# The histopathology markup language (HistoML)

## Language Specification for Level 1


| | |
|---|---|
| Chen Li | *Xi'an Jiaotong University, China* |
| Peiliang Lou | *Xi'an Jiaotong University, China* |
| Chunbao Wang | *The First Affiliated Hospital of XJTU, China* |
| Ruifeng Guo | *Mayo Clinic, USA* |
| Lixia Yao | *Temple University, USA* |
| Guanjun Zhang | *The First Affiliated Hospital of XJTU, China* |
| Jun Yang | *The Second Affiliated Hospital of XJTU, China* |
| Yong Yuan | *Shaanxi Provincial Tumor Hospital, China* |


05 December 2021

# Contents


1 Introduction ..... 3

1.1 Document Convention ..... 3

1.2 UML notation ..... 3

1.2.1 HistoML Class Diagram ..... 3

1.2.2 HistoML Object Property ..... 4

1.2.3 Inheritance ..... 4

2 HistoML Ontology Class Structure ..... 6

2.1 Top level entity class ..... 7

2.2 Entity ..... 8

2.3 PhysicalEntity ..... 9

2.4 Phenotype ..... 11

2.5 Diagnosis ..... 12

2.6 Quantitative_Metric ..... 13

2.7 PhysicalEntity Subclass ..... 14

2.7.1 Tumor ..... 15

2.7.2 Parenchyma ..... 16

2.7.3 Stroma ..... 20

2.7.4 NormalEntity ..... 24

2.7.5 Substance ..... 27

2.8 Phenotype Subclass ..... 28

2.8.1 Cellular_Appearances ..... 28

2.8.2 Architectural_Pattern ..... 30

2.8.3 Product_or_Reserve ..... 30

2.9 Diagnosis Subclass ..... 31

2.10 Utility ..... 32

2.10.1 EntityAttribute ..... 33

2.10.2 EntityReference ..... 34

2.10.3 Evidence ..... 39

2.10.4 Provenance ..... 40

2.10.5 Quantification ..... 41
2.10.6 Relationship ..... 42
2.10.7 Xref ..... 43
2.10.8 ControlledVocabulary ..... 46
2.10.9 DiagnosisProcess ..... 47
2.10.10 DiagnosisStep ..... 47
2.11 Data ..... 48
2.11.1 Slide ..... 48
2.11.2 PathologyReport ..... 49

# 1 Introduction

This document defines the HistoML level 1 ontology. It provides definitions of the HistoML ontology classes, object properties and data properties. As for how to use HistoML, we provide some example representations and best practice recommendations. HistoML level 1 OWL file could be downloaded from https://histoml.com/.

## 1.1 Document Convention

We use the following typographical conventions to distinguish classes, properties from other entities:

a) **`Class`**: Class names start with an uppercase character and are highlighted in bold in the text of this document;

b) *`property`*: Property names start with a lowercase character. *`Object properties`* are italicized while `datatype properties` are not;

c) The structure “object:ClassName” refers to an individual of a class, used to illustrate the range of an object property.

## 1.2 UML notation

In this document, we use UML, the Unified Modeling Language, as a basis for defining HistoML classes and properties.

### 1.2.1 HistoML Class Diagram

HistoML uses UML class notation to define HistoML ontology classes. Classes in UML class notation are drawn as simple tripartite boxes, as UML allows for operators as well as data attributes to be defined. But HistoML only uses data attributes (i.e.

datatype properties), so all HistoML class diagrams use only the top two portions of a UML class box as Figure 1-1 shows.

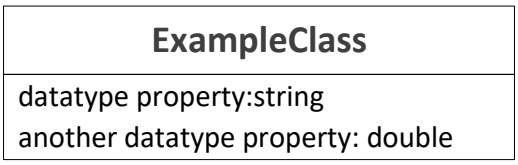


Figure 1-1 Example of a HistoML class diagram.

### 1.2.2 HistoML Object Property

HistoML uses UML Composition to represent a HistoML object property which links a HistoML class to another one. In HistoML, UML Composition represents different object properties in addition to the composition relationship. Figure 1-2 gives an example.


Figure 1-2 Example illustrating a HistoML object property

The line with the black diamond indicates a HistoML object property. If this property links Class1 to Class2, the diamond would locate on the Class1 side. The name of this property is on the line. Numbers are placed above the line near the Class2 side to indicate how many instances can be contained. The common cases in HistoML are the following: [0..*] signifies a list containing zero or more; [1..*] signifies a list containing at least one; and [0..1] signifies exactly zero or one. The absence of a numerical label means “exactly 1”.

### 1.2.3 Inheritance

HistoML classes can inherit properties from other classes, and inheritance in HistoML involves object and datatype properties from a parent class being inherited by

the child classes. Inheritance is indicated by a line between two classes, with an open triangle next to the parent class. Figure 1-3 gives an example.


Figure 1-3 Inheritance

# 2 HistoML Ontology Class Structure

In this section, we define each class of HistoML Level 1. Text definitions of the classes are provided along with the synonyms, comments and examples to help readers understand the definition and intended use of each class. The most specific class available should be used.

HistoML Level 1 ontology has three root classes which are **Entity**, **Utility** and **Data**. Figure 2-1 shows a high-level view of **Entity** and its subclasses, Figure 2-2 shows a high-level view of **Utility** and its subclasses and Figure 2-3 shows a high-level view of **Data**. HistoML classes are shown as boxes and the arrows represent subclass relationships.

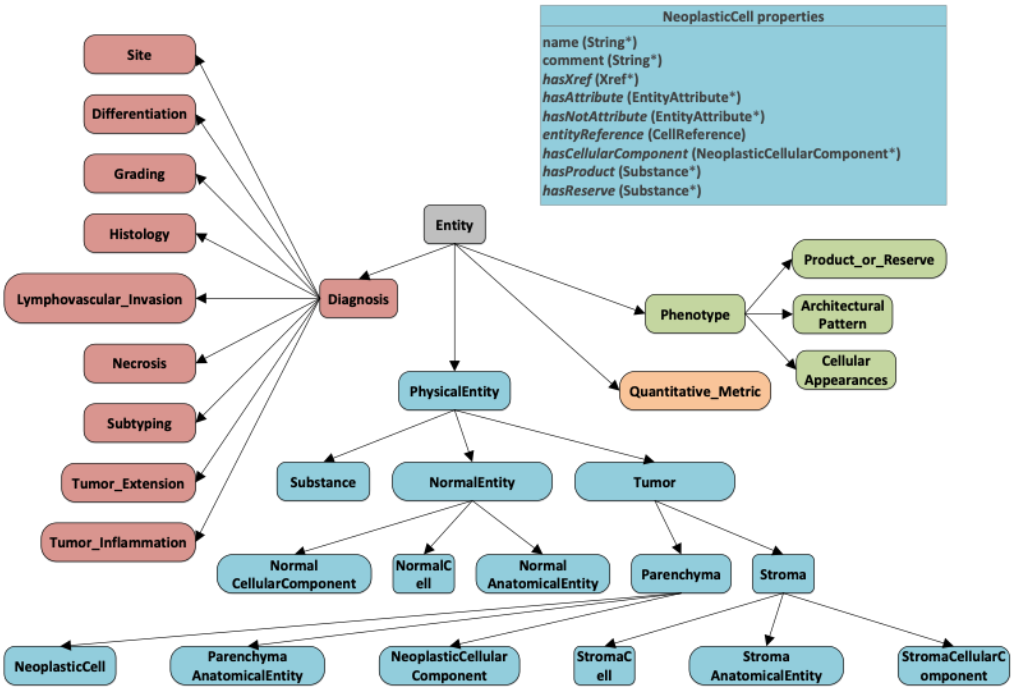


Figure 2-1 High-level view of **Entity** and its subclasses

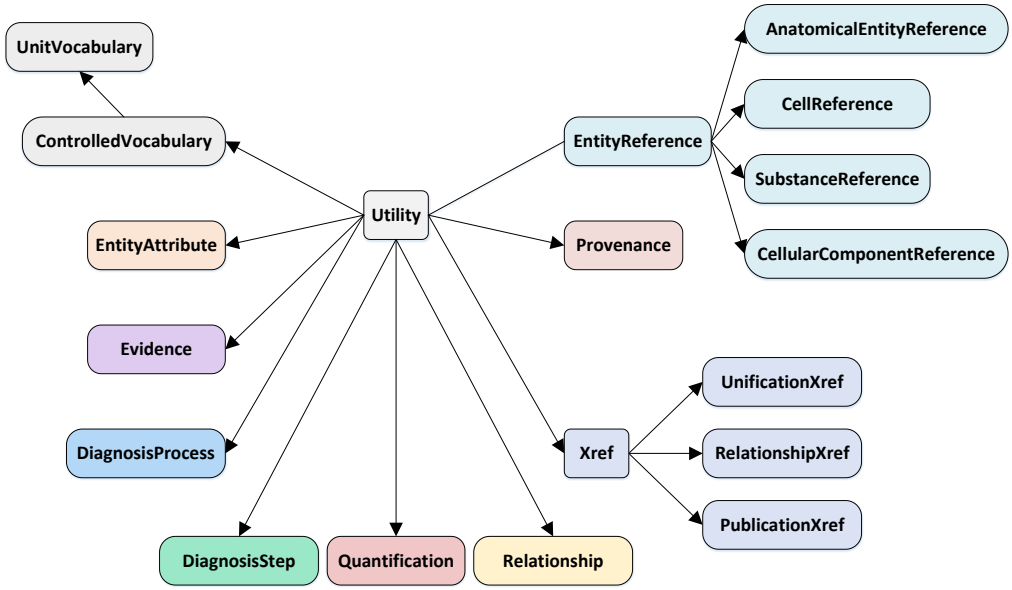


Figure 2-2 High-level view of **Utility** and its subclasses

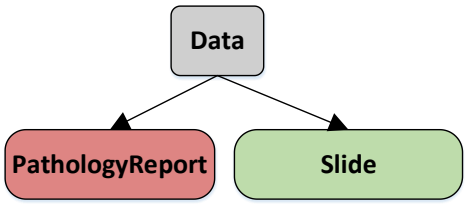


Figure 2-3 High-level view of **Data** and its subclasses

## 2.1 Top level entity class

The HistoML ontology defines five basic classes: the root level **Entity** class and its four subclasses: **Diagnosis**, **Phenotype** and **PhysicalEntity**.

A diagnosis could be regarded as a determination supported by a set of phenotypes such as grading, staging, and subtyping. A phenotype refers to a set of physical entities or one entity with some specific traits. Different types diagnoses are defined as children of **Diagnosis** class while different types of phenotypes are defined as children of the **Phenotype** class.

## 2.2 Entity

**Entity** is the root class of HistoML which includes **Diagnosis**, **Phenotype** and **PhysicalEntity.** Its definition is shown in Figure 2-4.

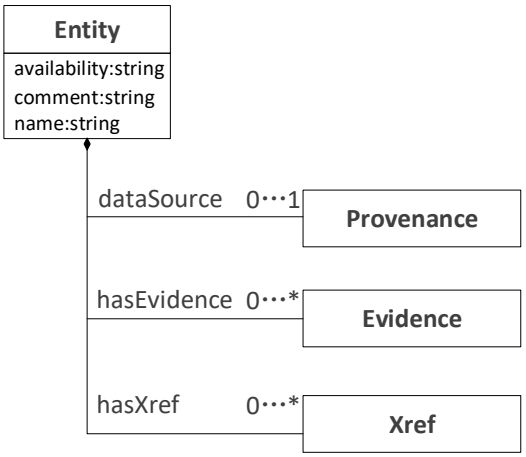


Figure 2-4

availability: (xsd:string) Describes the availability of this data (e.g. a copyright statement).

comment: (xsd:string) Comment on the data in the container class.

name: (xsd:string) One or more names of this entity. This will automatically include values of the displayName and standardName properties, as they are child properties of the name. displayName values are short names suitable for display in a graphic. standardName values are names that follow a standard nomenclature.

*datasource*: (0 or 1 object:Provenance) A description of the source of this data, e.g. a database or person name.

*hasEvidence*: (0 or more object:Evidence) Scientific evidence supporting the existence of the entity.

*hasXref*: (0 or more object:Xref) Values of this property define external cross-references from this entity to entities in external databases (e.g. controlled vocabulary).

## 2.3 PhysicalEntity

**PhysicalEntity** class is used to represent microscopically observable entities in a slide. According to their different types, **PhysicalEntity** has several subclasses covering cells, cellular components, substances (e.g. product or reserve of a cell), tissues and other anatomical structures (e.g. a cavity or duct). It's further categorized into ‘NormalEntity’ and ‘Tumor’ based on whether an entity is neoplastic or induced by tumor or not (e.g. blood vessels induced by tumor belong to ‘Tumor’ instead of ‘NormalEntity’).

To represent a phenotype, a physical entity in diverse states should be used. For example, a tumor cell is a generic physical entity while there are different forms of it such as a multinucleate giant tumor cell, a tumor cell having cytoplasm with a cyst, a tumor cell having evenly distributed chromatin. Under one slide, it is likely for you to observe more than one of those entities. To support representing physical entities in diverse states, a generic physical entity is represented using the **EntityReference** class which stores constant attributes of the entity while its different forms are represented using **PhysicalEntity** class which stores the variable features. The **EntityReference** is referenced from the **PhysicalEntity**. This design makes it easier to create different forms of an entity while not duplicating information common to all forms and explicitly linking all forms of an entity together through the shared **EntityReference**. Section 2.10.2 provides more information about **EntityReference** and principles for the use of this class. The definition of **PhysicalEntity** is shown in Figure 2-5.

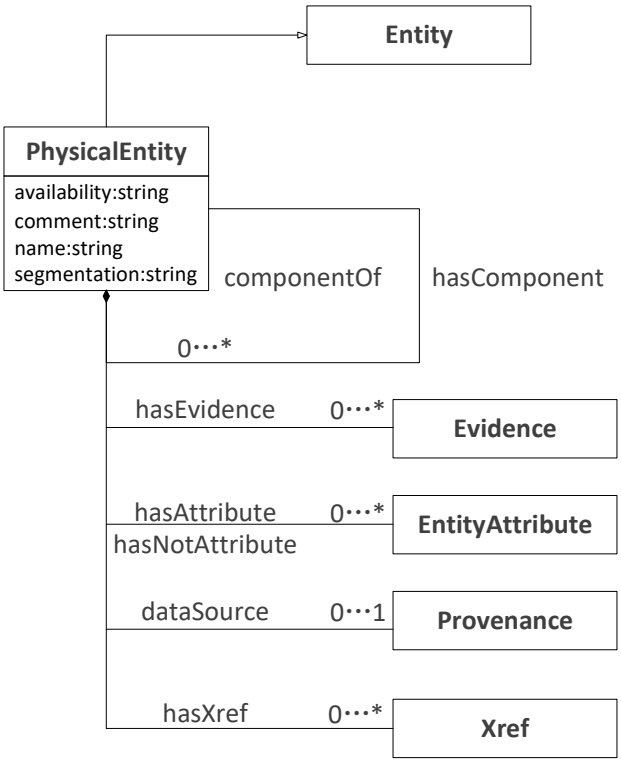


Figure 2-5

*hasComponent*: (0 or more object:PhysicalEntity) Used to define the components of a physical entity. Take a cell as an example, a cell consists of cytoplasm, membrane, nucleus, nucleolus and chromatin etc. *hasComponent* has three child properties which are *hasCell*, *hasCellularComponent* and *hasAnatomicalEntity*.

*componentOf*: (0 or more object:PhysicalEntity) Inverse property of *hasComponent*. Similarly, *componentOf* has three child properties which are *cellOf*, *cellularComponentOf* and *anatomicalEntityOf*. Since *componentOf* is inverse property of *hasComponent*, when representing an entity using HistoML, either *hasComponent* or *componentOf* is specified, the other could be inferred.

*hasAttribute*: (0 or more object:EntityAttribute) Attributes of the owner physical entity. For example, shape, size or other chemical attributes such as eosinophilic. A set of attributes helps define the state of an entity.

*hasNotAttribute*: (0 or more object: EntityAttribute) Attributes of this physical entity

which are known to be lacking. Attributes not specified are not known to be absent and only Attributes known to be lacking should be specified using this property.

`segmentation`: (xsd:string) The identifier of the segmentation or the annotation mask of this entity in the digital slide.

## 2.4 Phenotype

**`Phenotype`** class in HistoML is used to represent microscopic physical changes in slides that prompt pathologists to look more closely, typically the ones suggesting malignancy. **`Phenotype`** class has three subclasses, **`Cellular_Appearances`**, **`Product_or_Reserve`** and **`Architecturel_Pattern`**, covering all different levels of phenotypes in a slide. **`Cellular_Appearances`** and **`Product_or_Reserve`** are used to describe a physical change of a cell or a subcellular structure observed microscopically at high power. **`Architecturel_Pattern`** is used to describe histologic patterns of cell populations and tumor behaviors (e.g. extension, invasion) which are usually observed at medium and low power. In HistoML, a phenotype is represented in which each of the individual components is described, including their properties (e.g. size, length), relationships (e.g. membership) and behaviors (e.g. invasion). More information about these HistoML classes is given in Section 2.8. The definition of **`Phenotype`** is shown in Figure 2-6.

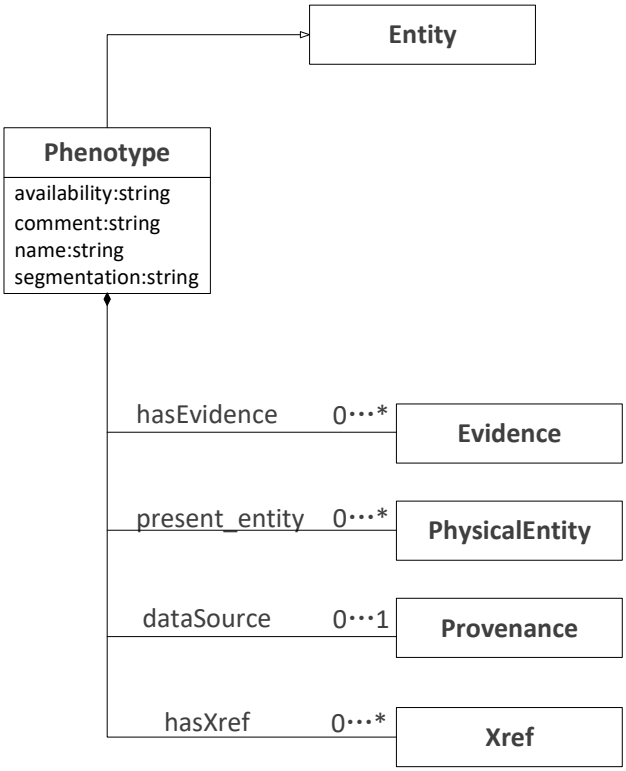


Figure 2-6

## 2.5 Diagnosis

**Diagnosis** class is designed to cover the diagnostic results included in a standard pathology report. It has several sub-classes including **Grading**, **Staging**, **Subtyping** etc. The definition of **Diagnosis** is shown in Figure 2-7.

*has_Morphologic_Evidence*: (0 or more object:Microscopic_Morphology) The set of supporting phenotypes of this diagnosis. Take ISUP grade 4 of clear cell renal cell carcinoma (ccRCC) as an example, the grading is supported by several phenotypes including nuclear pleomorphism, multinucleate giant cells, and/or rhabdoid and/or sarcomatoid differentiation.

*has_Diagnosis_Evidence*: (0 or more object:Diagnosis) A diagnosis could be supported by other previous diagnoses.

*has_Quantitative_Metric_Evidence*: (0 or more object:Quantitative_Metric) The

quantitative histomorphometry (QH) measurements of phenotypes (e.g. tumor-stroma ratio for measuring intra-tumoral stroma) which could be used as the supporting evidences of histopathological diagnoses

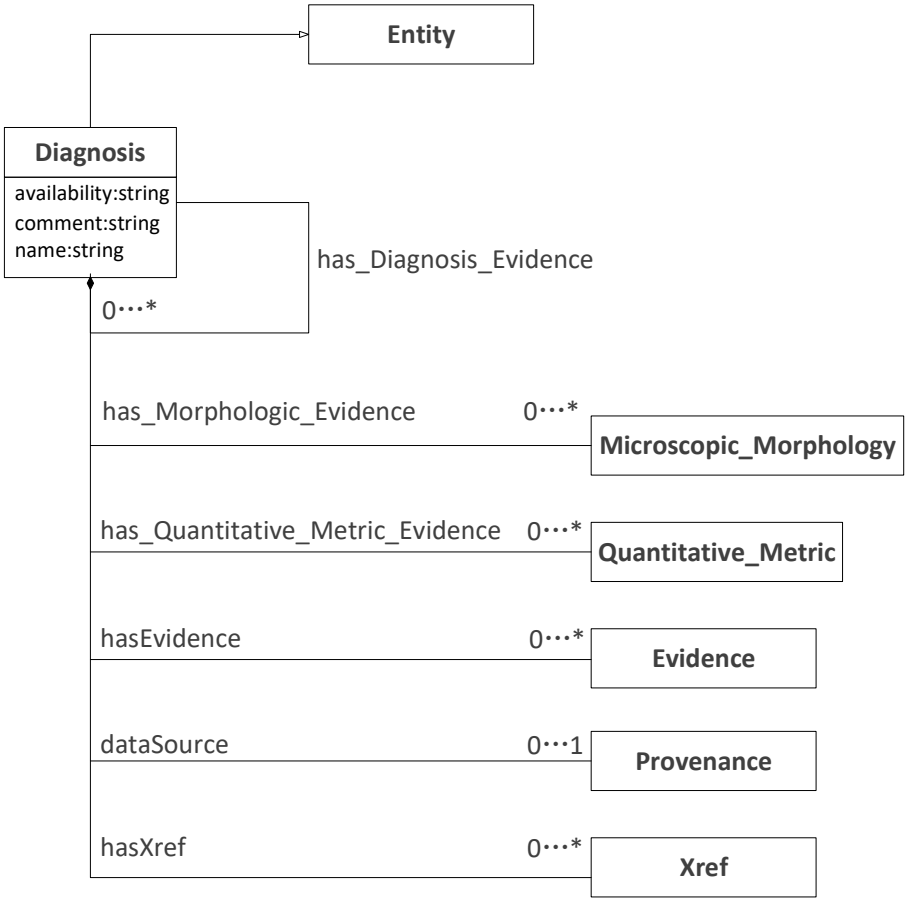


Figure 2-7

## 2.6 Quantitative_Metric

As quantitative histomorphometry (QH) measurements of phenotypes (e.g. tumor-stroma ratio for measuring intra-tumoral stroma) are gradually applied by many diagnostic criteria (e.g. the World Health Organization guideline) and used in real-world practice, HistoML represents these measurements using **`Quantitative_Metric`** and they could be used as the supporting evidences of histopathological diagnoses. The definition of **`Quantitative_Metric`** is shown in Figure 2-8.

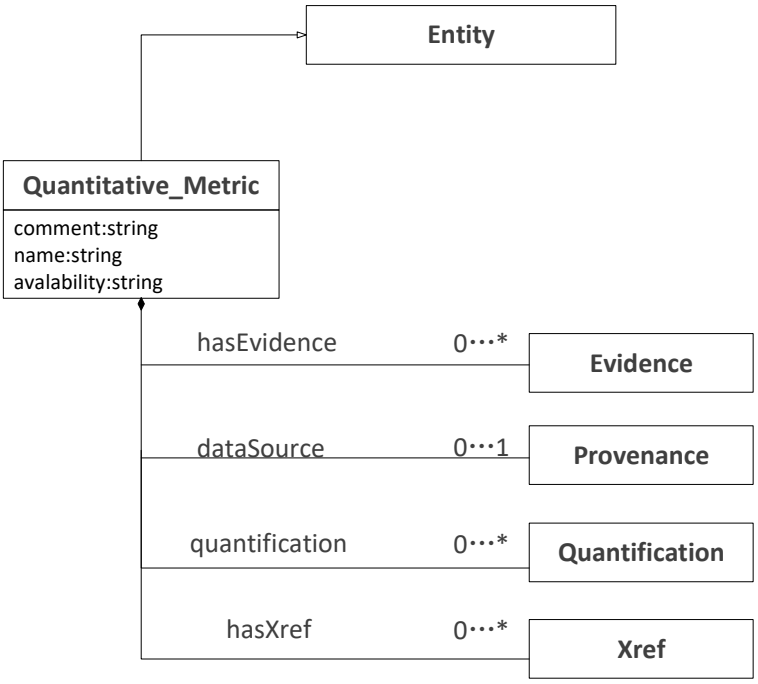


Figure 2-8

## 2.7 PhysicalEntity Subclass

The structure of **PhysicalEntity** is shown in Figure 2-9. It has three child classes which are **Tumor**, **NormalEntity** and **Substance**. The entities which compose a tumor are categorized as **Tumor** whereas the others are categorized as **NormalEntity**; any molecular entity is categorized as **Substance**. Based on the two compartments of a tumor, **Tumor** are further divided into **Parenchyma** and **Stroma**. Tissues and other anatomical structures (e.g. a cavity or duct) are described using **AnatomicalEntity** (e.g. **NormalAnatomicalEntity**). Correspondingly, **EntityReference** has four child classes which are **AnatomicalEntityReference**, **CellReference**, **CellularComponentReference**, and **SubstanceReference**. **Tumor**, **Parenchyma**, **Stroma** and **NormalEntity** don't have corresponding **EntityReference** classes.

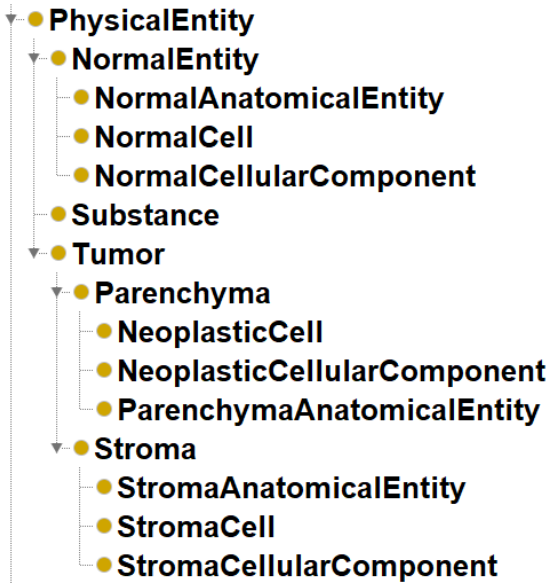


Figure 2-9

### 2.7.1 Tumor

**Tumor** refers to a benign or malignant pathologic structure in any part of the body, resulting from a neoplastic accumulation of cells. The definition of **Tumor** is shown in Figure 2-10.

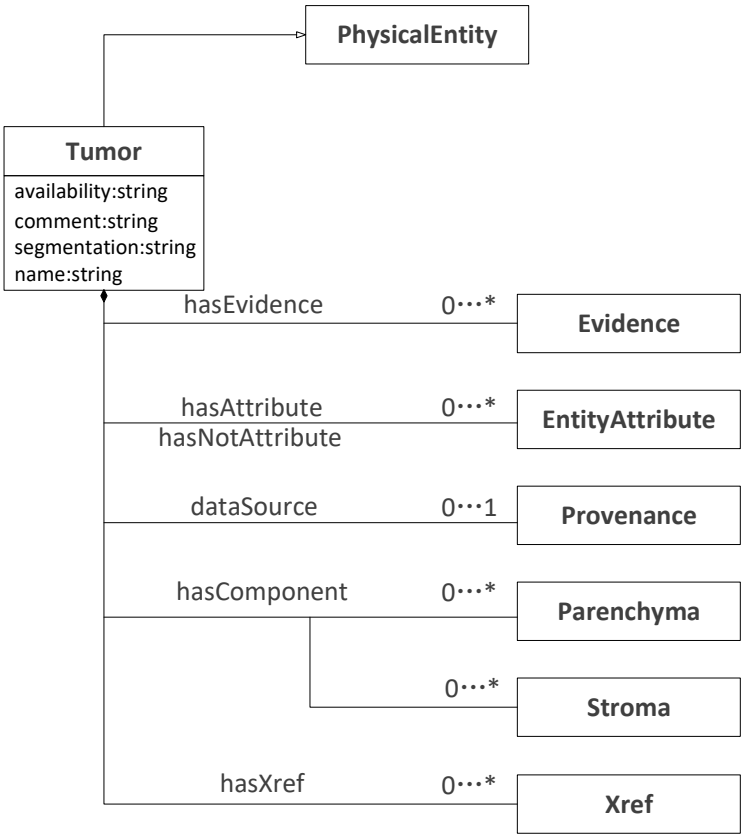


Figure 2-10

*hasComponent*: (0 or more object:Prenchyma or object:Stroma) Used to define tumor components such parenchyma or stroma.

### 2.7.2 Parenchyma

The parenchyma of a tumor is one of the two distinct compartments in a tumor; the other compartment is the stroma. The parenchyma is made up of neoplastic cells. Neoplastic cells and morphologic structures in the parenchyma are described using **Parenchyma**. **Parenchyma** has three child classes which are **NeoplasticCell**, **NeoplasticCellularComponent** and **ParenchymaAnatomicalEntity**. The definition of **Parenchyma** is shown in Figure 2-11.

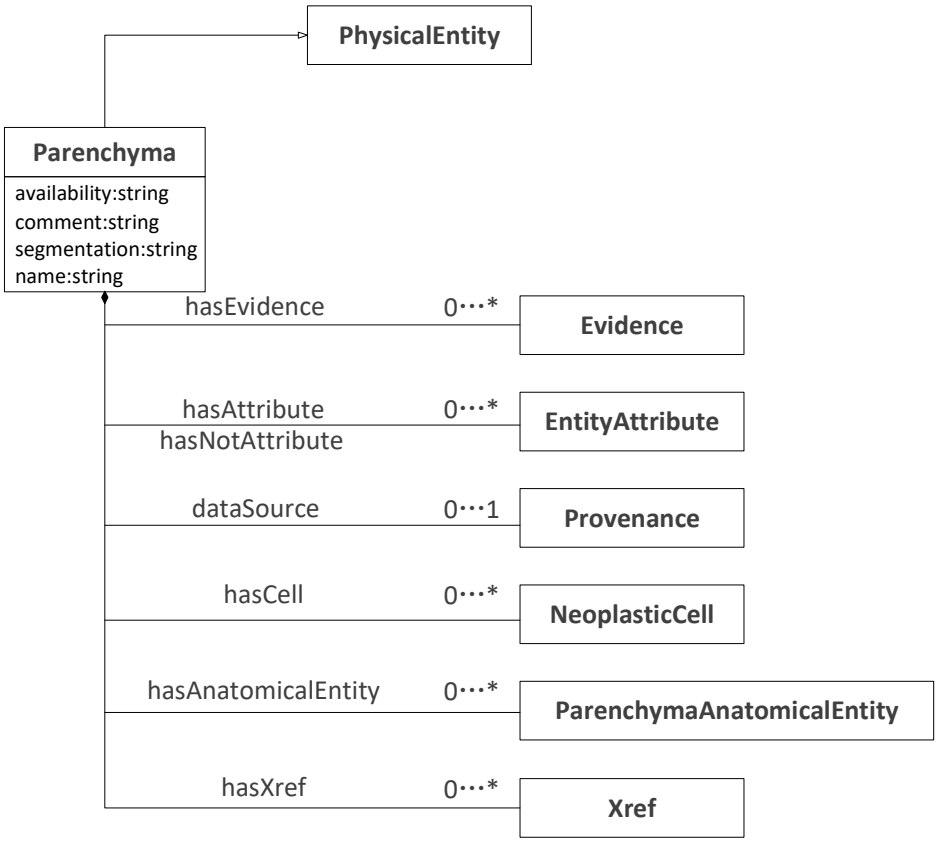


Figure 2-11

*hasCell*: (0 or more object:NeoplasticCell) Used to define tumor cells in the parenchyma.

*hasAnatomicalEntity*: (0 or more object:ParenchymaAnatomicalEntity) Used to define morphologic structures in the parenchyma.

### （1）NeoplasticCell

An instance of **NeoplasticCell** refers to a tumor cell. Its definition is shown in Figure 2-12.

*hasCellularComponent*: (0 or more object:NeoplasticCellularComponent) Used to define cellular components of the cell including its cytoplasm, nucleus, nucleolus.

*hasAnatomicalEntity*: (0 or more object:ParenchymaAnatomicalEntity) Used to define morphologic structures (e.g. cavity) in the tumor cell.

*hasProduct*: (0 or more object:Substance) Used to define the product of the cell.

*hasReserve*: (0 or more object:Substance) Used to define the reserve of the cell.

*entityReference*: (0 or 1 object: CellReference) The entity reference stores the base definition of the cell.

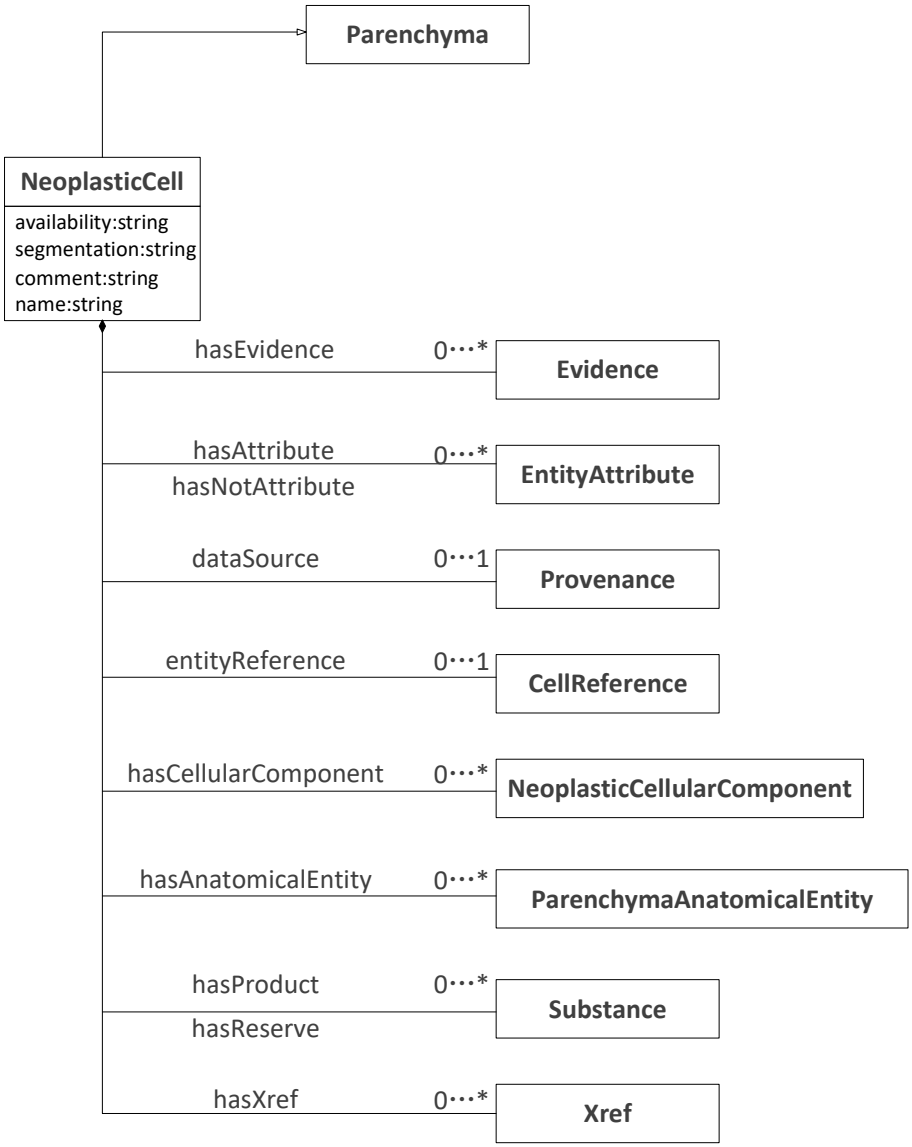


Figure 2-12

## （2）NeoplasticCellularComponent

An instance of **NeoplasticCellularComponent** refers to one of the cellular

components of a tumor cell. Its definition is shown in Figure 2-13.

*hasCellularComponent*: (0 or more object:NeoplasticCellularComponent) Some cellular components are even formed by several parts. For example, nucleus has chromatin and nucleolus etc. *hasCellularComponent* is used to define these components.

*hasAnatomicalEntity*: (0 or 1 object:ParenchymaAnatomicalEntity) Used to define morphologic structures in the cellular component.

*entityReference*: (0 or 1 object: CellularComponentReference) The entity reference stores the base definition of the cellular component.

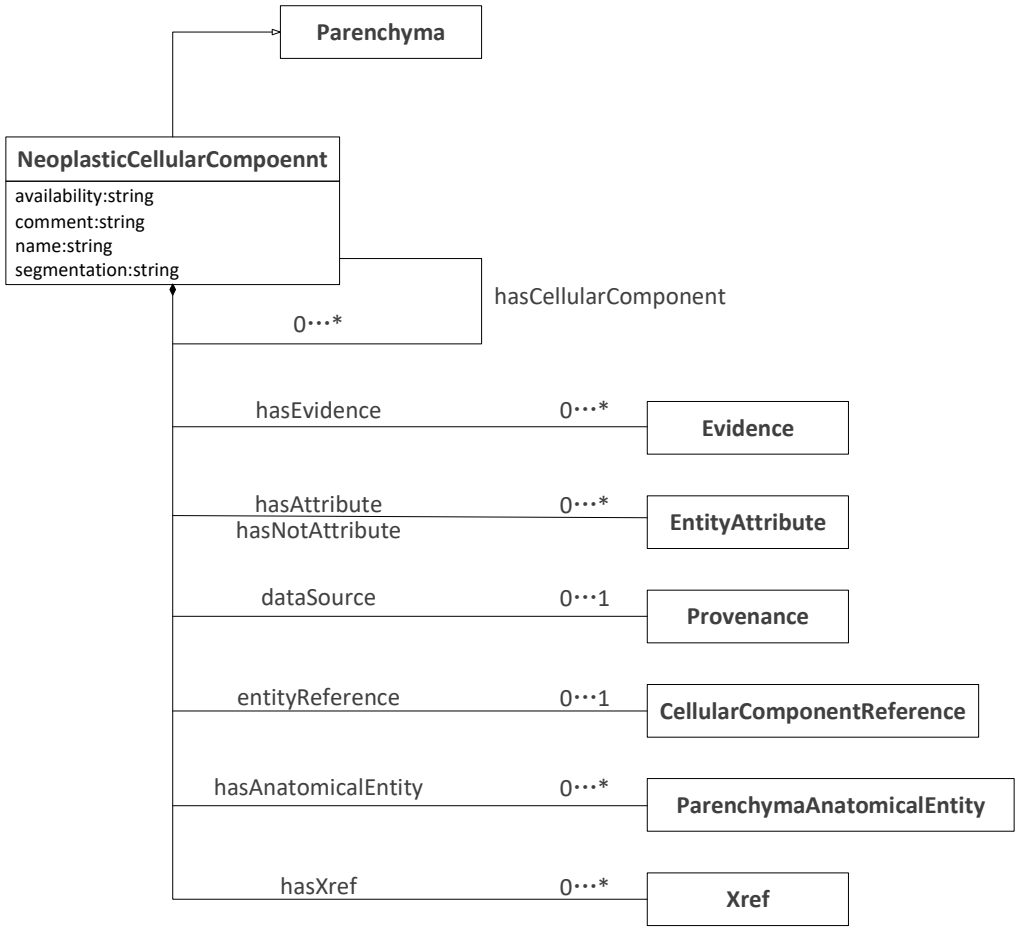


Figure 2-13

### （3）ParenchymaAnatomicalEntity

**ParenchymaAnatomicalEntity** is used to describe morphologic changes within tumor cells like scattered small cystic spaces in the cytoplasm. Its definition is shown in Figure 2-14.

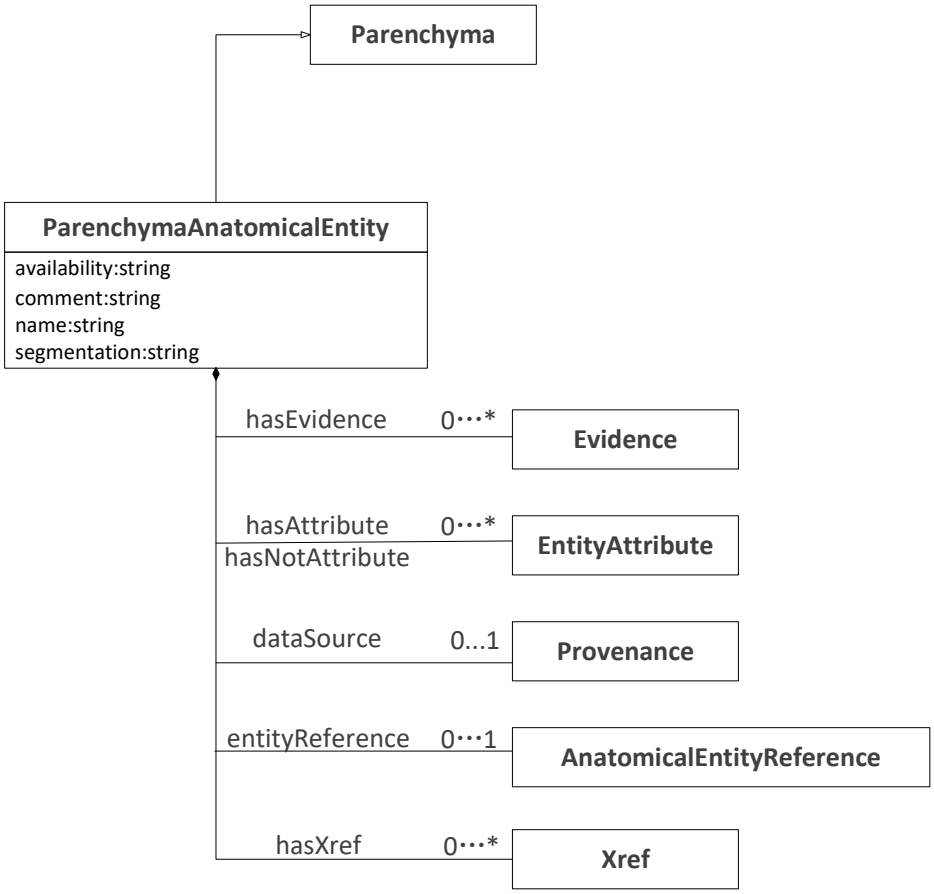


Figure 2-14

*entityReference*: (0 or 1 object:AnatomicalEntityReference) The entity reference stores the base definition of the morphologic structure.

## 2.7.3 Stroma

Tumor stroma is primarily composed of the basement membrane, fibroblasts, extracellular matrix, immune cells, and blood vessels. Cells, components and morphologic changes existed in the stroma should be described using **Stroma**. **Stroma**

has three child classes which are **StromaAnatomicalEntity**, **StromaCell**, **StromaCellularComponent**. Its definition is shown in Figure 2-15.

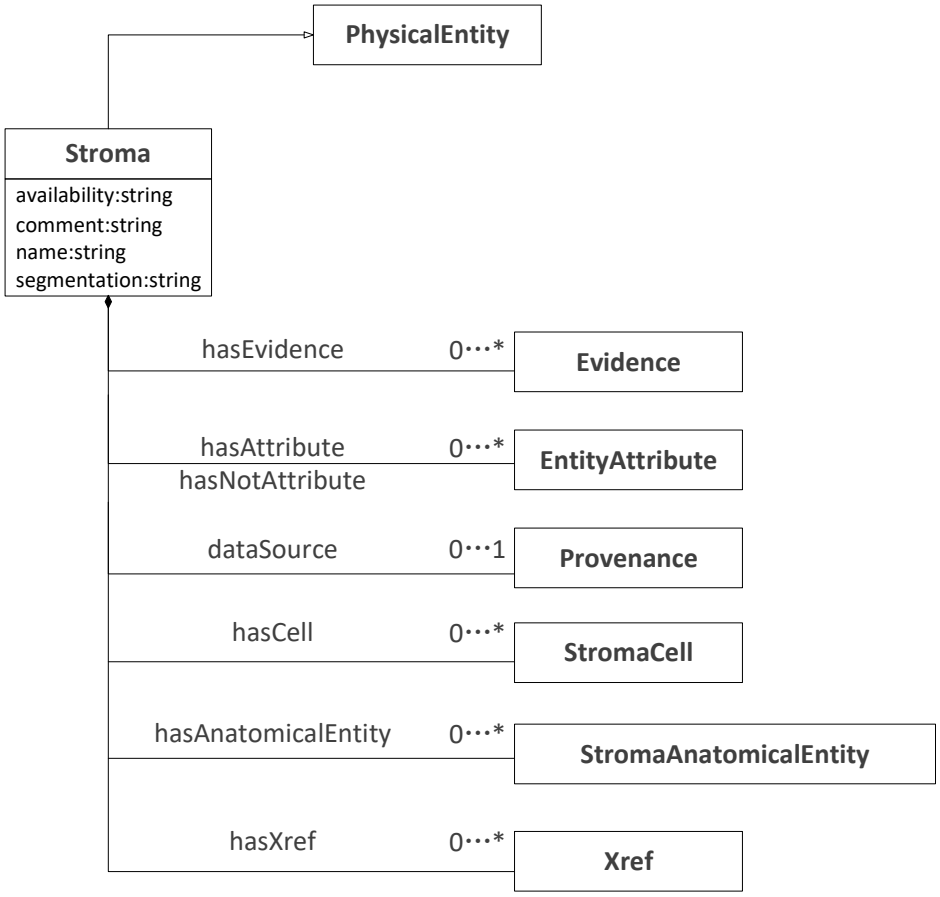


Figure 2-15

*hasCell*: (0 or more object:StromaCell) Used to define cells in the stroma such as immune cells, fibroblasts, endothelial cells etc.

*hasAnatomicalEntity*: (0 or more object:StromaAnatomicalEntity) Used to define tissues (e.g. blood vessels) or morphologic structures in the stroma.

### （1）StromaAnatomicalEntity

Within the stroma, other than cells and their cellular components, any tissue which has a biological function, such as blood vessels and fibers, should be represented using **StromaAnatomicalEntity**, as well as other morphological structures, which are

phenotypically meaningful while don't have a biological function or its function is not clear such as cavity or duct. Its definition is shown in Figure 2-16.

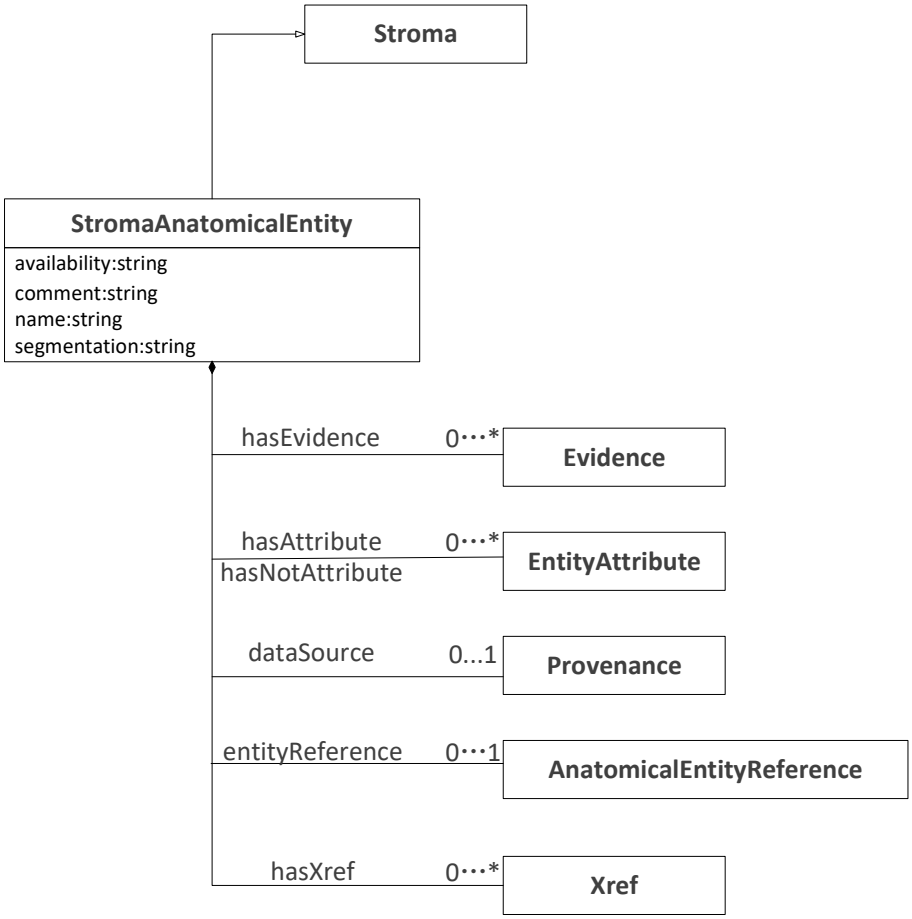


Figure 2-16

## （2）StromaCell

Cells in the stroma should be represented using **`StromaCell`** such as immune cells, fibroblasts, endothelial cells etc. Its definition is shown in Figure 2-17.

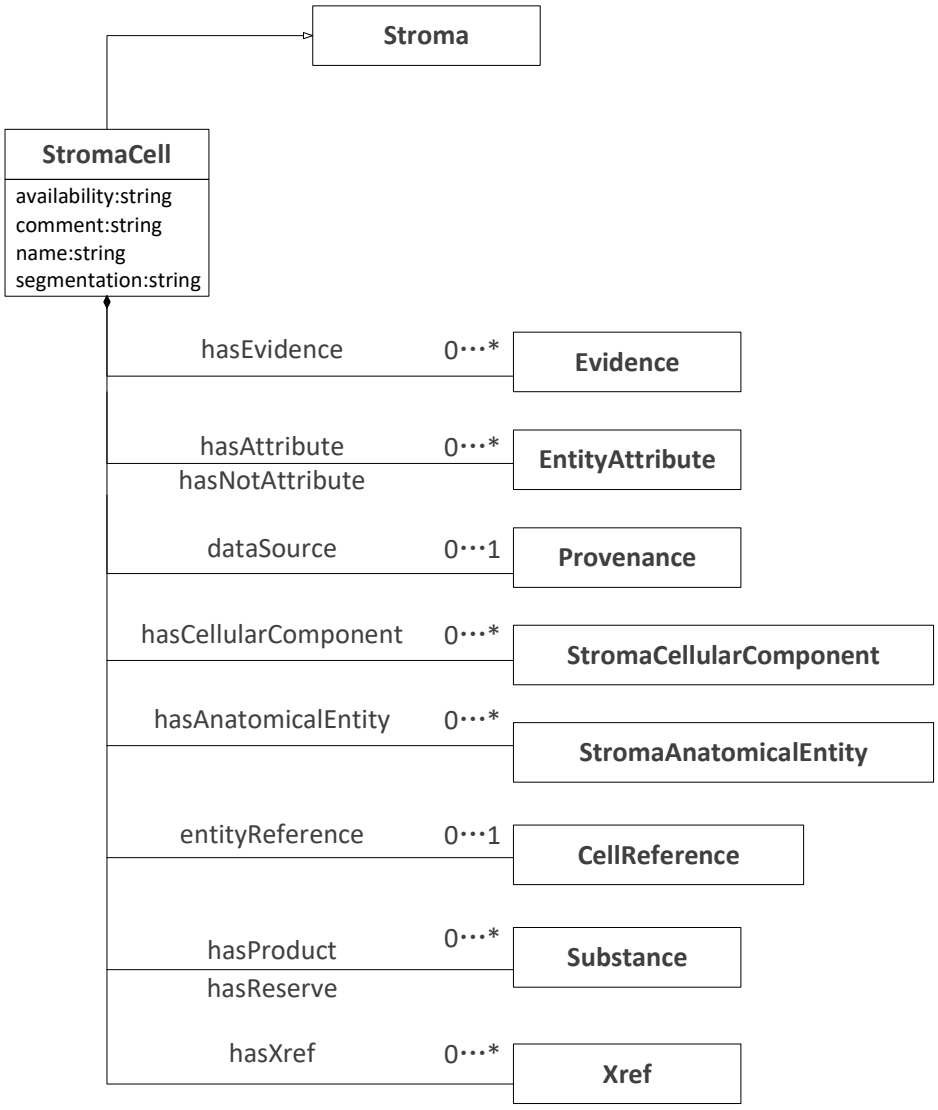


Figure 2-17

*hasCellularComponent*: (0 or more object:StromaCellularComponent) Used to define cellular components of a stroma cell.

*hasAnatomicalEntity*: (0 or more object:StromaMorphologicStrucutre) Used to define morphologic structures in a stroma cell.

*hasProduct*: (0 or more object:Substance) Used to define the product of the cell.

*hasReserve*: (0 or more object:Substance) Used to define the reserve of the cell.

### （3）StromaCellularComponent

Cellular components of the stroma cells should be represented using **StromaCellularComponent**. Its definition is shown in Figure 2-18.

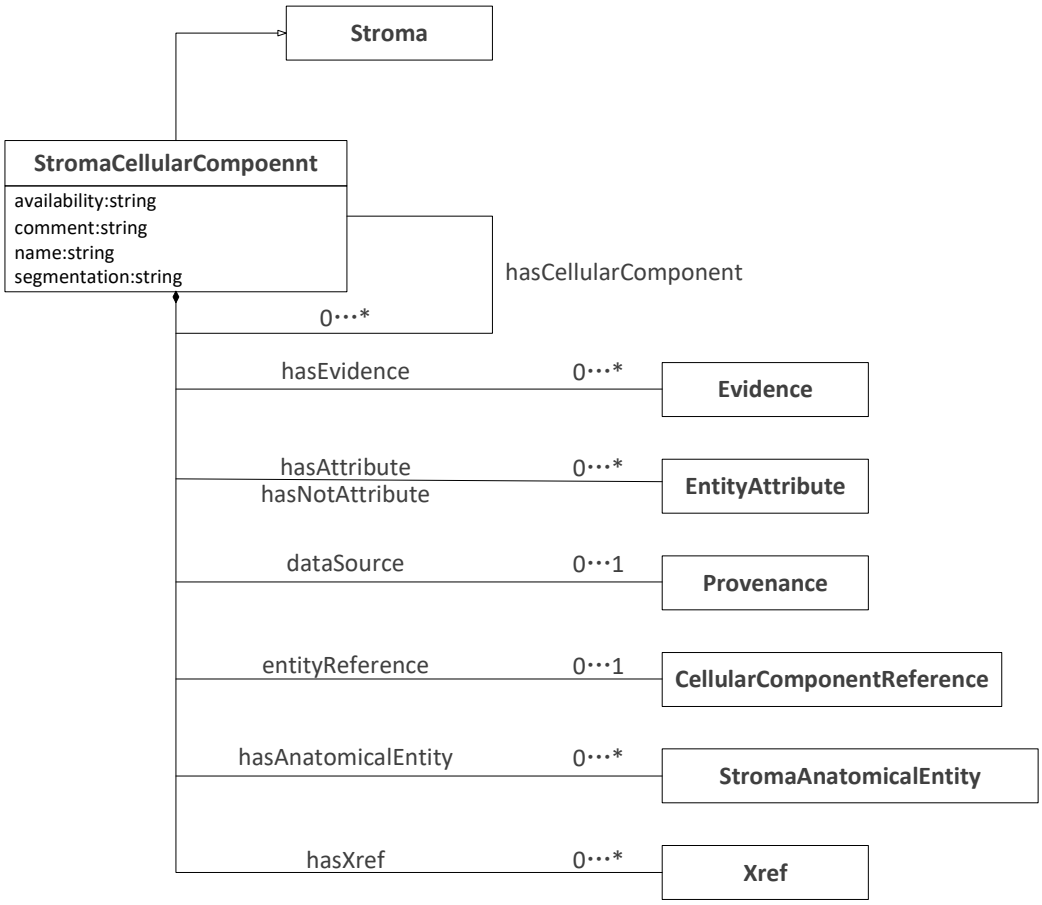


Figure 2-18

*hasCellularComponent*: (0 or more object:StromaCellularComponent) Some cellular components are even formed by several parts. For example, nucleus has chromatin and nucleolus etc. *hasCellularComponent* is used to define these components.

*hasAnatomicalEntity*: (0 or more object:StromaMorphologicStrucutre) Used to define morphologic structures in the cellular component.

## 2.7.4 NormalEntity

**NormalEntity** is used to describe normal cells and their cellular components and

tissues and morphologic structures in the normal regions other than tumors.

### （1）NormalCell

Its definition is shown in Figure 2-19.

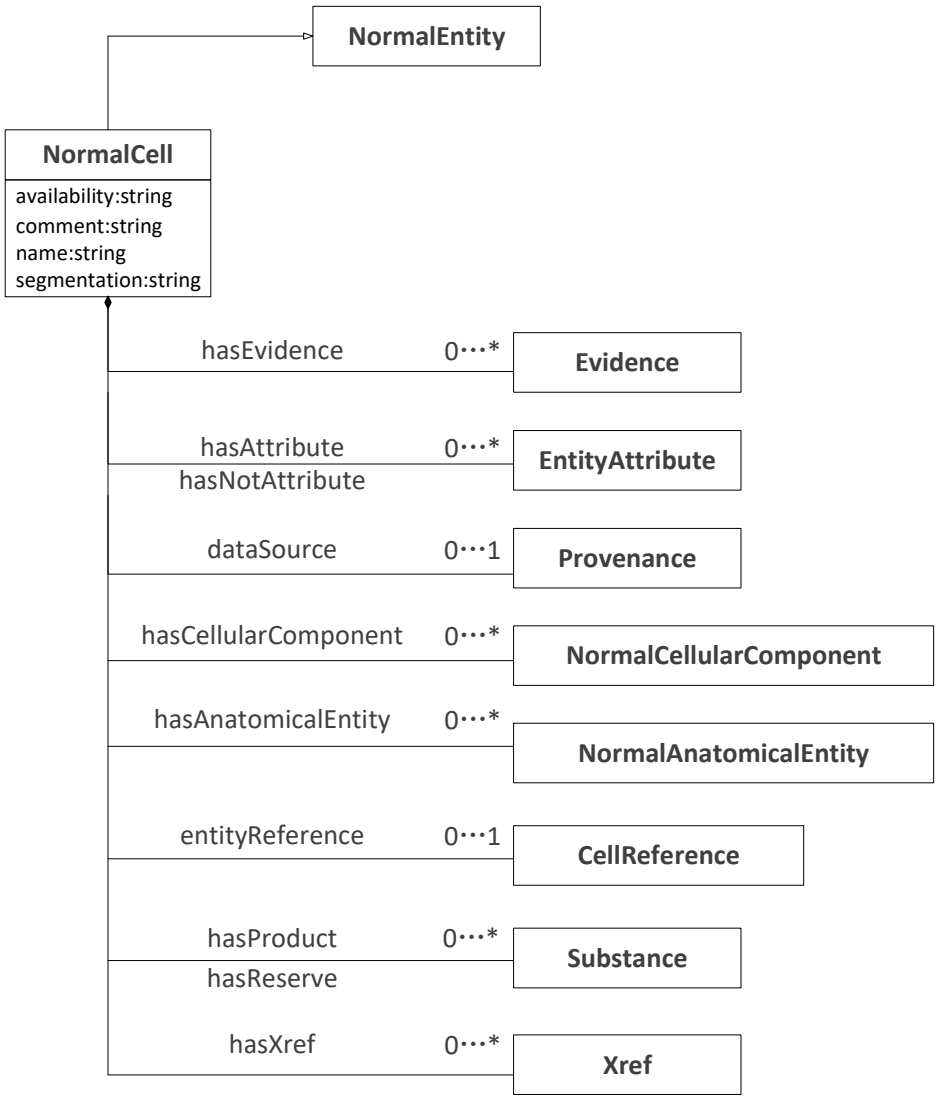


Figure 2-19

*hasCellularComponent*: (0 or more object:NormalCellularComponent) Used to define cellular components of a normal cell.

*hasAnatomicalEntity*: (0 or more object:NormalMorphologicStrucutre) Used to define morphologic structures in a normal cell.

*hasProduct*: (0 or more object:Substance) Used to define the product of the cell.

*hasReserve*: (0 or more object:Substance) Used to define the reserve of the cell.

### （2）NormalCellularComponent

Its definition is shown in Figure 2-20.

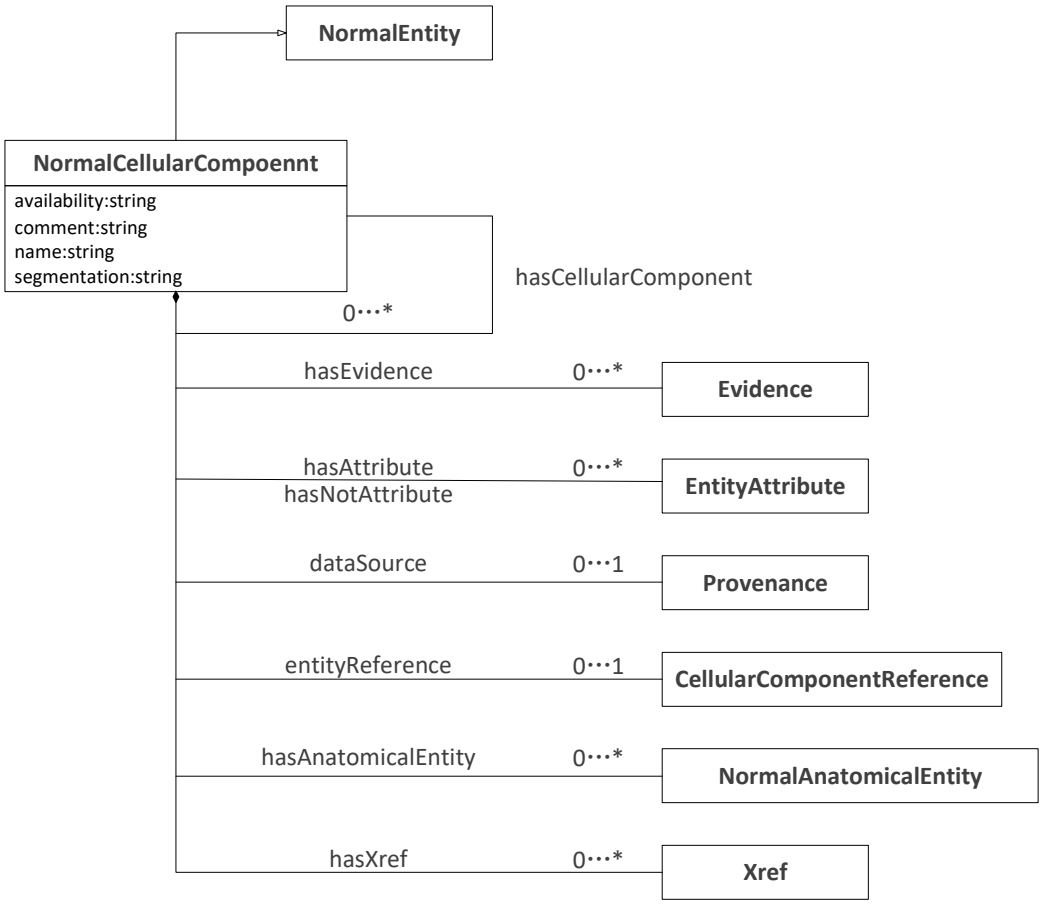


Figure 2-20

*hasCellularComponent*: (0 or more object:NormalCellularComponent) Some cellular components are even formed by several parts. For example, nucleus has chromatin and nucleolus etc. *hasCellularComponent* is used to define these components.

*hasAnatomicalEntity*: (0 or more object:NormalMorphologicStrucutre) Used to define morphologic structure in the cellular component.

### （3）NormalAnatomicalEntity

Tissues and morphological structures in the normal regions are represented using **NormalAnatomicalEntity**. Its definition is shown in Figure 2-21.

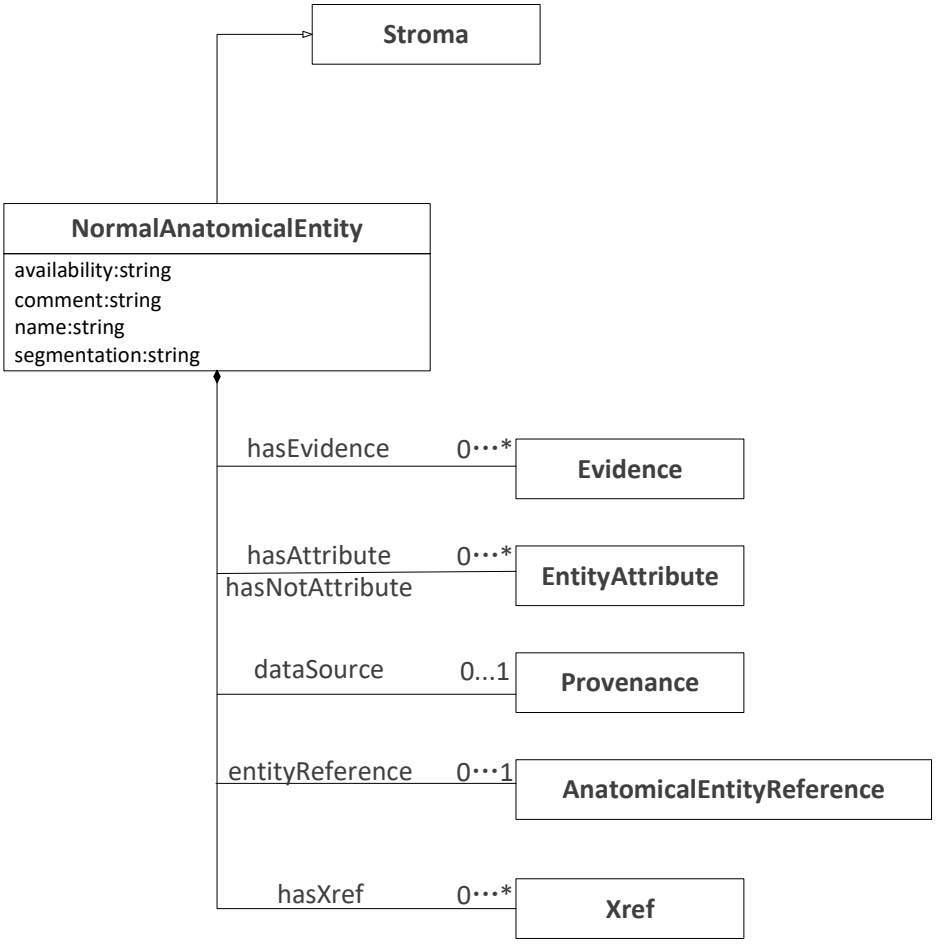


Figure 2-21

*hasCell*: (0 or more object:NormalCell) Used to define cells of a tissue in a normal region.

*hasAnatomicalEntity*: (0 or more object:NormalAnatomicalEntity) Used to define tissues and morphologic structures in a normal region.

## 2.7.5 Substance

**Substance** is used to describe any chemical substance including a cell's product

or reserve. Its definition is shown in Figure 2-22.

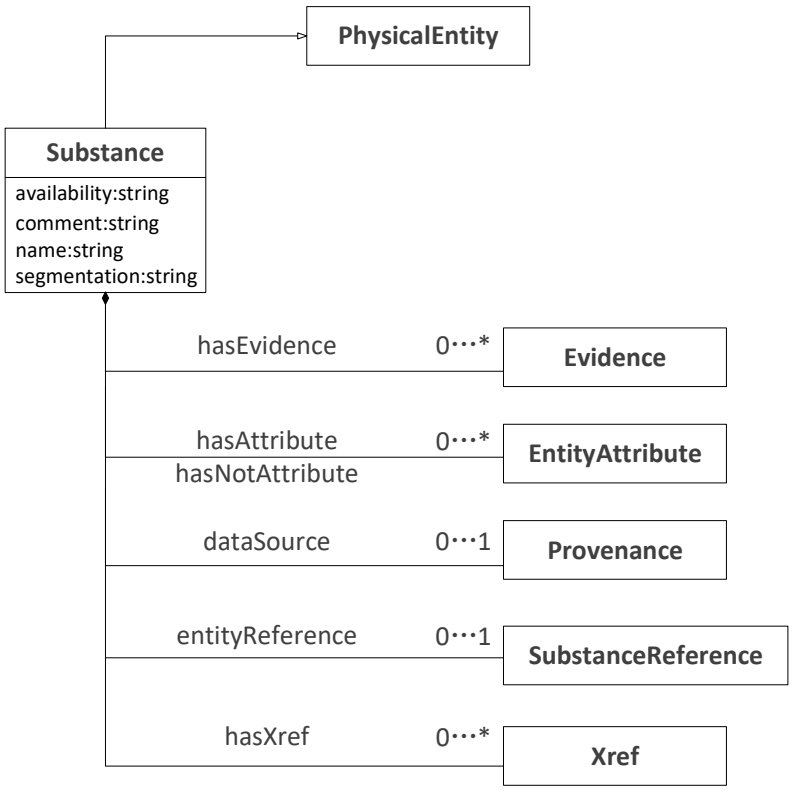


Figure 2-22

## 2.8 Phenotype Subclass

**Phenotype** has three child classes including **Architectural_Pattern**, **Cellular_Appearances** and **Product_or_Reserve**. The structure of Phenotype is shown in Figure 2-23.

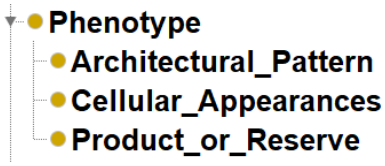


Figure 2-23

### 2.8.1 Cellular_Appearances

**Cellular_Appearances** describes a physical change of a cell or a subcellular

structure which are usually observed microscopically at high power. Its definition is shown in Figure 2-24.

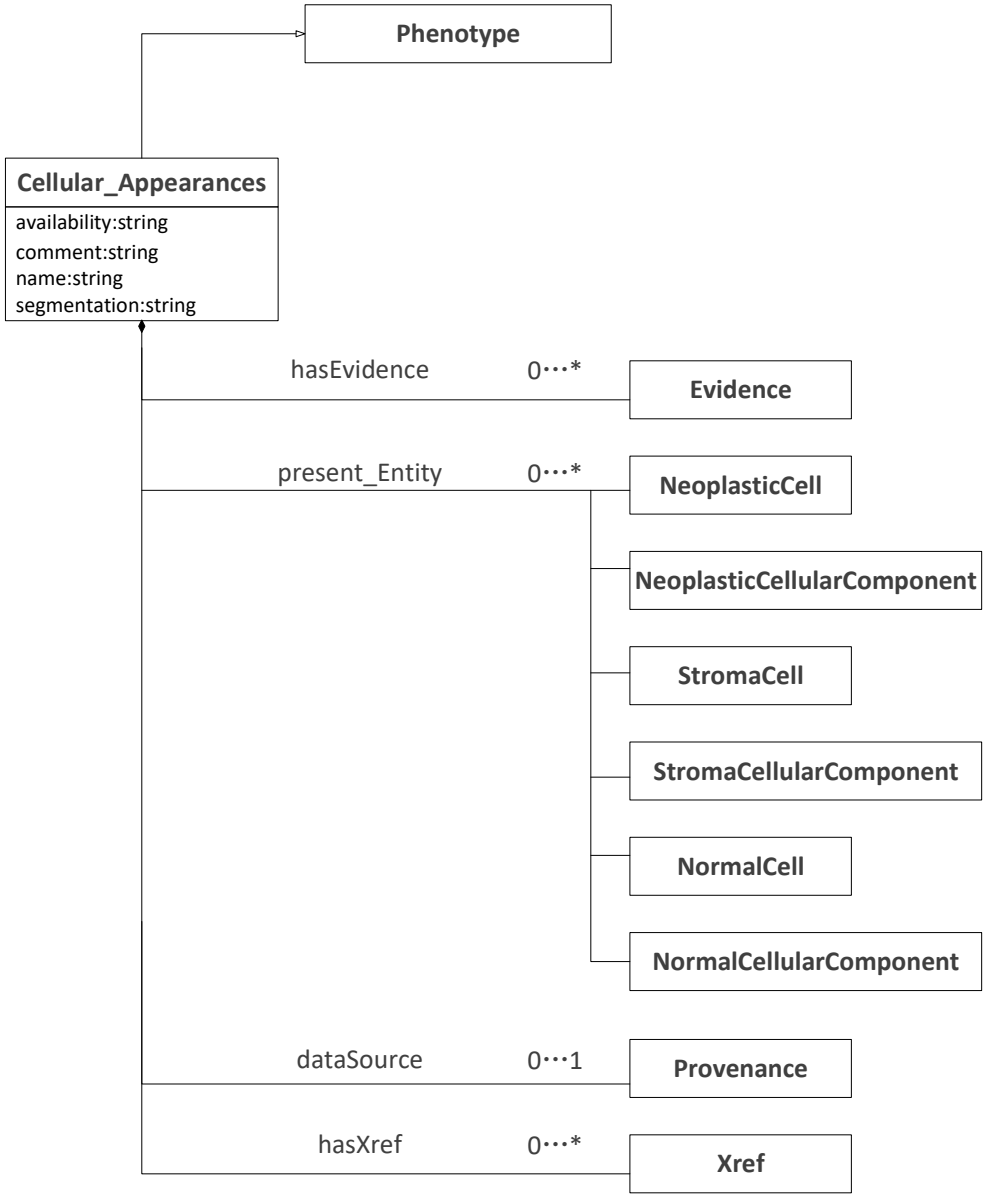


Figure 2-24

*present_Entity*: (0 or more object:Cell or object:CellularComponent) Used to describe the appearance by linking it to the corresponding **PhysicalEntity** instance which defines the cell or cellular component..

## 2.8.2 Architectural_Pattern

**Architectural_Pattern** describes histologic patterns of cell populations and tumor behaviors (e.g. extension, invasion) which are usually observed microscopically at medium and low power. Its definition is shown in Figure 2-25.

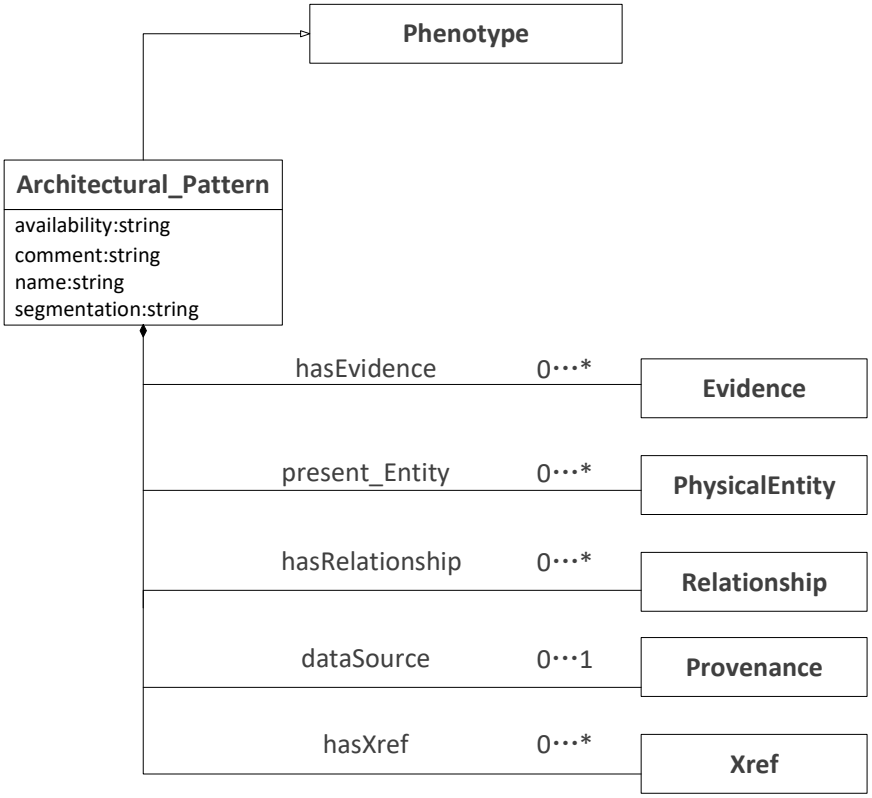


Figure 2-25

*present_Entity*: (0 or more object:PhysicalEntity) The physical entity that is present in the architectural pattern, e.g., in an acinar pattern, the present entities are erythrocytes, acidophilic fluid, a lumen, tumor cells, a capillary and endothelial cells.

*hasRelationship*: (0 or more object:Relationship) The relationships between the present entities which are used to describe tumor behaviors such as invading, extending, being limited to etc.

## 2.8.3 Product_or_Reserve

Some phenotypes observed under the slide refer to the products or reserves of

tumor cells such as neutral fat, glycogen, pigment or crystal etc. These phenotypes are described using **Product_or_Reserve**. Its definition is shown in Figure 2-26.

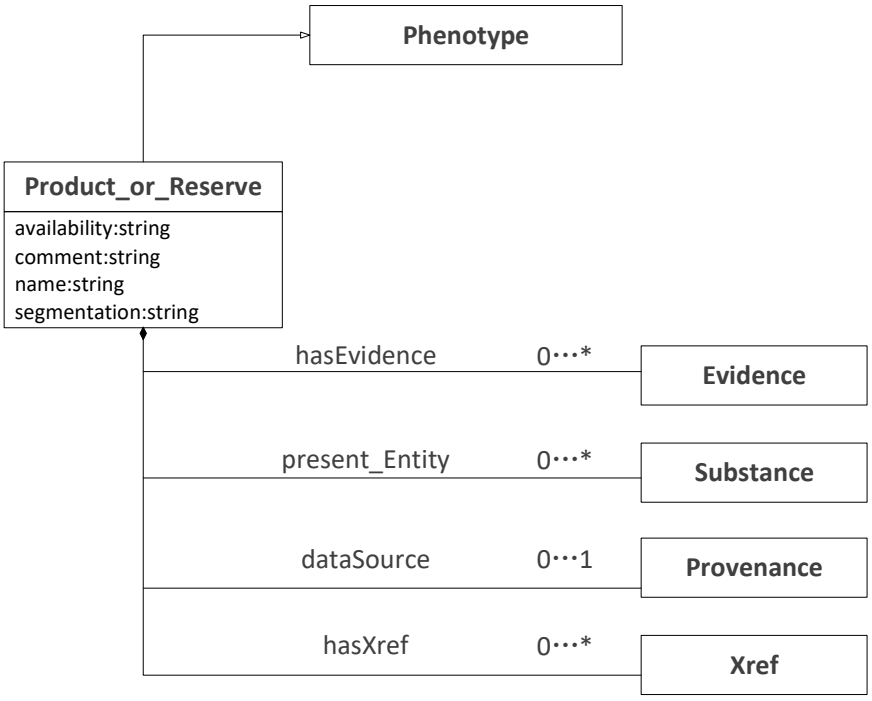


Figure 2-26

*present_Entity*: (0 or more object:Substance) The substance that is present in the slide, e.g., in the area of calcification, the present substance is calcium. .

## 2.9 Diagnosis Subclass

**Diagnosis** has nine child classes including **Differentiation**, **Grading**, **Histology**, **Lymphovascular_Invasion**, **Necrosis**, **Site**, **Subtyping**, **Tumor_Inflammation**, **Tumor_Extension**. The structure of **Diagnosis** is shown in Figure 2-27. The object properties and data properties and their usage of these subclasses are same as **Diagnosis**.

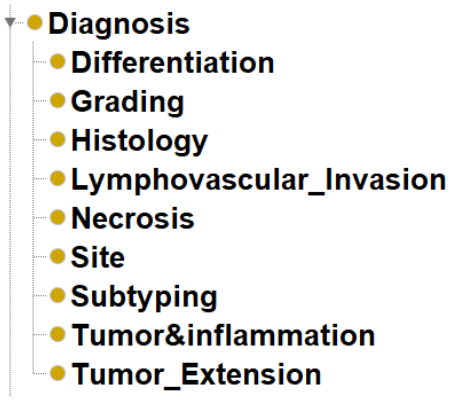


Figure 2-27

## 2.10 Utility

Several object properties of **Entity** subclasses accept instances of **Utility** classes as values. **Utility** classes are used to annotate the **Entity** subclasses. Examples include references to external databases, controlled vocabularies, evidence and provenance. The structure of Utility class is shown in Figure 2-28.

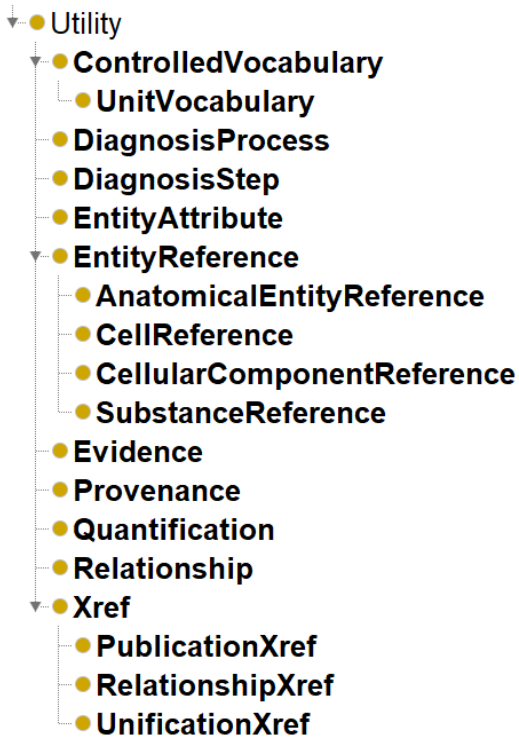


Figure 2-28

### 2.10.1 EntityAttribute

An attribute of a physical entity that can be changed while the entity still retains its biological identity. Entity attributes could also be generic across physical entities that are in a generic grouping. This allows generic attributes to be defined on a generic physical entity in **EntityReference**. Its definition is shown in Figure 2-29.

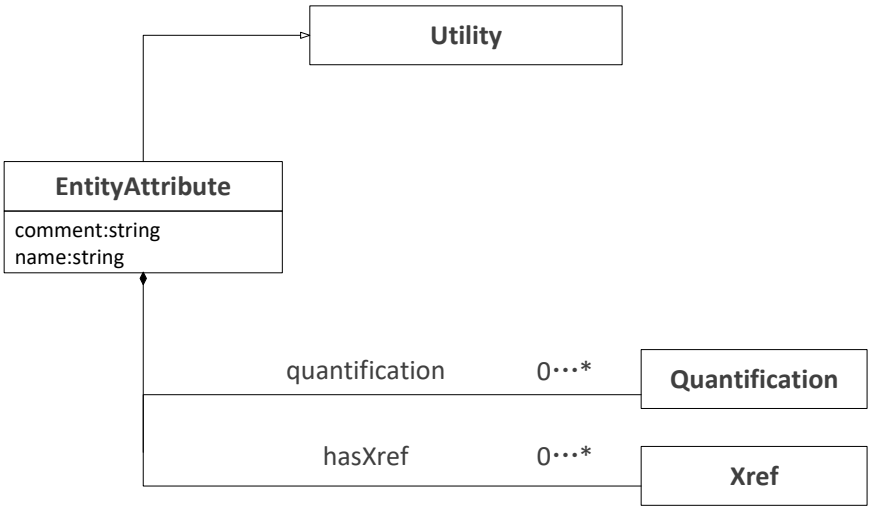


Figure 2-29

*quantification*: (0 or more object:Quantification) Used to quantitatively describe physical entities' attributes.

*hasXref*: (0 or more object:Xref) Used to clarify the attributes by providing their semantic cross-references to the controlled vocabulary such as The Phenotype And Trait Ontology (PATO).

### 2.10.2 EntityReference

An entity reference is a grouping of several physical entities with multiple states that are often named and treated as a single entity by pathologists. **EntityReference** instances store the information common to a set of entities in various states. For example, the morphology of a tumor cell of clear cell renal cell carcinoma (ccRCC) undergoes changes as it evolves including nuclear changes, cytoplasmic changes; any ccRCC tumor cell in one of the states is represented as an individual of **NeoplasticCell** while a generic ccRCC tumor cell is represented as an individual of **CellReference** (child of **EntityReference**). The **EntityReference** is important because it explicitly links multiple physical entities representing different states of a generic entity, which would otherwise be difficult to recognize as related. The definition

of EntityRefrence is shown in Figure 2-30.

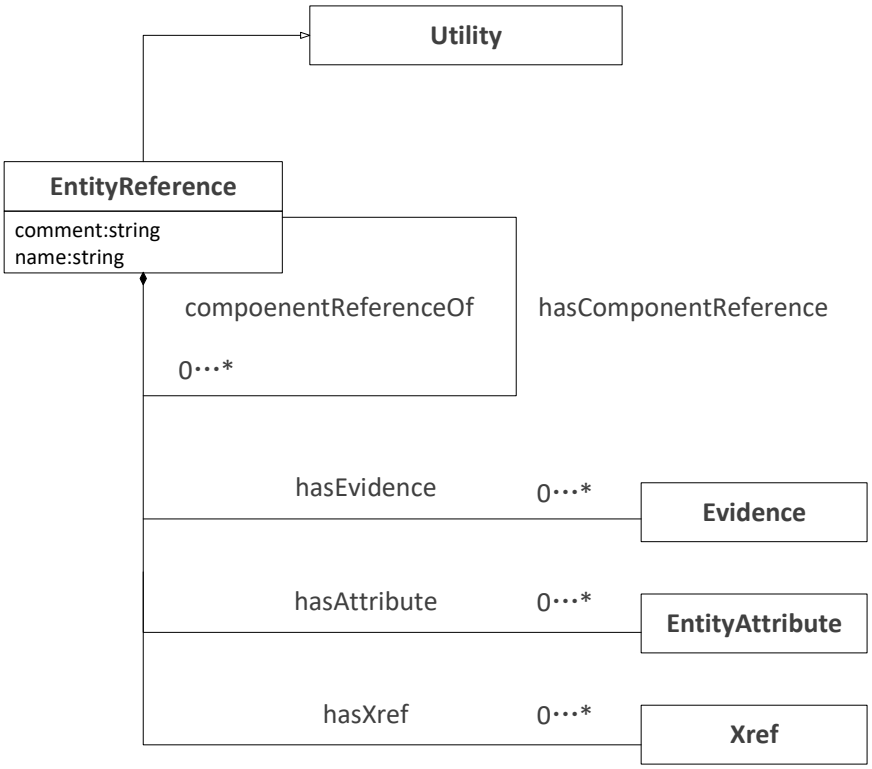


Figure 2-30

*hasAttribute*: (0 or more object:EntityAttribute) This property is used to define generic attributes for a generic physical entity. Other attributes which distinguish the specific physical entities representing different states of the generic entity are defined in **PhysicalEntity**.

*hasComponentReference*: (0 or more object:EntityReference) Whereas *hasComponent* is used to define components of a physical entity in a specific state, *hasComponentReference* is used to define components of the generic entity. It has 3 sub-properties which are *hasAnatomicalEntityReference*, *hasCellularComponentReference* and *hasCellReference*. For example, in order to represent a tumor cell with eosinophilic cytoplasm, a tumor cell is represented as a **NeoplasticCell**, and the cytoplasm is represented as a **NeoplasticCellularComponent**; *hasCellularComponent* is used to link the tumor

cell to the cytoplasm. While *hasCellularComponentReference* is used to link the entity reference of the tumor cell to the reference of the cytoplasm since cytoplasm, in any morphologic state, is the component of a cell.

*compoenentReferenceOf*: (0 or more object:EntityReference) Inverse property of *hasComponentReference*. Similar to *hasComponent* and *componentOf*, either *hasComponentReference* or *compoenentReferenceOf* is specified, the other could be inferred.

*hasXref*: (0 or more object:Xref) External cross-references to a controlled vocabulary.

### （1）CellReference

Used to store shared information about a set of related cells differing in states. Its definition is shown in Figure 2-31.

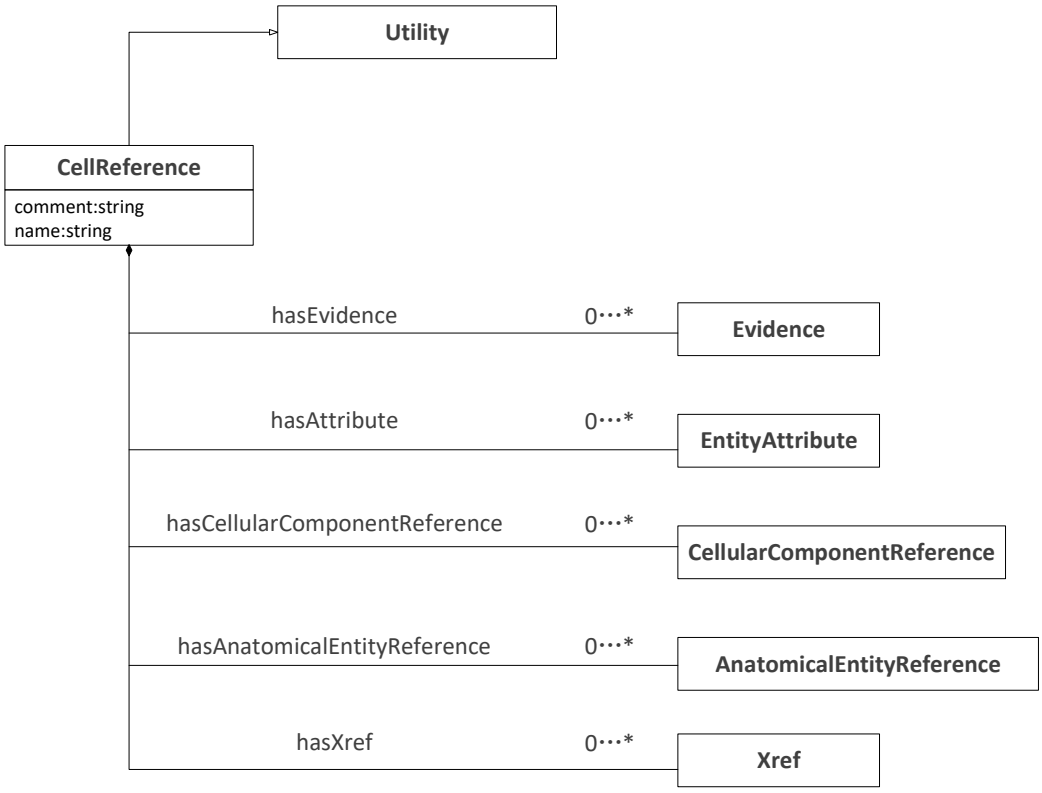


Figure 2-31

## （2） CellularComponentReference

Used to store shared information about a set of related cellular components differing in states. Its definition is shown in Figure 2-32.

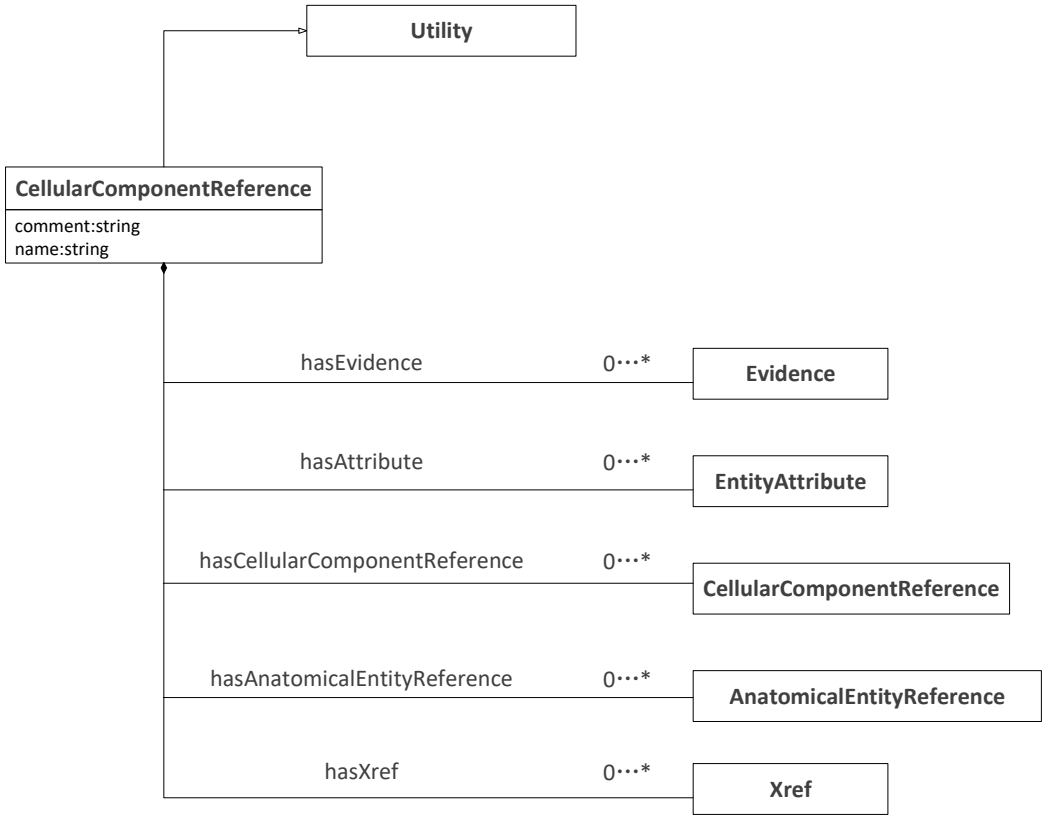


Figure 2-32

## （3） AnatomicalEntityReference

Used to store shared information about a set of related tissues or morphologic structures differing in states. Its definition is shown in Figure 2-33.

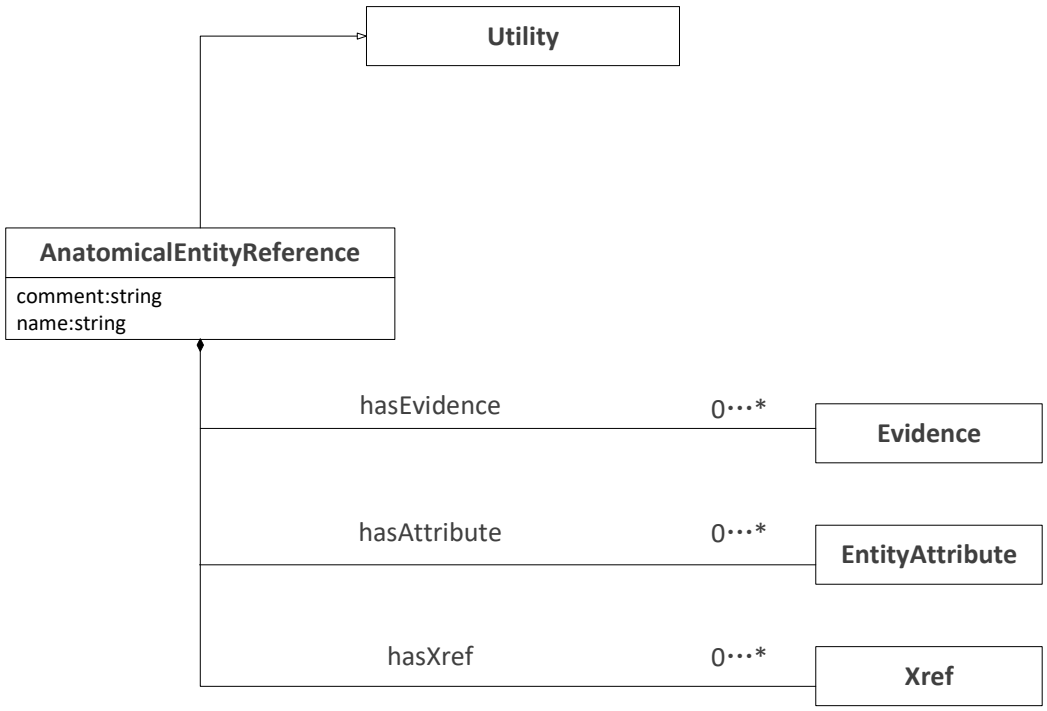


Figure 2-33

## （4）SubstanceReference

Used to store shared information about a set of related substance differing states such as protein molecules encoded by the same gene. Its definition is shown in Figure 2-34.

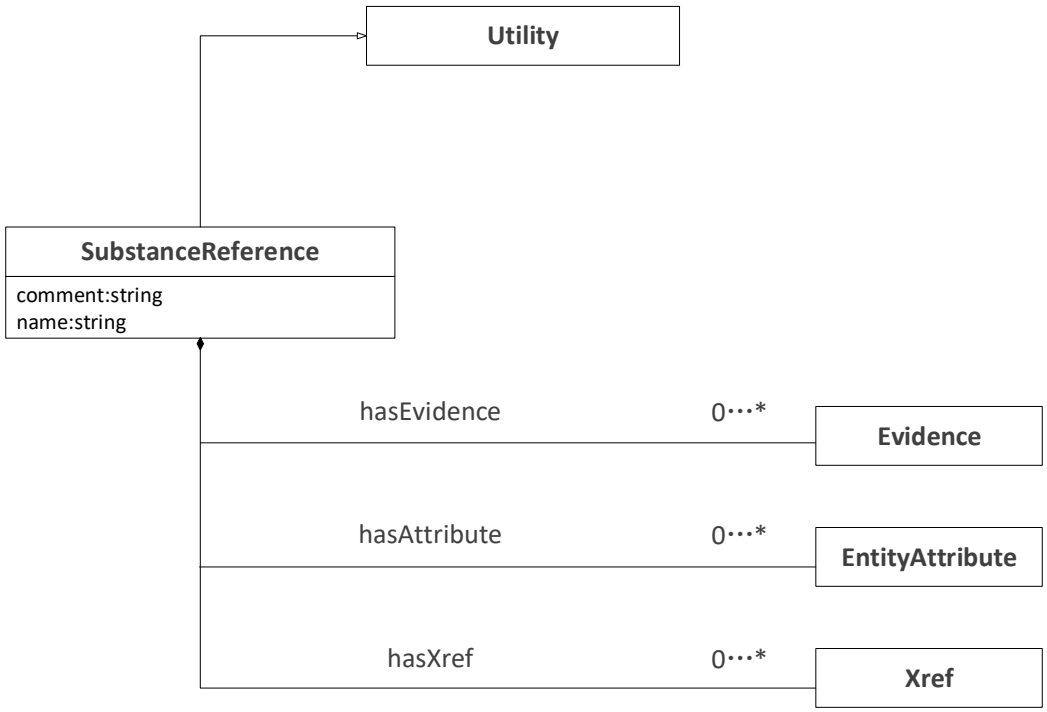


Figure 2-34

### 2.10.3 Evidence

The scientific support for an assertion in a HistoML representation, such as a histopathological diagnosis. Evidence class has 2 properties: *hasXref* and *comment*. *hasXref* would reference a publication describing the evidence using a **PublicationXref**; the content of the evidence should be the value of *comment*. For example, when you subtype a slide and represent the diagnosis which is "Human Papillomavirus-Related Endocervical Adenocarcinoma" in the HistoML representation. The value of *hasEvidence* in the instance of **Diagnosis**, which is an **Evidence** individual, should include a **PublicationXref** to the book of "Female Genital Tumours: WHO Classification of Tumours (Medicine) 5th Edition" and the value of *comment* should include the definition of the conditions to make this diagnosis in the book which is "Hallmarks of HPV-associated endocervical adenocarcinoma architecture include apical mitoses and karyorrhexis, conspicuous and identifiable at low-power magnification". An assertion might have more than one **Evidence**; the

relevant guidelines and papers can be included to provide more information. Its definition is shown in Figure 2-35.

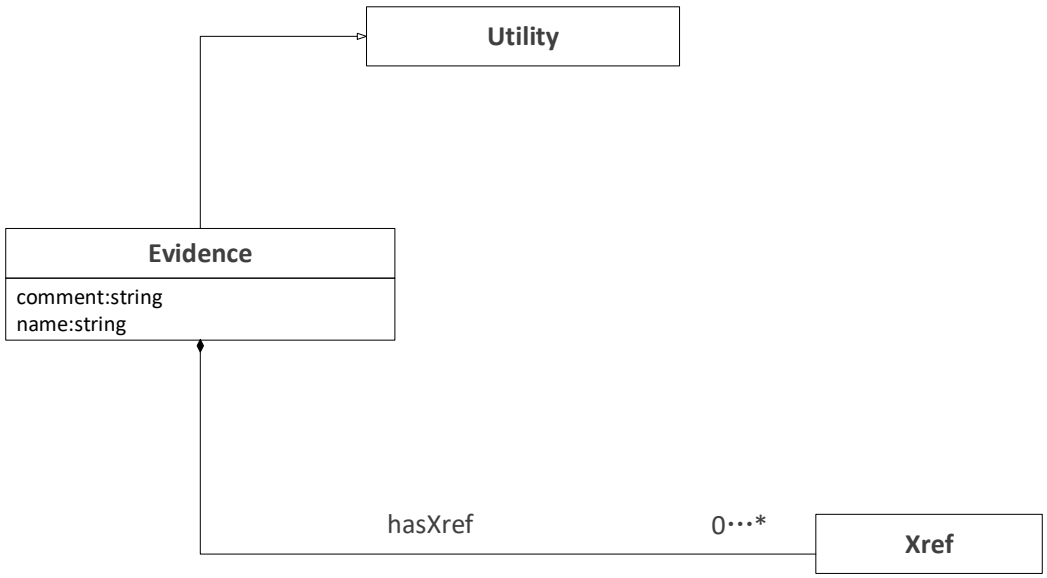


Figure 2-35

### 2.10.4 Provenance

The direct source of the data represented using HistoML such as a database. The *hasXref* property may contain a **PublicationXref** referencing a publication describing the data source (e.g. a database publication). Its definition is shown in Figure 2-36.

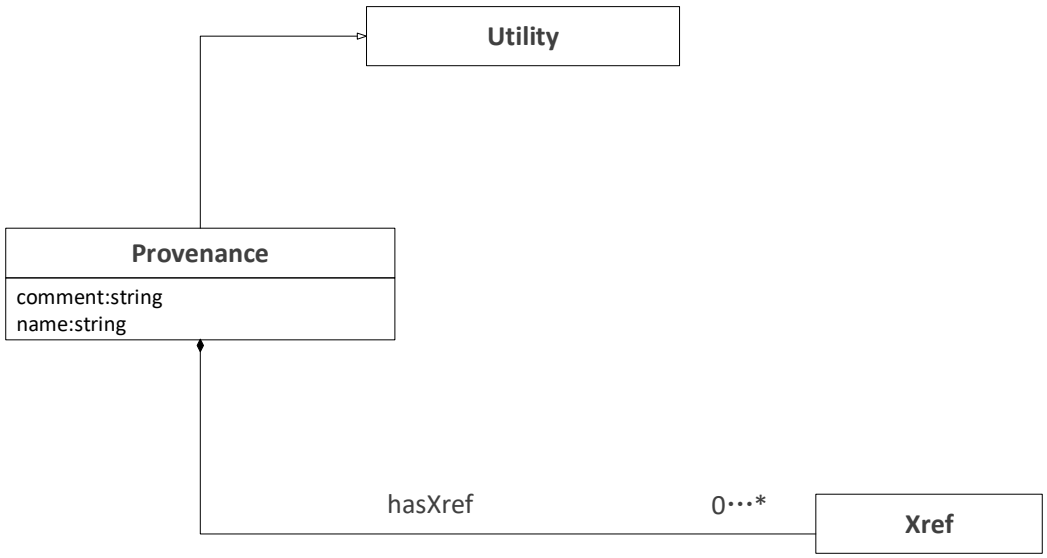


Figure 2-36

### 2.10.5 Quantification

**Quantification** is used to quantitatively describe an attribute of an entity which provides the absolute amount or the numerical range of the attribute. Its definition is shown in Figure 2-37.

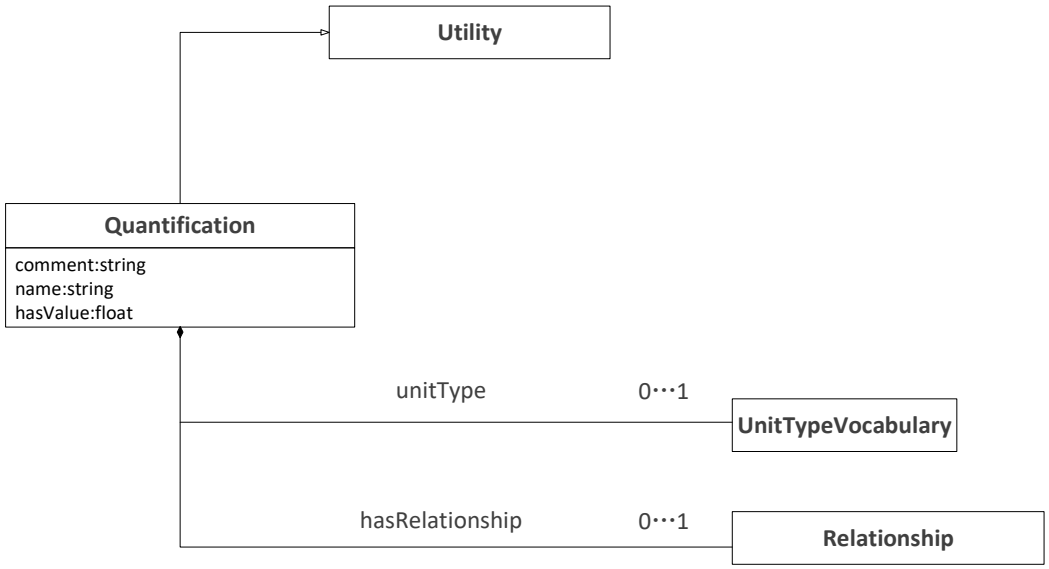


Figure 2-37

hasValue: (xsd:float) The absolute amount of the attribute.

*unitType*: (0 or 1 object:UnitTypeVocabulary) The unit of the attribute. The unit depends on the quality of the attribute. For example, if the quality is the diameter of a cell, then the unit is micrometer.

*hasRelationship*: (0 or 1 object:Relationship) The numerical relationship between the attribute and the amount such as “is equal to”, “is greater than or equal to”, and “is less than or equal to” etc. *hasXref* of the **Relationship** would reference the term of controlled vocabulary referring to this numerical relationship.

### 2.10.6 Relationship

The relationships between entities. The relationships are represented as triple. The head entity and tail entity are represented using *subject* and *object* respectively. Its definition is shown in Figure 2-38.

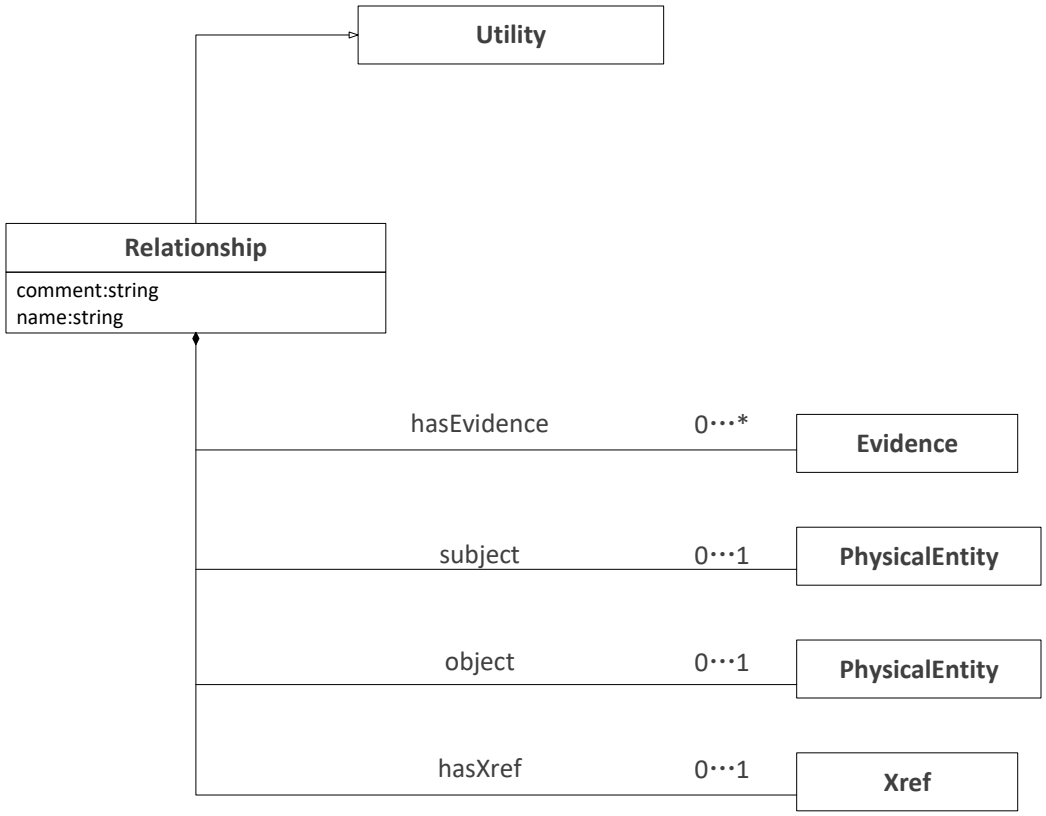


Figure 2-38

*subject*: (0 or 1 object:PhysicalEntity) The head entity of the relationship.

*object*: (0 or 1 object:PhysicalEntity) The tail entity of the relationship.

### 2.10.7 Xref

A reference from an instance of a class in HistoML to an object in an external resource (e.g. controlled vocabulary). Its definition is shown in Figure 2-39. **Xref** has three sub-classes which are **PublicationXref**, **RelationshipXref** and **UnificationXref**.

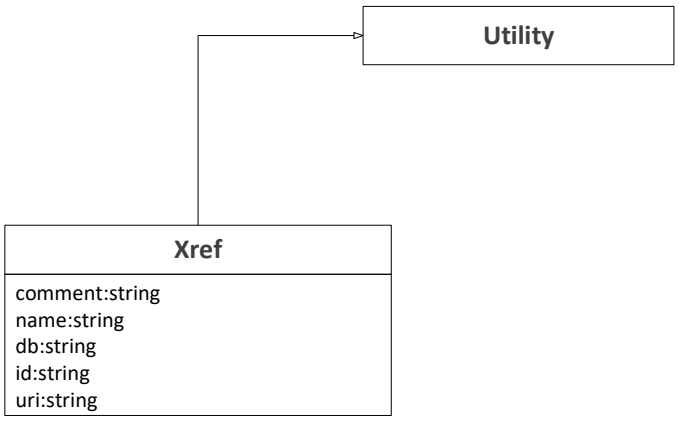


Figure 2-39

db: (xsd:string) The name of the external database to which this xref refers.

id: (xsd:string) The identifier in the external database of the object to which this xref refers.

uri: (xsd:string) Uniform resource identifier of referencing term, such as IRI (Internationalized Resource Identifier) of an ontology term.

#### （1）PublicationXref

An xref that defines a reference to a publication such as a journal article, book, web page, or software manual. References to PubMed are preferred when possible. Its definition is shown in Figure 2-40.

author: (xsd:string) The authors of this publication, one per property value.

db: (xsd:string) PubMed or ISBN.

id: (xsd:string) The PubMed of an academic paper or ISBN number of book if it is available.

title: (xsd:string) The title of the publication.

year: (xsd:string) The year when this publication published.

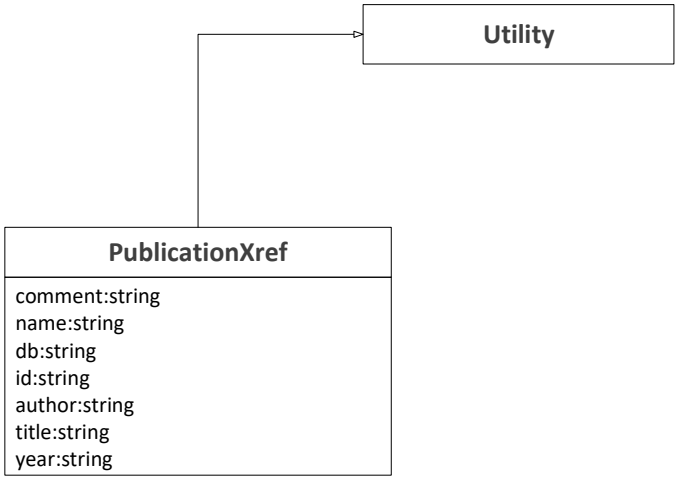


Figure 2-40

## （2）RelationshipXref

An xref that defines a reference to an entity in an external resource that does not have the same biological identity as the referring entity. Its definition is shown in Figure 2-41.

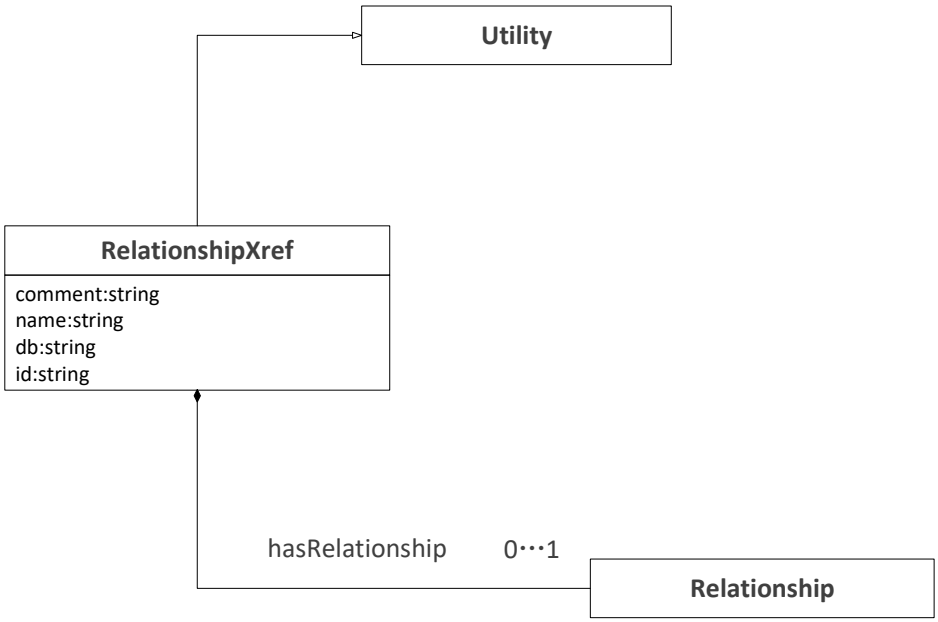


Figure 2-41

*hasRelationship*: (0 or 1 object:Relationship) This property names the type of relationship between the HistoML object linked from and the external object linked to.

### （3）UnificationXref

A **UnificationXref** defines a reference to an entity in an external resource that has the same biological identity as the referring entity. **UnificationXref** should be used whenever possible since it improves data integration and semantic interoperability. Its definition is shown in Figure 2-42.

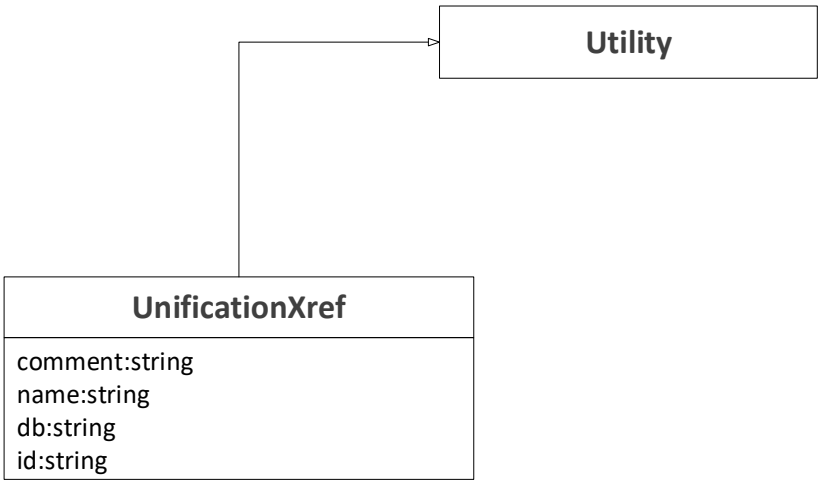


Figure 2-42

### 2.10.8 ControlledVocabulary

**ControlledVocabulary** is used to define HistoML's own controlled vocabulary terms. **UnitTypeVocabulary** is the sub-class of it which provides reference to the Units of measurement ontology (UO) to quantitatively describe the attributes of an entity. Its definition is shown in Figure 2-43.

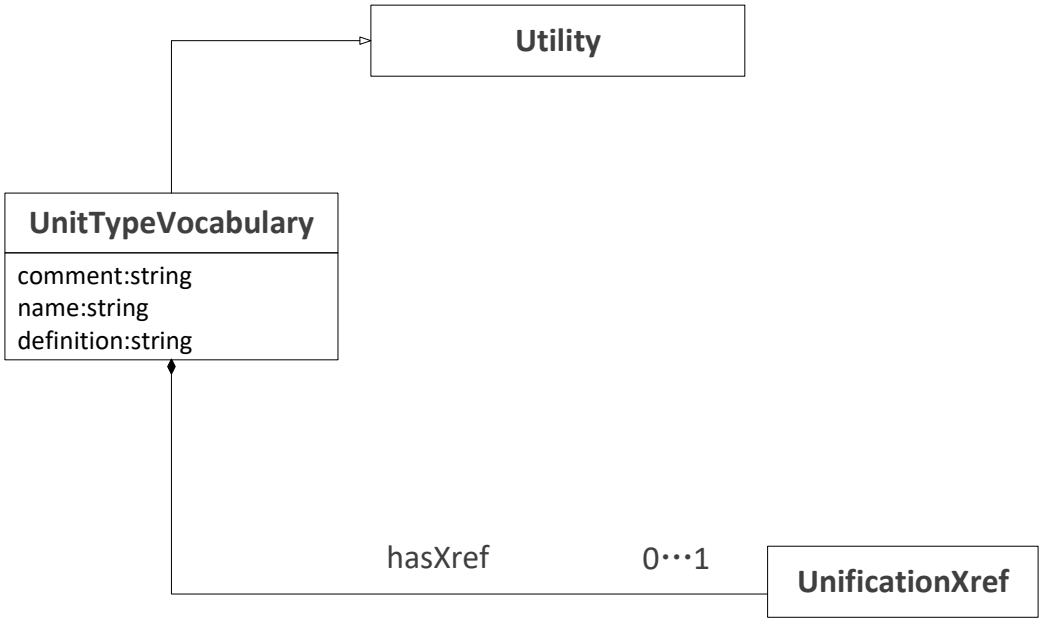


Figure 2-43

*hasXref*: (0 or 1 object:UnificationXref) the cross-reference to the term in Units of

measurement ontology (UO).

`name`: (xsd:string) name of this unit.

`definition`: (xsd:string) definition of this unit.

### 2.10.9 DiagnosisProcess

HistoML could represent the diagnostic decision-making process of a pathologist which is composed by several diagnoses by using **`DiagnosisProcess`** and **`DiagnosisStep`**, depicting the cause-and-effect relationships between these diagnoses. The definition of **`DiagnosisProcess`** is shown in Figure 2-44.

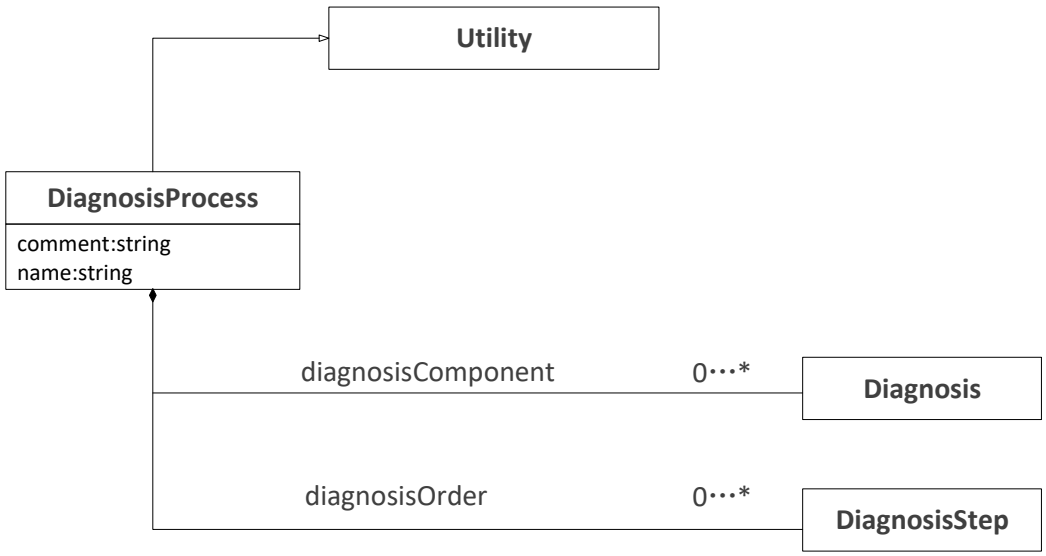


Figure 2-44

*`dianosisComponent`*: (0 or more object:Diagnosis) The set of diagnoses in the complete diagnostic decision-making process.

*`dianosisOrder`*: (0 or more object:DiagnosisStep) The ordering of the diagnoses in the complete diagnostic decision-making process.

### 2.10.10 DiagnosisStep

A step in a diagnostic decision-making process. The definition of **`DiagnosisStep`** is shown in Figure 2-45.

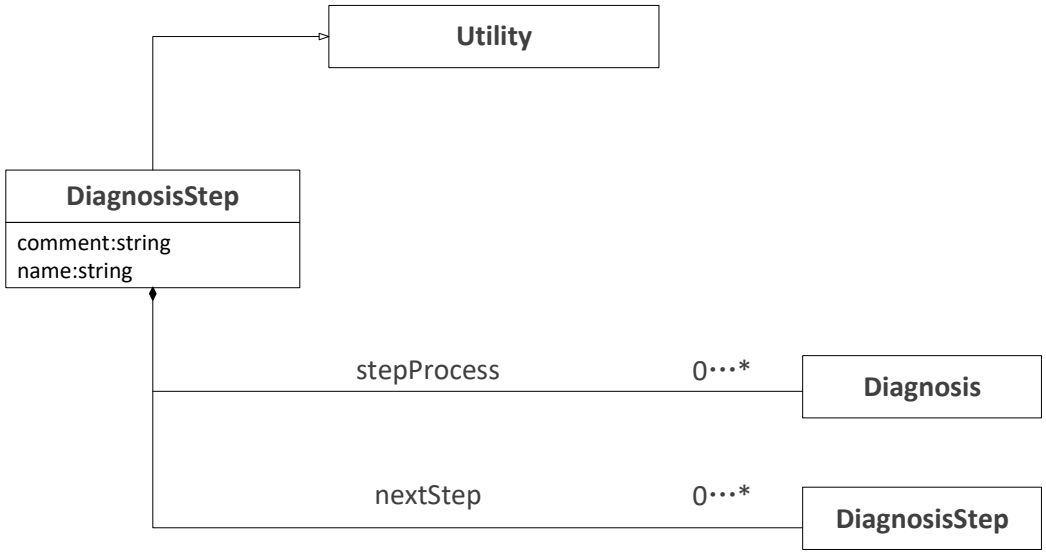


Figure 2-45

*stepProcess*: (0 or more object:Diagnosis) The diagnosis or diagnoses that is (are) made in the current step of the diagnostic decision-making process.

*nextStep*: (0 or more object:DiagnosisStep) The next step(s) of the diagnostic decision-making process. If there is no next step, this property is empty.

## 2.11 Data

**Entity** and **Utility** represent histopathological features of histopathology data, while **Data** stores metadata of the data files. Take digital slide as an example, the metadata include height, width, magnification, brand of the scanner and equipment settings used to capture the slide. **Data** has two sub-classes which are **Slide** and **PathologyReport**.

### 2.11.1 Slide

**Slide** stores metadata of a digital slide. Its definition is shown in Figure 2-46.

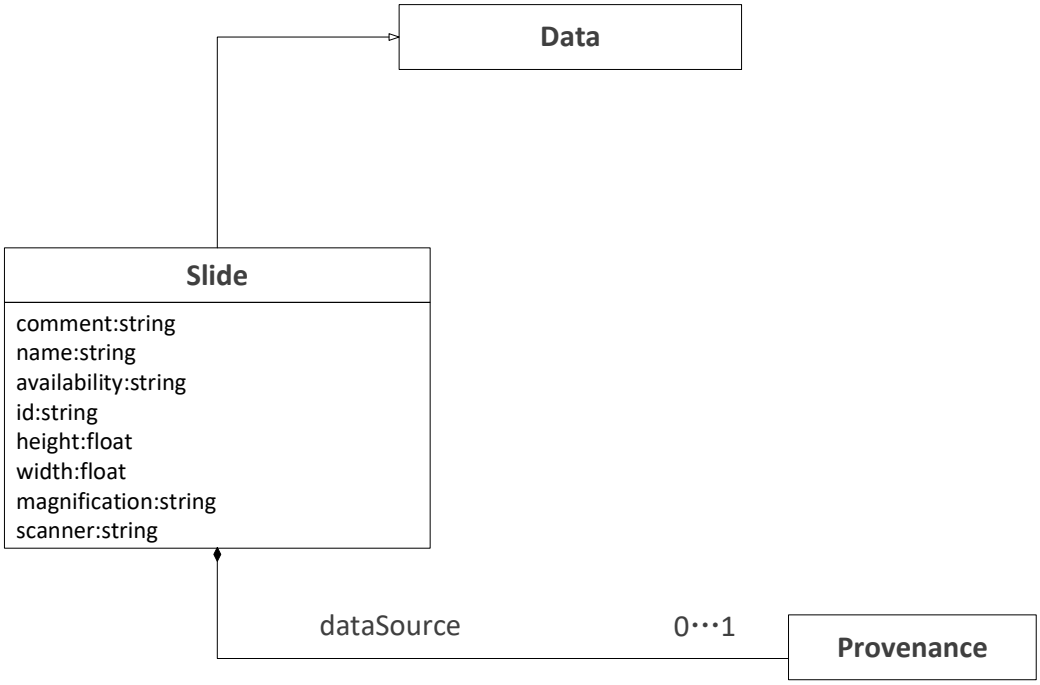


Figure 2-46

*dataSource*: (0 or more object:Provenance) A description of the source of this data such as a database name (The Cancer Genome Atlas).

`id`: (xsd:string) Identifier of this slide from the database.

`height`: (xsd:float) Height of the slide.

`width`: (xsd:float) Width of the slide.

`magnification`: (xsd:string) Magnification at which the slide was scanned.

`scanner`: (xsd:string) Model name of the scanner.

### 2.11.2 PathologyReport

**`PathologyReport`** stores metadata of a pathology report. Its definition is shown in Figure 2-47.

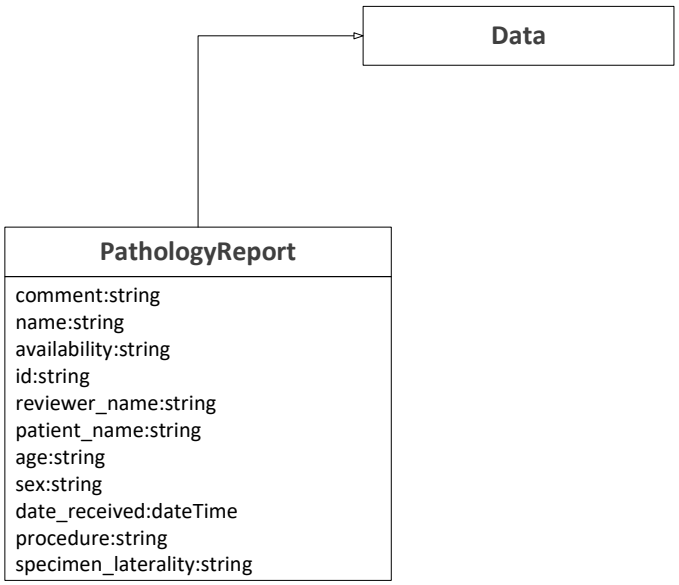


Figure 2-47

`id`: (xsd:string) Identifier of this report.

`reviewer_name`: (xsd:string) Name of the pathologist who wrote this report.

`patient_name`: (xsd:string) Name of the patient.

`age`: (xsd:string) Age of this patient.

`sex`: (xsd:string) Sex of this patient.

`date_received`: (xsd:dateTime) Received date of this patient.

`procedure`: (xsd:string) The specimen collection procedure such as partial nephrectomy.

`specimen_laterality`: (xsd:string) The laterality of the specimen (e.g. right, left or not specified).