Published in final edited form as:
J Biomed Inform. 2016 August ; 62: 90–105. doi:10.1016/j.jbi.2016.06.008.

A unified software framework for deriving, visualizing, and exploring abstraction networks for ontologies

Christopher Ochs[a,*], **James Geller**[a], **Yehoshua Perl**[a], and **Mark A. Musen**[b]

[a] Computer Science Department, New Jersey Institute of Technology, Newark, NJ 07102, USA

[b] Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, CA 94305, USA

Abstract

Software tools play a critical role in the development and maintenance of biomedical ontologies. One important task that is difficult without software tools is ontology quality assurance. In previous work, we have introduced different kinds of abstraction networks to provide a theoretical foundation for ontology quality assurance tools. Abstraction networks summarize the structure and content of ontologies. One kind of abstraction network that we have used repeatedly to support ontology quality assurance is the partial-area taxonomy. It summarizes structurally and semantically similar concepts within an ontology. However, the use of partial-area taxonomies was *ad hoc* and not generalizable. In this paper, we describe the Ontology Abstraction Framework (OAF), a unified framework and software system for deriving, visualizing, and exploring partial-area taxonomy abstraction networks. The OAF includes support for various ontology representations (e.g., OWL and SNOMED CT's relational format). A Protégé plugin for deriving "live partial-area taxonomies" is demonstrated.

Keywords

Ontology exploration; Ontology summarization; Abstraction network derivation; Visualization of ontology content; Ontology tools

1. Introduction

The development of biomedical ontologies depends on software tools like Protégé [1], WebProtege [2], and OBO Edit [3]. These tools allow a user to create, edit, and browse ontology content. Biomedical ontologies are typically large and complex knowledge representation systems. Even with well-established ontology editing tools such as Protégé, the size and complexity of many ontologies makes their maintenance difficult. In previous work, we introduced different kinds of abstraction networks [4], compact summaries of structure and content of an ontology, to support ontology maintenance [5], evolution

[*] Corresponding author at: Computer Science Department, New Jersey Institute of Technology, Newark, NJ 07102-1982, USA. cro3@njit.edu (C. Ochs).

Conflicts of interest
The authors have no conflicts of interest. All authors are aware of the submission of this manuscript.

tracking [6,7], and quality assurance (QA) [8–10], among other use cases. These abstraction networks are smaller and easier to comprehend than the original ontologies (for an example, see Fig. 3).

In the past, we developed abstraction networks and their associated quality assurance methodologies using a "one-at-a-time" approach; we designed one abstraction network for one ontology. It was not possible to create the same kind of abstraction network for different kinds of ontologies. As a corollary, it was not possible to derive the abstraction networks for different kinds of ontologies with the same software tool. For example, a *partial-area taxonomy* [9–11], a kind of abstraction network, summarizes concepts according to their structure and semantics (see Section 2.1.1). We created an appropriate tool for SNOMED CT [12] (see Section 2.2.1), but the method for deriving this abstraction network was not applicable to OWL format ontologies (e.g., the National Cancer Institute thesaurus (NCIt) [13] and the Ontology of Clinical Research (OCRe) [14]) due to differences in their structures.

From the point of view of quality assurance, we strongly subscribe to the principle that "You can't improve what you don't see." [15]. Thus, our ontology quality assurance framework is based on the development of a software tool for deriving and *visualizing* abstraction networks of ontologies. The visualization software that we developed for abstraction networks was implemented using the same "one-at-a-time" approach, resulting in limited applicability.

For example, in previous work we developed the Biomedical Layout Utility for SNOMED CT (BLUSNO) [16], a software tool for deriving and visualizing SNOMED CT partial-area taxonomies. However, when we developed new methodologies to create various kinds of partial-area taxonomies for ontologies in OWL and OBO Format, it was necessary to develop different software tools. Furthermore, as we developed additional kinds of abstraction networks, each of which summarizes a different aspect of an ontology's structure, it was necessary to develop new software systems for each abstraction network. For example, the tribal abstraction network (TAN) [17], which summarizes points of intersections among subhierarchies in an ontology, required the development of another *ad hoc* software tool.

To improve the efficiency of developing and deriving abstraction networks, we introduced a family-based methodology [18] for the ontologies in the NCBO BioPortal [19]. Following the family-based approach, we identify *families* of structurally similar ontologies [18]. Ochs et al. [20] introduced the *structural meta-ontology*, a flexible methodology for classifying ontologies into such families. For each ontology in a family of structurally similar ontologies, the same abstraction network derivation methodologies are applicable. The family-based approach saves a significant amount of time and effort, since abstraction network methodologies are developed such that they are applicable to many structurally similar ontologies. Accordingly, a generic software tool has to be capable of working with ontologies in various formats (*e.g.*, OWL [21] and OBO Format [22]) and it must be able to derive different kinds of abstraction networks and create appealing, accessible visualizations for all of them.

In this paper, we describe the Ontology Abstraction Framework (OAF), a software system for deriving, visualizing, and exploring abstraction networks, especially different versions of partial-area taxonomies. This is a worthwhile endeavor, as we have successfully used partial-area taxonomy abstraction networks (described in detail in Section 2.1) to support ontology quality assurance [8–10,18,23], evolution tracking [5–7], and other use cases. Thus, the OAF is expected to support unified abstraction-network-based quality assurance methodologies, where the same quality assurance techniques will be applicable to all or most of the ontologies in the same family.

Within the OAF, ontologies in various formats are represented using a standardized ontology model. The partial-area taxonomy derivation methodology is defined generically in terms of this standardized ontology model, making it applicable to ontologies released in various formats. Furthermore, processes that can be applied to partial-area taxonomies (*e.g.*, aggregation [7]) are defined in terms of the generic partial-area taxonomy representation, making the processes applicable to all types of partial-area taxonomies that can be derived in the OAF. The "define once, apply everywhere" approach that is enabled by the OAF saves significant amounts of time and research effort. One no longer needs to create different software tools to derive partial-area taxonomies for ontologies in different formats. The OAF includes support for OWL, OBO Format, SNOMED CT's relational formats [24], and Apelon Distributed Terminology System (DTS) format [25].

Fig. 1 illustrates the process of creating a partial-area taxonomy in the OAF. Each of the major components of the process, (a) the generic ontology representation, (b) the generic partial-area taxonomy derivation framework, and (c) the visualization method are described in detail in Section 3.

The development of the OAF solves several significant problems that we have encountered during our research into the applications of partial-area taxonomies. First, the OAF combines all partial-area taxonomy tools (*i.e.*, those we previously developed for SNOMED CT, OWL ontologies, etc.) into one consistently defined framework. In the past, tools for deriving and visualizing partial-area taxonomies were developed independently without interoperability. Most functionality was not shared among the different tools. For example, disjoint partial-area taxonomies [26], a useful refinement of the partial-area taxonomy methodology, could be derived for SNOMED CT but not for OWL ontologies, since the functionality was implemented in a SNOMED CT-specific fashion.

Secondly, the development of the OAF for partial-area taxonomies is the starting point for a more general abstraction network tool. We have designed the OAF such that modules for other kinds of abstraction networks, such as the tribal abstraction network [17] and ingredient abstraction network [27], can be "plugged into" the OAF, enabling the derivation of these different kinds of abstraction networks.

We have implemented the OAF in two versions. The first is a standalone tool that includes support for OWL ontologies, OBO Format ontologies, SNOMED CT (in its relational format), and Apelon DTS format ontologies. For each of these ontology representations it is possible to derive partial-area taxonomies [11], disjoint partial-area taxonomies [26], aggregate partial-area taxonomies [7], and others (see Section 3) using the OAF.

In the second version, the OAF is a Protégé [1] plugin, supporting OWL and OBO Format ontologies. By integrating the OAF into Protégé (see Fig. 2), one of the most widely used ontology development tools, partial-area taxonomies can now be utilized during the ontology development process ("live" partial-area taxonomies). Prior to our implementation of the Protégé plugin, partial-area taxonomies (and abstraction networks in general) were derived in an *a posteriori* process. A partial-area taxonomy was created for a fixed version of an ontology, after it had undergone development or revision.

With the Protégé plugin version of the OAF, partial-area taxonomies are generated "on the fly," while the ontology is being edited. This enables an editor to see the broad impact of different modeling decisions on the complexity of an abstraction network and the ontology that it summarizes.

This paper is organized as follows. In Section 2 we review our previous work on abstraction networks, particularly partial-area taxonomies. We also review ontology editing tools and BLUSNO, our previously developed partial-area taxonomy tool for SNOMED CT. In Section 3 we describe how the OAF generically represents ontologies, partial-area taxonomies, and the partial-area taxonomies processing steps. In the Implementation section we review the functionality of the OAF, the user interface of the OAF, and the implementation of the Protégé Plugin with its "live partial-area taxonomy" functionality.

2. Background

2.1. Abstraction networks

We define an abstraction network [4] as a compact summary of an ontology's structure and content. Structurally, an abstraction network consists of nodes, connected by hierarchical *child-of* relationships, where each node represents a set of similar concepts from the original ontology. The nature of similarity varies from one abstraction network to another. For a review of the various types of similarity used in abstraction network derivation see work by Ochs et al. [20]. For an abstraction network to function as an effective summary of an ontology, the number of nodes is expected to be significantly smaller than the number of concepts in the underlying ontology. Nodes are organized into a hierarchy based on the subsumption relationships of the ontology. As mentioned in the Introduction section, we have previously utilized Abstraction Networks to support various use cases, such as ontology quality assurance.

In previous studies, we have created various kinds of abstraction networks that summarize different aspects of an ontology's structure and content. For example, we have developed the tribal abstraction network (TAN) [17], an abstraction network that summarizes groups of concepts based on their ancestor(s) within an ontology. Specifically, the TAN summarizes sets of concepts that exist at points of intersection among ontology's subhierarchies. The ingredient abstraction network (IAbN) [27], derived from the NDF-RT [28], summarizes drug concepts according to their chemical ingredients. The ingredient abstraction network is based on sets of concepts that are targets of lateral relationships. Additional kinds of abstraction networks and their properties are discussed by Halper et al. [4].

2.1.1. Area taxonomies and partial-area taxonomies—

We have derived [9–11,23,26,29] abstraction networks called *area taxonomies* and *partial-area taxonomies* for SNOMED CT [12], the National Cancer Institute thesaurus (NCIt) [13], the Gene Ontology (GO) [30], and other OWL and OBO Format ontologies. We have used these taxonomies to support quality assurance of these ontologies. Due to their importance for this paper, we will now review the derivation method of the area taxonomy and of the partial-area taxonomy. Specifically, we will derive area taxonomies and partial-area taxonomies for OWL ontologies (see Fig. 3).

Throughout this paper we will use the terms “class” and “concept” interchangeably, as our goal is to provide a generic approach to ontology summarization. We will use “concepts” in general and “classes” when referring to an OWL ontology, such as in the definition of an area taxonomy for NCIt, an OWL format ontology, below.

We define an ***area*** as a set of structurally similar classes in an OWL ontology. This set is then represented by an area node in the abstraction network (see Fig. 3b). We have previously defined two kinds of areas: *domain-defined areas* [9] and *restriction-defined areas* [10]. A domain-defined area node is a node that summarizes the set of all concepts that are explicitly defined or inferred as being in exactly the rdfs:Domain axiom(s) of a given set of properties (object properties, data properties, or both). A restriction-defined area is defined as a node that summarizes the set of concepts that are explicitly defined or inferred to be bound by restrictions that use the same property types (again, either object properties or data properties). For both kinds of areas we only consider the type of object property, not the range.

A *root class* of an area is a class that does not have any superclass in its area. This definition implies that its set of properties must be different from the set of properties of its superclass(es) in other areas. (Alternatively, a root class of an area might not have any superclasses in the entire ontology, for example *owl:Thing*). Each area has one or more root classes. An area has multiple root classes when there are multiple classes that have no parents within the area (e.g., *Cardiovascular neoplasm* and *Thoracic neoplasm* in Fig. 3a).

An ***area taxonomy*** is an abstraction network where the nodes represent areas and these nodes are connected by *child-of* links that are defined based on the underlying subclass hierarchy. Specifically, an area node A *is child-of* another area node B if a root class in the area *A* has a parent in the area *B*. Hence, there is a subsumption relationship path from every class in *A* to some class in *B*.

Based on the above definitions of two kinds of areas, we further distinguish between two types of *area taxonomies*: domain-defined area taxonomies and restriction-defined area taxonomies. It is possible to derive both using either object properties or data properties. These derivation methodologies can be combined to derive, for example, an object property-defined area taxonomy (either domain-based or restriction-based). We previously derived such an area taxonomy for the Sleep Domain Ontology (SDO) [10].

The root classes within one area are used to define ***partial-areas***. A partial-area is the set of classes in one area consisting of one root class in the area and all of the root's descendant classes strictly in the same area. A partial-area summarizes semantically similar classes within one area, since all these classes are descendants of the same root.

A ***partial-area taxonomy*** is an abstraction network where the nodes represent partial-areas. These nodes are connected by *child-of* links derived from the IS-A relationships between the root class(es) and its (their) parent class(es) in partial-areas that are contained in other areas. Partial-areas are not necessarily disjoint, since a class may be a descendant of multiple roots in one area. To take advantage of this observation, we have introduced another abstraction network, called the *disjoint partial-area taxonomy* [26]. The word “disjoint” indicates that in such an abstraction network every class belongs to exactly one partial-area. A partial-area taxonomy is always visualized as an “overlay” on top of an area taxonomy, and the area taxonomy must always be derived first. The partial-area taxonomy contains more details than the area taxonomy, but fewer details than the original ontology. The derivation of a restriction-defined area taxonomy and partial-area taxonomy are illustrated using an excerpt from the *Disease, Disorder, or Finding* hierarchy of NCIt (Fig. 3).

2.2. Ontology software

With over 290,000 registered users, Protégé, developed at Stanford University by Musen et al. [1], is one of the most widely used ontology development tools. Musen [31] provides an overview of the history of the Protégé project, which started in the 1980s. In its current incarnation, Protégé is used for developing OWL ontologies and it currently utilizes the OWL API [32]. Within Protégé, a user can create, edit, and browse ontology classes, properties, annotations, and other information. Protégé is extendable via plugins, enabling developers to add additional functionality into the Protégé system. The Protégé plugin library [33] lists over 100 plugins. Some examples of plugins include OWL Diff [34], for comparing different versions of an ontology, the HermiT reasoner [35], and OWLViz [36] for viewing graphical displays of ontologies. The Pizza Ontology Tutorial [37], which provides step-by-step instructions for developing OWL ontologies in Protégé, is a widely-used educational resource for new ontology developers.

WebProtégé [2] is a web-based version of Protégé that focuses on collaborative ontology development. Using WebProtégé a user can develop an ontology anywhere using a web browser. Notably, a version of WebProtégé is currently being used to support the development of the International Classification of Diseases 11th Revision (ICD-11) [2]. Similar to the desktop version of Protégé, WebProtégé is extendable via plugins. Other examples of ontology editing software tools include OBO Edit [3], an editor for OBO Format ontologies [22], the IHTSDO Workbench [38], which was used to develop SNOMED CT for several years, and the Apelon DTS Editor [39]. However, none of the above ontology editing systems includes functionality to create and visualize structural summaries of ontologies.

Motta et al. [40] describes KC-Viz, a system for navigating ontologies based on summaries created by identifying “key concepts” in an ontology. KC-Viz is part of the NeOn Toolkit [41], a tool for developing ontologies. Queiroz-Sousa et al. [42] describes OWLSumBRP, a system for deriving and visualizing personalized ontology summaries that are based on measurements of concept relevance. While these systems provide a user with an interactive visual summary of an ontology, they focus on identifying the “most important” concepts in the ontology using various centrality measures. In contrast, abstraction networks, such as partial-area taxonomies, summarize the structure and content of an ontology.

2.2.1. Biomedical Layout Utility for SNOMED CT (BLUSNO)—

Geller et al. [16] implemented the Biomedical Layout Utility for SNOMED CT (BLUSNO), a software system for deriving, visualizing, and exploring SNOMED CT area taxonomies and partial-area taxonomies. Prior to developing the BLUSNO system, we conducted SNOMED CT quality assurance research using algorithms that produced text-based descriptions of partial-areas. We created partial-area diagrams from these text files using generic diagram editing tools, a time consuming, error-prone and laborious task, often taking hours or even days. Implementing BLUSNO brought the process of creating partial-area taxonomy diagrams down to seconds. Furthermore, in the BLUSNO tool, a user could show and hide information on demand, providing an interactive browsing experience.

The BLUSNO system unified our disconnected tools into one consistent system and added partial-area taxonomy visualization functionality. The ability to quickly summarize and visualize SNOMED CT's content and structure in a variety of ways, using the BLUSNO system, has enabled much of our research into the quality assurance of SNOMED CT [6–8,17].

However, BLUSNO was designed only for SNOMED CT. It was not possible to derive partial-area taxonomies for other ontologies (*e.g.*, those released in OWL format). Thus, it was necessary to design a new system that would enable us to create and visualize partial-area taxonomies for ontologies released in various other formats.

2.3. Glossary of terms

To assist the reader in understanding the “scientific terminology” we have developed for abstraction networks, we provide a glossary of terms in Table 1.

3. Methods

The Ontology Abstraction Framework (OAF) is composed of three major “back end” components. The first component is a generic framework for representing ontologies. This component abstracts away differences and idiosyncrasies found among various ontology representations. Specifically, it removes syntactic and structural differences among different ontology representations that affect partial-area taxonomy derivation (and abstraction network derivation in general). The second component of the OAF is a generic partial-area taxonomy derivation module that utilizes the generic ontology representation to derive partial-area taxonomies for any ontology supported by the OAF. Various processes that can be applied to partial-area taxonomies are defined in terms of the generic partial-area taxonomy representation, making them applicable to all kinds of partial-area taxonomies. The third component is a generic user interface for visualizing and exploring partial-area taxonomies. The user interface provides the user with multiple, navigable views of the partial-area taxonomy that summarize the structure and content of the underlying ontology.

3.1. Generic ontology representation

While many current ontologies have the same core structure (*i.e.*, a hierarchy of concepts and lateral relationships used to further define concepts), among different ontology release formats there exist differences in how this information is represented. For example, in the SNOMED CT relational format, a hierarchical *IS-A* relationship between two concepts is represented as a *"concept$_1$, IS-A, concept$_2$"* row. In OWL, hierarchical rdfs:subClassOf relationships are expressed in an OWL class definition.

Furthermore, different ontology representations will have different types of information associated with each concept. In SNOMED CT a concept may have attribute relationships and several descriptions (*i.e.*, synonyms) [24]. In OWL, classes may have equivalence axioms, restrictions, and annotations; classes may be in the domain of one or more properties [21].

In the past, the differences in ontology representations necessitated that partial-area taxonomy derivation software be designed separately for each ontology representation. Each implementation of a partial-area taxonomy derivation algorithm only worked for the one ontology representation it was designed for. For example, we wrote one program (BLUSNO) for deriving partial-area taxonomies for SNOMED CT and another program for deriving partial-area taxonomies for OWL ontologies. To address this issue, the OAF provides functionality for representing any concept hierarchy using a set of standardized data types. Specifically, the OAF includes generic *hierarchy* and *concept* data types that abstract away the differences found among various ontology representations. A user only needs to define how the concepts (or classes) and hierarchical relationships are stored.

The derivation of partial-area taxonomies is dependent on lateral semantic relationships used to define the concepts of an ontology. As described in Section 2, the derivation methodology for OWL is based on object properties and data properties. For SNOMED CT, the derivation is based on the defining attribute relationships of each concept [11]. For OBO, the derivation is based on the relationships of the ontology [29]. To abstract away the differences between the various kinds of semantic relationships, we introduced a generic *inheritable property* data type. Again, within the OAF, a user only needs to define how the available items of information are stored within the ontology representation.

A user may also want to include information about the representational mechanisms of a specific ontology. For example, users may want to include the information whether a SNOMED CT concept is primitive or not and whether it is active or not, because SNOMED CT declares concepts as inactive in lieu of deleting them. For an OWL ontology, a user may want to store the annotations, individuals, etc. associated with each class. Within the OAF, the generic concept type can be extended to include such representation-specific information. For example, one could create a *SNOMED CT concept type* and an *OWL Class concept type* as children of the *generic concept type* of OAF.

Fig. 4 illustrates the process of representing an ontology in the generic OAF ontology representation using two examples: the SNOMED CT concept *Hematoma*, as stored in RF2 relational format [43] and the NCIt concept *Neoplasm*, as represented in OWL format [21]. SNOMED CT's RF2 format represents concepts, and the relationships between those concepts, in a set of tab-delimited text files.

3.2. Generic partial-area taxonomy derivation

Using the OAF's standardized ontology representation, it is possible to define a generic partial-area taxonomy derivation methodology “template” in terms of the standardized ontology representation. Thus, when support for a new ontology representation (*e.g.*, the Apelon DTS format) was needed, it was only necessary to define how the concepts are represented, how the concept hierarchy is represented, what the inheritable properties are, and how they are stored.

With the data types defined above, consider the following generic definition for the area taxonomy and the partial-area taxonomy. We note that it is very similar to the OWL derivation described for NCIt in Section 2, but now the derivation will work for any ontology with some kind of inheritable property. Table 2 provides four concrete examples of implementations. In the description below, we highlight the generic data types used in the derivation in bold.

An *area node* is an abstraction network node that summarizes all the **concepts** that have the same set of **inheritable property** types. An *area taxonomy* is an abstraction network where the area nodes are connected by *child-of* links that are derived according to the underlying **concept hierarchy**. An area *root* is a **concept** that has no parent **concepts** in its area. All areas are disjoint in terms of the **concepts** they summarize. An area is further refined into partial-areas, one for each root.

The root **concept(s)** of each area define *partial-area(s)*. A partial-area node summarizes a root **concept** and all of its descendant **concepts** in the same area. Partial-area nodes summarize semantically similar **concepts** within each area, since all are descendants of the same root **concept**. A *partial-area taxonomy* is an abstraction network where the nodes are partial-areas that are connected by *child-of* links derived from the hierarchical relationships between each root **concept** and its parent **concept(s)** in other partial-areas. The *partial-area taxonomy* is a refinement of the area taxonomy.

With this generic definition of an area taxonomy and a partial-area taxonomy, the same derivation algorithm can be applied to derive SNOMED CT partial-area taxonomies and partial-area taxonomies for OBO Format ontologies (e.g., the Gene Ontology). Furthermore, the same algorithm can derive the four kinds of OWL partial-area taxonomies described in Section 2 (domain-defined partial-area taxonomies and restriction-defined partial-area taxonomies, each using either object properties or data properties, or combinations thereof).

Just as many ontologies share the same basic structure, different kinds of partial-area taxonomies also exhibit the same basic structure (*i.e.*, a hierarchy of partial-area nodes within a hierarchy of area nodes). To simplify the process of representing partial-area taxonomies, the OAF uses a standardized representation that abstracts away differences found among different kinds of areas and partial-areas that come out of deriving generic partial-area taxonomies for SNOMED CT, OWL, etc. The generic representation of a partial-area taxonomy is the output of applying the above generic derivation methodology.

Within the OAF, each *area node* of an area taxonomy consists of **(1)** a set of concepts and **(2)** the set of *inheritable properties* used to define those concepts. The area taxonomy consists of a hierarchy of area nodes. In a generic partial-area taxonomy, a *partial-area node* represents a singly-rooted subhierarchy of concepts within an area. Thus, each partial-area node only needs to store that subhierarchy of concepts. The area nodes in a partial-area taxonomy, however, must be modified (compared to an area taxonomy) to include their sets of partial-areas.

3.3. Generic partial-area taxonomy processes

Various processes for partial-area taxonomies have been developed to control the types and the amount of information presented to a user. By defining a generic partial-area taxonomy representation, one can define processes that are applicable to all partial-area taxonomies by defining them accordingly. For example, we have developed the *disjoint* partial-area taxonomy [26] to "carve out" the concepts that are summarized by multiple partial-areas (what we call "overlapping concepts," *e.g.*, *Non-neoplastic Heart Disorder* in Fig. 3). We also introduced the *aggregate* partial-area taxonomy [7], which provides a parametric method for controlling the granularity and size of a "big picture" partial-area taxonomy display.

In the past, these processes had to be redefined and re-implemented in our software tools every time support for a new ontology representation was needed. For example, we could derive disjoint partial-area taxonomies and aggregate partial-area taxonomies for SNOMED CT but not for OWL ontologies. Since partial-area taxonomies are represented consistently within the OAF, it becomes possible to define these processes generically for all kinds of partial-area taxonomies. We will now describe some of the processes that are applicable to partial-area taxonomies within the OAF. They will be illustrated in more detail using examples from the NCIt in Section 4.

3.3.1. Disjoint partial-area taxonomy derivation—Partial-area taxonomies are not necessarily disjoint [26], as illustrated in Section 2. While areas are by definition disjoint, two or more partial-areas within an area may summarize the same concept, because this concept has IS-A paths to two or more root concepts within the same area. The disjoint partial-area taxonomy derivation methodology [26] identifies these "overlapping" concepts in a partial-area taxonomy and creates *disjoint partial-area nodes*. After this methodology is applied, each concept is guaranteed to be summarized by exactly one disjoint partial-area node. Due to space limitations we do not describe the disjoint partial-area taxonomy derivation methodology in this paper. However, the full methodology is described in detail by Wang et al. [26]. Disjoint partial-areas provide a more complete picture of the semantics of the concepts within an area. Disjointness is a desired property for certain use cases (*e.g.*, quality assurance [8,44]).

3.3.2. Aggregation of partial-area taxonomies—In a previous study [7], we described the *aggregate partial-area taxonomy*, which is a partial-area taxonomy where "small" partial-areas that summarize a number of concepts below some given threshold, are aggregated into their direct large ancestor partial-area(s). One *aggregate partial-area* summarizes one or more smaller partial-areas. Aggregation reduces the overall number of partial-area nodes displayed in a partial-area taxonomy abstraction network, providing an even more compact summary of an ontology. The hidden smaller partial-areas can be "recovered on demand" by choosing an aggregate partial-area node and creating (by mouse click) an *expanded subtaxonomy* from the subhierarchy of small partial-areas that it summarizes [7]. Aggregation is important for situations where the partial-area taxonomy, while significantly smaller than the underlying ontology, is still too overwhelming to be used. The aggregate partial-area taxonomy provides a user-controlled, compact "big picture" of the ontology.

3.3.3. Root subtaxonomies and ancestor subtaxonomies—Often a user working on quality assurance wants to focus on a small portion of a partial-area taxonomy [8,16,45]. For example, the user may be interested in only subjects corresponding to a particular partial-area and all of its descendant partial-areas [45], providing a picture of the structure and content of a subhierarchy of concepts. Alternatively, the user may want to view all of the ancestor partial-areas of a selected partial-area, summarizing how the selected partial-area obtained its structure by inheritance. The OAF includes support for deriving both kinds of subtaxonomies.

3.4. Partial-area taxonomy visualization

The user interface of the OAF is composed of two components. The first is the partial-area taxonomy display, which is modeled after our previously developed partial-area taxonomy visualizations (*e.g.*, Fig. 3(c) and those of Ochs et al. and Wang et al. [9,11,29]). This display is interactive and dynamic, allowing a user to navigate and select different partial-area taxonomy components (*e.g.*, area nodes and partial-area nodes). The second is an interface for displaying information about individual partial-area taxonomy elements. For example, when a user selects a partial-area node the user will be presented with appropriate information and provided with options that are relevant to the concepts summarized by this node.

3.4.1. Partial-area taxonomy display—Visualization of partial-area taxonomies is the most important part of the OAF, as it is how users interact with the system. The visualization component is designed around the model-view-controller (MVC) architecture [46]. In the OAF, the model is the partial-area taxonomy that is generated by the generic partial-area taxonomy derivation algorithm. This model is independent of any visualization of a partial-area taxonomy. The view component of the OAF is a system for visually representing partial-area taxonomy elements in a variety of ways. Specifically, each partial-area taxonomy element is associated with a *visualization* of the element.

For example, for each area in a partial-area taxonomy there is an associated instance of an *area visualization* in the view. An area visualization is what a user sees on the screen in the OAF. The visual "look and feel" of different partial-area taxonomy elements is customizable and can be easily changed to support visualization schemes for different kinds of partial-areas taxonomies.

In the default view, the visualization of each partial-area taxonomy element is based on the visual style for partial-area taxonomies defined by Wang et al. [11] (*e.g.*, Fig. 3) and other previous publications. However, this visualization may be changed as needed. For example, in comparison to the default view of partial-area taxonomies (see Fig. 2), the "look and feel" of partial-areas is different for disjoint partial-area taxonomies (see Fig. 5, which follows the visual style of Wang et al. [26]) and aggregate partial-area taxonomies (see Fig. 6, which follows the visual style of Ochs et al. [7]).

Each visualization element in a partial-area taxonomy view is selectable. Different user actions are available for different kinds of partial-area taxonomy elements. For example, when a user single-clicks on a partial-area the appropriate parent partial-areas and child partial-areas are highlighted by a color change, identifying the *child-of* links within the partial-area taxonomy (see Fig. 2).

The visual elements of a partial-area taxonomy are organized on screen via a *taxonomy layout*. A taxonomy layout is a generic way of representing the placement of partial-area taxonomy visualization elements on the screen. The default taxonomy layout, illustrated in Figs. 2 and 3(c), organizes area nodes into levels according to their numbers of inheritable properties. At each such level, area nodes are organized so that the largest (*i.e.*, those that summarize the most concepts) are in the middle, focusing a user's attention on the largest groups of structurally similar concepts in the ontology. Within each area node, partial-area nodes are organized into a grid with the largest partial-area nodes at the top left, focusing a user's attention on the larger groups of semantically similar concepts within each area node.

Different taxonomy layouts can be defined, providing different views of a partial-area taxonomy. For example, one can use a layout that organizes partial-areas into regions (partitions based on inheritance of *inheritable properties*, as defined by Wang et al. [11]) within each area. For the disjoint partial-area taxonomy, the layout organizes disjoint partial-areas according to their locations within an area (see Fig. 5).

3.4.2. Partial-area taxonomy element details display—Each partial-area taxonomy element (*i.e.*, each area node, partial-area node, *child-of* link, and even the partial-area taxonomy itself) is associated with a "details display" that provides information about the element and the underlying portion of the ontology that it summarizes. These displays are designed to provide a user with the most relevant information. The OAF includes a diverse set of generic user interface elements for displaying information about partial-area taxonomy elements and ontology elements (*e.g.*, searchable and sortable lists of partial-areas) that can be used throughout the OAF.

When no elements are selected within the partial-area taxonomy, a user is presented with the *partial-area taxonomy details display*, which shows a short textual summary of the partial-area taxonomy (*e.g.*, number of areas, number of partial-areas) and "help text" describing how the partial-area taxonomy was derived and what each element represents. Additionally, tabs that provide a summary of the structure of the partial-area taxonomy (*e.g.*, number of areas, partial-areas, overlapping concepts at a given partial-area taxonomy level) and a searchable list of areas and partial-areas in the partial-area taxonomy are displayed.

When a user selects an element from within the partial-area taxonomy view, the partial-area taxonomy details display is replaced with an *area details display* or a *partial-area details display*, depending on whether the user selected an area or a partial-area. These displays provide details about the concepts summarized by the area or partial-area. They also provide information about the location of the element in the partial-area taxonomy. For example, one can obtain a list of parent or child partial-areas in the partial-area details display. Both of these displays also provide the user with context-based option menus. For example, when a partial-area with descendant partial-areas is selected, it is possible to create a root subtaxonomy with that partial-area as the root node. If an area has any overlapping partial-areas (*i.e.*, there are concepts that belong to two partial-areas) then an option to derive a disjoint partial-area taxonomy will be available to the user.

3.5. Supporting ontology quality assurance

We are currently using the OAF to support several quality assurance studies of NCIt and other biomedical ontologies (e.g., SNOMED CT, FMA, and GO). One way of using the OAF to support ontology quality assurance is reviewing a partial-area taxonomy for anomalies. If there is something unusual in the partial-area taxonomy (e.g., an unusual grouping of concepts in an area or partial-area, or an irregularity in the hierarchy of *child-of* links between partial-areas) then this may indicate errors in the underlying ontology. Prototype versions of the OAF were used to review partial-area taxonomies for such inconsistencies in OCRe, SDO, CanCo, DDI, ERO, and GO, among others [9,10,18,29,47].

For example, in a partial-area taxonomy for OCRe [14] there was a partial-area Relative time point (1), which was the only partial-area with multiple *child-of* links. This can be considered an anomaly within OCRe's partial-area taxonomy and the underlying cause should be reviewed. Indeed, when we reviewed the root class of the partial-area (the concept *Relatively time point*) we identified an erroneous superclass *Time interval*. In Ochs et al. [9] we describe the cause of this error and how OCRe's curator corrected it, along with additional examples of errors and inconsistencies found in OCRe using the partial-area taxonomy.

Furthermore, in a partial-area taxonomy derived for GO, we identified [29] an area {*regulates*} which had three partial-areas: *regulation of biological process* (2901), *regulation of molecular function* (192), and regulation of mammary gland cord elongation by mammary fat precursor cell-epithelial cell signaling (1). The last partial-area summarized just one concept. When the modeling of that concept was reviewed by GO's curator, it was determined to be missing a sequence of ancestors to a concept in the *regulation of biological* *process* (2901) partial-area. For more details of this error, and other anomalies discovered in GO, see Ochs et al. [29].

In previous work we also developed several partial-area-taxonomy-based methodologies for identifying groups of concepts that are expected to have a higher rate of error. These methodologies are based on identifying sets of concepts that exhibit certain partial-area-taxonomy-defined characteristics. In previous studies we have found that concepts with certain characteristics are more likely to have an error than concepts that do not have the characteristic. Typically, these characteristics identify concepts with uncommon modeling or relatively complex modeling. In general, there are relatively few concepts with a given characteristic in an ontology (e.g., there are only 292(/25,680 = 1.1%) overlapping concepts in the partial-area taxonomy derived for NCIt's *Disease, Disorder, or Finding* hierarchy). For an overview of these characteristics see Table 2 in Ochs et al. [20].

The OAF includes functionality for identifying sets of concepts that exhibit these characteristics, enabling a user to investigate the modeling of these concepts and similarly modeled concepts (*i.e.*, those summarized by the same areas and partial-area(s)). This way, the OAF can be used to support ontology quality assurance and partial-area-taxonomy-based quality assurance studies. We will now provide examples of several such characteristics and preliminary resulting errors and inconsistencies from quality assurance studies of NCIt that were supported by the OAF (see Table 3). The full details of these studies are outside of the scope of this paper and they will be described in detail in future publications.

In previous studies [23,45,48] we found that concepts summarized by relatively small partial-areas (*i.e.*, those that summarize only a few concepts) have a statistically significant higher error rate than concepts summarized by larger partial-areas in NCIt and SNOMED CT. In a current study we are currently reviewing the concepts that are summarized by small partial-areas in NCIt's *Disease, Disorder, or Finding* hierarchy. Another methodology is enabled by the disjoint partial-area taxonomy. We have found [8,29,44] that overlapping concepts [26] (concepts summarized by multiple partial-areas) have a higher error rate than non-overlapping concepts in SNOMED CT and GO. In a current study, enabled by the OAF, we are investigating the error rate of overlapping concepts in NCIt's *Neoplasm* subhierarchy.

We have also observed that the concepts in a relatively large top area of a partial-area taxonomy (*i.e.*, all of the concepts in the hierarchy that have no relationships) are often more likely to have errors, particularly missing relationships. This characteristic was observed in a recent study we performed on GO [29]. We are currently reviewing the top area concepts in a partial-area taxonomy for NCIt's *Biological Process* hierarchy. The initial results of this study indicate that many of these concepts are missing relationships relative to concepts that have at least one relationships. Finally, we have found that concepts with relatively many kinds of relationships (*i.e.*, concepts summarized by areas in higher indexed levels in the partial-area taxonomy) may tend to have more errors. These concepts were observed to have higher error rates in GO [29] and in a recent study we have observed a similar phenomenon in NCIt's *Biological Process* hierarchy.

On its own the OAF does not identify or correct errors in ontologies. The purpose of the OAF is to provide ontology editors and domain experts with a different view of an ontology's structure. As noted by Lanzenberger et al. [49] and Fu et al. [50], different kinds of ontology visualizations have different advantages and support various use cases. In the studies mentioned above, we have shown the view provided by a partial-area taxonomy supports quality assurance in various ways. Thus, the OAF, which creates and enables exploration of these views, is critical for partial-area-taxonomy-based quality assurance.

The process of identifying and correcting errors using the OAF relies on a domain expert's time and expertise. It requires a manual review of the taxonomy and/or concepts in the taxonomy that exhibit certain characteristics. However, reviewing a partial-area taxonomy summary of an ontology, or a subset of concepts chosen from the ontology based on the partial-area taxonomy, typically requires significantly less work than exhaustively reviewing each concept.

When creating an ontology there are different ways of modeling a concept. There may not be agreement among domain experts and ontology editors about which is “most correct.” A similar phenomenon is observed in manual ontology quality assurance reviews (e.g., those supported by the OAF). In Gu et al. [51] we analyzed quality assurance reports from four domain experts. We found that, individually, none of the reviewers was reliable. Only after a round of consensus, where each domain expert agreed or disagreed with the errors identified by the other domain experts, was a reliable result obtained. A similar result is discussed by Mortensen et al. [52] in the context of crowdsourced ontology quality assurance.

In our OAF-supported quality assurance studies, such as those we are currently performing on NCIt, we take steps to guarantee the quality of error reports. First, we employ, whenever possible, multiple reviewers and create a consensus report (e.g., in [8,51]). Additionally, we work very closely with the curators of an ontology to verify the correctness of the errors we identified. For example, in [9,10,17,18,23,29,47] we collaborated with the curators of SNOMED CT, NCIt, etc., and only reported the errors the curators confirmed. At the end of Section 5.2 we briefly describe future functionality of the OAF that will support this consensus-based and collaborative quality assurance approach.

4. Implementation

The Ontology Abstraction Framework is implemented using Java 8. The OAF currently supports SNOMED CT (RF1 and RF2 relational formats, inferred and stated releases), OWL ontologies, OBO Format ontologies, and ontologies in Apelon DTS relational format. OWL and OBO support is enabled by the OWL API [32]. The OAF exists in two versions: a standalone tool that works with all ontology formats, and a Protégé plugin that supports OWL and OBO Format ontologies inside of the Protégé ontology editing environment. The user interface is essentially the same in both versions of the tool. The only difference is that the standalone version of the tool includes functionality to load SNOMED CT and Apelon DTS ontologies and a user interface for opening ontologies in various formats. Both versions of the OAF are currently available at [53] in “beta” form.

We will now illustrate the functionality of the OAF using NCIt's *Disease, Disorder, or Finding* partial-area taxonomy, as displayed in the Protégé plugin version of the tool (see Fig. 2). Within the OAF, a user can derive a partial-area taxonomy for the entire ontology, a specific hierarchy, or any subhierarchy of concepts at a chosen root (*i.e.*, subject subtaxonomies [8]).

The default user interface of the OAF, shown in Fig. 2, is organized as follows. On the left side the current view of the partial-area taxonomy is displayed to the user. The user can freely navigate the taxonomy using the mouse, keyboard, or the arrow buttons displayed as part of the user interface. A user can zoom in and out to view the taxonomy at different scales. Within the display, each taxonomic element is selectable and selecting it provides additional details about the concepts it summarizes. For example, selecting a partial-area will show the *partial-area details display* for that partial-area (right hand side of Fig. 2).

At the top of the display window the numbers of areas, partial-areas, and concepts are displayed. Additionally, various option buttons are shown. For example, the “Reports and Metrics” button displays a dialog with various additional metrics for the current taxonomy. One report explicitly identifies the individual overlapping concepts [26] in a partial-area taxonomy, along with what partial-areas they are summarized by. For partial-area taxonomies derived for OWL ontologies another report identifies where imported content [54] (*i.e.*, classes and properties imported from other ontologies) is summarized within the partial-area taxonomy.

The “Derivation Options” menu provides a user with various options for deriving a partial-area taxonomy. From this menu a user can derive aggregate partial-area taxonomies (see Fig. 6). For OWL partial-area taxonomies the user can select the type(s) of properties used in the derivation of the partial-area taxonomy (object properties or data properties) and the usage of that property (the property's domain or its use in restrictions) from this menu.

When an area is selected additional relevant information will be displayed, including a list of concepts summarized by the area and whether there are any overlapping concepts in the area. When overlapping concepts exist in an area, various metrics are provided (Fig. 7b). For example, the user can see how many overlapping concepts exist in each overlapping partial-area.

Selecting a partial-area with overlapping concepts will display a list of which partial-area(s) the concepts overlap with and how many concepts overlap. For example, in Fig. 7(b) the *Cardiovascular Disorder* partial-area, which summarizes 148 overlapping classes, has been selected. In the bottom window of Fig. 7(b) the list of partial-areas that overlap with *Cardiovascular Disorder* are shown. In this window the user can see that 126 classes overlap between the *Cardiovascular Disorder* and *Respiratory and Thoracic Disorder* partial-areas. This information makes it possible to determine where the majority of overlap occurs within the area. If a selected area has overlapping concepts then it is possible to derive a disjoint partial-area taxonomy in that area. Fig. 5 shows the disjoint partial-area taxonomy for the {*Disease has associated anatomic site*} area, which has 148 overlapping classes.

Fig. 6 illustrates an aggregate partial-area taxonomy derived for NCIt's *Disease, Disorder, or Finding* hierarchy, created using a bound of 20. For this example, the choice of 20 as a bound was arbitrary for this example to create an aggregate taxonomy that fits on one screen. The bound for aggregation is user-specified. In Ochs et al. [7] we describe some reasonable choices for bounds. Smaller partial-areas that summarize fewer than 20 classes each are no longer displayed. The user interface for *aggregate* partial-area taxonomies follows the user interface designed for regular partial-area taxonomies. However, additional information (*e.g.*, which smaller partial-areas are summarized by an aggregate partial-area) and additional functionality are provided. For example, users can create expanded subtaxonomies that display the small partial-areas summarized by a larger aggregate partial-area.

4.1. Live partial-area taxonomies

In our previous research, partial-area taxonomies were derived and used in an *a posteriori* process. Given a single, fixed release of an ontology, we derived a static partial-area taxonomy for that release, using our previous generation software tools. The partial-area taxonomy was then used to support the intended use case (*e.g.*, quality assurance). However, the partial-area taxonomy was not updated during the process. Whenever an editor used a partial-area taxonomy for quality assurance, identified a problem, and corrected the problem, the visual display of the partial-area taxonomy did not change. The editor had to rederive a new partial-area taxonomy and display it. This limitation significantly affected the utility of those tools; ontology editors could not see the effects of their changes on the partial-area taxonomy without performing a cumbersome multi-step process of recreating it.

By integrating partial-area taxonomies into an ontology editing tool (in this case Protégé) it becomes possible to provide an ontology editor with a structural summary of the ontology as the user is editing it. We define a *live partial-area taxonomy* as a partial-area taxonomy that is immediately updated when changes are made to the underlying ontology. For example, when a new concept is added to the ontology, the partial-area taxonomy is automatically updated to reflect which partial-area(s) that concept is summarized by. If an object property domain is modified then the tool immediately shows which concepts this change affects.

Live partial-area taxonomies are supported by the Protégé plugin version of the OAF. When a user edits an ontology, the OAF automatically updates the partial-area taxonomy view to reflect the changes. The partial-area taxonomy view “snaps” to the partial-area taxonomy element (*i.e.*, area or partial-area) that was affected by the editing operation. If multiple elements are affected than a user has the option of selecting which one they are interested in. This functionality allows an editor to quickly obtain the “big picture” of the effects of the changes that the user made to the ontology. In the Protégé plugin, a user can seamlessly transition between the standard editing view provided by Protégé and the partial-area taxonomy view provided by the OAF. Within the OAF a user can click on classes and properties to switch to Protégé's editing view. Similarly, from within Protégé's editing view an editor can navigate to the location where a given class is summarized in the partial-area taxonomy. Furthermore, with live partial-area taxonomies, a user can seamlessly transition back and forth between partial-area taxonomies for the stated version of an ontology and the inferred version of an ontology.

Different kinds of ontology editing operations affect a partial-area taxonomy differently. Ochs et al. have written an OAF tutorial [55] that parallels the Pizza Ontology tutorial [32], to help educate new OAF users. The Pizza Ontology tutorial describes the design of an ontology about different kinds of pizza. In the OAF tutorial, we illustrate how each editing operation applied to the Pizza Ontology is reflected in the live partial-area taxonomy display.

5. Discussion

The development of the Ontology Abstraction Framework (OAF) represents a significant advance in abstraction network tools (and tools for ontology summarization, in general). By formulating a standardized format for the representation of ontologies, we were able to define a uniform generic partial-area taxonomy derivation methodology that is applicable to many ontologies in various source formats. The OAF replaces several disconnected software tools that we previously used to derive and visualize partial-area taxonomies for SNOMED CT, OWL ontologies, and others. The OAF unifies these separate software systems into one consistent framework and tool. With the OAF, processes (*e.g.*, aggregation) can be uniformly defined in terms of the generic partial-area taxonomy representation, making them applicable to all types of partial-area taxonomies.

Traditional ontology visualization tools, like those provide by Jambalaya [56] and OWL Viz [36], present users with linked-node diagram visualizations of an ontology. These kinds of visualizations quickly become overwhelming as more concepts and relationships are added [50]. In comparison, the OAF is a system for creating and visualizing *summaries* of ontologies, with a focus on presenting users with visual information about the overall structure of an ontology in a manageable way. By using subtaxonomies and aggregation, an OAF user can obtain a compact summary that captures structural information about thousands of concepts on a single screen.

The OAF is significantly different from existing ontology summary visualization tools. KC-Viz [40] and OWLSumBRP [42] focus on identifying key concepts to provide an overview of the types of concepts in an ontology. The OAF can also support this kind of content comprehension, but its current use case is supporting ontology development and quality assurance via partial-area taxonomies. The visualizations of ontology summaries created by the other tools are displayed node-link diagrams, similar to those created by OWL Viz, where each node represents a key concept. In contrast, the OAF provides a greater amount of structural information, useful for ontology quality assurance, via its partial-area taxonomy display (e.g., information about relationship introduction and inheritance).

The most significant difference between these tools is that the OAF is a generic framework for creating summaries of ontologies in various formats. In a literature review we have found no comparable framework. The above mentioned ontology summarization tools are more *ad-hoc*; they are only applicable to OWL ontologies and they only support one kind of summarization. The OAF, on the other hand, supports various kinds of ontologies summaries (the status of additional OAF modules that implement different kinds of abstraction networks is described in Section 5.2). Furthermore, since the OAF is a framework, the summarization techniques implemented in KC-Viz and OWLSumBRP could be implemented as OAF modules.

The OAF is currently being used to support many aspects of our research on family-based [18] ontology quality assurance. By unifying all of our partial-area taxonomy tools into the OAF, we are now able to apply all of our previously developed partial-area taxonomy-based methodologies [8,9,11,48] for quality assurance to all of the ontologies supported by the OAF. For example, using the OAF we have investigated errors in overlapping concepts [26] in SNOMED CT [8], the Gene Ontology (GO) [29], and recently in NCIt. Prior to the OAF system it was impossible to apply such methodologies across ontologies released in different formats, since the various partial-area taxonomy tools did not share common functionality.

The OAF serves as both an end user tool (in the form of a standalone user interface and a Protégé plugin) and as an application programming interface (API) for developing software that utilizes abstraction networks and ontologies. For example, Ochs et al. [20] analyzed the structure of over 350 ontologies hosted on BioPortal and categorized them into structurally similar families. The structural analysis was enabled by various components developed for the OAF (e.g., the generic ontology data types introduced in Section 3.1). The next step in this line of research is to investigate the characteristics of the partial-area taxonomies for those 350 BioPortal ontologies. This will require a program to automatically derive various kinds of partial-area taxonomies for large sets of ontologies. This functionality is supported by the generic partial-area taxonomy derivation mechanisms developed for the OAF.

The integration of partial-areas taxonomies into the Protégé ontology editing environment and the development of live partial-area taxonomies represent major steps toward integrating abstraction networks into the ontology design and editing workflow. Prior to the OAF it was not possible to obtain a summary of the content and structure of an ontology “on the fly.” Partial-area taxonomies could only be derived for one specific ontology release. The opportunity of using partial-area taxonomies during the ontology development process, which is enabled by the OAF, opens up a significant new approach to ontology development. The summary provided by an abstraction network enables a “big-picture-guided” approach for developing the content of an ontology.

5.1. Preliminary user evaluation studies

In this paper we presented the theoretical foundation and implementation of a software system for deriving and visualizing abstraction networks. However, the usability of complex software systems like the OAF must be extensively evaluated. The OAF (and its predecessors, e.g., BLUSNO) has been used to support our abstraction network research for over six years. For the majority of that time the OAF system was not publicly available. Internally, we organized several feedback cycles to evaluate the functionality and usability of the OAF. These internal evaluations resulted in significant changes to the user interface and functionality of the OAF. However, these evaluations never considered external user feedback or feedback from user interface experts.

To thoroughly evaluate the usability of the OAF we have formed an interdisciplinary research team with members who have expertise in user evaluation and user interface design. This team has conducted two preliminary studies. In the first, a new user was asked to perform 32 common partial-area taxonomy tasks using the OAF for the HDOT ontology. The tasks varied in complexity and time required. Each task required the user to either perform a procedure (e.g., create a certain kind of partial-area taxonomy or find a partial-area that contains a certain concept) or query to answer a question (e.g., determine how many concepts are modeled using a specific set of relationship types).

The user's interactions with the OAF were recorded while solving the tasks and then analyzed. A goal-action coding scheme was employed to characterize the user's interactions [57]. Each task was evaluated according to four criteria: Goal, Duration, Problem, and Prompts. For example, one task was “determine how many areas have more than one partial-area” in the partial-area taxonomy. This task mimics searching for sets of relationships that are introduced at multiple points in an ontology's hierarchy. This task was applied in the context of quality assurance in our previous studies on OCRe [9] and the SDO [10]. Table 4 illustrates the goal-action coding for the user's completion of the task. Additionally, a user interface expert provided annotated screen captures of where the user encountered issues. Using this information we are able to identify areas of improvement for information layout and workflow. For example, while the user was able to complete most of the tasks relatively quickly, the task described in Table 4 took him a relatively long time.

In the second evaluation, two domain experts, who both have medical domain knowledge and are familiar with partial-area taxonomies, were asked to perform the same tasks as the user in the first study, as well as some tasks that utilized their medical expertise. These sessions were also recorded and significant usability issues were uncovered. Both domain experts expressed criticism of the OAF's interface from the “educated user” perspective and provided extensive feedback about how the OAF interface could be improved.

Based on the results of these preliminary studies, and the expertise of the research team, several major changes to the OAF user interface are currently being designed and implemented. A more extensive evaluation, with the goal of assessing the impact of the changes made as a result of the preliminary studies, is currently in progress. This evaluation expands on the procedures developed for the first evaluation described above. A set of 14 users are being assigned approximately 50 tasks to be completed in the OAF. Each user's interactions will be recorded and goal-action coding will be performed for each user performing each task. The 14 users are separated into two groups to evaluate variations on the partial-area taxonomy display and differences in the OAF user interface. Based on this study we will be able determine the types of usability and learnable issues still need to be addressed.

In addition to the current evaluation study we are planning additional evaluation studies involving external users who have extensive ontology development experience but are not necessarily familiar with abstraction networks.

5.2. Future work

In future studies, we will investigate the hypothesis that live partial-area taxonomies can support the ontology development process, leading to better ontologies and a shortened development cycle. Looking forward, the OAF is being designed as a general abstraction network derivation and visualization system. The partial-area taxonomy module of the OAF, the topic of this paper, was completed. Additional modules, adding support for other kinds of abstraction networks, are in various stages of development. For example, a tribal abstraction network (TAN) [17] module and an ingredient abstraction network module [27] for OAF are under development.

An especially important module that is in development is the *diff abstraction network* module. Introduced by Ochs et al. [5] as a standalone process, difference (diff) partial-area taxonomies summarize the structural differences between two ontology releases. This module will support the derivation of *diff partial-area taxonomies* as part of the OAF. Diff partial-area taxonomies represent a major new use case for ontology summarization. This functionality will be integrated with live partial-area taxonomies in OAF. With this extension, an editor of an ontology will be able to obtain a live, dynamic summary of which parts of the ontology were affected by each editing operation. The editor will see the “big picture” of the effects of the changes as they are being made, enabling the user to detect potential errors as the ontology is being edited, and take steps to avoid any newly created errors.

Many of the processes developed for partial-area taxonomies (*e.g.*, aggregation) are also applicable to other abstraction networks. For example, Ochs et al. [27] utilized aggregation to reduce the size of the NDF-RT ingredient abstraction network. This work was supported by an early version of the ingredient abstraction network module. Within the OAF it is possible to generalize the implementation of such processes to other kinds of abstraction networks.

Prior to the OAF, all of our abstraction network tools were not publicly available. This meant that users external to our research group, many of whom have extensive ontology design and development expertise, did not have access to these systems. Indeed, an external users’ only knowledge of partial-area taxonomies would come from one of the papers we have published on the topic. Due to limited exposure to the methodology, the utility of abstraction networks created by the OAF may not be obvious.

To help address this issue, we have integrated a system of “help text” into the OAF that explains and illustrates what each partial-area taxonomy element represents. Additionally, a series of short instructional videos have been created to illustrate the functionality of the OAF. Finally, we created a Pizza Ontology partial-area taxonomy tutorial [55] that illustrates the principles of partial-area taxonomies and the functionality of the OAF. This tutorial has the advantage that it mimics the existing pizza-based ontology tutorial [37] that is familiar to many workers in the field. Even to ontology engineers who have not seen that previous tutorial, the domain of pizza and pizza toppings is intuitively easy to grasp, and the reader can concentrate on understanding the partial-area taxonomy methodology, instead of struggling with an unknown (medical) domain *and* the OAF software tool.

To overcome issues of user acceptance, we are working on implementing a “dual facet user interface” for the OAF. One interface facet would be designed for new users and it would display a limited subset of options and information. The second facet view would be designed for experienced users and it would provide the full functionality and include complete information about the partial-area taxonomy. In the future, it may also be useful to create different user interface facets for different use cases.

We are currently planning a proper usability evaluation study to determine what additional steps are needed to assist ontology editors in integrating the OAF into their workflow. A group of ontology curators will be asked to evaluate the OAF and its utility for supporting ontology development.

The precursors of the OAF have been used extensively to support our quality assurance studies [8,9,17,29]. Thus, the OAF user interface (*e.g.*, the data provided in the various displays) has naturally evolved around supporting this use case. We have not yet accumulated experience with applying the OAF to other use cases, such as ontology development. In a future study, we will evaluate the types of information displayed within the OAF as required for other use cases (*e.g.*, evolution tracking, ontology development). Particularly, we will look at how well the visualization of partial-area taxonomies and the detailed information provided by the OAF support these use cases.

Another issue is related to the amount of information displayed. For example, certain partial-area taxonomies can be overwhelming in size, even though they are much smaller than the underlying ontology. The *Procedure* partial-area taxonomy and the *Clinical finding* partial-area taxonomy for SNOMED CT each contain over 10,000 partial-areas [8]. The partial-area taxonomy for the Gene Ontology *Biological process* hierarchy [29] has over 1,700 partial-areas. The partial-area taxonomy for the NCIt *Disease, Disorder, or Finding* hierarchy has over 5000 partial-areas (Fig. 2).

A display of these partial-area taxonomies is too large to be useful. Thus, it will be necessary to design a system that *automatically* presents a user with a view that is compact and useful. We are currently investigating various heuristics to derive an *aggregate partial-area taxonomy* automatically, so that the user will never see a partial-area taxonomy with “too many” partial-areas. Instead of showing a partial-area taxonomy with too many details and letting the user fend for himself, the system will automatically apply the aggregate operation when a partial-area taxonomy becomes too large. If the users then desire to get more details, they can turn off the aggregation functionality or only select certain partial-areas of interest for expansion.

The automated aggregation mechanism can be combined with root subtaxonomies and/or ancestor subtaxonomies. This will make it possible to automatically generate a partial-area taxonomy screen display that is both targeted to support a certain use case and not overwhelming. An extension of the OAF will use various heuristics, combined with metrics derived from the partial-area taxonomy, to automatically control the amount of information displayed to a user. Such a system will be extensively evaluated by usability experts.

One limitation of the OAF visualization is that the layout of information does not change in response to user actions. A user can scroll through the display, zoom in, zoom out, and select different elements, but the layout of the areas and partial-areas does not change. A more dynamic system that adapts as a user navigates could potentially further address the issue of displaying too much information. For example, when a user zooms out, instead of displaying a larger cross-section of the complete partial-area taxonomy it may be more useful to instead display progressively less information (*e.g.*, when the user zooms out, the system could only show the "most important" areas and partial-areas). A user should also be able to selectively "hide" whole areas that she is not interested in. We will investigate the possibility of providing a user with various options for how much information is displayed while navigating, enabling a controllable level of display granularity.

Finally, in the area of social information management, an annotation mechanism for partial-area taxonomies would allow several collaborating ontology editors to share information and insights into the ontology with each other. Such a mechanism would require an OAF implementation that is user-specific by providing every user with a log in. This kind of functionality can be used to support the consensus-based quality assurance reviews described in Section 3.5. Editors and domain experts could "tag" suspicious areas, partial-areas, and concepts in the OAF.

6. Conclusions

In this paper, we introduced the Ontology Abstraction Framework (OAF), a framework and software system for deriving, visualizing, and exploring partial-area taxonomy abstraction networks. In the OAF, ontologies, partial-area taxonomy derivation methodologies, and processes that can be applied to partial-area taxonomies are represented generically. This generic representation of ontologies and derivation methodologies enables the standardized creation of partial-area taxonomies for ontologies represented in widely varying formats. We demonstrated the standalone version and the Protégé plugin version of the OAF using the National Cancer Institute thesaurus. Additionally, we introduced live partial-area taxonomies that are updated instantaneously as a user is editing the underlying ontology.

Acknowledgments

Research reported in this publication was supported by the National Cancer Institute of the National Institutes of Health under Award Number R01CA190779. The content is solely the responsibility of the authors and does not necessarily represent the views of the National Institutes of Health.

The authors thank David Kaufmann and Stephanie Furniss for their work on the preliminary OAF user evaluations. The authors also thank David Daudelin, Christopher Castellano, Vladimir Ventura, and the many other student developers who have contributed their time and effort to the OAF project.

References

1. Noy NF, Crubézy M, Fergerson RW, et al. Protege-2000: an open-source ontology-development and knowledge-acquisition environment. AMIA Annual Symposium Proceedings. 2003:953. [PubMed: 14728458]
2. Tudorache T, Nyulas C, Noy NF, et al. WebProtégé: a collaborative ontology editor and knowledge acquisition tool for the web. Sem. Web. 2013; 4(1):89–99.

3. Day-Richter J, Harris MA, Haendel M, et al. OBO-Edit—an ontology editor for biologists. Bioinformatics. 2007; 23(16):2198–2200. [PubMed: 17545183]

4. Halper M, Gu H, Perl Y, et al. Abstraction networks for terminologies: supporting management of "Big Knowledge". Artif. Intell. Med. 2015; 64(1):1–16. [PubMed: 25890687]

5. Ochs C, Perl Y, Geller J, et al. Summarizing and visualizing structural changes during the evolution of biomedical ontologies using a diff abstraction network. J. Biomed. Inf. 2015; 56:127–144.

6. Wei D, Gu H, Perl Y, et al. Structural measures to track the evolution of SNOMED CT hierarchies. J. Biomed. Inf. 2015; 57:278–287.

7. Ochs, C.; Perl, Y.; Geller, J., et al. International Workshop on Biomedical and Health Informatics. Springer; 2015. Using aggregate taxonomies to summarize SNOMED CT evolution; p. 1008-1015.

8. Ochs C, Geller J, Perl Y, et al. Scalable quality assurance for large SNOMED CT hierarchies using subject-based subtaxonomies. J. Am. Med. Inf. Assoc. 2014; 22(3):507–518.

9. Ochs, C.; Agrawal, A.; Perl, Y., et al. AMIA Annual Symposium Proceedings. Springer; 2012. Deriving an abstraction network to support quality assurance in OCRe; p. 681-689.

10. Ochs C, He Z, Perl Y, et al. Choosing the granularity of abstraction networks for orientation and quality assurance of the sleep domain ontology. Proceedings of the 4th International Conference on Biomedical Ontology. 2013:4–89.

11. Wang Y, Halper M, Min H, et al. Structural methodologies for auditing SNOMED. J. Biomed. Inf. 2007; 40(5):561–581.

12. Stearns MQ, Price C, Spackman KA, et al. SNOMED clinical terms: overview of the development process and project status. Proceedings of AMIA Annual Symposium. 2001:662–666.

13. Fragoso G, de Coronado S, Haber M, et al. Overview and utilization of the NCI thesaurus. Comp. Funct. Genom. 2004; 5(8):648–654.

14. Sim I, Carini S, Tu S, et al. The human studies database project: federating human studies design data using the ontology of clinical research. AMIA Summits on Translational Science Proceedings. 2010:1–55.

15. Hammarberg, M.; Sundén, J. Kanban In Action. Manning Publications; 2014. p. 22

16. Geller J, Ochs C, Perl Y, et al. New abstraction networks and a new visualization tool in support of auditing the SNOMED CT content. AMIA Annual Symposium on Proceedings. 2012:37–246.

17. Ochs C, Geller J, Perl Y, et al. A tribal abstraction network for SNOMED CT hierarchies without attribute relationships. J. Am. Med. Inf. Assoc. 2014; 22(3):628–639.

18. He Z, Ochs C, Agrawal A, et al. A family-based framework for supporting quality assurance of biomedical ontologies in BioPortal. Proceedings of AMIA Annual Symposium. 2013:81–590.

19. Whetzel PL, Noy NF, Sham NH, et al. BioPortal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucl. Acids Res. (NAR). 2011; 39(Web Server issue):W541–W545. [PubMed: 21672956]

20. Ochs C, He Z, Zheng L, et al. Utilizing a structural meta-ontology for family-based quality assurance of the BioPortal ontologies. J. Biomed. Inf. 2016; 61:63–76.

21. Motik B, Patel-Schneider PF, Parsia B. OWL 2 web ontology language structural specification and functional style syntax. W3C recommendation. 2009; 27(65):159.

22. Smith B, Ashburner M, Rosse C, et al. The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotechnol. 2007; 25(11):1251–1255. [PubMed: 17989687]

23. Min H, Perl Y, Chen Y, et al. Auditing as part of the terminology design life cycle. J. Am. Med. Inf. Assoc. 2006; 13(6):676–6906.

24. IHTSDO. SNOMED Clinical Terms® Technical Reference Guide. Jan. 2009 International Release <http://www.ihtsdo.org/fileadmin/user_upload/Docs_01/SNOMED_CT_Publications/SNOMED_CT_Technical_Reference_Guide_20090131.pdf>

25. Apelon Distributed Terminology System (DTS). [9 January 2015] <http://www.apelondts.org/>

26. Wang Y, Halper M, Wei D, et al. Abstraction of complex concepts with a refined partial-area taxonomy of SNOMED. J. Biomed. Inf. 2012; 45(1):15–29.

27. Ochs C, Zheng L, Perl Y, et al. Drug-drug interaction discovery using abstraction networks for "National Drug File – Reference Terminology" chemical ingredients. AMIA Annual Symposium on Proceedings. 2015:973–982.
28. Brown SH, Elkin PL, Rosenbloom ST, et al. VA national drug file reference terminology: a cross-institutional content coverage study. Medinfo. 2004; 11:477–481.
29. Ochs, C.; Perl, Y.; Geller, J., et al. Quality assurance of the gene ontology using abstraction networks. J. Bioinf. Comput. Biol. 2015. http://dx.doi.org/10.1142/S0219720016420014
30. Ashburner M, Ball CA, Blake JA, et al. Gene ontology: tool for the unification of biology. Nat. Genet. 2000; 25(1):25–29. [PubMed: 10802651]
31. Musen MA. The protégé project: a look back and a look forward. AI Matters. 2015; 1(4):4–12. [PubMed: 27239556]
32. Horridge M, Bechhofer S. The OWL API: a java API for working with OWL 2 ontologies. OWLED. 2009; 529:11–21.
33. Protege Plugin Library – Protege Wiki. 2014. [20 December 2014] <http://protegewiki.stanford.edu/wiki/Protege_Plugin_Library>
34. Kremen P, Smid M, Kouba Z. OWLDiff: a practical tool for comparison and merge of OWL ontologies. 22nd International Workshop on Database and Expert Systems Applications. 2011:29–233.
35. Shearer R, Motik B, Horrocks I. HermiT: a highly-efficient OWL reasoner. Proceedings of 5th International Workshop on OWL: Experiences and Directions (OWLED). 2008
36. Horridge, M. OWLViz. 2010. [29 March 2016] <http://protegewiki.stanford.edu/wiki/OWLViz>
37. Horridge, M. A Practical Guide to Building OWL Ontologies Using Protege 4 and CO-ODE Tools. University of Manchester; 2011. [17 December 2015] <http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf>
38. IHTSDO Workbench. <http://www.ihtsdo.org/news/article/view/ihtsdo-launches-global-health-terminology-workbench/>
39. DTS Editor. [29 March 2016]. Available from: http://apelon-dts.sourceforge.net/dtseditor.html
40. Motta E, Mulholland P, Peroni S, et al. A novel approach to visualizing and navigating ontologies. The Semantic Web – ISWC. 2011:470–486.
41. Haase P, Lewen H, Studer R, et al. The neon ontology engineering toolkit. WWW Developers Track. 2008
42. Queiroz-Sousa PO, Salgado AC, Pires CE. A method for building personalized ontology summaries. J. Inf. Data Manage. 2013; 4(5)
43. Ceusters W. SNOMED CT's RF2: Is the Future Bright? Stud Health Technol. Inf. 2011; 169:829–833.
44. Wang Y, Halper M, Wei D, et al. Auditing complex concepts of SNOMED using a refined hierarchical abstraction network. J. Biomed. Inf. 2012; 45(1):1–14.
45. Ochs C, Perl Y, Geller J, et al. Scalability of abstraction-network-based quality assurance to large SNOMED hierarchies. AMIA Annual Symposium Proceedings. 2013:1071–1080. [PubMed: 24551393]
46. Krasner GE, Pope ST. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. J. Object Oriented Program. 1988; 1(3):26–49.
47. He Z, Ochs C, Soldatova L, et al. Auditing redundant import in reuse of a top level ontology for the drug discovery investigations ontology. VDOS. 2013
48. Halper M, Wang Y, Min H, et al. Analysis of error concentrations in SNOMED. AMIA Annual Symposium Proceedings. 2007:314–318. [PubMed: 18693849]
49. Lanzenberger M, Sampson J, Rester M. Visualization in ontology tools. International Conference on Complex Intelligent and Software Intensive Systems. 2009:705–711.
50. Fu B, Noy NF, Storey M-A. Indented tree or graph? A usability study of ontology visualization techniques in the context of class mapping evaluation. The Semantic Web–ISWC. 2013; 2013:117–134.
51. Gu H, Elhanan G, Perl Y, et al. A study of terminology auditors' performance for UMLS semantic type assignments. J. Biomed. Inf. 2012; 45(6):1042–1048.

52. Mortensen JM, Minty EP, Januszyk M, et al. Using the wisdom of the crowds to find critical errors in biomedical ontologies: a study of SNOMED CT. J. Am. Med. Inf. Assoc. 2015; 22(3):640–648.

53. Ontology Abstraction Framework Beta Release. 2016. [2 June 2016] <https://web.njit.edu/~cro3/oaf/>

54. Kamdar MR, Tudorache T, Musen MA. A systematic analysis of term reuse and term overlap across biomedical ontologies. Semantic Web J. 2016 (in press).

55. Ochs, C.; Geller, J.; Perl, Y. Summarizing the structure of the pizza ontology: ontology development with partial-area taxonomies and the ontology abstraction framework. 2016. <https://sites.google.com/a/njit.edu/saboc/documents/Pizza%20Ontology%20Taxonomy%20Tutorial%20v1.pdf?attredirects=0&d=1>

56. Storey M-A, Musen M, Silva J, et al. Jambalaya: interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. Workshop on Interactive Tools for Knowledge Capture. 2001

57. Horsky J, Kaufman DR, Oppenheim MI, et al. A framework for analyzing the cognitive complexity of computer-assisted clinical ordering. J. Biomed. Inf. 2003; 36:4–22.

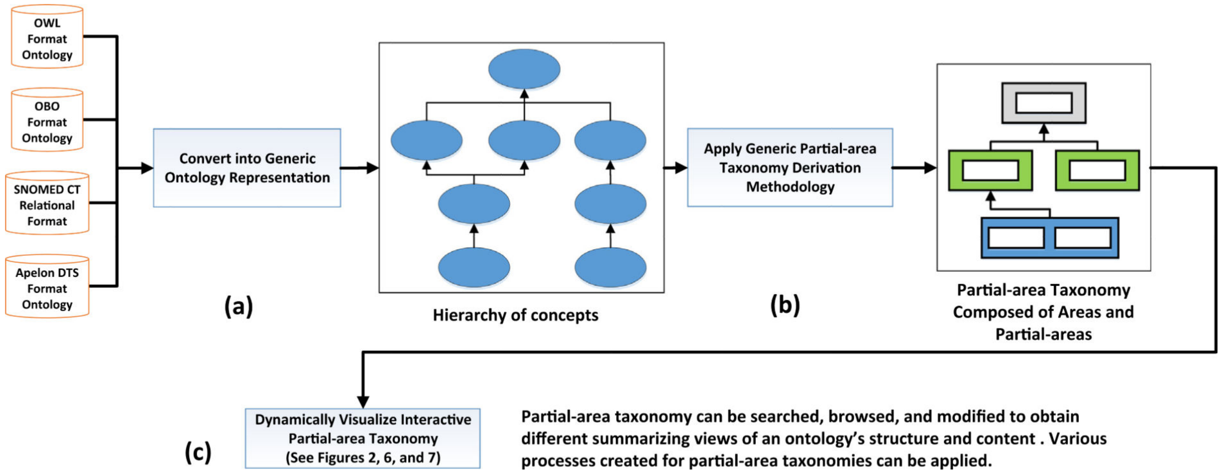
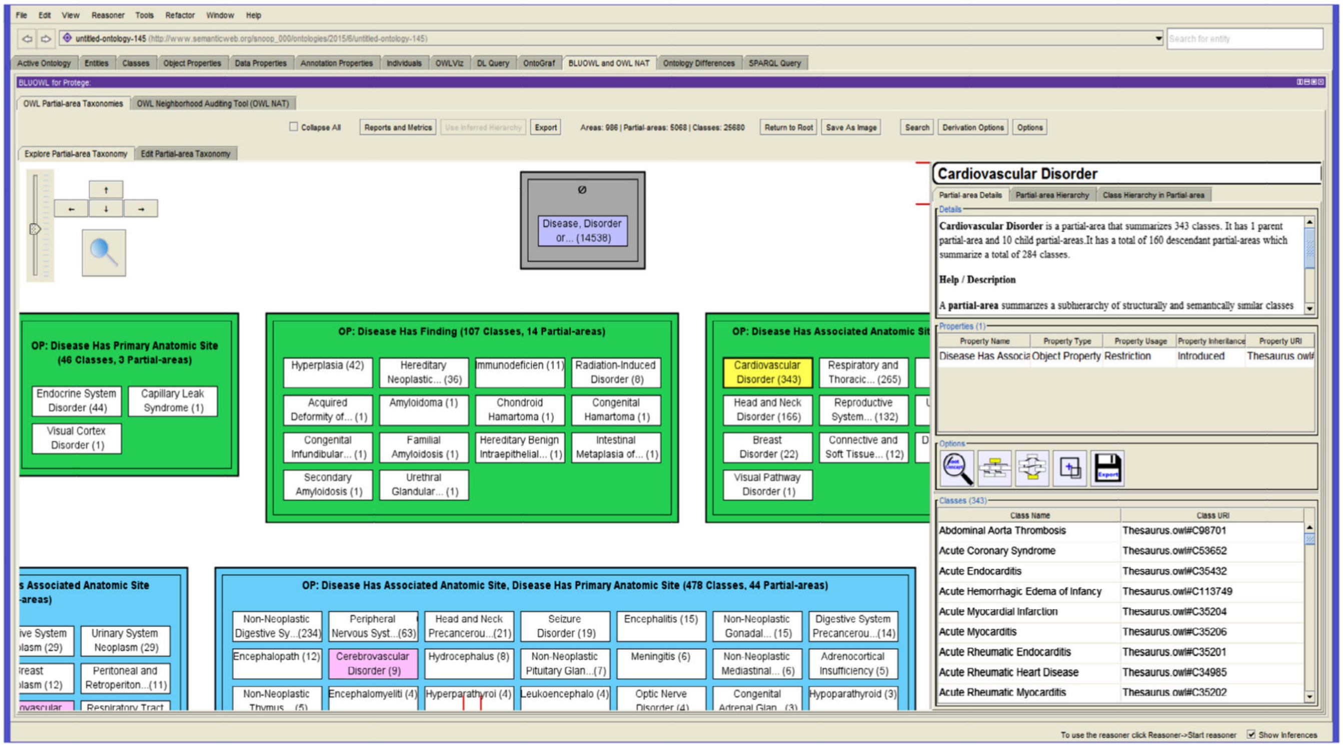

OWL Format Ontology
OBO Format Ontology
SNOMED CT Relational Format
Apelon DTS Format Ontology
Convert into Generic Ontology Representation
(a)
Hierarchy of concepts
Apply Generic Partial-area Taxonomy Derivation Methodology
(b)
Partial-area Taxonomy Composed of Areas and Partial-areas
(c)
Dynamically Visualize Interactive Partial-area Taxonomy (See Figures 2, 6, and 7)
Partial-area taxonomy can be searched, browsed, and modified to obtain different summarizing views of an ontology's structure and content . Various processes created for partial-area taxonomies can be applied.

Fig. 1.

An overview of the workflow of the Ontology Abstraction Framework. (a) All ontologies are converted into a standardized representation, abstracting away differences and idiosyncrasies. (b) The partial-area taxonomy derivation methodology is defined in terms of the standardized ontology representation, making it applicable to any ontology in a format that is supported by the OAF. (c) The partial-area taxonomy is visualized within the OAF, enabling a user to explore how the underlying ontology is summarized. Viewing the summary may provide users with new insights into the ontology, supporting quality assurance.


File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-145 (http://www.semanticweb.org/snoop_000/ontologies/2015/6/untitled-ontology-145)
Search for entity
Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query OntoGraf BLUOWL and OWL NAT Ontology Differences SPARQL Query
BLUOWL for Protege:
OWL Partial-area Taxonomies
OWL Neighborhood Auditing Tool (OWL NAT)
☐ Collapse All
Reports and Metrics
Use Inferred Hierarchy
Export
Areas: 986 | Partial-areas: 5068 | Classes: 25680
Return to Root
Save As Image
Search
Derivation Options
Options
Explore Partial-area Taxonomy
Edit Partial-area Taxonomy
∅
Disease, Disorder or... (14538)
OP: Disease Has Primary Anatomic Site (46 Classes, 3 Partial-areas)
Endocrine System Disorder (44)
Capillary Leak Syndrome (1)
Visual Cortex Disorder (1)
OP: Disease Has Finding (107 Classes, 14 Partial-areas)
Hyperplasia (42)
Hereditary Neoplastic... (36)
Immunodeficien (11)
Radiation-Induced Disorder (8)
Acquired Deformity of... (1)
Amyloidoma (1)
Chondroid Hamartoma (1)
Congenital Hamartoma (1)
Congenital Infundibular... (1)
Familial Amyloidosis (1)
Hereditary Benign Intraepithelial... (1)
Intestinal Metaplasia of... (1)
Secondary Amyloidosis (1)
Urethral Glandular... (1)
OP: Disease Has Associated Anatomic Sit
Cardiovascular Disorder (343)
Respiratory and Thoracic... (265)
Head and Neck Disorder (166)
Reproductive System... (132)
Breast Disorder (22)
Connective and Soft Tissue... (12)
Visual Pathway Disorder (1)
s Associated Anatomic Site
-areas)
ive System plasm (29)
Urinary System Neoplasm (29)
reast plasm (12)
Peritoneal and Retroperiton...(11)
ovascular
Respiratory Tract
OP: Disease Has Associated Anatomic Site, Disease Has Primary Anatomic Site (478 Classes, 44 Partial-areas)
Non-Neoplastic Digestive Sy...(234)
Peripheral Nervous Syst...(63)
Head and Neck Precancerou...(21)
Seizure Disorder (19)
Encephalitis (15)
Non-Neoplastic Gonadal... (15)
Digestive System Precancerou...(14)
Encephalopath (12)
Cerebrovascular Disorder (9)
Hydrocephalus (8)
Non-Neoplastic Pituitary Glan...(7)
Meningitis (6)
Non-Neoplastic Mediastinal... (6)
Adrenocortical Insufficiency (5)
Non-Neoplastic Thymus... (5)
Encephalomyeliti (4)
Hyperparathyroi (4)
Leukoencephalo (4)
Optic Nerve Disorder (4)
Congenital Adrenal Glan... (3)
Hypoparathyroid (3)
Cardiovascular Disorder
Partial-area Details
Partial-area Hierarchy
Class Hierarchy in Partial-area
Details
Properties (1)
Options
Classes (343)
To use the reasoner click Reasoner->Start reasoner
☑ Show Inferences

Cardiovascular Disorder is a partial-area that summarizes 343 classes. It has 1 parent partial-area and 10 child partial-areas.It has a total of 160 descendant partial-areas which summarize a total of 284 classes.

Help / Description

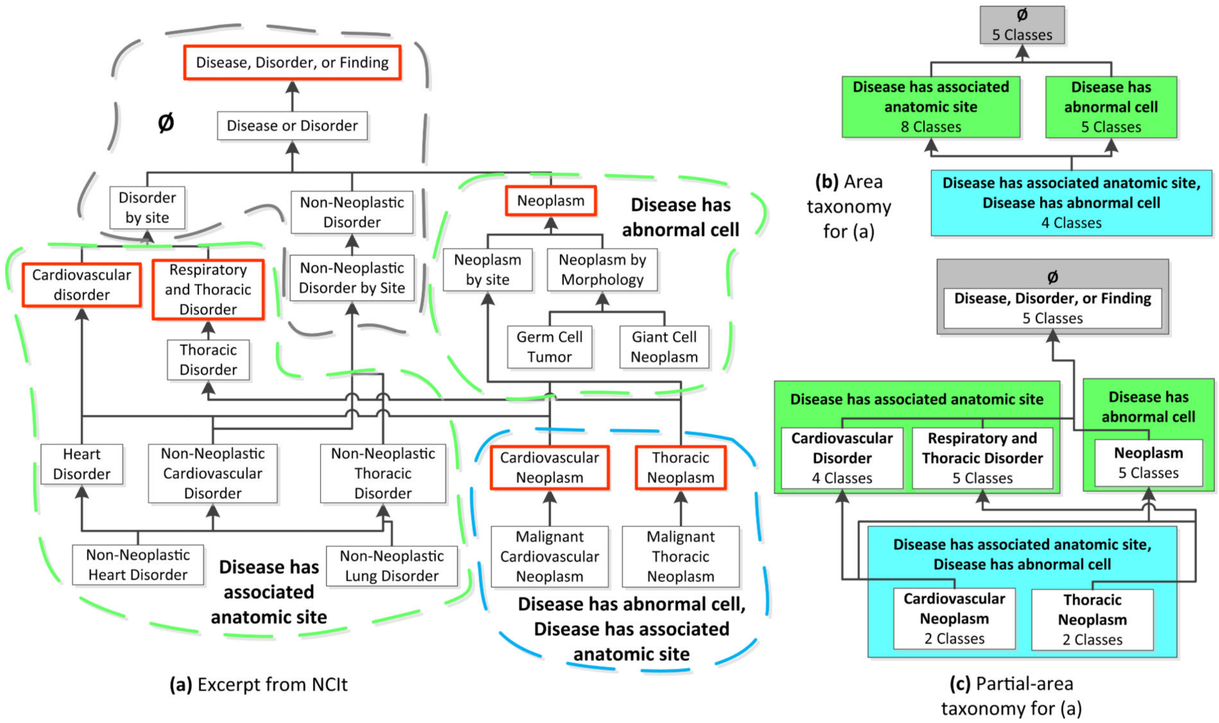
A **partial-area** summarizes a subhierarchy of structurally and semantically similar classes

Property Name	Property Type	Property Usage	Property Inheritance	Property URI
Disease Has Associa	Object Property	Restriction	Introduced	Thesaurus.owl#

Class Name	Class URI
Abdominal Aorta Thrombosis	Thesaurus.owl#C98701
Acute Coronary Syndrome	Thesaurus.owl#C53652
Acute Endocarditis	Thesaurus.owl#C35432
Acute Hemorrhagic Edema of Infancy	Thesaurus.owl#C113749
Acute Myocardial Infarction	Thesaurus.owl#C35204
Acute Myocarditis	Thesaurus.owl#C35206
Acute Rheumatic Endocarditis	Thesaurus.owl#C35201
Acute Rheumatic Heart Disease	Thesaurus.owl#C34985
Acute Rheumatic Myocarditis	Thesaurus.owl#C35202

Fig. 2.

A screenshot of the current version of the Ontology Abstraction Framework (OAF) Protégé Plugin. A partial-area taxonomy of an excerpt of the National Cancer Institute thesaurus (NCIt) *Disease, Disorder, or Finding* hierarchy is displayed in the plugin. The partial-area taxonomy summarizes sets of NCIt classes that are modeled using the same types of object properties. The interactive display provides information about the concepts summarized by each node in the partial-area taxonomy. In this example, a user has selected the *Cardiovascular Disorder* partial-area (highlighted in yellow, right side) and information about this partial-area is shown on the right. The ten child partial-areas of *Cardiovascular disorder* (mentioned in upper right window) are highlighted in pink. For more details see Fig. 3.


Disease, Disorder, or Finding
Ø
Disease or Disorder
Disorder by site
Non-Neoplastic Disorder
Neoplasm
Disease has abnormal cell
Cardiovascular disorder
Respiratory and Thoracic Disorder
Non-Neoplastic Disorder by Site
Neoplasm by site
Neoplasm by Morphology
Thoracic Disorder
Germ Cell Tumor
Giant Cell Neoplasm
Heart Disorder
Non-Neoplastic Cardiovascular Disorder
Non-Neoplastic Thoracic Disorder
Cardiovascular Neoplasm
Thoracic Neoplasm
Non-Neoplastic Heart Disorder
Disease has associated anatomic site
Non-Neoplastic Lung Disorder
Malignant Cardiovascular Neoplasm
Malignant Thoracic Neoplasm
Disease has abnormal cell, Disease has associated anatomic site
Ø
5 Classes
Disease has associated anatomic site
8 Classes
Disease has abnormal cell
5 Classes
Disease has associated anatomic site, Disease has abnormal cell
4 Classes
Ø
Disease, Disorder, or Finding
5 Classes
Disease has associated anatomic site
Cardiovascular Disorder
4 Classes
Respiratory and Thoracic Disorder
5 Classes
Disease has abnormal cell
Neoplasm
5 Classes
Disease has associated anatomic site, Disease has abnormal cell
Cardiovascular Neoplasm
2 Classes
Thoracic Neoplasm
2 Classes

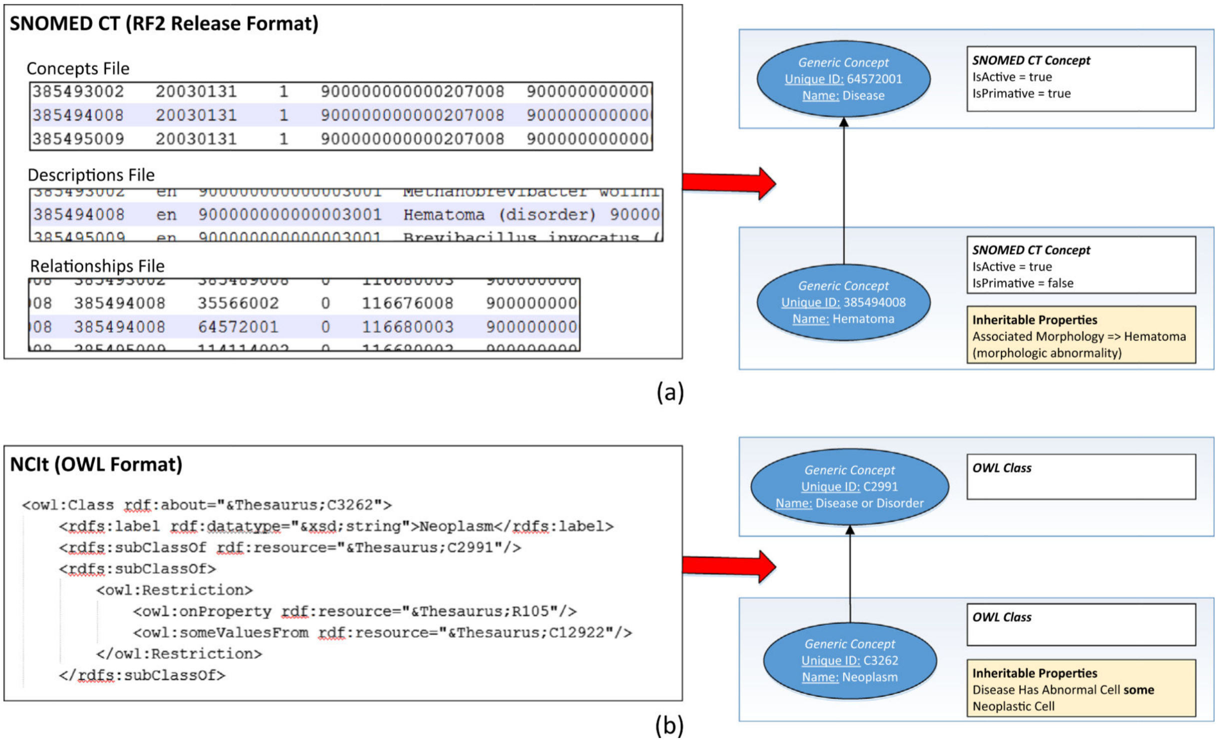
(a) Excerpt from NCIt

(b) Area taxonomy for (a)

(c) Partial-area taxonomy for (a)

Fig. 3.

(a) An excerpt of 22 classes from the NCIt *Disease, Disorder or Finding* hierarchy. Classes are shown as boxes labeled with concept names. Upward directed arrows are the subsumption relationships between concepts. Bold text in colored, dashed bubbles indicates which properties are used in the restrictions for a given set of concepts. Each root class is identified by a red outline. (b) The area taxonomy for the classes in (a), derived according to the properties in restrictions. The 22 classes are summarized by four area nodes. Areas are organized into three color-coded levels according to their numbers of property types. The top level (gray) area node represents classes with no properties. Sets of properties are used as area names. Arrows between areas are *child-of* links (*e.g.*, the area {Disease has abnormal cell, Disease has associated anatomic site} is a *child-of* the area {*Disease has associated anatomic site*} and also of the area {*Disease has abnormal cell*}). (c) The partial-area taxonomy for the classes in (a). Each white box represents a partial-area. Partial-area nodes are labeled with their root class' name and the total numbers of classes summarized by them. Upward directed arrows represent *child-of* links between partial-area nodes (*e.g.*, the *Neoplasm* partial-area node is *child-of* the *Disease, Disorder, or Finding* partial-area node). The partial-area taxonomy appears as an overlay of the area taxonomy in (b). Partial-areas may overlap (*i.e.*, two partial-areas may summarize the same class). The classes *Heart Disorder* and *Non-neoplastic Heart Disorder* are summarized by both the *Cardiovascular Disorder* and *Respiratory and Thoracic Disorder* partial-area nodes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



SNOMED CT (RF2 Release Format)

Concepts File

```
385493002   20030131    1    900000000000207008   900000000000
385494008   20030131    1    900000000000207008   900000000000
385495009   20030131    1    900000000000207008   900000000000
```

Descriptions File

```
385493002   en   900000000000003001   Methanobrevibacter wolini
385494008   en   900000000000003001   Hematoma (disorder) 90000
385495009   en   900000000000003001   Brevibacillus invocatus (
```

Relationships File

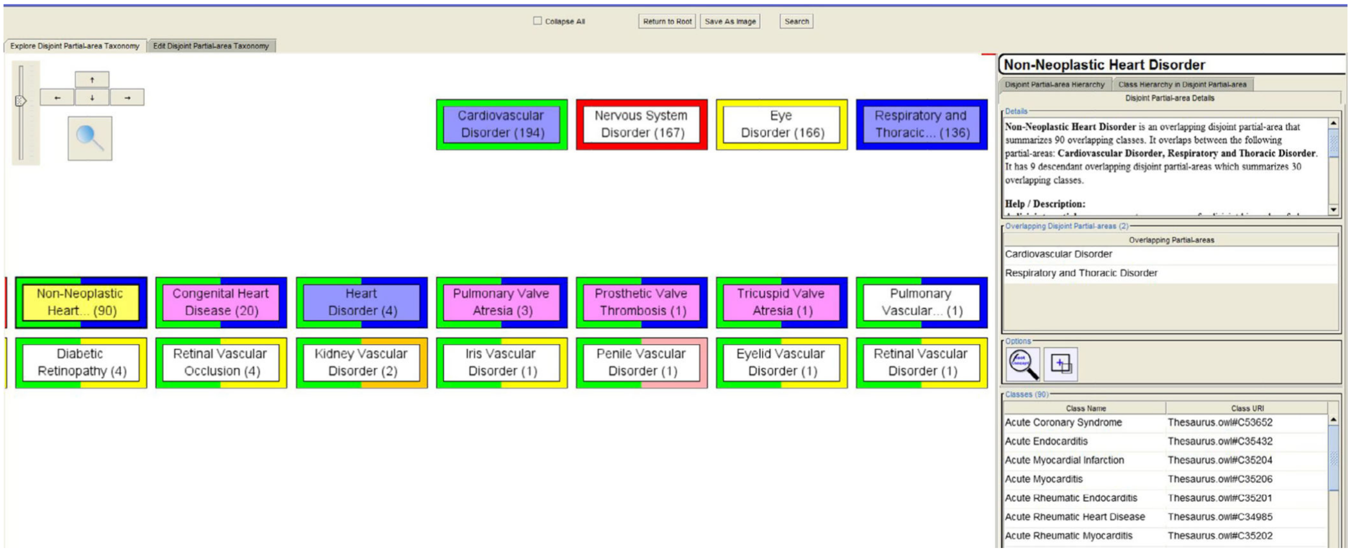
```
008   385493002   385469008   0   116680003   900000000
008   385494008   35566002    0   116676008   900000000
008   385494008   64572001    0   116680003   900000000
008   385495009   114114002   0   116680003   900000000
```

NCIt (OWL Format)

```
<owl:Class rdf:about="&Thesaurus;C3262">
    <rdfs:label rdf:datatype="&xsd;string">Neoplasm</rdfs:label>
    <rdfs:subClassOf rdf:resource="&Thesaurus;C2991"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&Thesaurus;R105"/>
            <owl:someValuesFrom rdf:resource="&Thesaurus;C12922"/>
        </owl:Restriction>
    </rdfs:subClassOf>
```

Fig. 4.

(a) The process of representing the SNOMED CT concept *Hematoma*, stored in RF2 relational release format in the generic OAF ontology representation. The "generic concept" data type of OAF is extended into a "SNOMED CT concept" data type that stores *Hematoma* with the status "active" and "non-primitive." This concept has one attribute relationship, *Associated Morphology*, with a target of *Hematoma (morphologic abnormality)*. It is stored as an inheritable property. (b) The process of representing the NCIt concept *Neoplasm*, as given in OWL format, in the generic ontology representation. The "generic concept" data type is extended into an "OWL Class" data type, storing annotations such as the class label (not shown). The restriction on *Neoplasm* (*Disease Has Abnormal Cell* **some** *Neoplastic Cell*) is stored as an inheritable property.


□ Collapse All
Return to Root
Save As Image
Search
Explore Disjoint Partial-area Taxonomy
Edit Disjoint Partial-area Taxonomy
↑
←
↓
→
Cardiovascular Disorder (194)
Nervous System Disorder (167)
Eye Disorder (166)
Respiratory and Thoracic... (136)
Non-Neoplastic Heart... (90)
Congenital Heart Disease (20)
Heart Disorder (4)
Pulmonary Valve Atresia (3)
Prosthetic Valve Thrombosis (1)
Tricuspid Valve Atresia (1)
Pulmonary Vascular... (1)
Diabetic Retinopathy (4)
Retinal Vascular Occlusion (4)
Kidney Vascular Disorder (2)
Iris Vascular Disorder (1)
Penile Vascular Disorder (1)
Eyelid Vascular Disorder (1)
Retinal Vascular Disorder (1)
Non-Neoplastic Heart Disorder
Disjoint Partial-area Hierarchy
Class Hierarchy in Disjoint Partial-area
Disjoint Partial-area Details
Details
Overlapping Disjoint Partial-areas (2)
Options
Classes (90)

Non-Neoplastic Heart Disorder is an overlapping disjoint partial-area that summarizes 90 overlapping classes. It overlaps between the following partial-areas: **Cardiovascular Disorder, Respiratory and Thoracic Disorder**. It has 9 descendant overlapping disjoint partial-areas which summarizes 30 overlapping classes.

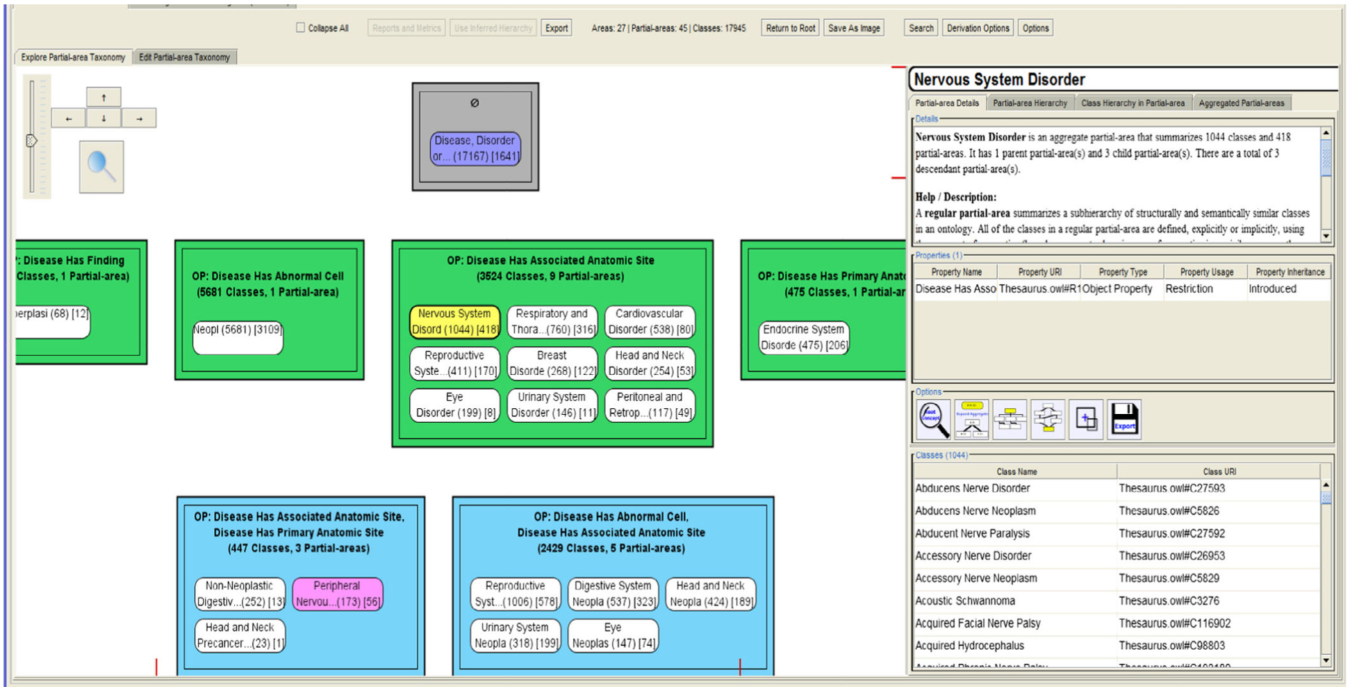
Help / Description:

Overlapping Partial-areas
Cardiovascular Disorder
Respiratory and Thoracic Disorder

Class Name	Class URI
Acute Coronary Syndrome	Thesaurus.owl#C53652
Acute Endocarditis	Thesaurus.owl#C35432
Acute Myocardial Infarction	Thesaurus.owl#C35204
Acute Myocarditis	Thesaurus.owl#C35206
Acute Rheumatic Endocarditis	Thesaurus.owl#C35201
Acute Rheumatic Heart Disease	Thesaurus.owl#C34985
Acute Rheumatic Myocarditis	Thesaurus.owl#C35202

Fig. 5.

The disjoint partial-area taxonomy derived from the {*Disease has associated anatomic site*} area selected in Fig. 7. The disjoint partial-areas with a single frame color (*e.g.*, *Cardiovascular Disorder* (194)) summarize non-overlapping classes. The disjoint partial-areas that summarize overlapping classes are color coded according to the partial-areas the classes overlap between. For example, a green and blue frame (*e.g.*, *Heart Disorder*) indicates that the classes summarized by these disjoint partial-areas overlap between the *Cardiovascular Disorder* and the *Respiratory and Thoracic Disorder* partial-areas. The disjoint partial-area *Non-Neoplastic Heart Disorder* (90) has been selected by the user with a mouse click. This turns its background temporarily into yellow. The child disjoint partial-areas of *Non-Neoplastic Heart Disorder* (*e.g.*, *Congenital Heart Disease*) turn pink and the parent disjoint partial-areas (*e.g.*, *Heart Disorder*) turn light blue. The *Disjoint Partial-area Details* display, on the right, lists the concepts that are summarized by the disjoint partial-area along with the partial-areas with which it overlaps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)


☐ Collapse All
Reports and Metrics
Use Inferred Hierarchy
Export
Areas: 27 | Partial-areas: 45 | Classes: 17945
Return to Root
Save As Image
Search
Derivation Options
Options
Explore Partial-area Taxonomy
Edit Partial-area Taxonomy
∅
Disease, Disorder or... (17167) [1641]
: Disease Has Finding Classes, 1 Partial-area)
erplasi (68) [12]
OP: Disease Has Abnormal Cell (5681 Classes, 1 Partial-area)
Neopl (5681) [3109]
OP: Disease Has Associated Anatomic Site (3524 Classes, 9 Partial-areas)
Nervous System Disord (1044) [418]
Respiratory and Thora...(760) [316]
Cardiovascular Disorder (538) [80]
Reproductive Syste...(411) [170]
Breast Disorde (268) [122]
Head and Neck Disorder (254) [53]
Eye Disorder (199) [8]
Urinary System Disorder (146) [11]
Peritoneal and Retrop...(117) [49]
OP: Disease Has Primary Anat (475 Classes, 1 Partial-ar
Endocrine System Disorde (475) [206]
OP: Disease Has Associated Anatomic Site, Disease Has Primary Anatomic Site (447 Classes, 3 Partial-areas)
Non-Neoplastic Digestiv...(252) [13]
Peripheral Nervou...(173) [56]
Head and Neck Precancer...(23) [1]
OP: Disease Has Abnormal Cell, Disease Has Associated Anatomic Site (2429 Classes, 5 Partial-areas)
Reproductive Syst...(1006) [578]
Digestive System Neopla (537) [323]
Head and Neck Neopla (424) [189]
Urinary System Neopla (318) [199]
Eye Neoplas (147) [74]
Nervous System Disorder
Partial-area Details
Partial-area Hierarchy
Class Hierarchy in Partial-area
Aggregated Partial-areas
Details
Properties (1)
Options
Root Concept
Expand Aggregate
Export
Classes (1044)

Nervous System Disorder is an aggregate partial-area that summarizes 1044 classes and 418 partial-areas. It has 1 parent partial-area(s) and 3 child partial-area(s). There are a total of 3 descendant partial-area(s).

Help / Description:

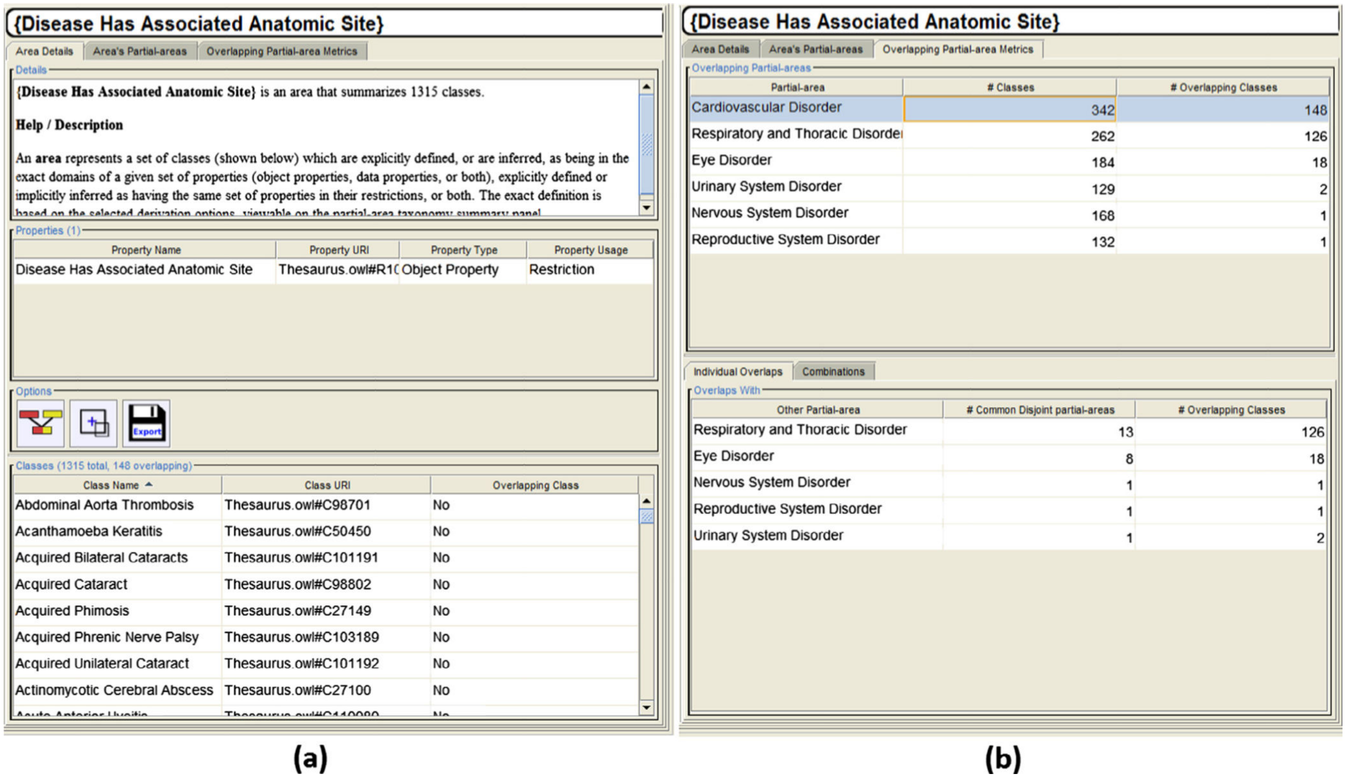
A **regular partial-area** summarizes a subhierarchy of structurally and semantically similar classes in an ontology. All of the classes in a regular partial-area are defined, explicitly or implicitly, using

Property Name	Property URI	Property Type	Property Usage	Property Inheritance
Disease Has Asso	Thesaurus.owl#R1	Object Property	Restriction	Introduced

Class Name	Class URI
Abducens Nerve Disorder	Thesaurus.owl#C27593
Abducens Nerve Neoplasm	Thesaurus.owl#C5826
Abducent Nerve Paralysis	Thesaurus.owl#C27592
Accessory Nerve Disorder	Thesaurus.owl#C26953
Accessory Nerve Neoplasm	Thesaurus.owl#C5829
Acoustic Schwannoma	Thesaurus.owl#C3276
Acquired Facial Nerve Palsy	Thesaurus.owl#C116902
Acquired Hydrocephalus	Thesaurus.owl#C98803

Fig. 6.

An aggregate partial-area taxonomy, derived using a bound of 20, for the NCIt *Disease, Disorder, or Finding* hierarchy. All partial-areas that summarize fewer than 20 classes in the complete partial-area taxonomy are not shown and have been aggregated into their closest larger ancestor partial-areas. For example, in the aggregate partial-area taxonomy the *Nervous System Disorder* aggregate partial-area (selected by the user and therefore highlighted in yellow) summarizes 536 classes and 418 "small" partial-areas that would summarize fewer than 20 classes each.



{Disease Has Associated Anatomic Site}

Details

{Disease Has Associated Anatomic Site} is an area that summarizes 1315 classes.

Help / Description

An **area** represents a set of classes (shown below) which are explicitly defined, or are inferred, as being in the exact domains of a given set of properties (object properties, data properties, or both), explicitly defined or implicitly inferred as having the same set of properties in their restrictions, or both. The exact definition is based on the selected derivation options, viewable on the partial-area taxonomy summary panel.

Properties (1)

Property Name	Property URI	Property Type	Property Usage
Disease Has Associated Anatomic Site	Thesaurus.owl#R1(	Object Property	Restriction

Options

Classes (1315 total, 148 overlapping)

Class Name	Class URI	Overlapping Class
Abdominal Aorta Thrombosis	Thesaurus.owl#C98701	No
Acanthamoeba Keratitis	Thesaurus.owl#C50450	No
Acquired Bilateral Cataracts	Thesaurus.owl#C101191	No
Acquired Cataract	Thesaurus.owl#C98802	No
Acquired Phimosis	Thesaurus.owl#C27149	No
Acquired Phrenic Nerve Palsy	Thesaurus.owl#C103189	No
Acquired Unilateral Cataract	Thesaurus.owl#C101192	No
Actinomycotic Cerebral Abscess	Thesaurus.owl#C27100	No
Acute Anterior Uveitis	Thesaurus.owl#C110080	No

{Disease Has Associated Anatomic Site}

Overlapping Partial-areas

Partial-area	# Classes	# Overlapping Classes
Cardiovascular Disorder	342	148
Respiratory and Thoracic Disorde	262	126
Eye Disorder	184	18
Urinary System Disorder	129	2
Nervous System Disorder	168	1
Reproductive System Disorder	132	1

Overlaps With

Other Partial-area	# Common Disjoint partial-areas	# Overlapping Classes
Respiratory and Thoracic Disorder	13	126
Eye Disorder	8	18
Nervous System Disorder	1	1
Reproductive System Disorder	1	1
Urinary System Disorder	1	2

Fig. 7.

The area details display for {*Disease has associated anatomic site*}. (a) A summary of the {*Disease has associated anatomic site*} area. The classes summarized by {*Disease has associated anatomic site*} are listed in alphabetical order. Additionally, overlapping classes, along with the partial-area(s) summarizing them, are identified in the area. (b) The overlapping class metrics display. When an area has overlapping classes this window indicates which partial-areas have overlapping classes with it, and how many classes are overlapping.

Table 1

A glossary of terms and definitions used in this paper.

Term	Definition
abstraction network	A summary of an ontology. A hierarchy of nodes where each node summarizes sets of "similar" concepts/ classes. The definition of similarity is based on the type of abstraction network being derived
child-of	A hierarchical connection between two abstraction network nodes. Typically based on the ontology's IS-A/ subclass relationships
area	The set of concepts/classes that are defined using the same set of semantic relationships (e.g., attribute relationships in SNOMED CT [11] and object properties in OWL [9,10])
area node	A node in an abstraction network that summarizes the classes of an area. For brevity this is sometimes abbreviated to "area"
area taxonomy	An abstraction network where the area nodes connected by child-of links (see Fig. 3(b) and [9–11])
root concept/root class	A concept/class in an area that has no parent concept/class in its area
partial-area	A set of semantically similar concepts within an area. Each partial-area consists of a root concept/root class and all of its descendant concepts/classes in the area
partial-area node	A node of an abstraction network that summarizes the classes of a partial-area. For brevity this is sometimes abbreviated to "partial-area"
partial-area taxonomy	An abstraction network where the partial-area nodes are embedded within their respective area nodes. Partial-area nodes are connected by child-of links. See Fig. 3(c) and [9–11] for a complete description
disjoint partial-area taxonomy	A partial-area taxonomy where each concept is summarized by exactly one partial-area (a disjoint partial-area). See [26]
aggregate partial-area taxonomy	A partial-area taxonomy where partial-areas that summarize a number of concepts/classes below a chosen bound b are combined into their larger ancestor partial-area(s). See Ochs et al. [7]
tribal abstraction network	An abstraction network that summarizes points of intersection among subhierarchies of concepts/classes in an ontology. See Ochs et al. [17]
ingredient abstraction network	An abstraction network based on the targets of semantic relationships (e.g., targets of SNOMED CT's attribute relationships and ranges of OWL properties). See Ochs et al. [27]

Table 2

Four examples of implementations of the partial-area taxonomy derivation methodology.

Ontology/format	Concept Type	Inheritable Property
SNOMED CT (relational format)	SNOMED CT concept	Attribute relationship [11]
OWL	OWL class	Object property or Data property (or both), with assigned domains [9] or used in class restrictions (or both) [10]
OBO format	OBO term	Relationship [29]
NDF-RT (Apelon DTS format)	NDF-RT concept	Role relationship

Table 3

Examples of errors found in concepts with different kinds of partial-area-taxonomy-defined characteristics. The OAF was used to identify which characteristic(s) each concept exhibited and to browse the relevant partial-area taxonomies.

Characteristic	NCIt Subhierarchy	Erroneous Concept(s)	Error
Concept summarized by a small partial-area	*Disease, Disorder, or Finding*	*Ameloblastic Carcinoma*	Missing several *Disease may have finding* restrictions (e.g., to *Oral Hemorrhage*). Several incorrect restrictions (e.g., *Disease has primary anatomic site* should have a range of *Oral Cavity*, not *Lip and Oral Cavity*)
		Brain Astrocytoma	Missing superclass Malignant Brain Neoplasm, missing several *Disease may have finding* restrictions
Overlapping concept	*Disease, Disorder, or Finding*	*Refractory Adult Spinal Cord Neoplasm* and *Refractory Childhood Spinal Cord Neoplasm*	Both concepts are missing *Disease has primary anatomic site* restrictions with a range of *Spinal cord*
		Splenic B Lymphoblastic Lymphoma	The *Abnormal cell* restriction does not express malignancy
Concept summarized by root area	*Biological Process*	*ATP Hydrolysis*	Missing an *Is part of process* restriction with a range of *Energy Metabolism Process*
		DNA Folding	Missing a *Has associated location* restriction with a range of *Nucleus*
Concept in high-indexed area in partial-area taxonomy	*Biological Process*	*Granulocyte Differentiation*	Missing superclass *Myeloid Cell Differentiation*
		Nuclear Division	Missing a *Has associated location* restriction with range of *Nucleus*

Table 4

Example of a goal-action encoding for identifying all of the areas that contain more than one partial-area.

Task:	Determine how many areas have more than one partial-area
Goal:	Count the number of boxes with more than one partial area.
	Visually, areas are identified as a colored box and partial-areas are identified as white boxes within the colored area box. The user must identify the colored area boxes with more than one white partial-area box
	Alternatively, the user can identify the list of areas in the “Areas in Taxonomy” list in the Partial-area Taxonomy Details Display and sort them according to their number of partial-areas.
Duration (m:s)	4:32 (*video time stamp*: 13:36–18:08)
Problems	Usability problem: Screen complexity, Granularity
	• User looks for a way to identify solution without having to count boxes. Selects button “Derivation Options” on top of screen and reads options. Does not identify a way to help him solve a problem, so the user counts the boxes on the screen.
	• User counts three times to obtain correct result. Inconsistency/difficulty is likely because participant can't view the entire partial-area taxonomy while completing the task.
Prompts	The user was not prompted with any information
Result	The user was able to identify the areas that contained more than one partial-area