



Semantic Modeling for Information Federation

Cory Casanave

(cory-c at modeldriven dot com)

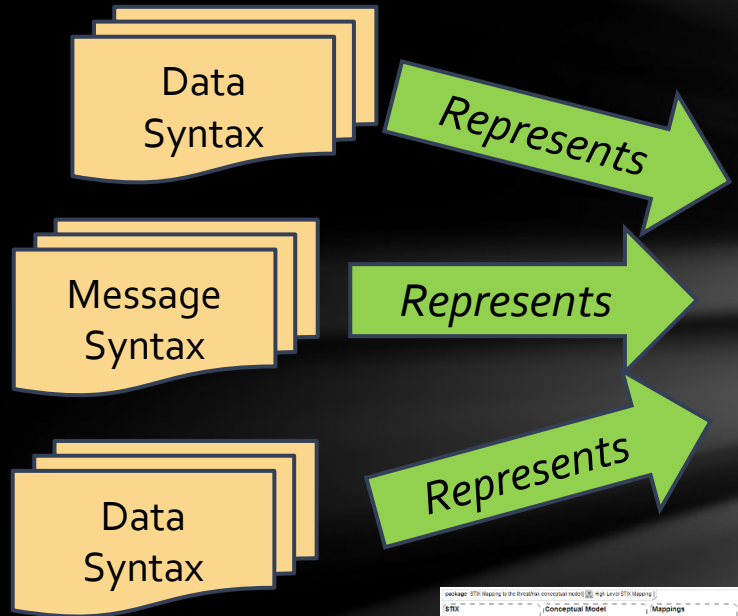
Conceptual Modeling Tools



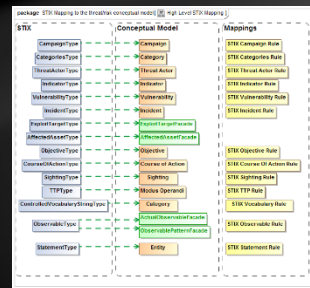
Semantic Mediation

We need to understand

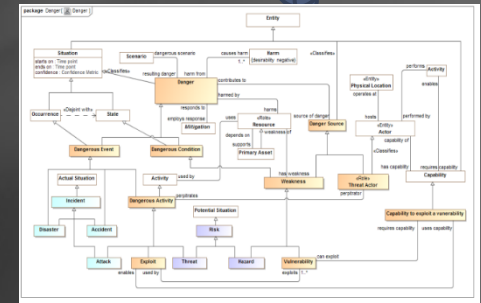
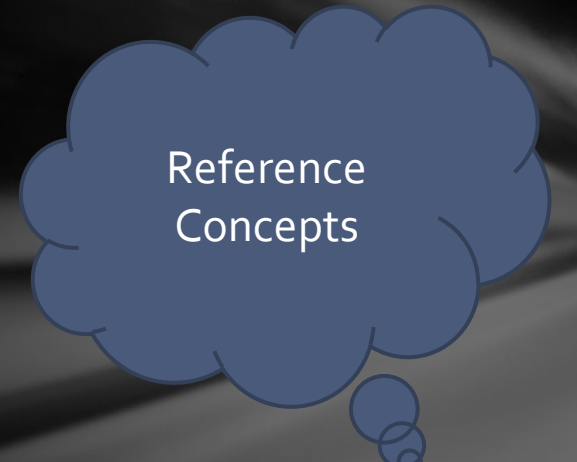
- What are the common concepts
- How do the various information syntaxes and systems represent those concepts
- What are the rules for translating between them in various context
- How information and data are governed



Data & Message Model



Mapping Rules



Conceptual Reference Model

Kinds of models

Conceptual reference models

- A.K.A. “Concept Models” or “Business Information Models”
- Model a subject area, such as a kind of business
- The subject of the model is business concepts

Data and Message Models

- Model data structures that store or transmit facts
- May be optimized for an application or purpose
- The subject of the model is data structures

Mapping Models

- Model how data structures represent business concepts
- May be elaborated to support automated translation, integration or federation of data
- The subject of the model is mapping

Important conceptual modeling and mapping capabilities

As easy as possible for domain stakeholders to understand the models as diagrams, tables or text

“First class” n-ary relationships that may have properties and participate in other relationships

Recognition and representation of time, provenance and context – most facts are only true for a limited time and in specific situations

Roles, phases and other “non rigid” classifications that also may be time or situationally dependent

Hierarchies of types, relationship types and properties

Business values as represented by various systems of units

Important features of SMIF concept modeling

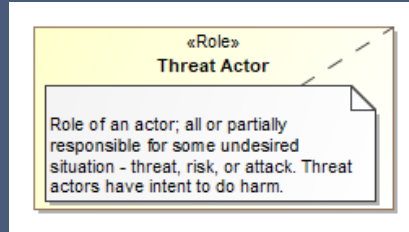
*Using the SMIF UML profile – based
on the UML (Unified Modeling
Language) standard*

Foundational Conceptual Tools (Meta-Concepts)

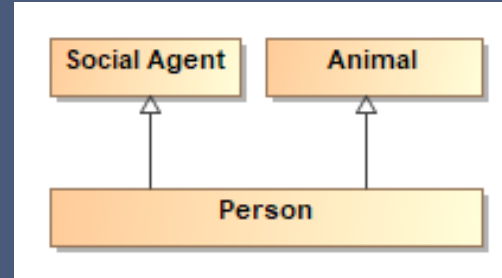
Entity Types



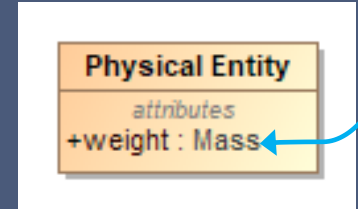
Definitions



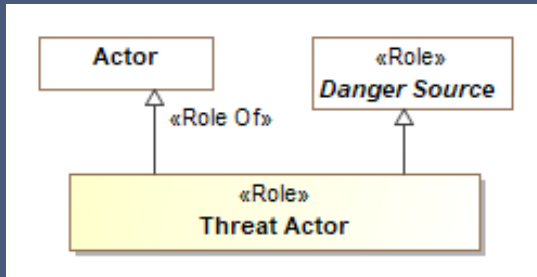
Specialization



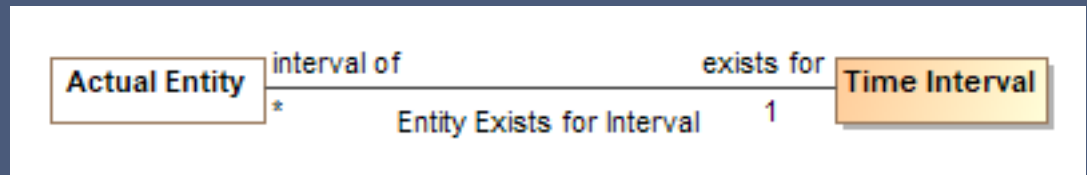
Characteristics



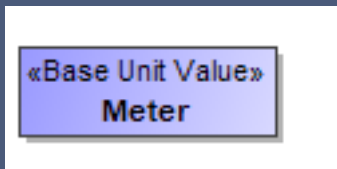
Roles



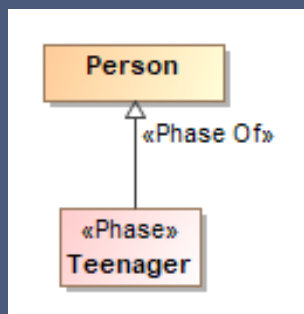
Associations (Always true between entities)



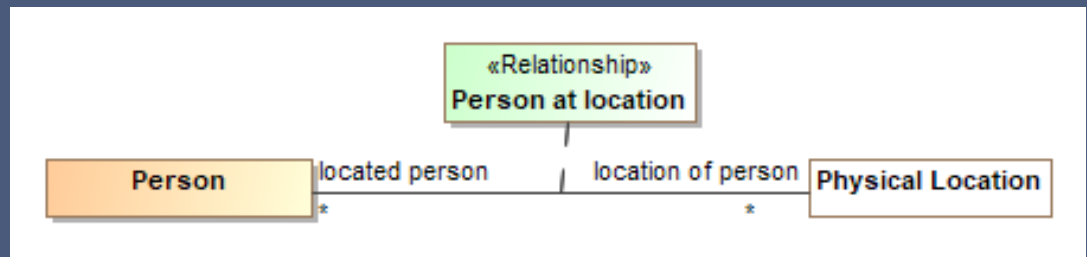
Values & Units



Phases



Relationships (Independent/Contextual)



Subject Areas - Organizing Concepts

Subject Areas provide context for the definition and management of a set of concepts

Subject areas may also be used for governance

Subject areas may use or depend on other subject areas

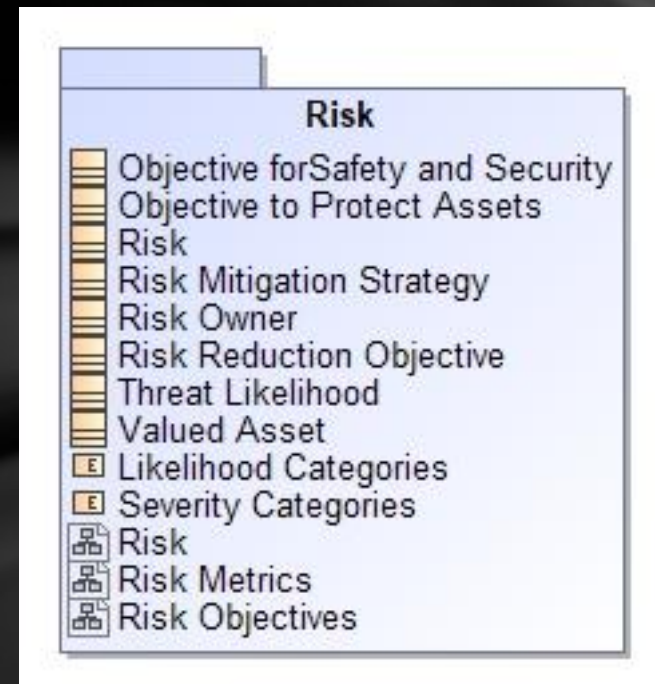
Subject areas may contain other subject areas

Subject areas may be business domain focused or technology focused

- Don't mix-up what is being modeled, the business, data, or applications

Models typically define and/or use multiple subject areas

Subject Areas



Things and their types

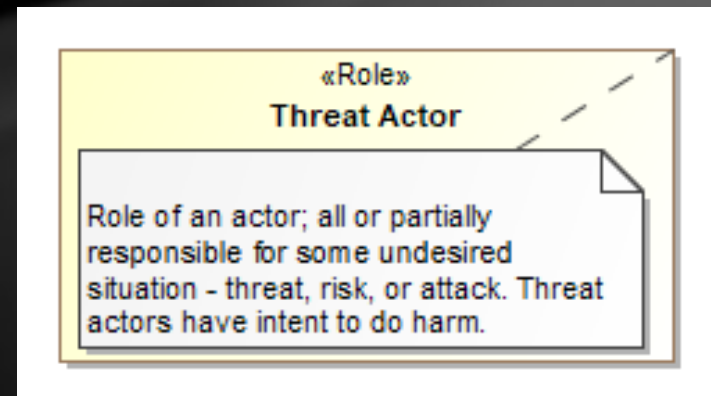
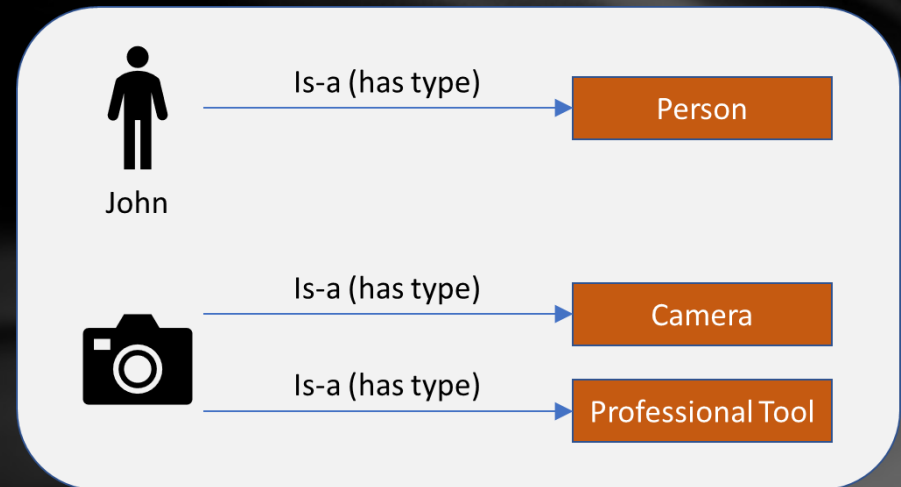
Everything in a model has more or more types

Types define a set of things that qualify as that type

A synonym for type is “Class” or “Classification”. Class is typically used in a more technical context.

Every type has a business focused definition of what it is or means

- Not how it is used for a particular application.
- Not how it is represented as data.



Specialization

Types may specialize other types

Most types specialize at least one other type and may specialize more than one

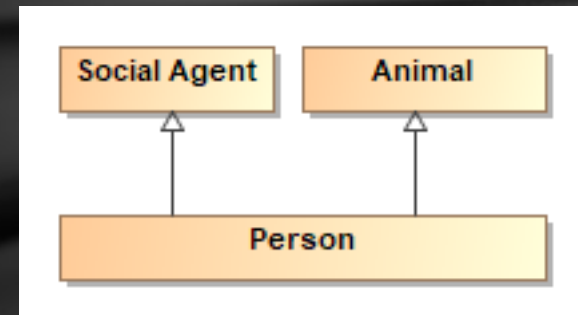
A specialized type is known as the “subtype”

A more general type is known as the “supertype”

Everything that is a subtype is implicitly also all of the supertypes

- E.g. all people are also animals and social agents

The inverse term “Generalization” is also commonly used



Roles

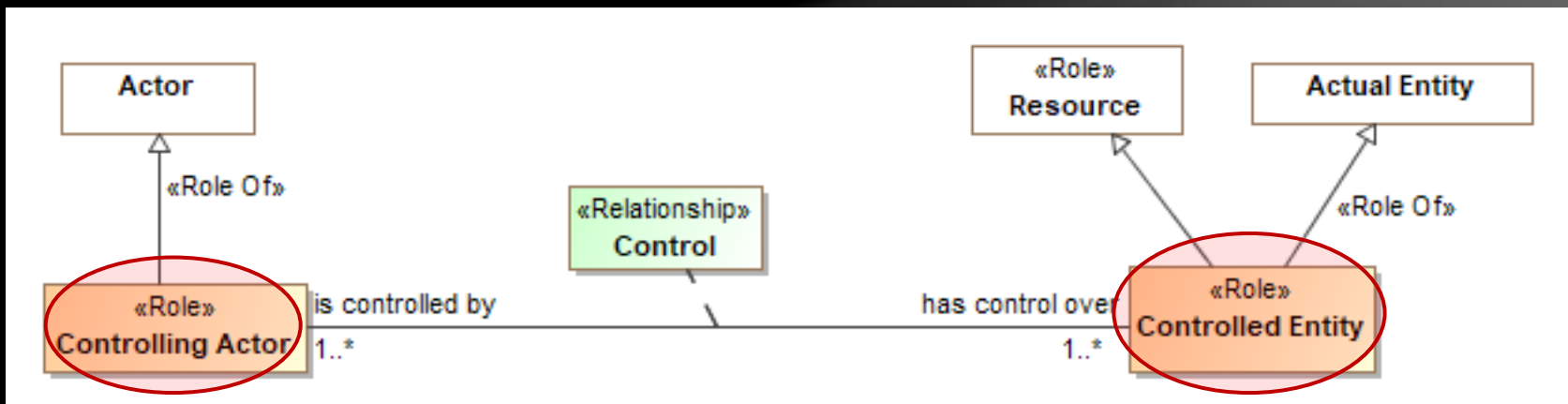
Roles define what something is for or how it behaves in a certain context, not “what it is”. A “relative thing”

A role is a <<Facet Of>> what it can be a role of.

An entity can play any number of roles and these roles may change over time.

Roles can be contextual and specialize other roles

Roles are usually established by relationships

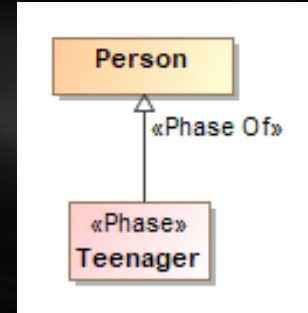


Phases

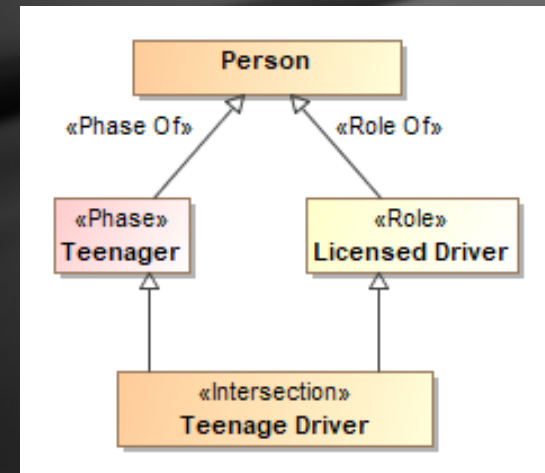
Phases (or states) are classifications of objects over their lifetime.

- Examples: Child, Teenager, Adult or Invoiced, Late, Paid

May be combined with other types using unions and intersection (e.g. teenage driver)



Teenager is a phase of a person



All teenagers who are licensed drivers are teenage drivers

Relationships

Relationships are meaningful atomic concepts involving a set of other entities or relationships. A “Mediating thing”.

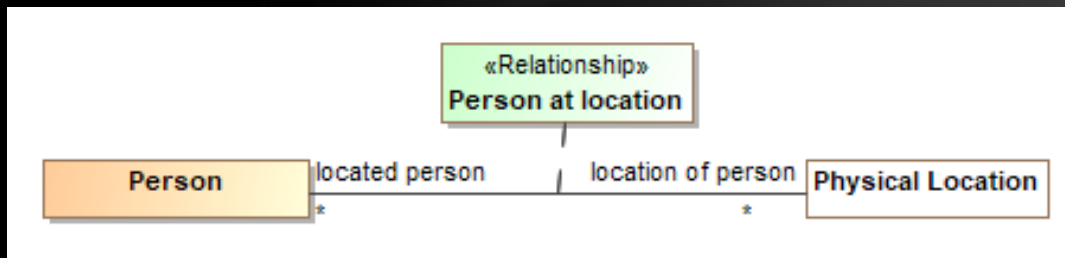
There can be any number of related “ends”, but two ends is most common

Relationships are atomic & static “Situations”, the involved ends do not change over the lifetime of the relationship. The context and “truth value” of the relationship may change.

Relationships are temporal – exist for a timeframe. The timeframe of a relationship may or may not match the timeframe of the “ends”

Relationships may be involved in other relationships and may have characteristics

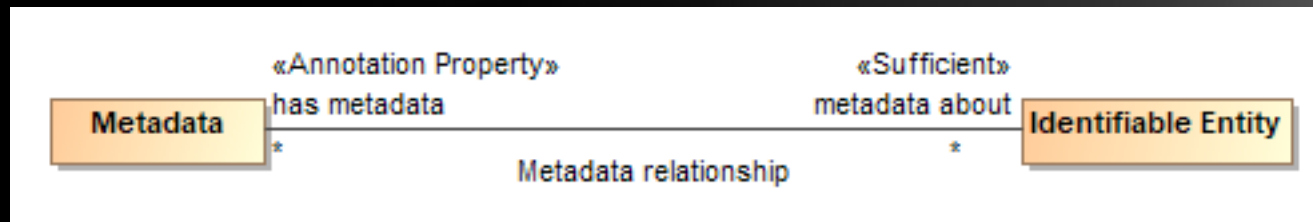
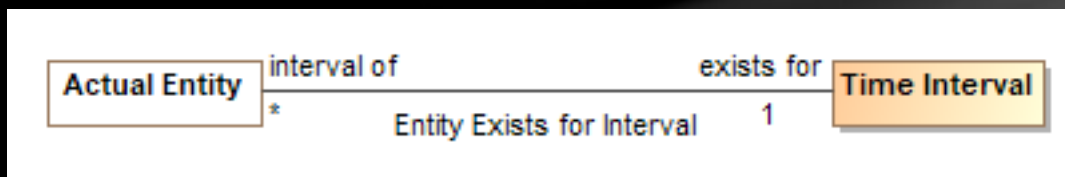
We refer to these independent relationships as “first class” relationships.



Associations

Associations are similar to relationships but:

- Not temporal – they exist for the lifetime of the related “ends”
- Do not have characteristics
- Limited to binary
- Also known as “Formal Relations”
- Map most directly to simple properties (e.g. OWL or Java)



Characteristics / Attributes

Characteristics (attributes) are features inherent in another type

- Most like “properties” in many languages

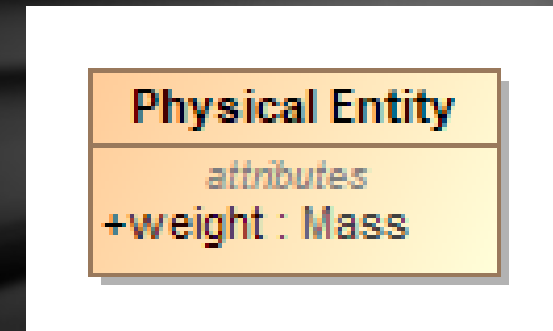
Usually have a value type as their range

- John has a weight of 60 KG

Characteristics are temporal (have a timeframe) & identifiable

- John weighed 60kg on March 3rd, 2011

In conceptual reference models, quantity kinds (e.g. Mass) are preferred over specific units (e.g. kg) or data types (e.g. “Real”) as the types of characteristics



Specializing properties

Properties encompass the “ends” of associations and relationships, and characteristics

Like everything else – properties have types (property types) and instances (called bindings)

Properties can be specialized and restricted

Models typically define property types

Specialized properties may be “virtually derived”, not define a new concept but restrict an existing one. These have no name or the same name as their super-property

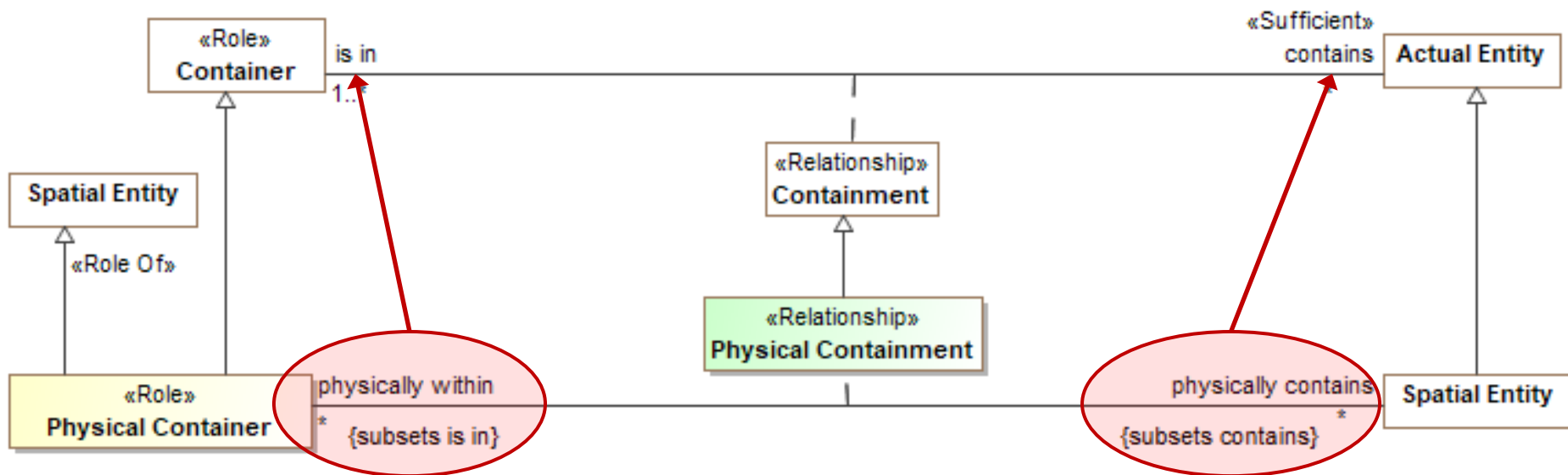
Property specialization is either a “subset” or redefines (equivalent set)

Subset properties

Subset properties define subtypes of other property types (ends of associations or relationships, characteristics, etc).

Extent of subset property is a subset of super-property

May tighten constraints – multiplicity, type, etc.

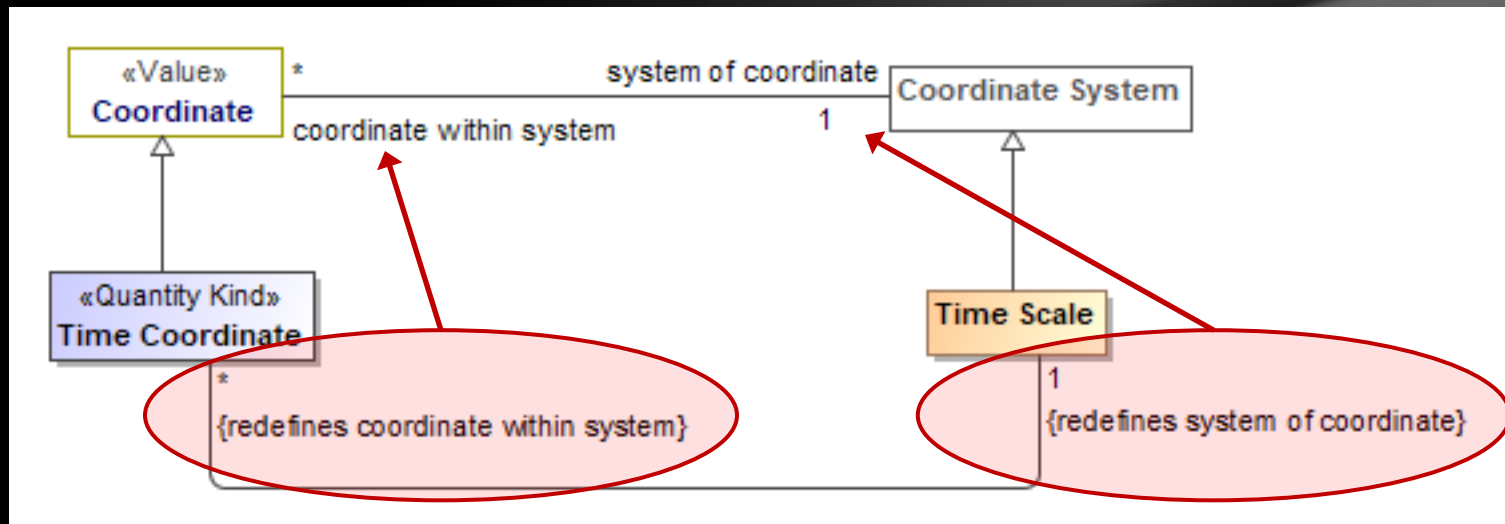


Redefined Properties

Redefined properties define subtypes of other property types (ends of associations or relationships, characteristics, etc) and **replace the super-property in the given context**.

Extent of redefined property is the same as the extent of super-property

May tighten constraints – multiplicity, type, etc.



Restriction Example

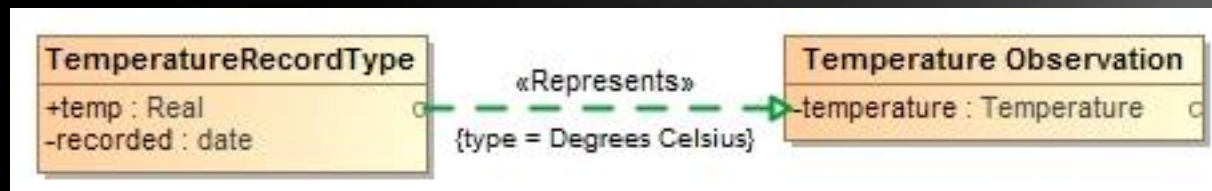
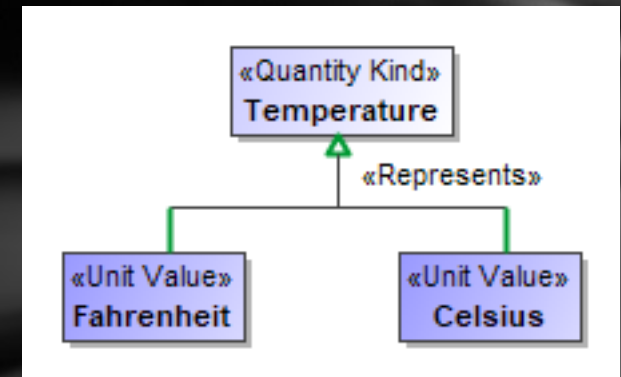
Quantity Kinds & Units

For numeric characteristics, we want to know what it means (e.g. Temperature), not the kind of number (Real).

<<Quantity Kind>> is an aspect common to mutually comparable quantities represented by one or more units.

A “unit value” represents a quantity kind, there are multiple units representing temperature.

A physical representation would then represent the unit as some kind of number in a specified unit.



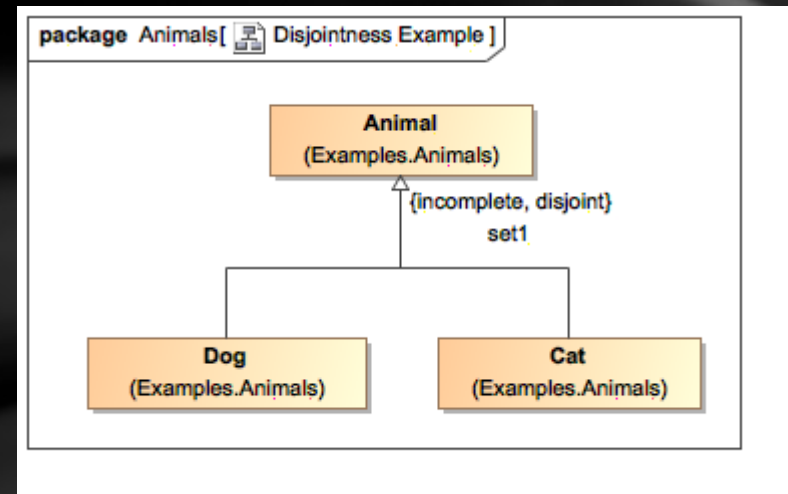
Incomplete and Disjoint

Subtypes are “complete” if there can be no more subtypes of the supertype

- Subtypes of “Animal” are incomplete – there are other kinds of animals

Subtypes are “disjoint” if they can’t be mixed

- E.g. the same thing can’t be a cat and a dog



Note: Disjoint may also use a dependency

Type Based Model Theory

- Every type represents some kind of thing that exists in our conceived world
- Reference models are conceptual, information models are application specific
- Multiple inheritance & multiple (instance) classification is supported
- Everything has metadata – source (including derivations), provenance & timeframe
- Things change – time matters (the world is not static or a snapshot)
 - E.G. Relationships, situations and characteristics are temporal
- Information is contextual (the world is not first order)
- Statements are not “deleted”, they go in and out of context
- Facts about the world is open, conclusions (and computations) are contextual
- Trust in information varies, not everything that can be inferred should be
- Independence from representation (e.g. schema) and processing technologies (including inference engines)
 - But, we can bind to any of these technologies