true*, this **Activity** is only activated when a **Compensation Event** is detected and initiated under **Compensation Event** visibility scope
loopCharacteristics: LoopCharacteristics [0..1]	An **Activity** MAY be performed once or MAY be repeated. If repeated, the **Activity** MUST have `loopCharacteristics` that define the repetition criteria (if the `isExecutable` attribute of the **Process** is set to *true*).
resources: ResourceRole [0..*]	Defines the resource that will perform or will be responsible for the **Activity**. The resource, e.g., a performer, can be specified in the form of a specific individual, a group, an organization role or position, or an organization.
default: SequenceFlow [0..1]	The **Sequence Flow** that will receive a *token* when none of the `conditionExpressions` on other *outgoing* **Sequence Flows** evaluate to *true*. The *default* **Sequence Flow** should not have a `conditionExpression`. Any such Expression SHALL be ignored.
ioSpecification: Input OutputSpecification [0..1]	The InputOutputSpecification defines the *inputs* and *outputs* and the `InputSets` and `OutputSets` for the **Activity**.
properties: Property [0..*]	Modeler-defined `properties` MAY be added to an **Activity**. These `properties` are contained within the **Activity.**
boundaryEventRefs: BoundaryEvent [0..*]	This references the **Intermediate Events** that are attached to the boundary of the **Activity**.
dataInputAssociations: DataInputAssociation [0..*]	An optional reference to the `DataInputAssociations`. A `DataInputAssociation` defines how the `DataInput` of the **Activity's** `InputOutputSpecification` will be populated.

Name	Description (OMG 2013)
dataOutputAssociations: DataOutputAssociation [0..*]	An optional reference to the `DataOutputAssociations`.
startQuantity: integer = 1	The default value is 1. The value MUST NOT be less than 1. This attribute defines the number of *tokens* that MUST arrive before the **Activity** can begin. Note that any value for the attribute that is greater than 1 is an advanced type of modeling and should be used with caution.
completionQuantity: integer = 1	The default value is 1. The value MUST NOT be less than 1. This attribute defines the number of *tokens* that MUST be generated from the **Activity**. This number of tokens will be sent done any *outgoing* **Sequence Flow** (assuming any **Sequence Flow** conditions are satisfied). Note that any value for the attribute that is greater than 1 is an advanced type of modeling and should be used with caution.
state: string = None	The lifecycle of an **Activity** is described; see (OMG 2013).
triggeredByEvent: boolean = false	A flag that identifies whether this **Sub-Process** is an **Event Sub-Process**. ➢ If *false*, then this **Sub-Process** is a normal **Sub-Process**. ➢ If *true*, then this **Sub-Process** is an **Event Sub-Process** and is subject to additional constraints In UAM only *false* is supported, Event Sub-Processes are not used.
artifacts: Artifact [0..*]	This attribute provides the list of **Artifacts** that are contained within the **Sub- Process**.

Notation

Constraints

- Shall not have any owned operations;
- Shall not have any owned behaviours;
- May only have relationships with Business Roles or with other Business Entities.

Business Perspective Language Summary

The table lists the modelling language elements for the business level viewpoints. These language elements have been selected in order to provide the constructs needed at the business level. It has purposefully been kept simple since the business level models should be simple and easy to understand.

	Aspect			
Perspective	**Data**	**Activity**	**Location**	**People**
▪ Business	**Business Entity View** ➢ Business Entity ➢ Entity Associations ▪ Simple	**Business Process View** ➢ Association ➢ Call Activity ➢ Data Object ➢ DataAssociation ➢ Event End ➢ Event Link Catch ➢ Event Link Throw ➢ Event Start ➢ Event Timer Start ➢ Gateway ➢ Group ➢ Lane / LaneSet ➢ Message Flow ➢ Message Receive ➢ Message Send ➢ Participant (Pool) ➢ Rule ➢ Sequence Flow ➢ Sub-process ➢ Task ➢ Text Annotation	**Business Locations View** ➢ Business Comms ➢ Business Domain ➢ Business Location ➢ Business Zone	**Business Roles View** ➢ Business Actor ➢ Business Role ➢ Task (from BPV) ➢ Relationships: ▪ Actor Generalization ▪ Role Aggregation

Bibliography

OMG. "Business Process Model and Notation (BPMN)." Version 2.0.1 (September 2013): 532.