# SIMF Meta Model::Values

The values package defines the concepts of values and quantities expressed in units.

Values may be differentiated from entities in that values have no independent lifetime or "identity" other than the value its self. E.g. the number 5 "just is" and can't be changed. Properties and relations referencing values can, of course, change but the values are constant.

The failure to properly express units in data models often results in errors, inefficiencies and risk. Translation and federations between models, schema and data sources that is not cognizant of the units used would be even more error prone and risky. For example, what does “Speed limit 50” mean? For these reasons the SIMF language provides specific support for specifying quantity kinds and unit types in conceptual, logical and physical models. The SIMF mapping rules may then perform the appropriate unit conversions.

The foundation of information specification in SIMF at all levels is the type system. Types specified for all properties and relations involving values must match the types of the related values. The concepts of units and values as defined in "VIM" [JCGM 200-2008] is used as the basis for defining the types used in SIMF to guarantee type safety of quantities across different representations. Since many existing models and schema do not include well defined units some effort may be required to find and then specify the implicit units based on documentation, SME interviews or inspection of data or source code. It is recommended that the units used by external models and schema be determined prior to attempting federation and integration of information based on those models or schema.

**VIM [JCGM 200-2008] concepts of quantities and units**

* **quantity:** property of a phenomenon, body, or substance, where the property has a magnitude that can be expressed as a number and a reference [ed. to a unit]
* **kind of quantity (quantity kind):** aspect common to mutually comparable quantities
* **measurement unit (unit):** real scalar quantity, defined and adopted by convention, with which any other quantity of the same kind can be compared to express the ratio of the two quantities as a number

**SIMF concepts of quantities and units**

SIMF uses the VIM concepts to define "quantity values" and types to capture the quantity kind and unit. The goals for this type based approach are:

* That it is clearly grounded in semantics as defined in VIM
* That a type may be used to specify the range of a property or relation involving unit based values.
* That a quantity value (e.g. 5 grams) be representable as a simple number with a type.
* That there is a clear type hierarchy starting with a representationally independent type in a conceptual model (e.g. mass) that can be further specialized to a specific unit in a logical model (e.g. grams) and further specialized to be represented by a physical data type (e.g. “double”).
* That external models and schema may have unit specifications asserted without changing the schema.
* That a quantity of an entity be able to be referenced without a specific quantity value being known (e.g. John’s weight).
* That systems of units such as [ISO-80000] or [OMG QUDV] (A part of SysML) be able to be directly referenced as the definition of a unit.

SIMF defines three types to realize the above goals: Quantity Kind, Unit Type, Base Unit Type. SIMF also defines Quantity Values, which are instances of unit types.

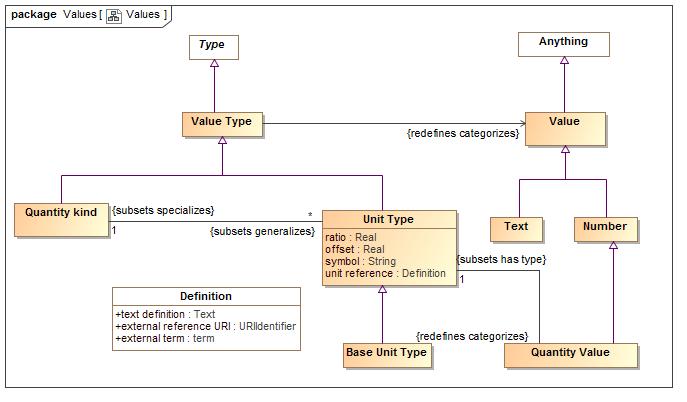
In VIM a quantity has a magnitude that is expressed as a number and a reference. The SIMF quantity value is the numeric value of such a quantity where the reference is specified by the “unit reference” property of the quantity value’s type. The quantity value’s type is a “Unit Type”. The Unit type has attributes for converting a unit to a base unit, a symbol and a unit reference. Based on VIM the unit reference may be “a measurement unit, a measurement procedure, a reference material, or a combination of such” and is specified with a description that contains reference information. In summary, the reference of a SIMF quantity value is determined indirectly through its unit type. A quantity value has exactly one unit type and exactly one Quantity Kind. A quantity value expressed in any unit of the same quantity kind may be converted to any other unit of the same quantity kind.

This type-based sapproach allows specification of a property at the conceptual (quantity kind) logical (unit type) or physical (unit type with a numeric type) levels. Such specifications use the same type-based approach used for other aspects of the models. Given this information a SIMF implementation may correctly and reliably convert between compatible types regardless of representation. Please see the specification of the value types, attributes and relationships for more detail.

**Example:**

* A specification for a road segment has a property “Speed limit”.
* The type of this property in a reference conceptual model is “Speed:Quantity Kind”.
* A unit “Kilometer per Hour:Unit Type” is defined as a subtype of “Speed:Quantity Kind” with a “unit reference” of “[ISO-80000.4] Kilometer per Hour”. Note that quantity kinds and unit types would normally be defined in reference models that correspond to a “system of units”.
* Miles per hour is also defined as a subtype of Speed.
* A physical schema defines “Speed-KPH: Integer”.
* A SIMF mapping rule maps “Speed limit” to “Speed-KPH” and asserts a type of “Kilometer per Hour” on the “Speed-KPH” end.
* A data file described by the schema defines a road “Route One” with a Speed-KPH of 100.
* When converted to a U.S. application this speed limit of route one can be viewed as 62 MPH.

## Values



1. Values

### Class Base Unit Type

One unit type of a quantity kind may be marked as the base unit. The base unit provides the basis for conversions between units of the same quantity kind. The base unit always has a ratio of one and an offset of zero.

#### Direct Known Superclasses

[Unit Type](#_9a97d5f73bf658c81147f5fab194bf88)

package SIMF Meta Model::Values

public class Base Unit Type

extends

Unit Type

### Class Number

The concept of a number that may be expressed in any way.

#### Direct Known Superclasses

[Value](#_a739673c8d53da123e392b7e5059ceec)

package SIMF Meta Model::Values

public class Number

extends

Value

### Class Quantity kind

[JCGM 200:2008] A Quantity Kind is an aspect common to mutually comparable quantities represented by one or more units. Units with a common quantity kind may be algorithmically converted to any other unit of that quantity kind. e.g. temperature.  
  
Quantity kinds are a supertype of unit types which are then a type of all quantity values, Quantity values are mutually comparable with all other quantity values categorized by the same quantity kind.

#### Direct Known Superclasses

[Value Type](#_b08132d9b30f1d47632a28aa6e4894bf)

package SIMF Meta Model::Values

public class Quantity kind

extends

Value Type

#### Associations

-726234319.jpg : [Unit Type](#_9a97d5f73bf658c81147f5fab194bf88) [\*]

A Unit type is a type of a quantity value referencing a specific unit and a required type of a property representing a quantity.   
  
Each quantity value has a reference as defined by the "unit reference" property of the quantity value's type.  
  
[JCGM 200:2008] A Unit is a real scalar quantity, defined and adopted by convention, with which any other quantity of the same quantity kind can be compared to express the ratio of the two quantities as a number. e.g. Degrees Centigrade, Miles.  
  
Each unit type represents refinement of a quantity kind using generalization and is thus substitutable for that quantity kind. Typically quantity kinds are used in conceptual models and unit types in physical or logical models.  
  
Note that unit types are not units, but the type of quantity values expressed in a common unit as defined in [JCGM 200:2008].

### Class Quantity Value

A quantity value is a numeric magnitude with a unit type that may be used as the value of a quantity property as defined by [JCGM 200:2008]. The reference of the quantity is defined by the "unit reference" property of the Unit Type.  
  
Each quantity value has exactly one Quantity Kind as a type.  
  
In a physical model a quantity value must have a type that specifies its unit (e.g. "Gram") and may have a data type specifying its numeric representation (e.g. "Double").  
  
[JCGM 200:2008] A quantity is a property of a phenomenon, body, or substance, where the property has a magnitude that can be expressed as a number and a reference.   
  
Note: A quantity as defined here is a scalar. However, a vector or a tensor, the components of which are quantities, is also considered to be a quantity.  
  
e.g. 5cm is an instance of the unit type "Centimeter"

#### Direct Known Superclasses

[Number](#_f5a86db7bd9636f0fa472c3859bc9c3c)

package SIMF Meta Model::Values

public class Quantity Value

extends

Number

#### Associations

-726234319.jpg : [Unit Type](#_9a97d5f73bf658c81147f5fab194bf88) [1]

A Unit type is a type of a quantity value referencing a specific unit and a required type of a property representing a quantity.   
  
Each quantity value has a reference as defined by the "unit reference" property of the quantity value's type.  
  
[JCGM 200:2008] A Unit is a real scalar quantity, defined and adopted by convention, with which any other quantity of the same quantity kind can be compared to express the ratio of the two quantities as a number. e.g. Degrees Centigrade, Miles.  
  
Each unit type represents refinement of a quantity kind using generalization and is thus substitutable for that quantity kind. Typically quantity kinds are used in conceptual models and unit types in physical or logical models.  
  
Note that unit types are not units, but the type of quantity values expressed in a common unit as defined in [JCGM 200:2008].

### Class Unit Type

A Unit type is a type of a quantity value referencing a specific unit and a required type of a property representing a quantity.   
  
Each quantity value has a reference as defined by the "unit reference" property of the quantity value's type.  
  
[JCGM 200:2008] A Unit is a real scalar quantity, defined and adopted by convention, with which any other quantity of the same quantity kind can be compared to express the ratio of the two quantities as a number. e.g. Degrees Centigrade, Miles.  
  
Each unit type represents refinement of a quantity kind using generalization and is thus substitutable for that quantity kind. Typically quantity kinds are used in conceptual models and unit types in physical or logical models.  
  
Note that unit types are not units, but the type of quantity values expressed in a common unit as defined in [JCGM 200:2008].

#### Direct Known Superclasses

[Value Type](#_b08132d9b30f1d47632a28aa6e4894bf)

package SIMF Meta Model::Values

public class Unit Type

extends

Value Type

#### Associations

-726234319.jpg : [Quantity kind](#_ca1c56b440439615024c837658185d15) [1]

[JCGM 200:2008] A Quantity Kind is an aspect common to mutually comparable quantities represented by one or more units. Units with a common quantity kind may be algorithmically converted to any other unit of that quantity kind. e.g. temperature.  
  
Quantity kinds are a supertype of unit types which are then a type of all quantity values, Quantity values are mutually comparable with all other quantity values categorized by the same quantity kind.

-726234319.jpg : [Quantity Value](#_746074a5bfbd6e26906da5d4bd0d2a7f)

A quantity value is a numeric magnitude with a unit type that may be used as the value of a quantity property as defined by [JCGM 200:2008]. The reference of the quantity is defined by the "unit reference" property of the Unit Type.  
  
Each quantity value has exactly one Quantity Kind as a type.  
  
In a physical model a quantity value must have a type that specifies its unit (e.g. "Gram") and may have a data type specifying its numeric representation (e.g. "Double").  
  
[JCGM 200:2008] A quantity is a property of a phenomenon, body, or substance, where the property has a magnitude that can be expressed as a number and a reference.   
  
Note: A quantity as defined here is a scalar. However, a vector or a tensor, the components of which are quantities, is also considered to be a quantity.  
  
e.g. 5cm is an instance of the unity type "Centimeter"

### Class Value

A value is an atomic piece of information without a specific lifetime or identity independent of the value. Values include numbers, strings and other atomic "primitive" data.

#### Direct Known Superclasses

[Anything](#_a52cb0ff6e414b3170b58afe10b6afcb)

package SIMF Meta Model::Top level

public class Value

extends

Anything

### Class Value Type

A type categorizing values where a value is an atomic piece of information without a specific lifetime or identity independent of that value. Values include numbers, strings and other atomic "primitive" data.

#### Direct Known Superclasses

[Type](#_dfe1514224ca21cedba7b2b29802db50)

package SIMF Meta Model::Values

public class Value Type

extends

Type

#### Associations

418297915.jpg : [Value](#_a739673c8d53da123e392b7e5059ceec)

A value is an atomic piece of information without a specific lifetime or identity independent of the value. Values include numbers, strings and other atomic "primitive" data.