

1. Overview of UML Syntax Used For The Models

The models for the Topic Map standard uses a very restricted subset of the UML language. The subset used enables the definition of "static data models", that is, data models that simply define data types and the possible relations between them. A type is simply a named collection of attributes. (Note that the terms "attribute" and "property" are used interchangeably in static data models.) The model does not define any methods or actions. [Figure 1](#) shows a simple model that contains all of the graphical elements used in the Topic Map models.

Types in UML are organized into packages. Each package establishes a distinct name space of types. Types from one package can be used by reference from other packages. When types are used from other packages, the names of the used types are qualified in the diagrams with the package names.

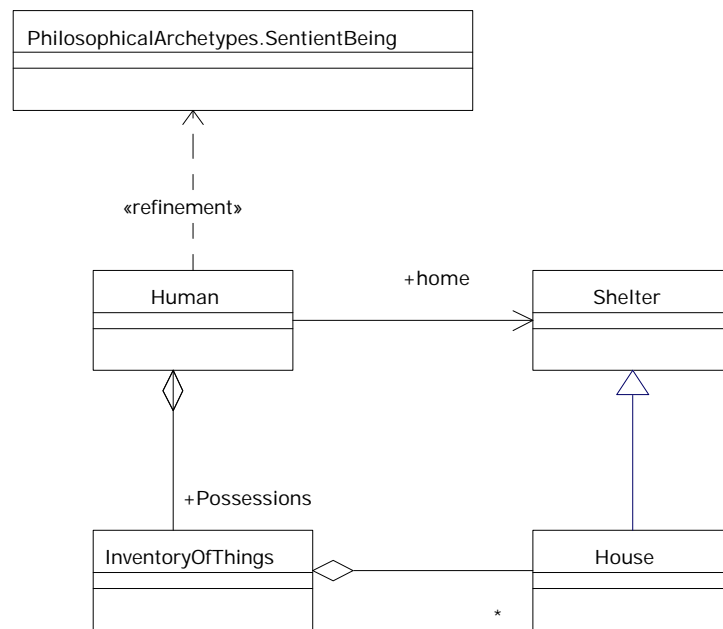


Figure 1 — Sample UML Model

Starting in the middle of the model, the box labeled "Human" represents the type "Human". The line to the right establishes a relation between the type "Human" and the type "Shelter". The relation is labeled "home", indicating that the type "Human" has an attribute (or property) named "home" whose value is a single Shelter object. The arrow on the line indicates "navigability", meaning that the Human knows about the Shelter but the Shelter doesn't know about the Human. If there are no arrows, then the relationship can be navigated in both directions.

The Human type has a "black-diamond" relation to the type "InventoryOfThings". The black diamond represents strict containment or ownership. For the purposes of the Topic Map models, containment means that the contained type has no meaningful existence

outside the context of the thing that contains it. In this simple model, it means that one cannot talk about InventoryOfThing objects without talking about the human that owns the inventory.

The InventoryOfThings type has an "open diamond" relation to the type "House". The open diamond indicates "aggregation", rather than containment. Aggregation means that the types aggregated may have a primary location somewhere else. The "*" on the House end of the relation indicates that zero or more House objects may be aggregated. Because this relation has no explicit label, it is implicitly labeled with the name of the related type, so this relation can be read as "InventoryOfThings" has an aggregation attribute "houses" (the plural is implied by the "*").

The House type has a "subtype of" relation to the type "Shelter" (this is also called "generalization" in UML). A subtype of a type inherits all of its attributes. A type may have multiple supertypes. The result of having multiple supertypes is that the type is the union of the attributes provided by all of its supertypes.

Returning to the Human type, the dashed arrow indicates a dependency relationship, in this case to a type from another package. The label "<<refinement>>" is a "stereotype", which indicates the specific type of dependency the arrow represents. The notion of "refinement" means that the type Human is at a lower level of abstraction than the refined type, "SentientBeing". Thus, the package "PhilosophicalArchetypes" represents a set of higher-level abstractions. This idea of refinement from higher-level abstraction to lower-level abstraction is roughly analogous to SGML Architectures, where the architectural forms are higher-level abstractions and the element types derived from those forms are lower-level refinements of the architectural forms. Refinement is different from subclassing in that for subclass/superclass relationships, both types are in the same package at the same level of abstraction, whereas for refinement relationships, the two types are always at different levels of abstraction. In short, refinement establishes a semantic correspondence without necessarily requiring a syntactic or structural correspondence. For example, a single abstract type might be refined using multiple concrete types. In SGML terms, you can think of the SGML representation of a topic map as an implementation-level refinement of the abstract Topic Map model.