

received as output from an operation. A parameter descriptor has a relationship with a datatype descriptor, a reference type, a direction, and may have relationships with the following elements: precondition value constraint descriptors and postcondition value constraint descriptors.

**[0157]** A parameter descriptor has a one-to-one aggregation relationship with a datatype descriptor that is the type of value(s) allowed to be assigned to the parameter. A parameter descriptor has a one-to-one aggregation relationship, with a reference type that is the manner in which the parameter is referenced. Generally, a parameter is accessed by reference or by value. Access by reference points to the real value directly. Access by value is a shallow copy of the real value (allowing the child values to be changed by reference, but not allowing the parameter value to be changed). A parameter descriptor has a one-to-one aggregation relationship with a direction that is one of: In, Out, or InOut. The direction determines the originator of the parameter reference. "In" means an external process passes in the parameter. "Out" means this process must set the parameter before completion. InOut means an external process passes in the parameter and this process is allowed to change the parameter reference before completion. Out and InOut require the Reference Type to be "by reference". A parameter descriptor has a zero-to-many association relationship, with a precondition value constraint descriptors that are each a description of the restrictions on allowed values held by the parameter implementation. A parameter descriptor has a zero-to-many association relationship with postcondition value constraint descriptors that are each a description of the restrictions placed on the values allowed for each parameter after the execution of the operation.

**[0158]** A parameter descriptor describes a parameter, but does not implement that parameter, hold the parameter value, or know how to access or change that value. No additional events are added by the parameter descriptor.

**[0159]** A method descriptor of the present invention is a feature descriptor and an operation descriptor. A method descriptor is an operation participating in a model. The method descriptor is logically related to the model in which the method descriptor participates. Instance methods must be performed using an instance of the model to which it belongs. Instance methods may access both instance and static attribute values from the model to which it belongs. Static methods do not require an instance of a model and may only access static attributes of the model. A method descriptor has a relationship with a return data descriptor and may have a relationship with the following elements: precondition constraint descriptors, postcondition constraint descriptors, and signal descriptors.

**[0160]** A method descriptor has a one-to-one aggregation relationship with a return data descriptor that is the type of the value(s) allowed to be returned from an execution of the operation. The default type for return type is void. A method descriptor has a zero-to-many association relationship with precondition constraint descriptors that are each description of constraints placed on the system, model or parameters that must be met before the execution of the operation can begin. If precondition constraints are not met, a failure is generated by the constraint and the operation never executes. A method descriptor has a zero-to-many association relationship with postcondition constraint descriptors that are

each a description of constraints placed on the system, model or return value that must be met in order for the operation to be considered a successful execution. If these constraints are not met, the condition fires a failure and the operation fails. A method descriptor has a zero-to-many association relationship with signal descriptors that are each a description of events that may be fired during the execution of the operation and the associated methods to register interest in receiving these event notifications.

**[0161]** A method descriptor describes an operation with a set of attributes, but does not implement that method or know how to invoke that method. No additional events are added by the method descriptor.

**[0162]** A signal descriptor of the present invention is a classifier descriptor. Signal descriptors describe event notifications and the mechanism for registering to receive these notifications. A signal descriptor has a relationship with a listener type and registration operation descriptors and may have a relationship with the following elements: registration constraint descriptors, listener access operation descriptors, and deregistration operation descriptors.

**[0163]** A signal descriptor has a one-to-one association relationship with a listener type that is the interface type required by the generator of the events in order to notify the listener when the event occurs. A listener type uses an identity to lookup the interface descriptor for that type. A signal descriptor has a zero-to-many association relationship with registration constraint descriptors that are constraints that further restrict the models that can register to receive event notification. For example, a registration constraint may limit registration to a single listener. A signal descriptor has a one-to-many aggregation relationship with registration operation descriptors that are the operations implemented by the source of events to allow listeners to register interest in receiving events. A signal descriptor has a one-to-many aggregation relationship with deregistration operation descriptors that are the operations implemented by the source of events to allow listeners to remove themselves from receiving events. A signal descriptor has a zero-to-many aggregation relationship with listener access operation descriptors that are the operations available to investigate the listeners registered to receive event notifications.

**[0164]** A signal descriptor describes a signal, but does not implement that signal. No additional events are added by the signal descriptor.

**[0165]** A constructor descriptor of the present invention is an operation descriptor and a feature descriptor. A constructor descriptor describes the available mechanism to create a new instance of a particular classifier. A constructor descriptor inherits attributes from operation descriptor and feature descriptor, but adds no additional attributes. A constructor descriptor describes a constructor with a set of attributes, but does not implement that constructor or know how to invoke that constructor. No additional events are added by the constructor descriptor.

**[0166]** A destructor descriptor of the present invention is an operation descriptor and a feature descriptor. A destructor descriptor describes the manner in which an instance of a particular model is disposed. A destructor descriptor has a zero-to-many association relationship with precondition constraint descriptors that are each a description of con-