

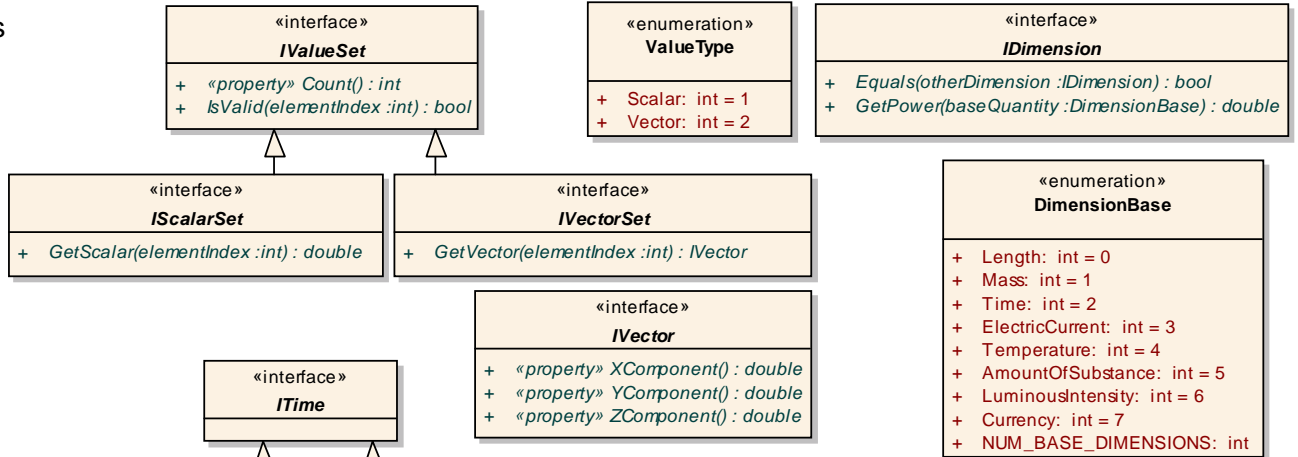


## data definitions

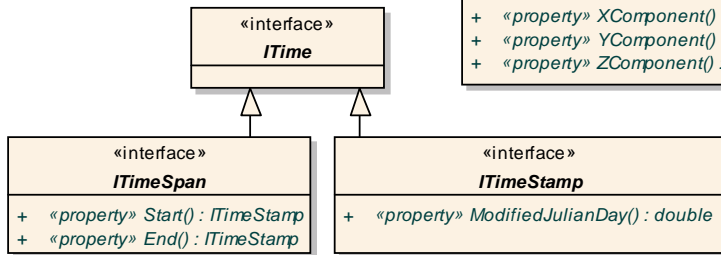
What

# OpenMI

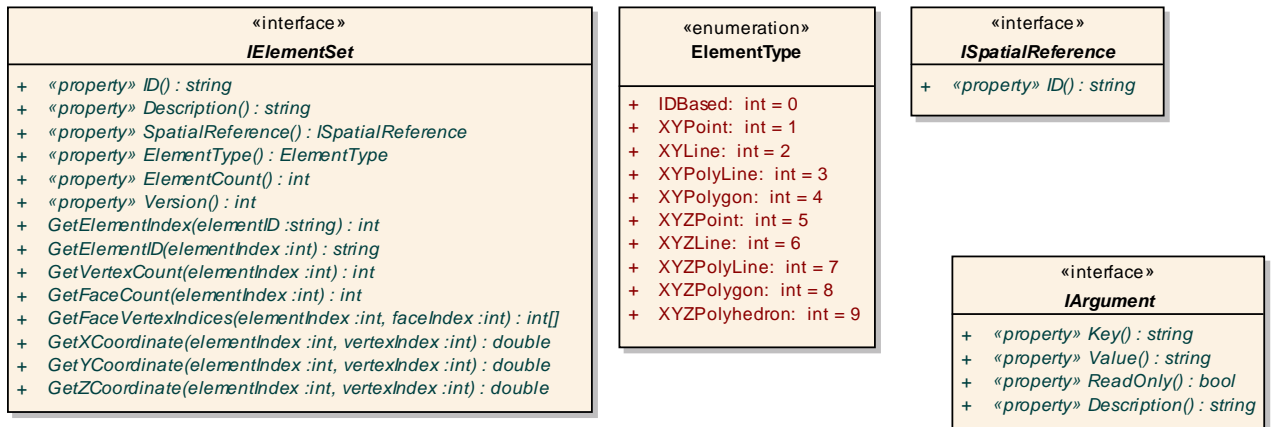
Values



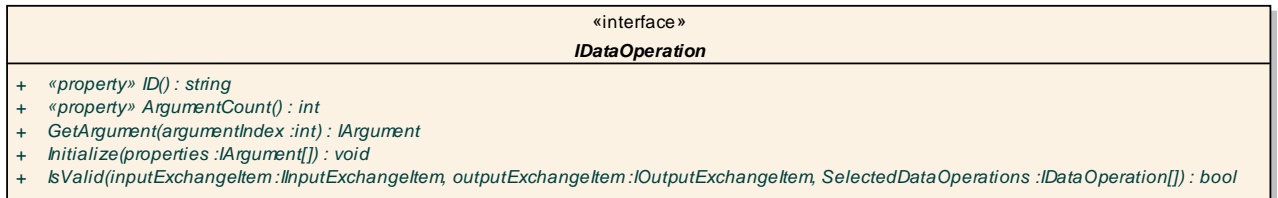
When



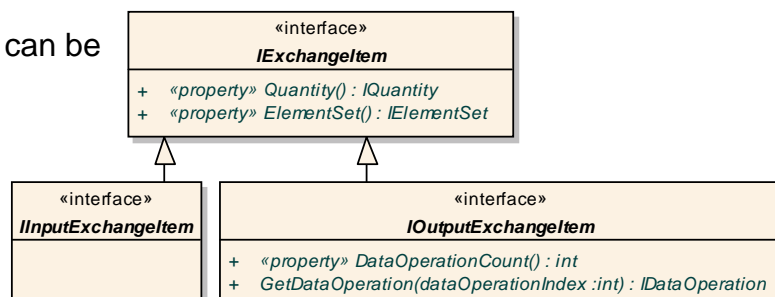
Where



How



meta data to express what can be exchanged





# OpenMI

messaging definitions

specification what will be exchanged and how

```

«interface»
ILink
+ «property» ID() : string
+ «property» Description() : string
+ «property» SourceComponent() : ILinkableComponent
+ «property» TargetComponent() : ILinkableComponent
+ «property» TargetQuantity() : IQuantity
+ «property» TargetElementSet() : IElementSet
+ «property» SourceElementSet() : IElementSet
+ «property» SourceQuantity() : IQuantity
+ «property» DataOperationsCount() : int
+ GetDataOperation(dataOperationIndex :int) : IDataOperation
  
```

```

«interface»
IEvent
+ «property» Type() : EventType
+ «property» Description() : string
+ «property» Sender() : ILinkableComponent
+ «property» SimulationTime() : ITimeStarrp
+ GetAttribute(key :string) : object
  
```

```

«enumeration»
EventType
+ Warning: int = 0
+ Informative: int = 1
+ ValueOutOfRange: int = 2
+ GlobalProgress: int = 3
+ TimeStepProgres: int = 4
+ DataChanged: int = 5
+ TargetBeforeGetValuesCall: int = 6
+ SourceAfterGetValuesCall: int = 7
+ SourceBeforeGetValuesReturn: int = 8
+ TargetAfterGetValuesReturn: int = 9
+ Other: int = 10
+ NUM_OF_EVENT_TYPES: int = 2005
  
```

component interfaces for generic component access

obligatory interface

```

«interface»
IPublisher
+ Subscribe(listener :IListener, eventType :EventType) : void
+ UnSubscribe(listener :IListener, eventType :EventType) : void
+ SendEvent(Event :IEvent) : void
+ GetPublishedEventTypeCount() : int
+ GetPublishedEventType(providedEventTypeIndex :int) : EventType
  
```

```

«interface»
IListener
+ OnEvent(Event :IEvent) : void
+ GetAcceptedEventTypeCount() : int
+ GetAcceptedEventType(acceptedEventTypeIndex :int) : EventType
  
```

```

«interface»
ILinkableComponent
+ «property» ComponentID() : string
+ «property» ComponentDescription() : string
+ «property» ModelID() : string
+ «property» ModelDescription() : string
+ «property» InputExchangeItemCount() : int
+ «property» OutputExchangeItemCount() : int
+ «property» TimeHorizon() : ITimeSpan
+ «property» EarliestInputTime() : ITimeStamp
+ Initialize(properties :IArgument[]) : void
+ GetInputExchangeItem(inputExchangeItemIndex :int) : IInputExchangeItem
+ GetOutputExchangeItem(outputExchangeItemIndex :int) : IOutputExchangeItem
+ AddLink(link :ILink) : void
+ RemoveLink(linkID :string) : void
+ Validate() : string
+ Prepare() : void
+ GetValues(time :ITime, linkID :string) : ValueSet
+ Finish() : void
+ Dispose() : void
  
```

optional extensions

```

«interface»
IManageState
+ KeepCurrentState() : string
+ RestoreState(stateID :string) : void
+ ClearState(stateID :string) : void
  
```

```

«interface»
IDiscreteTimes
+ HasDiscreteTimes(quantity :IQuantity, elementSet :IElementSet) : bool
+ GetDiscreteTimesCount(quantity :IQuantity, elementSet :IElementSet) : int
+ GetDiscreteTime(quantity :IQuantity, elementSet :IElementSet, discreteTimeIndex :int) : ITime
  
```



Starting point:  
The OMI-file

