Analysis of the NCI Thesaurus

Natasha Hoy, Stanford Medical Informatics

September 9, 2005

Abstract

After extensive discussions with NCI Thesaurus editors, we have analyzed the NCI Thesaurus and its use cases. We analyzed the version of the NCI Thesaurus in OWL that was automatically generated from the version developed in Apelon's TDE. We paid particular attention to the current modeling problems in the thesaurus, concerns of the editors for the completeness and usability of the representation, logical consistency and correct use of language primitives. The analysis includes recommendations for the overall structure of the Thesaurus, and specific recommendations, considerations, and questions for many of the parts in the Thesaurus. The analysis also includes discussion on the limitations of the Ontylog representation language vis-à-vis requirements of the NCI Thesaurus, implications of moving to OWL and Protégé, and limitations of the OWL language for NCI Thesaurus representation. We also suggest possible ways to overcome these limitations, and discuss the necessary tool support for that.

1 Background, methods, and motivation

We analyzed the OWL version of the NCI Thesaurus in order to understand the modeling problems that currently exist and to recommend possible solutions.

1.1 Methods

We have conducted meetings with editors responsible for various parts of the NCI Thesaurus to discuss issues specific to their particular domains. We had general meetings to discuss high-level issues and issues with the NCI Thesaurus as a whole. With each group, we have considered use cases, and the main questions that the users ask about the corresponding part of the thesaurus. We have discussed what hurdles in modeling the editors encountered, what are the things they wished they were able to express but cannot or do not express currently.

Furthermore, we conducted an informal analysis of the OWL version of the NCI Thesaurus in Protégé, noting problems, gaps, and inconsistencies. The analysis presented in this report contains suggestions, recommendations, different solutions options, as well as issues and questions to consider.

1.2 The version of the NCI Thesaurus used

We analyzed the OWL version of the NCI thesaurus, mostly considering the baseline version from May 2005. This version was generated automatically from the NCI Thesaurus developed in the TDE environment.

1.3 Summary of Recommendations

The document contains a number of recommendations, alternatives to consider, questions to address, tools to develop. These recommendations fall into several categories:

- improve or correct **the scripts for conversion** of the NCI Thesaurus from Ontylog DL to OWL with the goal of
    - using the OWL DL semantics correctly in the NCI Thesaurus
    - correcting some of the incorrect use of the Ontylog DL semantics (e.g., some vs all)
    - exploiting the greater expressiveness of OWL compared to Ontylog (separating restrictions into necessary and necessary&sufficient, using datatype properties, hasValue restrictions, etc.)
    - cleaning up some of the components of the NCI Thesaurus OWL representation that no longer appear necessary in OWL (e.g., use of Kinds)
- perform **post-processing** (or incorporating the actions into the conversion scripts) to improve the precision of modeling, including (but not limited to)
    - breaking up the NCI Thesaurus into mutually importing modules to facilitate reuse, improve the efficiency of classification, and so on
    - making domains and ranges of properties more precise
    - linking parts of the NCI Thesaurus to other upper-level or core ontologies, such as the Gene Ontology (for biological processes) or DOLCE
- develop **custom-tailored tool components** (such as Protégé plug-ins) to facilitate the editing of the NCI Thesaurus in OWL, to reduce the editors' effort, and to make editing less error-prone:
    - perform formal usability studies to determine which features of the tools the editors use most often and facilitate access to these features (limiting access to the features that are not used)
    - develop editing templates to reflect common ontology-design patterns for the NCI Thesaurus; it is conceivable that there can be a single template for each subtree (Kind) in the NCI Thesaurus, containing the properties used, correct types of restrictions, required restrictions, the break-down into necessary and necessary&sufficient conditions, and so on.
    - simplify the user interface to make editing easier (e.g., switch to infix notation for existential and universal restrictions, etc.)
- use **specific modeling solutions and patterns** to represent precisely the semantics of various components of the NCI Thesaurus; these solutions (or possible alternatives) are based on our analysis of use cases, discussions with editors, and analysis of the current representation in the Thesaurus. These specific modeling suggestions cover **the following categories** (Kinds):
    - Findings and Diseases (Section 5)
    - Genes and Gene Products (Section 6)
    - Anatomy and EO_Anatomy (Section 7)
    - Chemicals and Drugs (Section 8)
    - Molecular abnormalities and Abnormal Cells (Section 9)
    - Biological Processes (Section 10)

The recommendations and the sources of problems fall into several different categories:

1. Better understanding of some modeling issues and relations between different Kinds in the Thesaurus
2. Problems in the initial modeling in the TDE environment stemming from incomplete understanding or incorrect interpretation of the semantics if the Ontylog DL
3. Problems in the OWL file resulting from the incorrect or incomplete conversion script.
4. New requirements stemming from the larger acceptability and wider use of the NCI Thesaurus
5. Imprecise or incomplete modeling stemming from the limited expressivity of the Ontylog DL compared with OWL DL; additional opportunities for expressive power resulting from the adoption of OWL.
6. Requirements for the user interface to facilitate the editing of the NCI Thesaurus.

Many of the problems in items 1-3 can be fixed in the current TDE environment. Some of the new requirements (item 4), such as modularization are unlikely to be available in TDE. All of the issues in item 5 and 6 require transition to OWL.

2 Languages Semantics: Ontylog DL and OWL DL

In view of the possible transition to use OWL DL rather then Ontylog DL as the primary representation language for the NCI Thesaurus, we consider the features of the two languages that are appropriate for the representation needs of the NCI Thesaurus, as well as features that are lacking in either of the languages. We also sketch possible workarounds for the latter in the case of OWL. In summary, OWL DL is a more expressive language than Ontylog DL, and many of the additional features are in fact required for the proper representation of the information in the NCI Thesaurus dictated by the use cases and potential usage scenarios.

2.1 OWL Language features used (directly converted from the Ontylog DL)

At the basic level, there is a considerable overlap in the features of the Ontylog DL and OWL DL. Translation of many of these features, covered by the current conversion scripts is straightforward and is described elsewhere (Golbeck et.al. 2003)

- Classes, class hierarchy, multiple inheritance
- Annotation properties
- Domains and ranges of object properties
- Primitive and defined classes (although currently all restrictions are either necessary or necessary and sufficient – artifact of the Ontylog DL)
- Existential (*some*) and universal (*all*) restrictions (although not all of them are used correctly in the Ontylog DL version)

2.2 Limitations of the Ontylog DL for the representation of the NCI Thesaurus

A number of things that appear in the OWL file seem incorrect semantically, but they are artifacts of the more limited expressiveness of the Ontylog DL compared to OWL. Some

of these, at least in the first approximation, can be repaired automatically on translation from Ontylog to OWL. The automatic repair will take care of most of the cases (90%?). Some cases will need to be looked at manually. For each of the specific categories we discuss (Sections 7-10), we list the specific actions or changes to include in the translation scripts to make modeling of those categories correct semantically or more precise. Below is the list of some artifacts of the limitations of the Ontylog DL that appear in the OWL DL version of the NCI Thesaurus:

- **Definitional (necessary & sufficient) versus necessary conditions**: In Ontylog DL, all restrictions for a class must either all be necessary & sufficient (thus comprising a definition for a class) or all be necessary (thus making the class primitive). For example, in the definitions of Genes, the definition includes not only the organism, chromosomal location, and processes where the gene plays a role (indeed, definitional components of a gene), but also such non-definitional characteristics as links from genes to the corresponding gene products, abnormalities, or diseases. OWL does not have such a restriction and we should separate which properties constitute the definition of a class and which describe the necessary conditions. In most cases, this separation can be done automatically by a script since it will depend on the property itself, and we can usually define in advance the list of the definitional properties for each category (Kind). We consider specific suggestions on such separation in the sections corresponding to each category (Sections 7-10).
- **Negation**: There is no notion negation in Ontylog DL. As a result, the restrictions that by nature represent *lack of* something, encode this negative characteristic as part of the name. These properties all have the substring `Excludes` in their names (e.g., `Disease_Excludes_Finding`). There are currently 8 such properties. Semantically, these properties are negations of the corresponding properties with `Has` in their names (e.g., `Disease_Has_Finding`). However, this relation between the two corresponding properties is not made explicit in the NCI Thesaurus and cannot be used for classification or any other form of reasoning. We propose specific transformation that converts the properties with `Excludes` to the corresponding negations in OWL and discuss the implications of alternative representations (Section 5.4.1).
- **Restrictions for datatype properties**: In Ontylog DL, datatype properties (properties with strings or numbers as values) are not used for classification and their values are not inherited. These properties are essentially analogous to annotation properties in OWL. In Genes, for example, it would be desirable to represent chromosomal and physical locations as strings or XML Schema datatypes (Section 6.2)—it simply does not make sense to have a class such as `_13_102296421-102326346` (a physical location string). We also want these values to inherit to subclasses and to use them in classification.
- **`hasValue` restrictions**. The use of datatype properties also makes it natural to use `hasValue` rather than `some` restrictions with these properties (Section 6.2). There are no `hasValue` restrictions in Ontylog DL, however. We discuss the use of `hasValue` restrictions for Genes and the specific updates to the conversion script to use this restriction along with datatype property values in Section 6.2.

- **Unions**: There are no unions (`OR` operation) in Ontylog DL. Therefore, role groups, for example, are treated as intersections. Similarly, one cannot have multiple *sets of defining conditions* for a class if one does not have unions. We can use such multiple sets of defining conditions to define, for example, a gene by its Organism and location, or by its physical location (Section 6.2)

2.3 Features of Ontylog DL not directly present in OWL: Role Groups

There is a small number of Ontylog DL features that don't have direct equivalent in OWL. **Role groups** appears to be one feature that is used in the NCI Thesaurus representation and that does not have a direct OWL DL counter part. In future work, we will consider other features such as associations and modifiers.

Role groups are used in the NCI Thesaurus to represent the type of information illustrated in Figure 1. In the example in the figure, the role groups define some of the characteristics of the disease: each molecular abnormality corresponds to a specific cytogenetic abnormality; they are two phases of the same phenomenon. Therefore, we would like to group them together.

It is not clear what the formal semantics of role groups are, but within a single role group, it appears to be a simple AND of the restrictions within the role group. In other words, the necessary condition for the disease in Figure 1 is to have both cytogenetic abnormality `t_8_21_q22_q22` **`AND`** to have molecular abnormality `AML1-ETO_Fusion_Protein_Expression`.

However, the *only* available semantics in Ontylog DL for multiple role groups within the same class is that of **intersection** (**`AND`**). Therefore, the definition in Figure 1 implies, that all *four* conditions must be true for the disease. However, the *desired* semantcis is often to have **union (`OR`)**: we want to say that in the case of this disease we have

either

- cytogenetic abnormality `t_8_21_q22_q22` **`AND`** molecular abnormality `AML1-ETO_Fusion_Protein_Expression`

  `OR`
- cytogenetic abnormality `Structural_Chromosomal_Abnormality` **`AND`** molecular abnormality `Fusion_Protein_Expression`

This representation is readily available in OWL DL because OWL allows arbitrary mixture of intersections and unions in expressions. Therefore we can represent the desired expression in OWL as:

```
    Disease_Has_Cytogenetic_Abnormality some t_8_21_q22_q22
    AND
    Disease_Has_Molecular_Abnormality some
                        AML1-ETO_Fusion_Protein_Expression
OR
    Disease_Has_Cytogenetic_Abnormality some
                        Structural_Chromosomal_Abnormality
    AND
    Disease_Has_Molecular_Abnormality some
                        Fusion_Protein_Expression
```

While semantically correct and valid in OWL DL, such an expression would be rather onerous to create in the Protégé OWL plugin. Thus, role-group expressions are prime

targets for the templates representing common patterns to simplify editing. The template should make it easy to enter the role group and should translate the role group into valid OWL behind the scenes.

Acute_Myeloid_Leukemia_with_t_8_21_q22_q22_AML1_CBFa_ETO

Role group 1

Disease_Has_Cytogenetic_Abnormality: *t_8_21_q22_q22*
Disease_Has_Molecular_Abnormality: *AML1-ETO_Fusion_Protein_Expression*

Role group 2

Disease_Has_Cytogenetic_Abnormality: *Structural_Chromosomal_Abnormality*
Disease_Has_Molecular_Abnormality: *Fusion_Protein_Expression*

Figure 1. Role groups in the NCI Thesaurus: Two sample role groups in a definition of a disease. The Molecular abnormality corresponds to the Cytogenetic Abnormality.

2.4 Limitations of the Ontylog DL for the representation of the NCI Thesaurus

While much more expressive than Ontylog DL, OWL lacks a number of features that would be useful in representation of the NCI Thesaurus. For most, if not all of these features, we can instrument the tools to provide workarounds, post-processing capabilities (possibly through use of rules and rule engines, but not necessarily), or convenient access to circumvent the limitations. These limitations include lack of direct means to

- express **imprecise information** or information about what is typical, or common, or possible, but not always necessarily true
- represent **exceptions** (such as properties of genes that are changed in alleles)
- define **role chains** (e.g., representing a link from gene to gene product and then to the disease, but being able to query the link between the gene and the diseases, bypassing the gene product)
- access **inverses on restrictions** (e.g., representing that gene X plays role in a process Y by using an existential restriction on the class for gene X, but being able to query which genes play role in process Y, from the process point of view)
- represent **numeric ranges** in restrictions (e.g., feature present in 20 to 50 percent of cells)

We discuss these limitations in detail and recommend ways to circumvent them in the rest of this section.

2.4.1 Expressing imprecise, possible information in OWL

A number of properties, in particular in the Diseases hierarchy represent the information that is typical, or possible, for a particular disease, but is not necessary. Some of the subclasses may not exhibit this property at all, and in some this property may become necessary (These roles are *poss roles* in Ontylog).

In general, using existential restrictions for representing these typical roles may carry incorrect semantics. For instance, suppose we say that:

```
Astroblastoma:
     Disease_May_Have_Finding some Necrotic_Change
```

Such a statement replies that any instance of these disease has the instance of the finding as the value for the property `Disease_May_Have_Finding`. However, from the meaning of the property, not all instances of this disease have this finding, and, in fact, some of the subclasses may explicitly *exclude* this finding.
Given that the goal of the logical constraints in the NCI Thesaurus is *not* to provide diagnostic classification for patients, but rather to ensure logical consistency of the terminology, using existential restrictions for these "possible" properties does not cause a problem: We do indeed want to classify a disease that may have finding `subclass_of_X` as a subclass of the disease that may have finding `X` (given that all other conditions confirm the subclass relationship). We discuss these `poss` (in Ontylog terms) properties and their relations to properties for exclusion in detail in Section 5.4.2.

2.4.1 Exceptions

One of the problems editors encountered in modeling the NCI Thesaurus is inability to represent exceptions. Genes represent a good example where exceptions would have been helpful. In the Gene hierarchy, alleles are modeled as subclasses of the corresponding genes. In general, alleles inherit all the properties of the gene, but may occasionally change (override) essentially any of the gene's properties, from chromosomal location to process in which the gene plays a role, to roles in pathways. Ideally, we would like to specify the properties of the wild-type gene, and then represent the properties that are changed (overridden) for each allele. However, OWL does not support exceptions. We suggest alternatives for modeling the inheritance of properties for genes and alleles in Section 6.3. One of the possible solution is to introduce properties such as `Gene_`**`Typically`**`_In_Chromosomal_Location` that are inherited to alleles and do not get overridden. If the allele has a different location, we can use the property `Allele_In_Chromosomal_Location` to express this information. Hence, the allele definition will have both properties:
`Gene_Typically_In_Chromosomal_Location` and `Allele_In_Chromosomal_Location`, which appears to reflect correctly the intended semantcis. See Section 6.3 for more details and other alternatives.

2.4.2 Role chains

A useful notion in modeling is the notion of role chains. In fact, it addresses one of prime concerns of the modelers for Genes: For example, the definitions of Genes often include links from a gene to a gene product (`Gene_Encodes_Product`), and then from the gene product to molecular abnormality (`Gene_Product_Has_Abnormality`). However, the users are usually interested in the direct link from gene to the abnormality or from gene to a disease. (The link from gene to a disease can also be modeled indirectly through gene product, and, possibly, through a molecular abnormality). Ideally, the modelers would like to have the following situation (Figure 16):
Specify:

1. `GeneX`:
   `Gene_Has_Product` **`some`** `ProductY`

2. `ProductY:`
   `Product_Has_Abnormality` **some** `AbnormalityZ`

Infer:

`GeneX:`
`Gene_Has_Abnormality` **some** `AbnormalityZ`

We can specify such inference as a rule in a rule language such as SWRL and then use a rule language to perform the additional inference. However, it appears that there is only a small, well specified number of cases in the NCI Thesaurus where such inference is required. Therefore, a more practical solution for the modeling of the NCI Thesaurus could be to have a special pattern to specify role chains and then to use a simple post-processing script to infer the additional restrictions. We discuss this solution, as well as other alternatives in detail in Section 6.4.

2.4.3 Inverses on restrictions

OWL has the notion of inverse properties. For example, we can declare the following two properties form the NCI Thesaurus to be inverse properties using `owl:inverseOf`[1] property:

- `Gene_Encodes_Product`
- `Gene_Product_Encoded_by_Gene`

Using the semantics of `owl:inverseOf`, we can make the following inference:

`GeneX Gene_Encodes_Product ProductY` ⇒
`ProductY Gene_Product_Encoded_by_Gene GeneX`

However, this inference is true only for *instances*: in other words, if we have a statement for the property value for a specific instance of a gene, we can fill in the gene for the corresponding product. However, the NCI Thesaurus does not represent instances. The relationship between classes is expressed primarily through existential restrictions such as:

`GeneX:`
`Gene_Encodes_Product` **some** `ProductY`

This restriction *does not* imply the inverse restriction:

`ProductY:`
`Gene_Product_Encoded_by_Gene` **some** `GeneX`

In fact, the first restrictions says that every instance of `GeneX` must encode some instance of `ProductY`. However, it says nothing that would be applicable to every instance of `ProductY`. It leaves open the possibility that some instances of `ProductY` are not encoded by any of the `GeneX` instances. The second restriction however states exactly that: every instance of `ProductY` is encoded by some instance of `GeneX`. So, lack of inverses of restrictions in OWL has sound semantics. However, in modeling the Thesaurus, we usually do want to conclude that the inverse is also true: it so happens that in the specific cases encoded in the Thesaurus, the "inverse restriction" also happens to hold. Not only that, but also many use cases *require* access to information from both directions: some use cases involve queries for genes given a disease and others query diseases associated with a given gene.

[1] http://www.w3.org/TR/owl-ref/#inverseOf-def

The Protégé user interface provides modelers with access to this information in both direction through the “find usage” button. Figure 2 gives an example of how this information can be accessed: From the definition of a gene, we can find what classes in the ontology reference this gene in their restrictions. The result will include gene products that have a restriction on the property `Gene_Encodes_Product` with our gene of interest as a value. Note that the link between the gene product and the gene is specified in the ontology only in one direction.

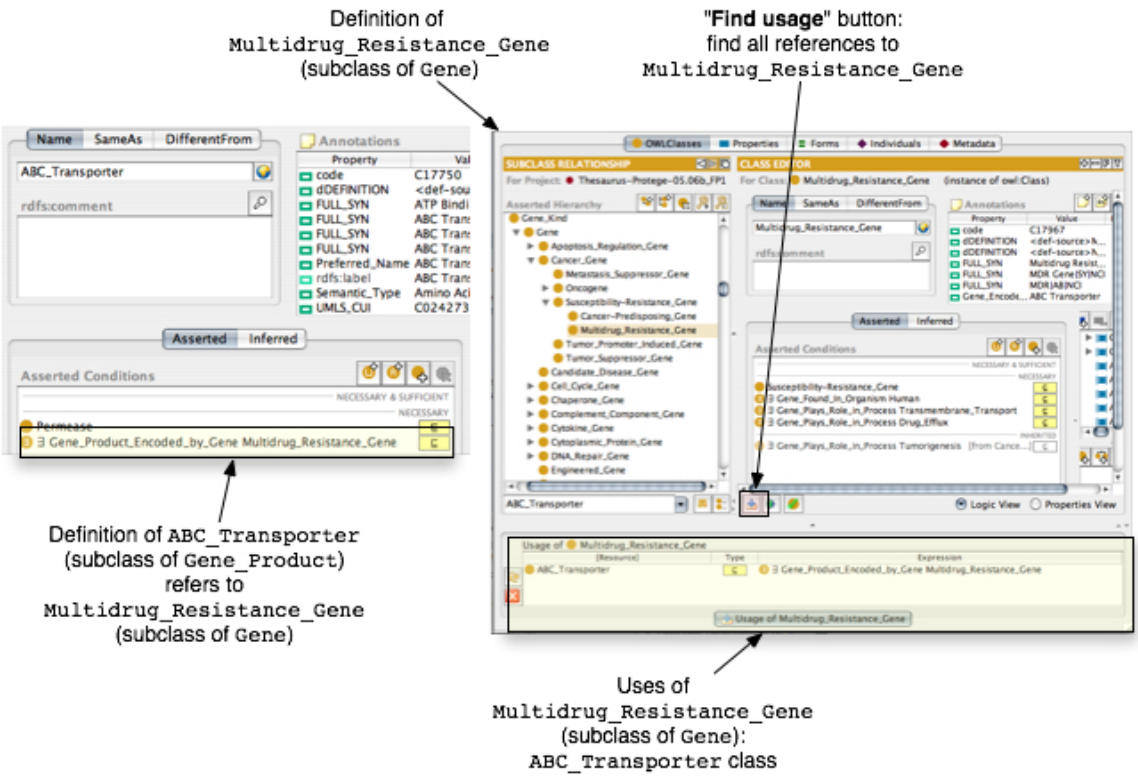


Figure 2. Finding inverse restriction links in Protégé. From the definition of a gene (iin this case, `Multidrug_Resistant_Gene`), we can find definitions of gene products where this gene is referenced,

2.4.4 Numeric ranges in restrictions

Lack of the ability to specify numeric ranges (e.g., saying that a teenager is a person whose age is between 13 and 19) in OWL has been notes numerous times. The NCI Thesaurus also requires the use of numeric ranges to represent some of its concepts.

Consider, for example, a class such as `Bone_Marrow_Dysplasia_Present_in_50_Percent_or_More_of_the_Cells_of_Two_Cell_Lines`.[2] This class inherently contains a numeric range in its definition: it is a value range for a property representing the percentage of specific cells

[2] Definition of this concept also raises other issues such as the use of n-ary relations (http://www.w3.org/TR/swbp-n-aryRelations/) and in general the need for such detailed classes in the NCI Thesaurus. We defer this discussion until later.

present in cell lines. And the restriction on this property must include a numeric range restriction. We have recently added the ability to represent numeric ranges to Protégé OWL by using user-defined XML schema datatypes (http://protege.stanford.edu/plugins/owl/xsp.html). The solution is conformant with OWL and RDF. However, there is no document yet from W3C describing the best way to represent numeric ranges in OWL, although one is in the works.[3] When such solution emerges, we will update the handling of numeric ranges in Protégé OWL to conform with the solution endorsed by W3C.

3 Tool Requirements

While Protégé OWL is currently able to handle many of the requirements of the NCI Thesaurus, significant improvements are needed to make the editing really smooth and easy for modelers. These improvements include development of special editing templates and wizards that facilitate entering of specific ontology patterns used in the NCI Thesaurus; improvements to scalability of classification; and simplification of the user interface in general.

3.1 Usability analysis and custom-tailored interface

In order to develop an interface that is custom-tailored to editing the NCI Thesaurus, we suggest to start with a **formal usability and requirements analysis** of the NCI Thesaurus editing process. This analysis would involve hiring a usability expert who would observe the editing process, interview the editors and come up with suggestions for the user interface. Such suggestions oculd include but are not limited to:

- a set of language **features** used by the NCI Thesaurus editors: these features could be made easier to use and other features could be hidden or disabled. Such approach would not only allow editors to focus on the features that they need, but also would serve as rudimentary quality control as the editors would not be able to use the features they are not supposed to use. This analysis would also determine which language features are not used by the editors. These features can be disabled in the custom-tailored interface.
- A set of ontology **patterns** that are common during editing and that are specific to the NCI Thesaurus. These patterns could be specific not only to the Thesaurus as a whole, but also to specific categories (Kinds). For instance, an editor editing the Gene hierarchy, could get a pre-defined list of necessary and sufficient conditions to fill in (such as physical location, organism, etc.) and a set of features for necessary conditions. In fact, we observed that we can define such a pattern essentially every category in the NCI Thesaurus. The patterns would serve dual role: they would simplify the editing process and make it more efficient. They would also reduce the number of errors and omissions since editors would get prompts on what they need to fill in.

The Protégé architecture is very amenable to these sorts of extensions and a tab plugin can be developed to implement the suggestions resulting from this analysis.

[3] http://lists.w3.org/Archives/Public/public-swbp-wg/2004Apr/0125.html

3.2 Templates and wizards for ontology patterns

Based on the results of the usability analysis suggested in Section 3.1, we suggest developing templates and wizards that support the editing of the specific ontology patterns identified during the analysis. In fact, we can develop a template for each of the categories in the NCI Thesaurus, as well as for such structures as role groups, numeric ranges, and so on. The editors will then be guided through the process of defining a new class in a specific hierarchy, requested to enter information that is mandatory, prompted to enter optional information, etc. Such template could also, for example, hide that an editor is creating an existential restriction and simply ask the editor to enter a property-value pair that will be translated into the restriction behind the scenes.
Furthermore, we can develop new wizards, such as the one developed by the CO-ODE group in Manchester for entering bulk information.[4]
As a research issue, we can identify specific patterns and develop some general mechanism to provide convenient user interface to fill in instances of these ontology patterns.

3.3 Classifying parts of the NCI Thesaurus

The sheer size of the NCI Thesaurus makes classification difficult and requires significant computational resources. In a later section (Section 4.1), we discuss the option of breaking up the NCI Thesaurus into multiple ontologies that mutually import one another.

However, currently, if an ontology with imports is sent to a classifier, all the imported ontologies are also loaded into the classifier and considered during classification. This approach is required for complete classification of the ontology with imports. However, there could be intermediate debugging steps that don't include imports in classification. For instance, if a concept is inconsistent even when imports are not included, it will continue to be inconsistent once imports are brought in. Therefore, these inconsistencies could be flagged and repaired with limited and more efficient classification.
Note however, that inferred classification (re-treeing of concepts) is very dependent on the imported ontologies. The inferred hierarchy without imports is not indicative of how a hierarchy with imports will look like. The limited classification would be helpful to identify inconsistent concepts, however.

3.4 Simplification of the user interface

OWL is a complex language and a generic interface for OWL is necessarily complex. However, even in a generic interface, many simplifications are possible. Some of the simple changes that have been brought up in the discussions with NCI modelers include the use of infix notation for existential and universal restrictions and simplifying the interface for inherited properties.

3.4.1 Use infix rather than prefix notation for restriction

The infix notation is much more intuitive for many users. In the restriction in Figure 3,

[4] http://www.co-ode.org/downloads/wizard/co-ode-index.php

the modifier **some** (□) actually modifies the value `Bone`, rather than the property `Antomical_Structure_Is_Physical_Part_Of` that it is directly adjacent to. The next release of Protégé (version 3.2beta) will include the syntax renderers developed by the CO_ODE group in Manchester and will use this syntax as a default for OWL ontologies. Figure 4 demonstrates this syntax: not only it uses infix notation, but also it uses words such as **some** and **all** rather than logic symbols □ and ∀, as well as **and** and **or** instead of ⊓ and ⊔ to represent the restrictions. We believe that this syntax would be easier for users to understand.

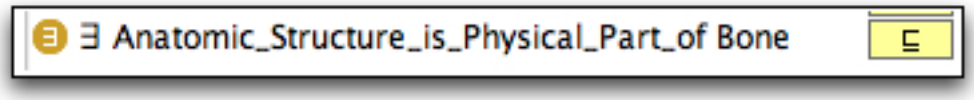


Figure 3. An existential restriction for the class Bone_surface using the prefix notation. The `some` modifier (□) actually applies to the value of the restriction (`Bone`) rather than to the property `Antomical_Structure_Is_Physical_Part_Of`.

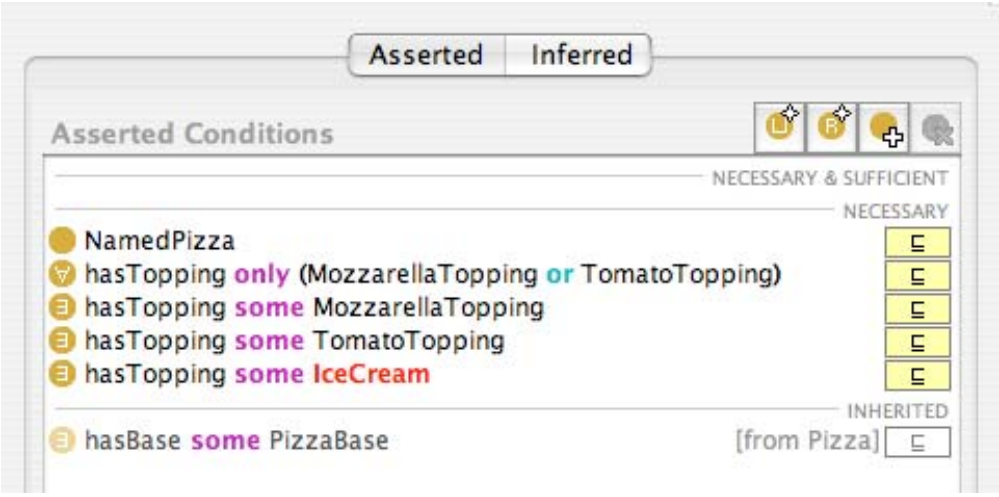


Figure 4. New "Manchester syntax" using infix notation.

3.4.2 Simplification of the interface for inherited restrictions

Many classes in the NCI Thesaurus inherit a large number of restrictions, often from multiple parents. Restrictions on a single property often subsume one another, and only one of the restrictions—the most specific one—is actually relevant to the class in question. Consider the example in Figure 5. There are five universal restrictions for the Property `Disease_Has_Abnormal_Cell`.[5] However, only one of the restrictions—the one that restricts all values to instances of `Neoplastic_T-Lymphocyte_and_Neoplastic_Natural_Killer_Cell`—is required. It subsumes all the other restrictions because:

- `Neoplastic_T-Lymphocyte_and_Neoplastic_Natural_Killer_Cell` is a subclass

[5] As we will discuss later, a universal restriction may not be appropriate here. However, the same argument applies if the restriction was an existential one.

of `Neoplastic_Lymphocyte`

- `Neoplastic_Lymphocyte` is a subclass of both, `Abnormal_Lymphocyte` and `Neoplastic_Hematopoietic_Cell`

Because the semantics of multiple universal (or existential) restrictions is intersection, only the restriction to the most specific class is of value and encompasses the other ones. It would be desirable if the tool could spot these additional restrictions and provided the option of hiding them.

We expect that other similar simplifications will be identified in the future.

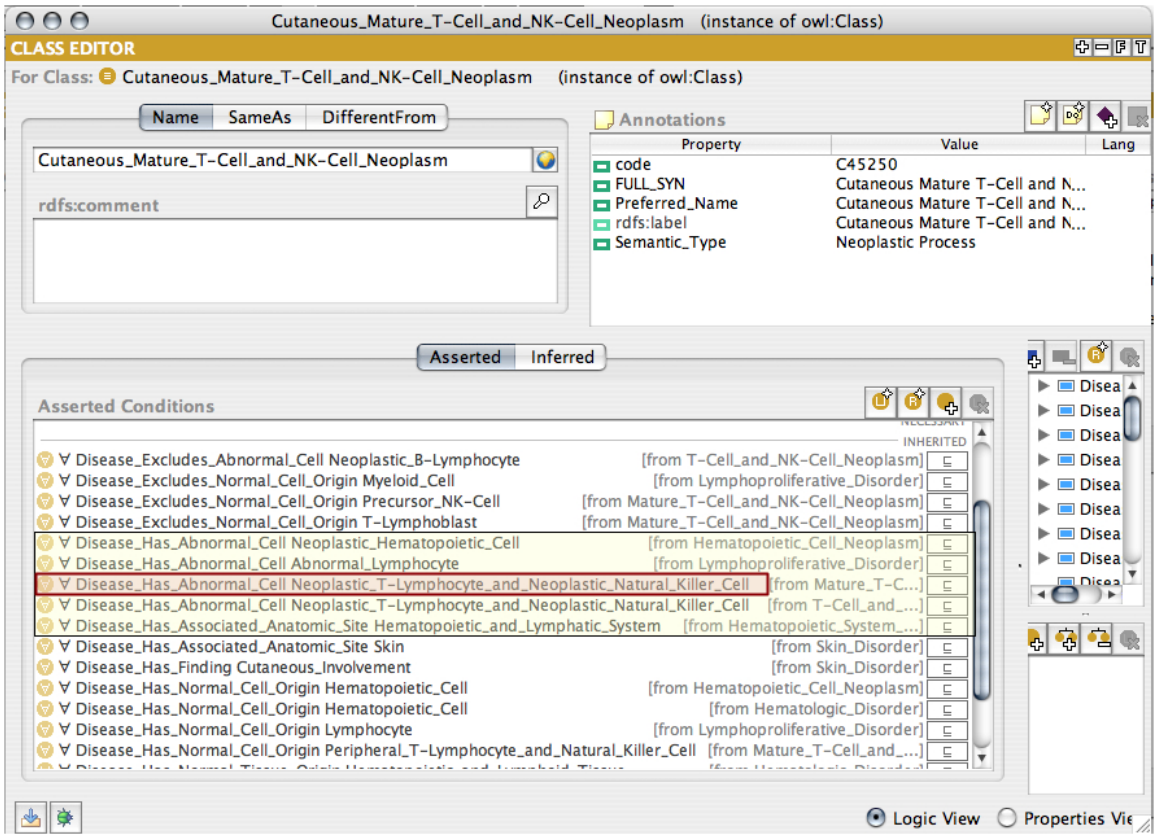


Figure 5. Inherited restrictions for the class Cutaneous_Mature_T-Cell_and_NK-Cell_Neoplasm. Only one of the five highlighted restrictions for the property Disease_Has_Abnormal_Cell is relevant for this class; it subsumes the other four.

4 Top-level Structure of the NCI Thesaurus

Before addressing specific modeling issues in suggestions for each of the categories (Kinds) in the NCI Thesaurus (Sections 7-10), we consider the overall high-level structure of the NCI Thesaurus, and suggest improvements and changes to consider. These changes include:

- division of the NCI Thesaurus into modular subontologies (Section 4.1)
- eliminating the extra level of Kinds in the OWL version of the NCI Thesaurus (Section 4.2)
- making domains and ranges of properties more specific (Section 4.3).

As we consider the structure of the NCI Thesaurus as a whole and the structure of its categories, we are driven primarily by the use cases, as well as the following consideration: The **main goal of inference on the NCI Thesaurus** is to ensure internal consistency of the Thesaurus itself. The definitions and restrictions are not intended to be used for example for making a diagnosis. Classification is used to ensure internal consistency, not to support inference by outside applications. In other words, for example, a definition of a disease and its necessary and sufficient conditions should ensure the correct place of the disease class in the inferred hierarchy and not to make a diagnosis of a specific patient with specific conditions by classifying a patient under a particular disease. The NCI Thesaurus represents the current understanding of this concept; useful information about it.

4.1 Division of the NCI Thesaurus into subontologies

The Thesaurus is currently one monolithic OWL ontology. We recommend that NCI considers modularization of the Thesaurus into subontologies that mutually import one another. There are several advantages to the modularization:

1. performance of classification;
2. tool performance during editing;
3. modularized editing and debugging;
4. ability to reuse parts of the NCI Thesaurus and to have NCI control the modules in which such partial reuse is performed.

Performance of classification

Some of the NCI Thesaurus subtrees don't depend on the whole NCI Thesaurus for classification. If they are separated into separate modules, these modules, or collections of dependent modules can be sent to a classifier separately, thus vastly improving performance. Figure 6 shows the analysis of the current state of the dependencies between the NCI Thesaurus subtrees. The subtrees above the first line are the ones that don't have any dependencies to other subtrees (there may have internal dependencies). Each of these can be classified separately. Admittedly, many of these don't have many restrictions in them and classification would be rather trivial. However, if there are any inconsistencies there, you could get the result much faster.

At the next level is the `Findings_and_Disorders` subtree. It depends only on four subtrees at the top level (`Anatomy`, `Molecular Abnormality`, `Abnormal Cells`, `Properties and Attributes`). Therefore, it needs to import only these four subontologies. When `Findings and Disorders` is then sent for classification, only the 5 subontologies need to be analyzed. At the next level are `EO findings and disorders`, that depends on `Findings and Disorders`, and one additional module from the top level (`EO Anatomy`). Again, this smaller part of the NCI Thesaurus can be sent to the classifier. The bottom level contains six classes that mutually depend on each other and on many other modules in the Thesaurus. To classify any of these six modules, the whole NCI Thesaurus will need to be classified (see, however, Section 3.3 for being able to determine inconsistencies in a single module, without analyzing the modules it imports).

Editing performance:

Any time we can load a smaller module or access a smaller module when using a *file* backend, the loading time would be significantly shorter

Modularized editing and debugging

Software-engineering community has over the years seen the benefits of modularized software development. Nowadays, no professional software engineer would write a whole application, or a significant part of it as a single class or a single method. Modularized software is much easier to write and to debug.

Reuse of Thesaurus parts by other parties

One of the complaints about the Thesaurus we often hear (and the same one we have often heard about the Foundational Model of Anatomy) is that it is very hard to reuse because it comes in one single extremely large piece. Many users don't need all of its components, and would like to reuse only some of them in tehir applications or ontologies. The breakdown into separate modules that import one another rprovides a natural division of the Thesaurus, enabling users to import only the components that they need. At the same time, NCI controls the granularity and the content of the components. In the absence of such division coming from NCI, users are likely to use ad-hoc methods to extract the parts that they need. This approach would lead to proliferation of differently divided and structure subparts of the Thesaurus, with mo coherence, or control from the NCI. Providing modules that are easily reusable would largely remove this problem.

How do we break up the Thesaurus into modules?

The simplest way to break up the Thesaurus into modules is to put each current Kind into a separate modules, with the import structure reflecting the structure in the diagram in Figure 6.

Another alternative is to use the modularization tool, such as the tool developed by Bernardo at the MindSwap lab. This tool breaks up the NCI Thesaurus to minimize the so-called eConnections (Grau 2004)—the special type of connections used by the Pellet reasoner to optimize reasoning over multiple ontologies. The algorithm breaks the Thesaurus along the lines close to breaking it up into Kinds. The possible disadvantge of this approach could be that it is too tied to the Pellet reasoner in terms of optimizing reasoning performance. At the same time, there is already an experimental DIG interface to Pellet available, and in theory, one can run Pellet from Protégé:
http://lists.mindswap.org/pipermail/pellet-devel/2005-July/000336.html
If this implementation also makes use of eConnections, then we can utilize the modularization optimized for eConncetions in the Thesaurus.

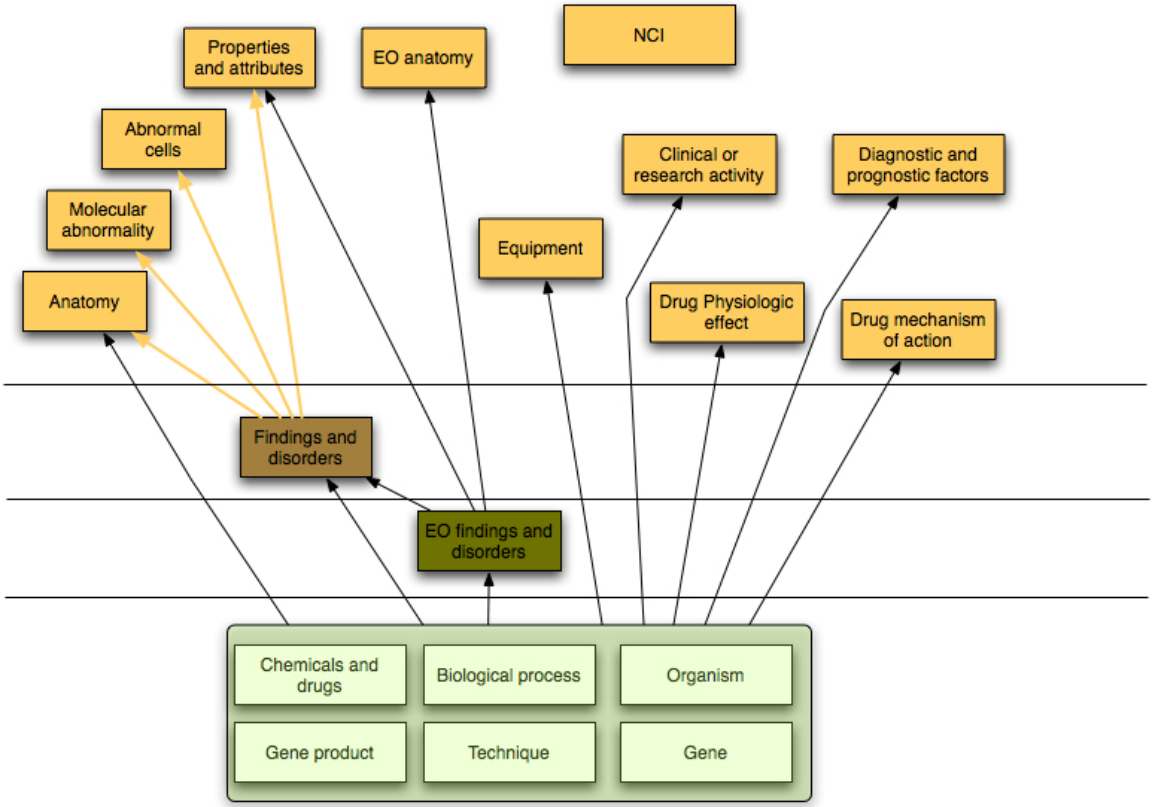


Figure 6. Dependency between Kinds in the Thesaurus based on the roles connecting them. Kinds in the top tier do not depend on any other kind and do not have to import any other ontologies. The Kind in the second tier (Findings and disorders) needs to import only four ontologies from the top tier. The Kind in the third tier, imports Findings and disorders and one more top-tier ontology. The Kinds in the bottom tier mutually depend on one another and cannot been separated. They also need to import all the higher-tier ontologies, except NCI Kind.

4.2 Kinds in the OWL File

Kinds are used in the TDE environment for several purposes:

1. declare disjointness between different branches
2. provide domains and ranges for roles

In the OWL file, the Kinds carry essentially no information, with a couple of small exceptions below. The two purposes above that Kinds are used for in TDE can be achieved simply by making the top-level classes in each branch subclasses of owl:Thing and removing the classes corresponding to Kinds altogether. This step would simplify the top-level structure of the OWL file. As far the user of the OWL version of the NCI Thesaurus is concerned, the presence of Kinds does not add anything to the Thesaurus but creates confusion.

Specifically, removing the Kinds would involve the following steps (all of these should be done in the conversion script that creates the OWL file);

- Delete the Kind classes
- Make the top-level classes to be subclasses of owl:Thing

- Declare disjointness over the top-level classes
- Given the one-to-one correspondence between Kinds and the top-level classes, replace the domain and ranges of properties with the corresponding top-level classes
- Ensure that the Kind definitions become definitions of the top-level classes (It is already the case for most classes)

The following special case should be handled separately. There is one Kind that doesn't have one-to-one correspondence with a top-level class: NCI_Kind. It has two subclasses: Conceptual_Entities and NCI_Administrative_Concepts. The following solution is possible:

- Create a new top-level class: NCI_Concept (or a similar name)
- Make this class a superclass of Conceptual_Entities and NCI_Administrative_Concepts
- Make this class a domain and range of the property Conceptual_Part_Of—the only property where NCI_Kind is involved

4.3 Domains and ranges of properties

Currently, domains and ranges of properties are limited to Kinds (or will be to roots of the corresponding subtrees). In many cases, however, domains and ranges could be more specific. For instance, `Gene_Product_Expressed_In_Tissue` can have `Tissue` as range (rather than `Anatomy`), `Technique_Has_Target_Protein` can have `Protein` as its range (rather than `Gene_Product`), etc. Assigning these more specific domains and ranges to properties has at least two advantages for the quality of editing:

- **During editing**, editors are given fewer options when specifying restrictions. For instance, only properties that have a specific class in the domain will appear in the list of properties to restrict. Therefore, for instance, allelic properties will appear only for Allele classes and not for Genes
- **After editing and during classification**, restrictions that violate these more specific domains would cause wrong hierarchical inferences and therefore, the editors will be able to identify the mistake. For instance, if one restricts the value of the property `Gene_Product_Expressed_In_Tissue` to something other than a `Tissue` (let's say a `Cell`), that `Cell` would be classified under `Tissue` and this wrong classification would point out the erroneous restriction. There can also be an option in the user interface to prevent creation of such restrictions in the first place. Note that in general in OWL, you can create an existential restriction for a property with the value that is outside of the property's declared range: such declaration will cause the restriction value to be classified as a subclass of the range. Therefore, an option to restrict the allowed values for restriction can be a convenient configurable shortcut, but not a universal editor feature.

4.4 Summary of suggestions for the top-level structure of the NCI Thesaurus

- Consider splitting the NCI Thesaurus into subontologies, corresponding to Kinds
- Remove the classes for Kinds from the OWL version of the NCI Thesaurus during the conversion from Ontylog to OWL

- Consider having more specific domains and ranges for properties.

5 Findings and Diseases

[*There is a document **NCI Thesaurus Disease Model** that covers many of the issues. Here we simply summarize them and address the difficult/controversial ones, or the ones not raised in the document*]

5.1 Goals of the Disease Model

[*From the **NCI Thesaurus Disease Model** document*]
The model of neoplasms and related diseases addresses two basic needs:

1. To define, code, and retrieve neoplasms according to their essential aspects and criteria; and
2. To represent other associations important for clinical or research purposes, including normal values, prognostically significant features, and important diagnostic criteria found in only some cases.

[*end quote*]

Part of the goal is also to provide clear definitions and descriptions of cancers, so that people use those terms consistently. In theory, they can be used to help with diagnosis, particularly for rare forms of cancers. However, in general, it is not the goal of disease definitions in the NCI Thesaurus to serve as the basis for a classification system for diagnosis.

5.2 Links between diseases and other categories

In the NCI Thesaurus, definitions of diseases contain links to some of the other categories in the NCI Thesaurus, mainly:

- `Anatomy`
- `Molecular_Abnormality`
- `Abnormal_Cell`

There are links *to* diseases from other categories, such as genes and gene products. The use cases involving diseases exercise links in both directions. For instance:

- You have some therapy that targets a particular protein (`Gene_Product`), find out which cancers could be treated with that. The only explicit link in the Thesaurus that will help in this query is from `Gene` product to `Disease` (`Gene_Product_Associated_With_Disease`)
- Given a specific disease, find out diagnostic criteria and likely outcomes – for this, we can follow the direct links in the definition of disease
- Given a gene, find which cancers it is involved in and which other genes and gene products could be involved (for research purposes). Involves following the link from gene to disease, and then reverse links from disease to other genes.

Diseases therefore are primary use case for “inverses on restrictions” (see Section 2.4.3): we need to be able to traverse both, restrictions asserted directly at the disease class, and the ones that have the class as their value.

5.3 Defining Roles

The following roles are currently defining roles for `Diseases` and should become necessary&sufficient conditions, with existential restrictions. Other role restrictions are necessary conditions. The roles below are organized in the role hierarchy.

- `Disease_Has_Associated_Anatomic_Site`
  - `Disease_Has_Primary_Anatomic_Site`
  - `Disease_Has_Metastatic_Anatomic_Site`
- `Disease_Has_Normal_Cell_Origin`
- `Disease_Has_Normal_Tissue_Origin`
- `Disease_Has_Abnormal_Cell`
- `Disease_Has_Molecular_Abnormality`
- `Disease_Has_Cytogenic_Abnormality`
- `Disease_Has_Finding`
- `Disease_Has_Associated_Disease`
- `Disease_Is_Stage`
- `Disease_Is_Grade`

5.3.1 Exceptions to defining roles

In general, it will be possible to decide whether or not a property is definitional based on its name and to use this knowledge in the conversion script. However, there are some cases where whether or not the property is definitional will depend on the value of the property. For instance, some findings may be definitional of a disease and some may not. Therefore, at some point, this determination will need to be done by hand, probably after the transition to Protégé OWL as the primary editing environment.

5.4 Properties of Diseases

We consider several issues with representing properties of diseases. These issues include:

- properties implicitly representing a negation (i.e., properties with `Excludes` in the name)
- properties representing values that are possible and important to note but not always present (i.e., properties with `May_Have` in the name)
- properties for representing anatomic sites of diseases and the use of suproperty hierarchy there.

5.4.1 Properties with Excludes. Negation.

There is a number of properties with "`_Excludes_`" in their name that are used because Ontylog DL does not have negation. All these properties appear to be definitional. Because OWL and Protégé have negation, it will make sense to translate these properties as negations of the corresponding "`_Has_`" properties. Consider the definition in Figure 7, for example. We can replace the highlighted properties with `_Exclude_` in the name with the corresponding negations. To define the negation, we create a new restriction on the property:

```
not (Disease_Has_Abnormal_Cell some
                    Neoplastic_Ganglion_Cell)
```

Figure 8A shows this representation in Protégé OWL.

Note that it might be tempting to express the negation above in way that looks very similar:

```
Disease_Has_Abnormal_Cell some
                        (not Neoplastic_Ganglion_Cell)
```

However, the semantics of the second expression are different than the semantics of the first one: In the first case, we are saying that the property `Disease_Has_Abnormal_Cell` does not have a value that is `Neoplastic_Ganglion_Cell`. We are not saying anything about whether or not this property must have any value at all. Thus, a class that has no value for the property `Disease_Has_Abnormal_Cell` satisfies this restriction. In the second case, we are saying that there *must be* a value for the property `Disease_Has_Abnormal_Cell`, but this value must be something other than `Neoplastic_Ganglion_Cell`. Thus, a class that has no value for the property `Disease_Has_Abnormal_Cell` does not satisfy this restriction.

However, if the class definition already has another existential restriction already requiring that there is at least some value for the property `Disease_Has_Abnormal_Cell`, as is the case in Figure 8A, we *can* put the negation inside the value of the existential restriction, since a class with no value for the property `Disease_Has_Abnormal_Cell` is not going to satisfy the definition anyway (Figure 8B)

It appears that in the NCI Thesaurus, for every property with `_Excludes_`, there is a corresponding `_Has_` property in the definition of the same calss. If this is really the case universally, then we can choose either of the two ways to specify the negation. The decision then will probably be driven by what is more convenient to the editors.

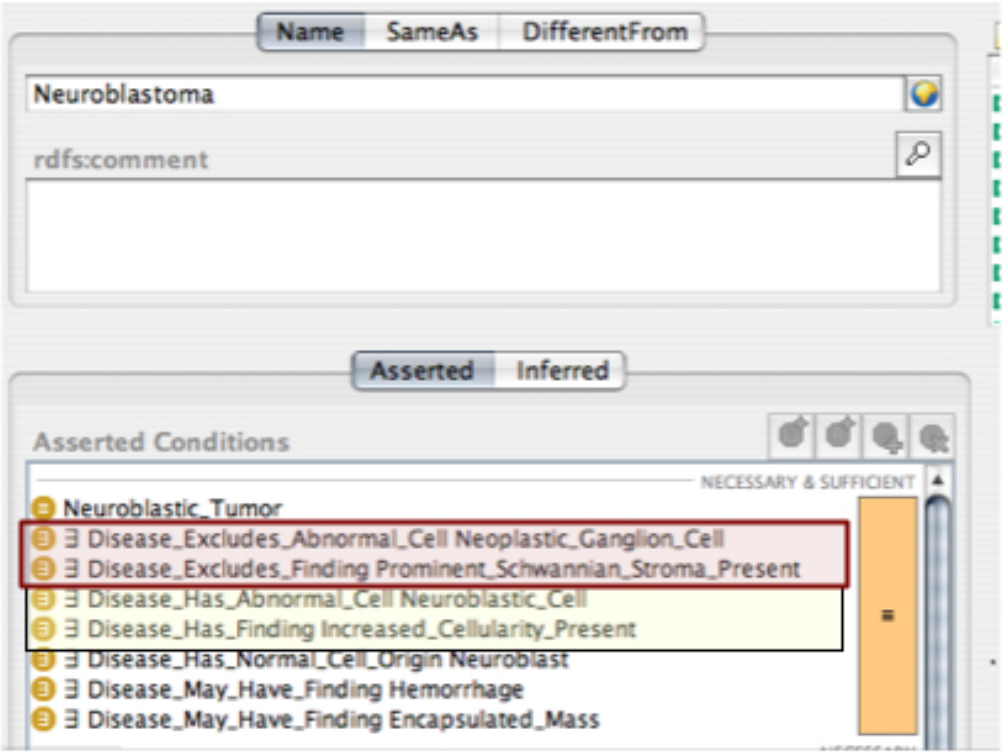


Figure 7. Definition of `Neuroblastoma` that has properties both `_Excludes_` and the corresponding `_Has_`. (The example in the figure was changed from the version in the NCI Thesaurus to contain only existential restrictions)

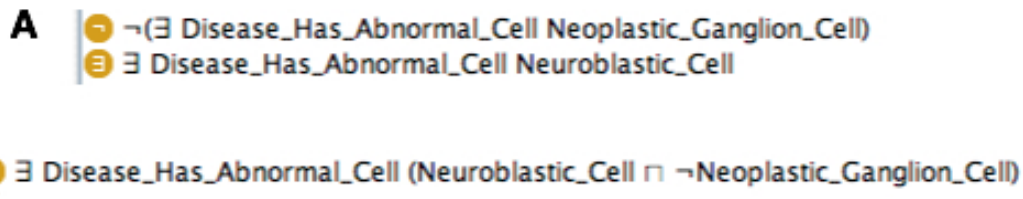


Figure 8. Two ways to model `_Exclude_` roles with negation. A. We put negation outside the whole restriction saying that it is not true that such a value exists. B. We put negation on the value itself, saying that the value does exist, but it must not be `Neoplastic_Ganglion_Cell`. In the case in the figure (where there is also a "positive" restriction on the same role), the two alternatives are semantically equivalent. In general, however, simply putting negation inside the restriction "`(property` **`some`** `(`**`not`** `X))`" is not equivalent to saying that "**`not`** `(property` **`some`** `X)`"

5.4.2 Roles with "_May_Have_"

Certain features are typical characteristics of some kinds of cancers. These features often occur when the cancer is present (e.g., possible outcome), but some instances of this cancer may not have this feature. These features are identified with properties that have "`_May_Have_`" in their name:

- `Disease_May_Have_Abnormal_Cell`
- `Disease_May_Have_Associated_Disease`
- `Disease_May_Have_Cytogenetic_Abnormality`
- `Disease_May_Have_Finding`
- `Disease_May_Have_Molecular_Abnormality`
- `Disease_May_Have_Normal_Cell_Origin`
- `Disease_May_Have_Normal_Tissue_Origin`

Furthermore, a class high in the hierarchy may have a particular feature as a typical feature, but some of its subclasses may not exhibit this feature at all, and, in fact, explicitly exclude this feature; conversely, they may have this feature necessarily for all instances, rather than have it as a typical feature.

Currently, for each of these types of features, there are three roles defined in the NCI Thesaurus:

1. `Disease_May_Have`
2. `Disease_Has`
3. `Disease_Excludes`

As discussed in Section 5.4.1, `Disease_Excludes` should be replaced with negation of `Disease_Has` in OWL.

`Disease_Has` can be a subproperty of `Disease_May_Have`. This solution would handle the case where the feature is typical or optional at the higher level of the hierarchy, but is necessary at the lower levels.

Using the property hierarchy for `Disease_May_Have` as a superproperty and `Disease_Has` a subproperty requires that we give special thought to the representation of negation for `Excludes`. We've discussed earlier (see Section 5.4.1) the implications of the different ways to represent negation for the existential restrictions (with negation on the whole restriction or negation on the value of the restriction). Depending on which approach we take, implications will be slightly different here. Note, that given definition of subproperties, the following implication is true:

```
Disease_Has_Finding some X ⇒
     Disease_May_Have_Finding some X
```

(because `Disease_Has_Finding` is a subproperty of `Disease_May_Have_Finding`)

For the same reason,

```
Disease_Has_Finding some (not X) ⇒
     Disease_May_Have_Finding some (not X)
```

However, the following implication does *not* hold:

```
not (Disease_Has_Finding some X) ⇒
     not (Disease_May_Have_Finding some X)
```

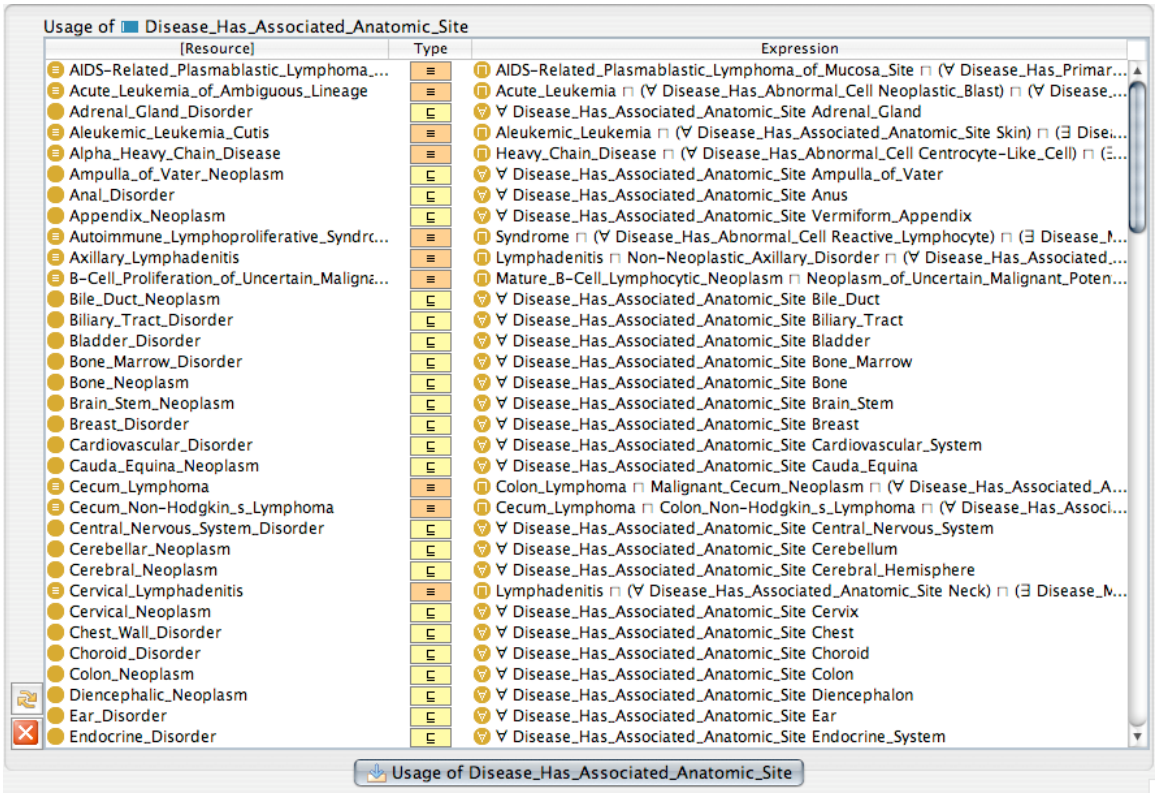
Knowing that there is no value of type X for the property `Disease_Has_Finding`, does not allow us to conclude anything for the possible values of the superproperty. In other words, implementing negation outside of the existential restriction, does not allow us to use the property hierarchy to imply anything useful.

5.4.3 Using suproperties to specify anatomic sites

It would make sense to use a property hierarchy to specify anatomic site of the disease:

- `Disease_Has_Associated_Anatomic_Site`
  - `Disease_Has_Primary_Anatomic_Site`
  - `Disease_Has_Metastatic_Anatomic_Site`

*<u>Question to consider</u>: Would it be fair to say that each site **must** be either a primary or a metastatic site? If the answer is yes, we can get a list of classes where the superproperty is used (partial list is available in Figure 9) and an editor will have to go through these manually distinguishing whether the site is primary or secondary. To get this list, select the property* `Disease_Has_Associated_Anatomic_Site` *in the Properties tab, and click the "Find Usage" button (cf.* Figure 2*).*



Usage of Disease_Has_Associated_Anatomic_Site

[Resource]	Type	Expression
AIDS-Related_Plasmablastic_Lymphoma_...	≡	AIDS-Related_Plasmablastic_Lymphoma_of_Mucosa_Site ⊓ (∀ Disease_Has_Primar...
Acute_Leukemia_of_Ambiguous_Lineage	≡	Acute_Leukemia ⊓ (∀ Disease_Has_Abnormal_Cell Neoplastic_Blast) ⊓ (∀ Disease_...
Adrenal_Gland_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Adrenal_Gland
Aleukemic_Leukemia_Cutis	≡	Aleukemic_Leukemia ⊓ (∀ Disease_Has_Associated_Anatomic_Site Skin) ⊓ (∃ Dise...
Alpha_Heavy_Chain_Disease	≡	Heavy_Chain_Disease ⊓ (∀ Disease_Has_Abnormal_Cell Centrocyte-Like_Cell) ⊓ (∃...
Ampulla_of_Vater_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Ampulla_of_Vater
Anal_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Anus
Appendix_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Vermiform_Appendix
Autoimmune_Lymphoproliferative_Syndrc...	≡	Syndrome ⊓ (∀ Disease_Has_Abnormal_Cell Reactive_Lymphocyte) ⊓ (∃ Disease_N...
Axillary_Lymphadenitis	≡	Lymphadenitis ⊓ Non-Neoplastic_Axillary_Disorder ⊓ (∀ Disease_Has_Associated_...
B-Cell_Proliferation_of_Uncertain_Malignа...	≡	Mature_B-Cell_Lymphocytic_Neoplasm ⊓ Neoplasm_of_Uncertain_Malignant_Poten...
Bile_Duct_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Bile_Duct
Biliary_Tract_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Biliary_Tract
Bladder_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Bladder
Bone_Marrow_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Bone_Marrow
Bone_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Bone
Brain_Stem_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Brain_Stem
Breast_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Breast
Cardiovascular_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Cardiovascular_System
Cauda_Equina_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Cauda_Equina
Cecum_Lymphoma	≡	Colon_Lymphoma ⊓ Malignant_Cecum_Neoplasm ⊓ (∀ Disease_Has_Associated_A...
Cecum_Non-Hodgkin_s_Lymphoma	≡	Cecum_Lymphoma ⊓ Colon_Non-Hodgkin_s_Lymphoma ⊓ (∀ Disease_Has_Associ...
Central_Nervous_System_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Central_Nervous_System
Cerebellar_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Cerebellum
Cerebral_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Cerebral_Hemisphere
Cervical_Lymphadenitis	≡	Lymphadenitis ⊓ (∀ Disease_Has_Associated_Anatomic_Site Neck) ⊓ (∃ Disease_N...
Cervical_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Cervix
Chest_Wall_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Chest
Choroid_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Choroid
Colon_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Colon
Diencephalic_Neoplasm	⊑	∀ Disease_Has_Associated_Anatomic_Site Diencephalon
Ear_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Ear
Endocrine_Disorder	⊑	∀ Disease_Has_Associated_Anatomic_Site Endocrine_System

Usage of Disease_Has_Associated_Anatomic_Site

Figure 9. The use of the super-property Disease_Has_Associated_Anatomic_Site. Perhaps consider if each of these restrictions should instead distingiush whether it is primary or secondary site.

5.5 Partonomy

Partonomy is usually the primary hierarchy in anatomy and the disease properties are often inherited through partonomy. For example, we want to say that

- `Gastrointestinal_Carcinoma` occurs in `Gastrointestinal_System`
- `Small_intestine_Carcinoma` (a subclass of `Gastrointestinal_Carcinoma`) occurs in `Small_Intestine`, which is a part of `Gastrointestinal system` (but not its subclass)

Suppose we use the following restriction to define the site of `Gastrointestina_Carcinoma`:

```
Disease_Has_Associated_Anatomic_Site some
                        Gastrointestinal_system
```

Then we define the following restriction for `Small_Intestine_Carcinoma`:

```
Disease_ Has_Associated_Anatomic_Site some
                        Small_Intestine
```

The class `Small_Intestine_Carcinoma` also inherits the first restriction from `Gastrointestinal_Carcinoma`. Since `Small_Intenstine` is not a *subclass* of `Gastrointestinal_System` (nor should it be), but rather its part, the

combination of these two restrictions actually states that there are really *two* associated anatomic sites for `Small_Intestine_Carcinoma`: one that is a `Gastrointestinal_System` or its subclass, and another one is a `Small_Intestine` or its subclass. This interpretation seems to be wrong.

Using universal restrictions here instead of existential ones is even worse, since it would imply that the site is something that is *both* `Gastrointestinal_System` and a `Small_Intestine`, which, after all disjoints have been declared, will make the `Small_Intestine_Carcinoma` class inconsistent.

The solution to representing partonomy in classification is provided by Alan Rector and Chris Welty in the following draft note for the Semantic Web best-practices group at W3C:
http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/index.html

Here is how the solution would look for the NCI Thesaurus: We define the anatomic site for `Gastrointestinal_System` as `Gastrointestinal System` *or any of its parts*:

```
Gastrointestinal_Carcinoma:
     Disease_Has_Associated_Anatomic_Site some
          (Gastrointestinal_System
           or
          (some Anatomic_Structure_is_Physical_Part_of
                         Gastrointestinal_System))
```

Then we define `Small_Intestine_Carcinoma` in a similar way:

```
Small_Intenstine_Carcinoma:
     Disease_Has_Associated_Anatomic_Site some
          (Small_Intestine
           or
          (some Anatomic_Structure_is_Physical_Part_of
                         Small_Intestine))
```

Entering these restrictions by hand is tedious and we can employ one of the two solutions:

1. develop user interface templates for filling in instance of ontology patterns (see Section 3.2); or
2. use a post-processing script that process each of the restrictions on anatomical sites of diseases and automatically converts restriction of the form "`Disease_Has_Associated_Anatomic_Site` **some** `X`" into "`Disease_Has_Associated_Anatomic_Site (X` **or** `(`**some** `X))`"

5.6 Findings

Many of the classes with findings have numerical characteristics in their names (Figure 10). These classes are hard to maintain and compare. Furthermore, the semantics are encoded in syntactic names and are not accessible for direct machine processing. Given that OWL allows using numeric properties in definitions, and Protégé OWL now supports definitions of numeric ranges for properties as restrictions, it may be advisable

to consider defining properties such as `Percentage_Of_Blast_Cells`, and to define the classes in Figure 10 in terms of these properties and the explicit restrictions on their ranges (Section 2.4.4).

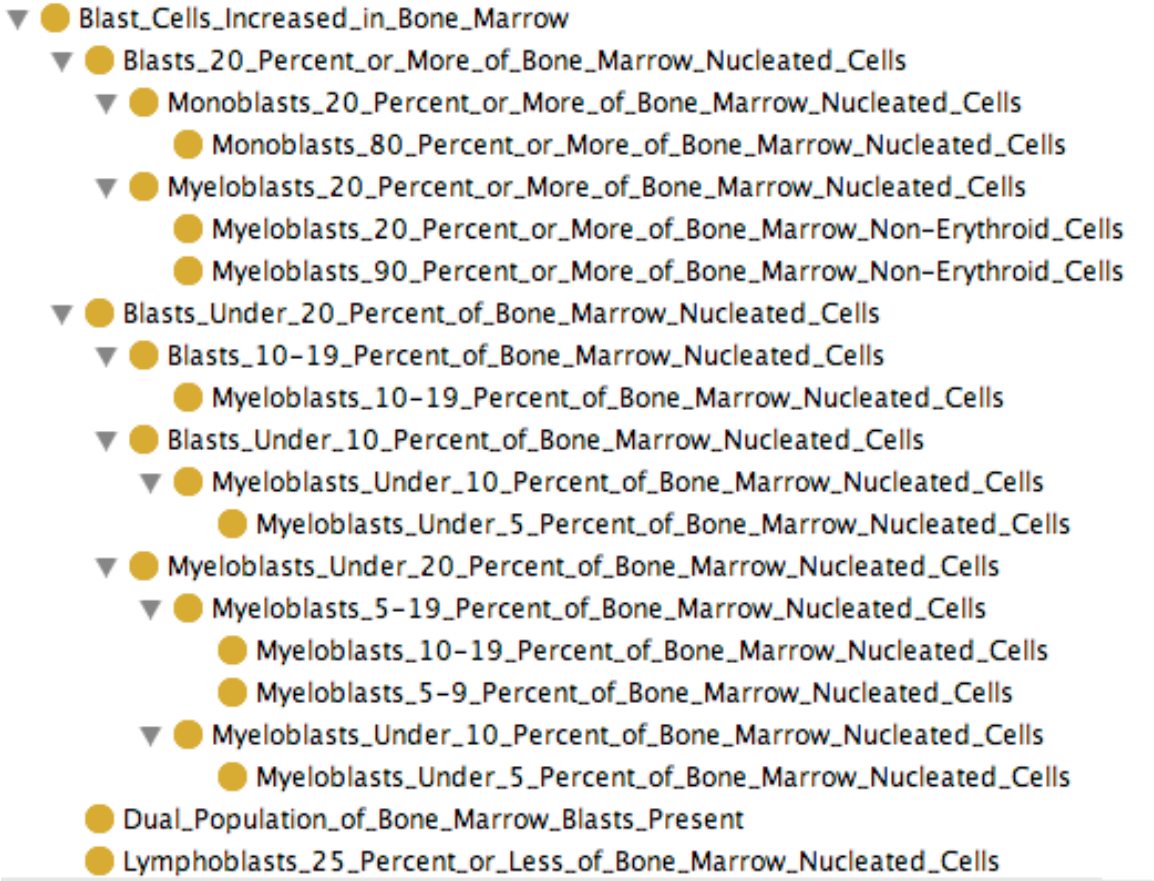


Figure 10. Findings with numeric values in their names.

5.7 Summary of suggestions for changes in Findings and Diseases

- Update the transformation script to determine which conditions in Diseases definitions should be necessary&sufficient, and which should be necessary, based on the property, thus creating defined classes and correct definitions.
- Make all restrictions existential restrictions for the moment
- Replace all the roles with "`_Excludes_`" in their name with the appropriate negation pattern. The script will have to have a mapping table to record which of the Excludes role corresponds to which of the "`_Has_`" roles
- Make properties with "`Disease_Has_`" to be subproperties of the corresponding "`Diseases_May_Have_`" properties.
- Consider if each of the anatomic sites of a disease should be specified as either primary or metastatic
- Make the properties `Disease_Has_Primary_Anatomic_Site` and `Disease_Has_Metastatic_Anatomic_Site` to be subproperties of `Disease_Has_Associated_Anatomic_Site`

- Define sites of diseases to refer to the site and its parts
- Consider using numeric definitional properties for classes of Blast cells

6 Genes and Gene Products

Genes and gene products present a number of modeling challenges, some of them we address here:

- definitional versus non-definitional roles
- describing physical and chromosomal locations (taking advantage of the availability of inheritable datatype properties in OWL)
- representing relations between genes and alleles
- representing relations between genes, gene products, abnormalities, and diseases

6.1 Definitional roles

6.1.1 Definitional roles for Genes

The following roles are defining roles for `Genes` and should be necessary & sufficient conditions in the definitions of genes:

- `Gene_Found_In_Organism` (In 99% of the cases, the organism is `Human`, therefore, this characteristic is not really distinguishing)
- `Gene_Has_Chromosomal_Location`
- `Gene_Plays_Role_In_Process`
- `Gene_In_Physical_Locaiton` (present only if the previous three characteristics are not distinguishing enough)

The rest of the roles in the `Gene` definitions should become necessary conditions.

6.1.2 Definitional roles for Gene products

The following roles are the definitional roles for `Gene_Products` and should be necessary & sufficient conditions in the definitions of gene products:

- `Gene_Product_Encoded_by_Gene`
- `Gene_Product_Plays_Role_In_Biological_Process`
- `Gene_Product_Has_Biochemical_Function`

The rest of the roles in the `Gene_Product` definitions should become necessary conditions.

<u>Question to consider</u>: Is it correct to say that, for example

- *any Ribosomal_RNA that*
- *is encoded by the RNR1_Gene, and*
- *has biochemical function Ribosomal_RNA (???), and*
- *plays role in the Translation biological process*

is a Ribosomal_RNA-1? Are all of the conditions above necessary to identify something as Ribosomal_RNA-1? If not, which subset is sufficient? Is the biochemical function correct?

Any `Gene_Product` that has restrictions on each of the three properties defined should probably become a defined class. This transformation can be performed in the conversion script. More specifically the conversion script should do the following:

- For each subclass of `Gene_Product`, consider if there are existential restrictions on each of the three defining roles: `Gene_Product_Encoded_by_Gene`, `Gene_Product_Plays_Role_In_Biological_Process`, `Gene_Product_Has_Biochemical_Function`
- If there are restrictions for each of the three roles above, move these restrictions, and the named superclass of the class to necessary&sufficient conditions.

Currently, whether or not a `Gene_Product` is defined is inconsistent. For instance, `Ribosomal_RNA-1` is a defined class, whereas a class with restrictions in identical format is primitive (Figure 11).
Similarly, `Gene_Products` that are defined, have all their properties (including the ones that are not definitional) as necessary&sufficient (Figure 18). This structure is the result of the limitation of the Ontylog DL: a class must have all its properties either as definitional or as non-definitional. A mixture of definitional and non-definitional is not allowed. The break-down of which properties go into which category depends on the property type and therefore can easily be done by a script. Therefore, a conversion script can place the definitional properties of `Gene_Products` (when all three are present) as necessary&sufficient conditions, and the rest of the properties (such as `Gene_Product_Is_Element_in_Pathway`) will be necessary.

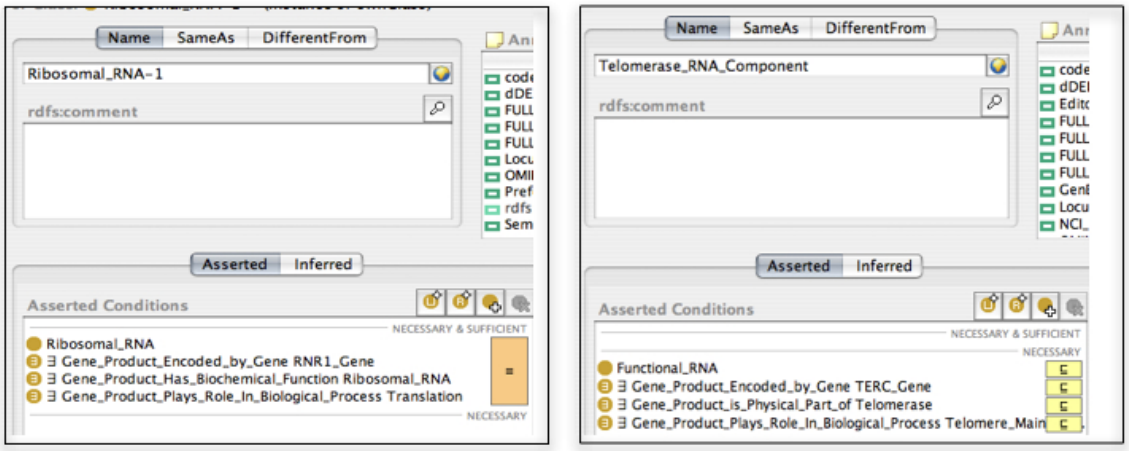


Figure 11. Definitions of two `Gene_Product` classes, one primitive and one defined. `Ribosomal_RNA-1` is a defined class, whereas a class with restrictions in identical format is primitive.

6.2 Describing physical and chromosomal locations

There are two properties in gene definition associated with locations:

- `Gene_in_Chromosomal_Location` links gene to the chromosomal band position (infrequently, to chromosome number or arm).
- `Gene_Has_Physical_Location` links the gene with its actual physical location (region) of a chromosome.

Both of these roles are among the defining roles for `Genes`. We consider if the types of restrictions on these properties should be changed and corresponding value classes deleted from the NCI Thesaurus.

6.2.1 Representing the property Gene_Physical_Location

Because the Ontylog DL does not allow defining roles to be datatype properties, the values for the restrictions must necessarily be defined as classes in the Ontylog DL. This requirement leads to classes such as `Gene_Physical_Location` and its subclasses such as `_13_102296421-102326346`. These classes are not really classes but rather designations of locations

There seems to be a general agreement among modelers that it would be more appropriate to model the physical location as a simple string value. OWL allows such solution. Furthermore, for String values, the appropriate restriction to use is **`hasValue`** rather than **`some`**. Figure 12 shows the current definition of a gene and the definition after we created a new property `Gene_Has_Physical_Location_Value` to replace `Gene_Has_Physical_Location` and changed the physical-location definition for the same gene.

In practice, we will simply change the property `Gene_Has_Physical_Location` to become a datatype property and update the corresponding restrictions. Note however that in Protégé, if you simply change an object property to a datatype property, all its values and restrictions will be lost, because they are no longer valid. Thus, the way to perform the transformation, is through a script that does the following (this script could be a part of the conversion script):

- replace each restriction of the form
  `Gene_Has_Physical_Location` **`some`** `X`
  with
  `Gene_Has_Physical_Location` **`hasValue`** `Y`
- change the definition of `Gene_Has_Physical_Location` to become a datatype property with the range of `xsd:String`

In addition, the class `Gene_Physical_Location` and its subclasses can be removed.

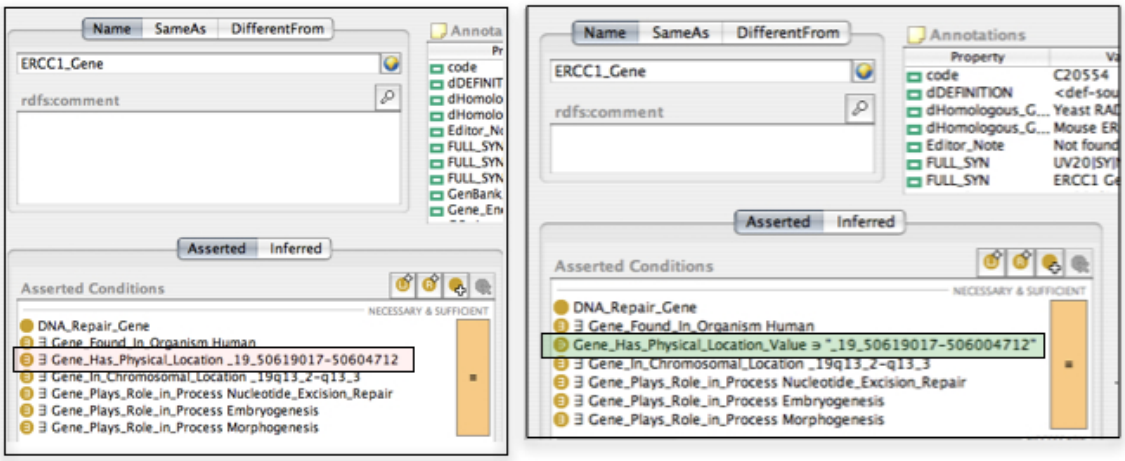


Figure 12. Defining `Gene_Physical_Location` as a value for a datatype property

6.2.2 Chromosome_Arm and Chromosome_Band

Currently, chromosome arms and chromosome bands are classes in the ontology. Chromosome bands are defined by their positions on the corresponding arms (Figure 13). The chromosome number and the physical location of the band on the arm is encoded in their name. There are some disadvantages to this solution: First, it is difficult to identify if a particular location or a band is a band on one of the known locations. Second, chromosomal locations of genes change, or become more precise. When this new information is entered in the NCI Thesaurus, the old location classes still remain there. As a result, the ontology contains a large flat listing of classes representing bands that happen to be chromosomal locations for some genes, or were ones thought to be chromosomal locations for genes. Third, the role of these classes is only to encode the locations on chromosomes, which have more of a flavor of datatypes, rather than classes in the ontology (similar to physical locations of genes).
More specifically, the following are common **use cases** that use the chromosomal location information:

- A person has an abnormality: they are missing a particular band on the chromosome. Which genes are affected?
- If the missing band is, for example, `_1p35-p32`, the result should include genes that have location chromosomal location `_1p34`

We suggest considering representation of chromosomal bands as XML Schema datatypes. A set of the XML Schema datatypes for describing chromosomal locations is described, for example, at the following location:
http://biodas.org/RFCs/king_das2/doc/schema/annotation.html
OWL allows ranges of properties to be XML Schema datatypes
http://www.w3.org/TR/2002/WD-owl-guide-20021104/#Datatypes1
Therefore, we can use an approach similar to the one suggested for gene physical location (Figure 12), by having the chromosomal location to be a datatype property with a special XML Schema datatype as its range; use **`hasValue`** restriction to specify genes chromosomal location.

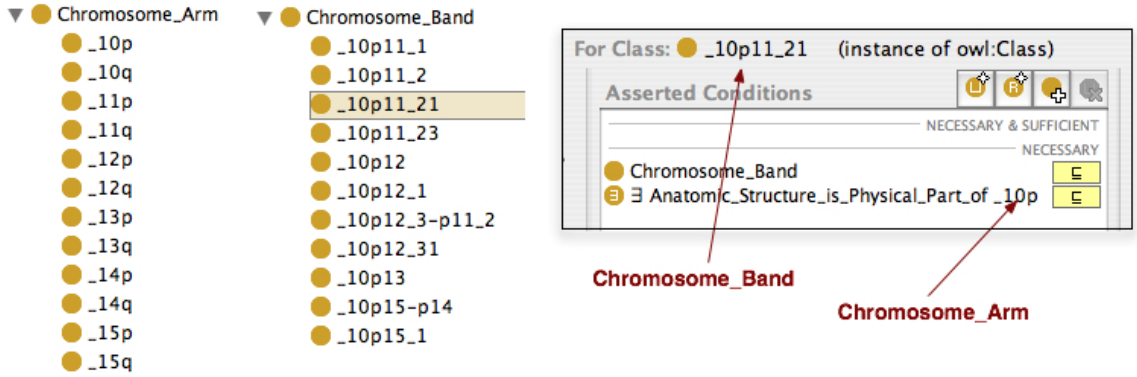


Figure 13. Excerpts of the subtrees for Chromosome_Arm and Chromosome_Band and the definition of Chromosome_Band referring to Chromosome_Arm

6.3 Genes and alleles

NCI Thesaurus contains classes for genes, their wild types, and alleles. Alleles are inheritable mutations in populations (these are not mutations that occur as a result of exposure to carcinogens). As a rule, alleles inherit properties of their parent gene: the chromosomal location, the role it plays in processes, etc. However, the small fraction of cases where alleles alter some of the properties of the gene are the really "interesting" ones.

The NCI Thesaurus models only a small number of alleles—only the ones for which there are use cases (diseases or abnormalities).

6.3.1 The current inheritance structure for genes and alleles

The following is the current solution for representing the relations between genes and alleles in the NCI Thesaurus.

- Represent `Genes` first at the level of Gene-as-class, encoding their typical (normal) properties.
- Represent wild-type genes and alleles as sibling subclasses of the corresponding genes (Figure 14)
- Use property inheritance to suppress inheritance of the features from the gene that are altered by `Allele`.

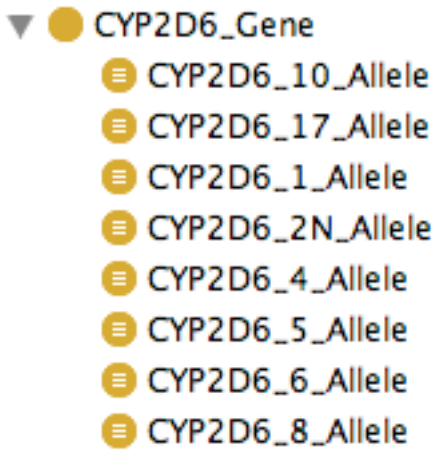


Figure 14. Hierarchical representation of a Gene and its alleles

6.3.2 Using property inheritance to define overriding

We first describe the current use of allele-related properties (and the corresponding properties of genes) and then discuss the semantic problems with such representation as well as propose a solution that is more sound.

6.3.2.1 Current representation of allele properties

Property inheritance is used to suppress the features of the parent gene that are altered by the corresponding allele. For instance, Gene-as-class has property `Gene_Associated_With_Disease`. An inherited allele may have a restriction on a property `Allele_Not_Associated_With_Disease`, which is supposed to suppress the associated disease inherited from the corresponding Gene (Figure 15). The same idea is used for other properties of a gene that can be altered by alleles.

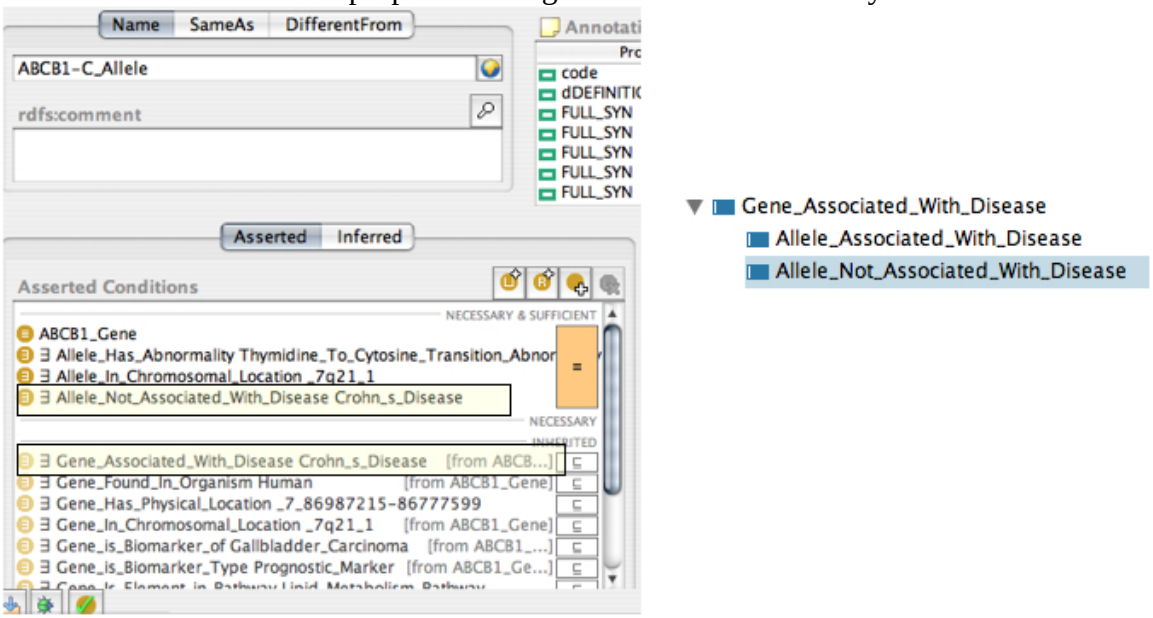


Figure 15. Using property inheritance to suppress properties allele inherit from genes, but override them.

Thus, the following role hierarchy has been developed:

1. `Gene_In_Chromosomal_Location:`
   `Allele_Absent_From_Wild-type_Chromosomal_Location`
   `Allele_In_Chromosomal_Location`

2. `Gene_Plays_Role_In_Process:`
   `Allele_Plays_Altered_Role_In_Process:`

3. `Gene_Is_Element_In_Pathway:`
   `Allele_Ceases_Function_In_Pathway:`

4. `Gene_Associated_With_Disease:`
   `Allele_Associated_With_Disease:`
   `Allele_Not_Associated_With_Disease:`

5. `Gene_Has_Abnormality:`
   `Allele_Has_Abnormality:`
   `Allele_Not_Associated_With_Abnormality:`

Currently, most of the “suppressing” subproperties are not used. There are no restrictions that use the following properties:

- `Allele_Not_Associated_With_Abnormality`
- `Allele_Absent_From_Wild-type_Chromosomal_Location`
- `Allele_Ceases_Function_In_Pathway`

For many of the allele-related roles that do have restrictions associated with them, these restrictions seem to be identical to the restrictions at the gene level:

- It appears (though we couldn't check completely) that the value for `Allele_In_Chromosomal_Location` is always the same as the inherited value for `Gene_In_Chromosomal_Location`

6.3.2.2 Problems with property inheritance

The property inheritance does not logically achieve the suppression. First, the inherited values are always true of the subclasses. For instance, suppose we have:

- `Gene_X:`
  - `Gene_Associated_With_Disease` **`some`** `Disease_Y`
- `Allele_Z`
  - **`subclassOf`** `Gene_X`
  - `Allele_Not_Associated_With_Disease` **`some`** `Disease_Y`

From the fact that `Allele_Not_Associated_With_Disease` is a subproperty of `Gene_Associated_With_Disease`, the following will be inferred in OWL:

- `Allele_Z:`
  - (inferred) `Gene_Associated_With_Disease` **`some`** `Disease_Y`
  - (asserted) `Allele_Not_Associated_With_Disease` **`some`** `Disease_Y`

The two conditions for `Allele_Z` above (the inferred and the asserted ones) are clearly in contradiction with each other.

The same is true for other allele-related subproperties: property inheritance cannot be used to suppress restrictions inherited from superclasses. In contrast, the implication from the property inheritance states the **opposite** of what the modeler intended. The allele that has the opposite role in pathways will be classified as `Allele_Z`.

Therefore, a different mechanism must be used to assert properties of alleles.

6.3.2.3 Alternatives for representing genes and alleles

We suggest two alternatives for representing the relationship between genes and alleles and their properties.

Alternative 1: Non-wild-type alleles in a separate tree from genes

As evident from use cases, not everything that is true about gene-as-class is true about the allele. Therefore, one could argue that alleles **should not** be subclasses of genes: After all, everything that is true of a superclass is also true of a subclass, by definition of class inheritance.

Therefore, we can have alleles in a separate tree, and have a property linking genes and alleles, such as `Allele_Is_Mutation_For_Gene`. This approach would address the problems with inheritance, but has one major drawback: in most cases, alleles *do* inherit properties of the parent gene. If alleles are not subclasses of the parent genes, such inheritance will not be automatic and will need to be inferred by a special-purpose reasoner, by defining special rules that describe how the gene properties are propagated to alleles. This reasoner will propagate the properties of the corresponding gene to the allele, unless there are conflicting properties in the allele's definition. Role chaining (Section 2.4.2) with some extensions can be used here.

Alternative 2: Typical and altered roles

One can say that the roles of the wild-type gene are **typical** roles for that gene. This "typicality" is still true of the allele, even though allele does something atypical. For instance, we can say the following:

- `Gene_X`:
  - `Gene_`**`Typically`**`_Associated_With_Disease` **`some`** `Disease_Y`
- `Allele_Z`:
  - `subclassOf Gene_X`
  - (inherited) `Gene_`**`Typically`**`_Associated_With_Disease` **`some`** `Disease_Y`
  - (asserted at `Allele_Z`): `Allele_Not_Associated_With_Disease` **`some`** `Disease_Y`

In this solution, there is no subproperty relation between properties `Gene_Typically_Associated_With_Disease` and `Allele_Not_Associated_With_Disease`. Both statements above are true about `Allele_Z`: it *typically* associated with a particular disease, but not in this specific case.

Ideally, we would also like to express that for any allele, the value for the property `Allele_Not_Associated_With_Disease` should be one of the values for the property `Gene_Typically_Associated_With_Disease`. However, there is no direct way to express this in OWL (you need SWRL for that).

More specifically, this solution would lead to having the following properties for genes and alleles replacing the property hierarchy above:

- Properties for `Genes`:
  - `Gene_Typically_In_Chromosomal_Location`
  - `Gene_Typically_Plays_Role_In_Process`
  - `Gene_Typically_Is_Element_In_Pathway`
  - `Gene_Typically_Associated_With_Disease`
  - `Gene_Typically_Has_Abnormality`
- Properties for `Alleles`:
  - `Allele_Absent_From_Wild-type_Chromosomal_Location`
  - `Allele_In_Atypical_Chromosomal_Location` (only to be asserted when it is different from the inherited one)
  - `Allele_Plays_Altered_Role_In_Process`

- Allele_Ceases_Function_In_Pathway
- Allele_Not_Associated_With_Disease
- Allele_Has_Abnormality
- Allele_Not_Associated_With_Abnormality

6.4 Genes, gene products, abnormalities, and diseases

Figure 16 shows the various classes of concepts affected by genes:

- Genes encode gene products
- Gene products cause molecular abnormalities (note that there is currently no role expressing this information)
- Molecular abnormalities cause diseases
- Gene products play roles in processes

Modeling relations between classes along the solid dark arrows in the Figure 16 would paint the most complete picture of what is going on. However, this detailed information (e.g., always going through gene products) either may not be available, or may be too tedious to enter. While technically it is the gene product rather than the gene itself that plays a role in the disease, we often want to bypass such detailed information. In fact, it is rare for people to ask what proteins (gene products) are involved in a particular disease; rather they ask about the genes themselves (that had encoded the product).

Moreover, many of the use cases involve “jumping over” gene products or molecular abnormalities to get directly from genes to processes, or genes to diseases, for example. The dashed green arrows in Figure 16 represents the properties that users want to query. Here are some use cases:

- Researchers often need a link from gene to abnormality
- Clinicians often need a link from gene to diseases: which disease is the gene associated with
- Clinicians need to treat patients and see if a particular patient carries this allele, look if protein activity changes in response to the drug, and decide on the treatment.

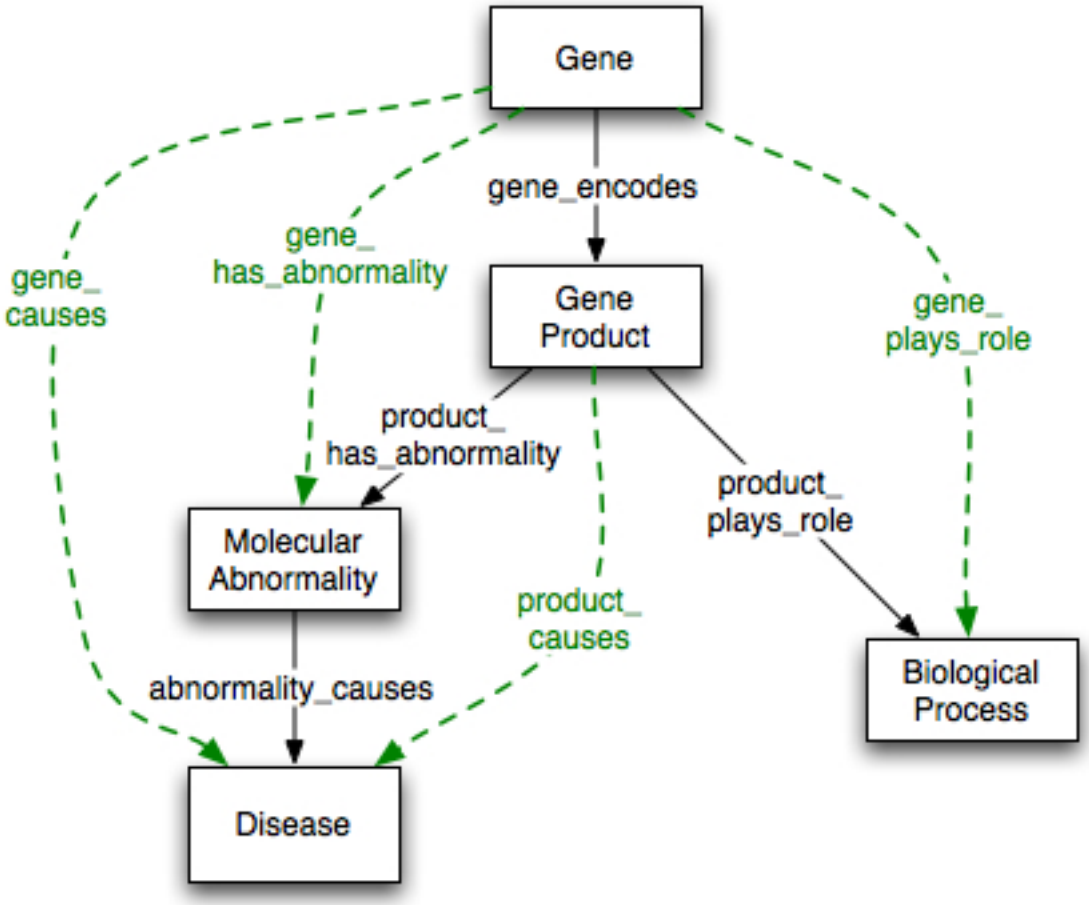


Figure 16. Relations between gene-related classes. The solid black arrows indicate properties directly represented in the Thesaurus. The dashed green arrows indicate properties that should be inferred.

What we really need here are chains of properties:
Given

```
1. GeneX: encodes some ProductY
2. ProductY: plays_role_in some ProcessZ
```

Infer:

```
GeneX: plays_role_in some ProcessZ
```

There are two possible solutions: one uses property inheritance and transitivity; the second uses role chains (Section 2.4.2).

6.4.1 Using property inheritance and transitivity

We can rename the properties on the green dashed lines slightly. Let's say we replace the property

`gene_plays_role_in_process`

in Figure 16 with a new (rather elaborate) property

`gene_encodes_product_or_product_plays_role_in_process`.

This new property has the union of `Genes` and `Gene_Products` as its domain and union of `Gene_Products` and `Biological_Processes` as its range.

First, we then define the following property inheritance structure:

- `gene_encodes_product_playes_role_in_process_or_product_plays_role_in_process`
  - `gene_encodes_product`
  - `product_plays_role_in_process`

Second, we define the property `gene_or_gene_product_plays_role_in_process` as *transitive*.
Then we infer, using property inheritance:

- `GeneX: gene_encodes_product_playes_role_in_process_or_product_plays_role_in_process` **some** `ProductY`
- `ProductY: gene_encodes_product_playes_role_in_process_or_product_plays_role_in_process` **some** `ProcessZ`

And then by transitivity:

- `GeneX: gene_encodes_product_playes_role_in_process_or_product_plays_role_in_process` **some** `ProcessZ`

which is exactly what we want.
The disadvantage of this solution is the creation of a new artificial and extremely cumbersome property. Note that queries will need to know about this bulky role and ask for it, instead of the original intuitive one (gene_plays_role_in_process). Furthermore, after the inference, is preformed, the Thesaurus will contain not only the last statement (which is the one that we really wangt), but also the two intermediate statements that were a direct inference of property inheritance. Thus, there will be a lot of extraneous statements that we are not really interested in.
The advantage of this solution is that we need to define only the property hierarchy and the transitivity of the property, and the restrictions will be inferred by the classifier—there is no need for any special-purpose reasoning.

6.4.2 Using role chains

Another possible solution is to define the desired properties represented by the dashed green arrows in the figure explicitly in the NCI Thesaurus. In other words, have all the three statements above in the NCI Thesaurus. However, rather than defining all of them manually, we can define them as *role chains* (Section 2.4.2): we will say that traversing two property (e.g., `gene_encodes` and `product_plays_role`) results in a new property (`gene_plays_role`). As suggested in Section 2.4.2, we can either use SWRL rules to define these conditions, or, use some post-processing to infer the additional properties. Given that we will have only a limited number of cases, a custom-tailored solution, such as post-processing, or an use of an ontology patterns may be preferable. An ontology pattern may define that when we have two such property statements, they should be expanded to include a third one.

6.5 Observations and analysis

This section includes a number of other observations from our analysis of the `Gene` and `Gene_Product` categories.

6.5.1 Protein vs Protein_Family

`Protein_Family` is a subclass of `Protein`. Logically, this is incorrect: `Protein_Family` is *not* a kind of `Protein`. Perhaps `Protein_Family` should be a direct subclass of `Gene_Product`. Even more correctly, it should be its own kind, but such drastic retreeing is not necessary, because, from the point of view of properties that are applicable to it, `Protein_Family` is a `Gene_Product`.
The same consideration probably applies to `Protein_Complex_Subunit`

6.5.2 Use of properties Gene_Is_Biomarker_Of and Gene_Is_Biomarker_Type

In the use cases, and in the modeling document on Genes and Alleles, it is asserted that the following two properties apply only to alleles, and not to genes as classes:

1. `Gene_Is_Biomarker_of:`
2. `Gene_Is_Biomarker_Type:`

However, currently these two properties are used primarily with `Genes` as classes (Figure 17). Either the description of the properties in the use cases is incorrect, or their usage is incorrect.

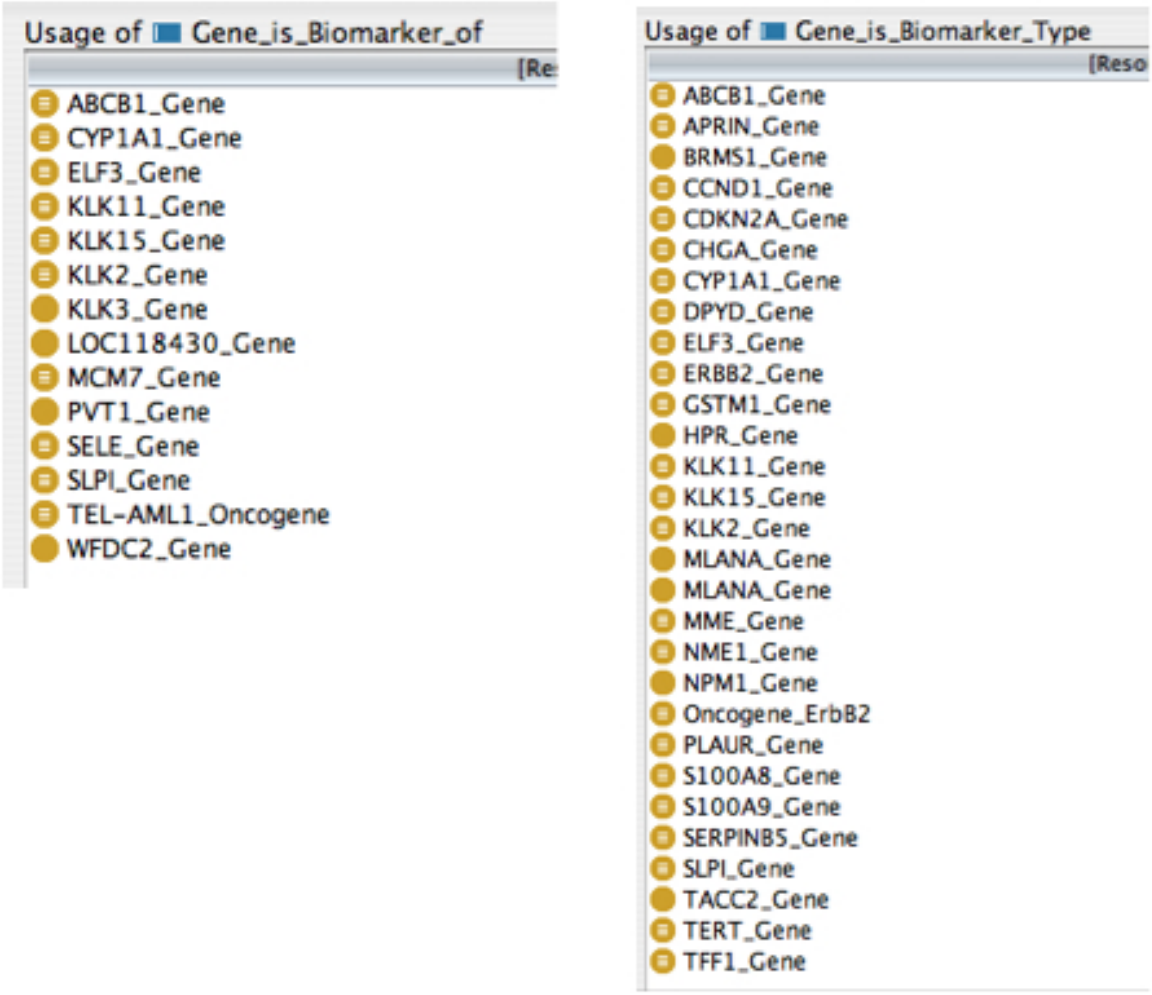


Figure 17. The usage of properties Gene_Is_Biomarker_Of and Gene_is_Biomarker_Type in the Thesaurus. The list on the left represents the classes where the property `Gene_Is_Biomarker_of` is used (generated automatically, by using the "Find Usage" button on the `Gene_Is_Biomarker_of` property). The list on the right represents the classes where the property `Gene_Is_Biomarker_Type` is used Most of the usage is with Genes as classes, rather than with alleles

6.5.3 Multiple Biological_Processes in definitions of Gene_Products and Genes

A number of `Gene_Product` classes have more than one restriction on the property `Gene_Product_Plays_Role_In_Biological_Process` (see Figure 18 for an example).

<u>Question to consider:</u> What is the presumed semantics of these multiple definitions? It can be either of the following:

1. *The specific Gene_Product (Calcineurin-A_Alpha) must play a role in each of the processes that are values for restrictions simultaneously. Any gene_product that does not play a role in any of these processes is not Calcineurin-A_Alpha. (This is the interpretation of the current definition)*
2. *The specific Gene_Product (Calcineurin-A_Alpha) must play a role in at least one of the processes in the restrictions. Is it correct to say that any Protein_Complex_Subunit that is encoded by gene PPP3CA_Gene, has biochemical function Phosphoprotein_Phosphatase, and plays a role in either of the biological processes in the restriction is definitely Calcineurin-A_Alpha?*

Figure 19 presents two variants of the definition for `Calcineurin-A_Alpha` corresponding to the two cases above. In both cases, we keep only the definitional components of the definition in Figure 18 by moving the non-definitional restriction into the necessary conditions. Figure 19a keeps all the definitional restrictions as part of a single intersection. The semantics of this variant corresponds to the case 1 above. In Figure 19b, there are four alternative blocks (sets) of necessary& sufficient conditions. If the `Gene_Product` satisfies the conditions in any of the four sets, it is then `Calcineurin-A_Alpha` (case 2 above). Naturally, any of the definitional blocks, can have more than one biological process.

A similar situation occurs in `Genes` for the restrictions of the property `Gene_Plays_Role_In_Process`. Currently, it is assumed that the `Gene` has to play role in *all* of the listed process to be classified as a particular gene. In some cases. The definition may need to be split up to create different blocks, as in Figure 19.

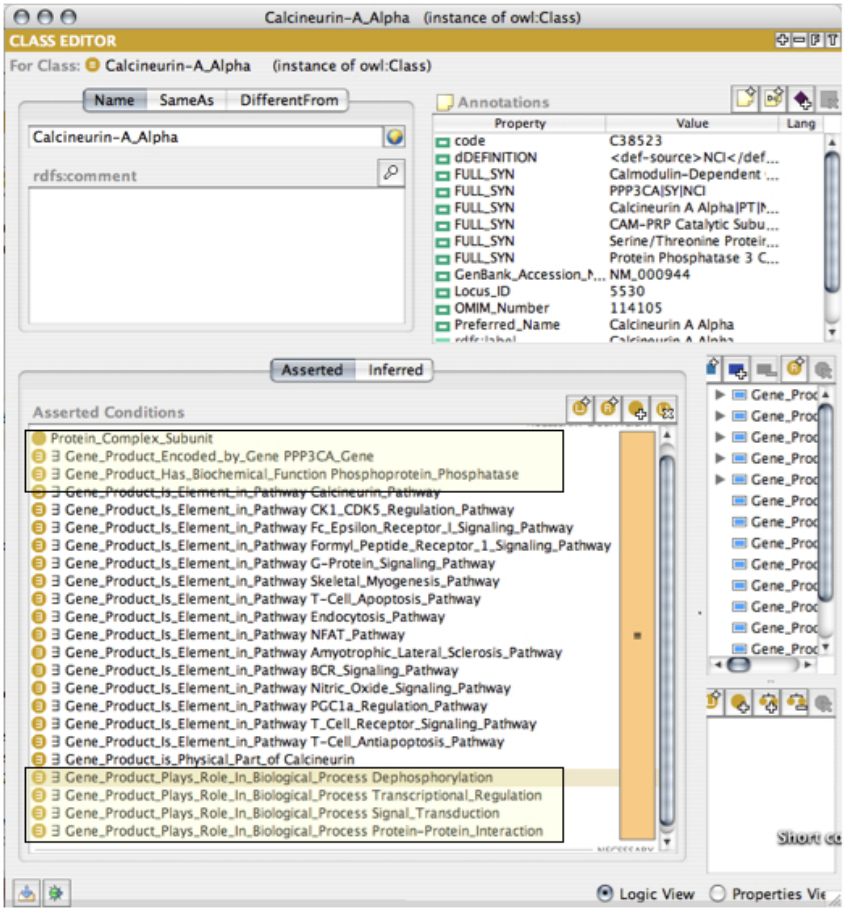
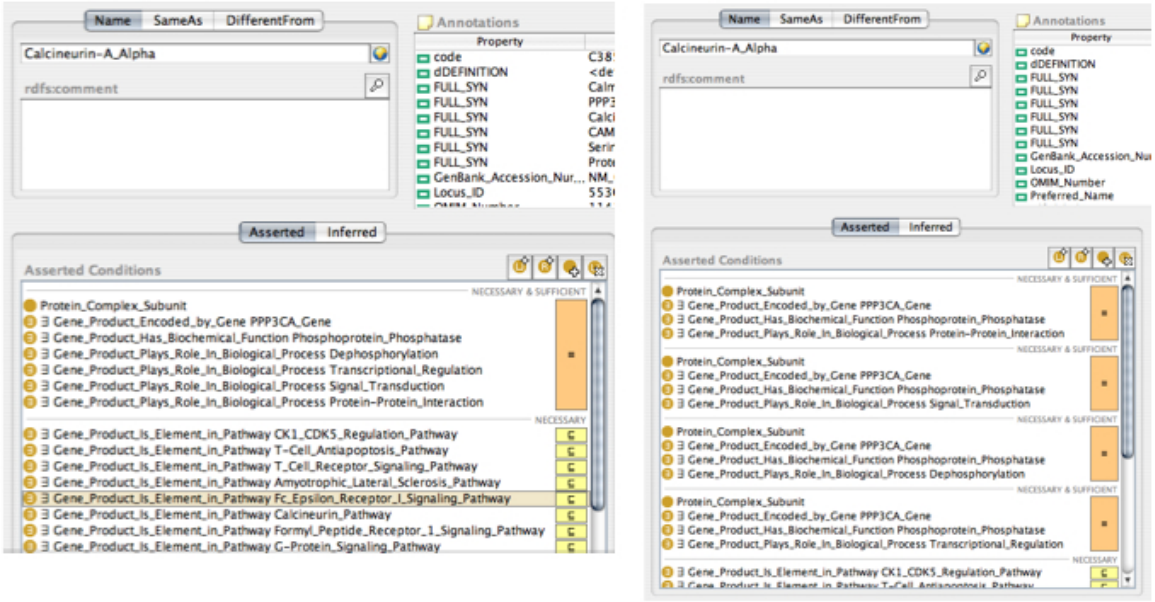


Figure 18. Definition of a Gene_Product. The class `Calcineurin-A_Alpha` is defined. However, only the highlighted restrictions should be necessary&sufficient conditions. The rest of the definitions should be necessary conditions**.**



(a) (b)

Figure 19. Different definitions corresponding to the different semantics of the multiple biological processes

6.6 Summary of suggestions for changes in the Gene and Gene_Product Kinds

- Update the conversion script to determine which conditions in `Gene` and `Gene_Product` definitions should be necessary&sufficient, and which should be necessary, based on the property, thus creating defined classes and correct definitions.
- Update the conversion script to change the type of the property `Gene_Has_Physical_Location` and the restrictions for this property.
- Consider using XML Schema datatypes for chromosomal locations of genes.
- Determine if it is necessary to assert a property on allele (e.g., `Allele_In_Chromosomal_Location`) if it is exactly the same as the corresponding property on the gene.
- Decide on the alternative to be used for modeling the allele roles suppressing the inherited gene roles; specific transition steps depend on the alternative pursued.
- Decide which alternative to use to infer the additional properties that link genes directly to abnormalities and diseases, when only the properties going through gene products are asserted
- Determine the correct use or definition of the properties `Gene_Is_Biomarker_*`: should they describe genes or alleles?
- Consider making `Protein_Family` and `Protein_Complex_Subunit` direct subclasses of `Gene_Product`
- Consider if multiple biological processes in the definitions of `Genes` and `Gene_Products` should be in the same definitional block or different ones. This decision may need to be done on a case by case basis

7 Anatomy and EO_Anatomy

7.1 Summary of the Anatomy Kinds

There are two categories related to Anatomy: `Anatomy`, which refers to Human Anatomy and `EO_Anatomy`, which refers to the anatomy of experimental organisms.
The primary experimental organism considered is mice.
The human anatomy contains both gross anatomy and embryonic anatomy.
The two relations in the Anatomy kind are:

(1) `Anatomic_Structure_is_Physical_Part_of` and
(2) `Anatomic_Structure_Has_Location`

The `EO_Anatomy Kind` has only one relation currently:
`EO_Anatomic_Structure_is_Physical_Part_of`

The Kind that is connected the most to the anatomy kind is Findings and Disorders. Other Kinds are related to Anatomy as well:

- Biological_Processes
- Chemicals and Drugs
- EO_Findings_and_Disorders
- Gene through `Gene_In_Chromosomal_Location` (although this will probably go away – see Section 6.2)
- Gene Product
- Technique

7.2 Analysis and Observations

7.2.1 Transitivity of Anatomic_Structure_is_Physical_Part_of

The relation `Anatomic_Structure_is_Physical_Part_of` is transitive, and should be defined so in OWL.

7.2.2 Domain and range of the property EO_Disease_Metastatic_to_EO_Anatomy

The property `EO_Disease_Metastatic_to_EO_Anatomy` has `Findings_and_Disorders_Kind` as its domain and `Anatomy_Kind` as its range. However, from the name of the property, this property should have different domain and range:

- Domain: `EO_Findings_and_Disorders`
- Range: `EO_Anatomy`

7.2.3 Defined classes in Anatomy

Currently, all classes in the two `Anatomy` categories are primitive classes, defined through their necessary conditions. Indeed, it appears that for the most part, anatomical

structures are hard to define through necessary and sufficient conditions. It may be useful to consider if any of the restrictions in the `Anatomy` subtree can be definitional (see the discussion on Embryonic structures in Section 7.2.4)

7.2.4 Embryonic structures

It is unclear if the following restriction on the class `Embryonic_Cell` is correct:

```
Anatomic_Structure_is_Physical_Part_of some
                                        Embryonic_Tissue
```

For example, `Blastomere` is a subclass of `Embryonic_Cell`, but it is physical part of `Blastocyst`, which is *not* defined as an `Embryonic_Tissue`. Similarly, `Neural_Crest_Cell`, which is a subclass of `Embryonic_Cell` is part of `Neural_Crest`. `Neural_Crest` is *not* defined as a tissue, but rather as `Other_Embryologic_Structure`. Should the restriction on the `Embryonic_Cell` instead be:

```
Anatomic_Structure_is_Physical_Part_of some
                                        Embryonic_Structure
```

`Embryonic_Cell` can be a defined class with the following necessary and sufficient condition:

```
Normal_Cell
and
Anatomic_Structure_is_Physical_Part_of some
                                        Embryonic_Structure
```

In other words, and normal cell that is part of an embryonic structure is an Embryonic cell.

7.2.5 Relations between Anatomy and EO_Anatomy

There is currently no relation between the two Anatomy-related categories and no properties that would link the two. There is currently some work on mapping between the human anatomy and mouse anatomy. In light of this work, several questions must be considered:

- How this mapping would be represented?
- Would this mapping be part of the Thesaurus?
- Would this be part of annotation?

7.2.6 Embryonic vs Gross Anatomy

It is important to have both embryonic and gross anatomy represented. Currently, there is no relation between the two parts of anatomy, but these will be needed eventually. These relationships will encode the relation between the gross structure and its precursors in the embryos

7.2.7 Fluids in the Anatomy

It is questionable whether classes such as `Body_Fluids` really belong to the `Anatomy` subtree because `Anatomy` should focus on structural parts, and an argument could be

made that fluids are not structural parts of the body (and they can exist outside of the body). However, having these classes in the NCI Thesaurus is important.

Consider the following **use case**: labs need controlled vocabulary to annotate where samples come from , and many of them come from body fluids. Therefore, the NCI Thesaurus should include terminology for fluids, and `Anatomy` is the most appropriate place for it. Note that FMA also include body fluids as part of Anatomy.

7.3 Summary of suggestions for changes in the Anatomy Kinds

- Add a relation between (human) `Anatomy` and `EO_anatomy`. For example: `Anatomic_Structure_Maps_To_EO_Anatomic_Structure`
- Add a relation between features of the gross anatomy and their embryonic precursors. For example: `Anatomic_Structure_Has_Precursor_Embryionic_Anatomic_Structure`
- When in OWL, remove the class `Gene_Physical_Location` and its subclasses, and consider removing subclasses of `Chromosome_Arm` and `Chromosome_Band`, since these are datatype values, specified directly in the restrictions as strings (see the discussion in Section 6.2)
- When in OWL, define `Anatomic_Structure_is_Physical_Part_of` as a transitive property
- Change the definition of the `Embryonic_Cell` class to define it as a physical part of `Embryonic_Structure`, rather than `Embryonic_Tissue`. Make the `Embryonic_Cell` class defined.
- Change domain and range of the property `EO_Disease_Metastatic_to_EO_Anatomy` to `EO_Findings_and_Disorders` and `EO_Anatomy` respectively.

8 Chemicals and Drugs

The category of chemicals and drugs includes chemicals, drugs, and food (perhaps it can be renamed to “Drugs, chemicals, and nutrient compounds”). One of the main concerns in this category is what should be the main organizing principle for the is-a hierarchy (currently, several different principles are used). We also consider several other issues:

- the relations between this category and some of the classes in the anatomy hierarchy representing substances, such as hormones, that can be both parts of anatomy and drugs.
- the use of primitive and defined classes in this category
- instantiating links between chemicals and drugs and diseases
- closing roles for properties such as FDA approval

8.1 Organizing the is-a hierarchy

Currently, the main concern in the representation of Chemicals and Drugs is determining what should be the main organizing principle for the is-a hierarchy.

8.1.1 Current approach

There are several possible axis for the main is-a organizing structure (Figure 20). All of them are currently used to some extent, and there is some overlap as some chemicals are treed in both places:

- function (about 90% of entities)
- structure (the remaining 10%)

The current rule of thumb is to tree a concept under function, it its function is known, use structure otherwise.

Figure 20. Top-level organization of Chemicals and drugs.

8.1.2 Problems with the current approach

There are several problems with the current approach:

1. Inconsistent modeling: there is no primary organizing principle, and a decision needs to be made for each component whether it should go under structure or function
2. No real use cases for structural organization: in most use cases for the NCI thesaurus, structure is not a concern at all. Chemists are the only ones concerned

with structure, but they are not NCI's customers. Besides, there are many chemical databases they can go to get the chemical structure

3. Mixed principles in modeling by function: There are two more axis within the functional hierarchy:
    - Therapeutic intent
    - Mechanism of action

    We observed that these two axis are used interchangeably. See for example Figure 21: `Anti-asthmatic_Agent` is therapeutic intent, but `Bronchodilator` is mechanism of action.

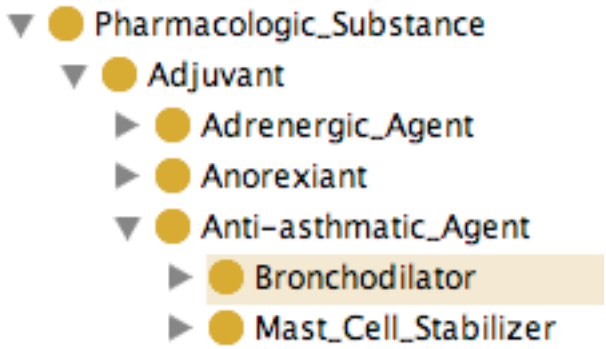


Figure 21. Two different classification principles in one tree: therapeutic intent (Anti-asthmatic agent) and Bronchodilator (mechanism of action)

8.1.3 Use cases for the hierarchy

There are different classes of users: some are interested in therapeutic intent; others are interested in molecular targets and they would like to see by mechanism of action. Hardly any users are explicitly interested in structure.
Users (such as CTAP) are also interested in links between drugs and specific cancers. There are chemical compounds that need to be in so that people can code their data with them, but very little is known about their function, you want them under structure.

8.1.4 Classification approach to consider

Separate one axis and use that one as the main categorization principle. Let the classifier generate the hierarchies under other axis if there is enough information. This approach is the most consistent from the point of view of principled design. Therapeutic intent seems to be the most promising axis to use as the primary classification.

Proposal: perform a pilot experiment, selecting a single axis, classify everything under tree, and provide information for treeing in other trees through properties.

8.2 Chemicals and anatomy

Some things, such as hormones. could be drugs if you take them out of an organism and put in a drug; when they are in the organism itself, they are treated differently, could be targets of drugs, etc. Example:

- `Therapeutic_Androgen` is a subclass of `Pharmacologic_Substance`
- `Androgen` is a subclass of `Physiology-Regulatory_Factor`

Currently there is no relation between these types of substances.
Consider if there should be a relationship between `Therapeutic_Androgen and Androgen`.

8.3 Defined versus Primitive classes

The `Chemicals_and_Drugs` category contains a number of defined classes. It is not clear how the decision on what is a defined and what is a primitive class is made. Consider for example the two classes in Figure 22: one of them is defined and one is primitive, but they have essentially the same sets of restrictions. In fact, it appears that, by their very nature, classes in this kind will be primitive rather than defined. In any case, it should be considered when classes in this hierarchy should become defined, and which restrictions are both necessary and sufficient and which are only necessary.

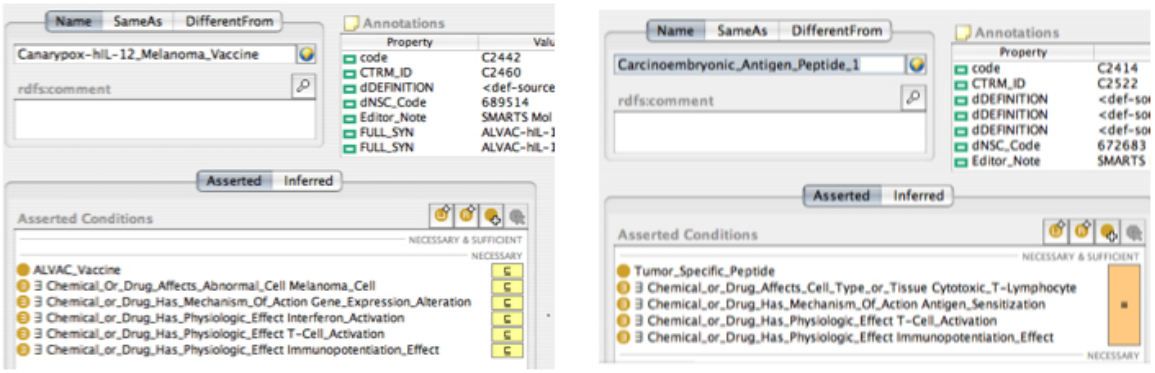


Figure 22. Two classes in the Chemicals_and_drugs kind: the class definitions are very similar, but one class is defined and one is primitive.

8.4 Links between Chemicals and Drugs and Diseases

It appears that many use cases require links from chemicals and drugs to disease: the use of chemicals and drugs in the NCI drug dictionary (enabling clinicians and patients to learn about the drugs), the use of the NCI Thesaurus for conducting clinical trials, etc. There are several properties in the NCI Thesaurus that link Chemicals and Drugs and Diseases:

- `Chemical_or_Drug_FDA_Approved_for_Disease`
- `Chemical_or_Drug_Has_Accepted_Therapeutic_Use_For`
- `Chemical_or_Drug_Has_Study_Therapeutic_Use_For`

However, in the current version not a single one of these roles is instantiated for any single drug or chemical. In fact, other roles that may lead to diseases, such as `Chemical_Or_Drug_Affects_Abnormal_Cell` are also instantiated very poorly.

8.5 Closing roles: FDA approval

While in general we don't suggest closing roles by stating both existential and universal restrictions the role `Chemical_or_Drug_FDA_Approved_for_Disease` is one

example where it should be closed: When we know that a particular drug is approved by FDA for disease X and Y, we also know that it is also *not* approved for anything else. Thus we must state the following for some `DrugZ`:

```
1. Chemical_or_Drug_FDA_Approved_for_Disease
                                some DiseaseX
2. Chemical_or_Drug_FDA_Approved_for_Disease
                                some DiseaseY
3. Chemical_or_Drug_FDA_Approved_for_Disease
                                only (DiseaseX or DiseaseY)
```

8.6 Summary of suggestions for Chemicals and Drugs

- Consider renaming to "Drugs, chemicals, and nutrient compounds"
- Consider performing a pilot exercise to use only therapeutic intent (or some other single principle) as a primary principle for classifying chemicals and drugs
- Consider creating a link between Chemicals and Drugs and Anatomy to indicate that some drugs are pharmacologic version of some anatomic components.
- Consider if any of the classes should be defined. If not, make the classes that are currently defined to be primitive. If yes, determine what principles should be used to make a class defined and which restrictions should be necessary and sufficient, and which should be only necessary.
- Close roles for FDA approval

9 Molecular Abnormalities. Abnormal Cells

The main issues in modeling abnormalities is creating links between abnormalities and other categories: mainly, links between abnormal cells and normal cells in the anatomy subtree; and links between abnormalities and other categories such as diseases that they cause.

9.1 Links between abnormalities and other categories

`Abnormal_Cells` currently don't have any properties of their own. Rather they are fillers for properties on `Diseases` and `Chemicals_and_Drugs`. `Molecular_Abnormalities` also don't have any properties of their own, but fill in properties for `Diseases`, `Genes`, and `Gene Products`. For the most part, abnormalities exists for molecular characteristics of diseases. However, it appears that there will need to be links from abnormalities to other kinds: for instance, abnormalities cause disease (see also Figure 16).

9.2 Links between abnormal and normal cells

Currently, there is no relation between the concepts in the `Abnormal_Cell` tree and the `Cells` and `Normal_Cells` in the `Anatomy` tree. `Normal_cell` and `Cell` are in `Anatomy`; `Abnormal_Cells` is a separate kind. In fact, there are places where the hierarchy of abnormal cells essentially mirrors the hierarchy of normal cells. The current thinking is that the `Anatomy` domain contains the naturally occurring substances, prototypical, non-pathological concepts. However, it may be useful to consider if such link should exist,

10 Biological Processes

It appears that there are no major modeling issues for Biological processes – their modeling is straightforward. All biological process classes are primitive (Consider if this should be the case). There are no specific use cases that require biological process directly: rather, these are used to define other classes, such as genes and gene products.

10.1 Links to Gene ontology and Upper ontologies

The Biological process kind seems to be the prime target for linking with ontologies and terminologies outside of the NCI Thesaurus. If such linking is to be considered, Biological process is the best place to do this and this is where the benefits of such links would be most apparent. Specifically, there are two types of links to consider: linking to Biological processes in the Gene ontology, and linking to the notion of Process in upper ontologies, such as DOLCE.

10.1.1 Linking to Gene Ontology

The Gene ontology also has the notion of biological process and in fact, appears to have a lot more of them than the NCI Thesaurus. The major difference in the representation is that the NCI Thesaurus focuses only on pathological processes, whereas GO represents normal processes.

In fact, linking to GO or reusing some of the Biological process hierarchy from GO for NCI Thesaurus has been considered before. At the time, the following reasons for *not* using GO were cited:

1. GO considers all organisms, not just humans
2. There was no true is-a hierarchy in GO
3. GO did not publish history, whereas NCI users need to be able to point to the original concept in the correct version
4. There was no formal mechanism in the language to link to an outside terminology
5. GO has only normal functions

Note, that 4 our of these 5 concerns are not valid today:

1. NCI Thesaurus is also branching out into other organisms, such as mice (although still not as widely as GO)
2. There is a true is-a hierarchy for processes in GO
3. GO is better about publishing history, and will be better still as it becomes part of the Stanford cBIO National Center for Biomedical Ontology
4. OWL has a formal import mechanism that can be used to import GO
5. It is still true that GO focuses primarily on normal functions whereas NCI Thesaurus focuses on pathological functions

Given that the situation has changed so drastically, t may be worthwhile reconsidering importing the Gene Ontology. Most of the Biological processes in the thesaurus would be subclasses of the GO processes, except for the pathological ones and maybe population processes.

The main advantage of using the Biological Processes hierarchy from GO is the immediate access to a wider terminology, without the need to develop one in-house: only extend the parts that are needed. GO is widely accepted and is used for annotating experimental data – NCI researchers can make use of these links

10.1.2 Linking to Upper Ontologies such as DOLCE

Many upper ontologies represent Processes in a very detailed way. We will use DOLCE as an example (it was also developed for OWL, which is an advantage over other upper ontologies, such as SUMO). They provide the notions of spatial and temporal locations of processes, the fact that processes can be broken up into subprocesses, etc. None of this is present in the Thesaurus, and it doesn't make sense to reinvent the wheel for these.

References

Golbeck, J., Fragoso, G., Hartel, F., Hendler, J., Oberthaler, J. and Parsia, B. (2003). The National Cancer Institute's Thesaurus and Ontology. *Journal of Web Semantics* **1**(1).

Grau, B.C., Parsia, B. and Sirin, E. (2004). Working with multiple ontologies on the semantic web. In: *Proceedings of the Third Internatonal Semantic Web Conference (ISWC2004)*.