A Basic Introduction to Biomedical Ontologies

Jyotishman Pathak Ph.D.

cabIG® Vocabulary Knowledge Center

Mayo Clinic, Rochester, MN

Version 1.0

June 28, 2010
# Table of Contents

1. Introduction to Ontologies: the What? ................................................................. 3
   2.1 Indexing and Data Annotation ........................................................................ 5
   2.2 Semantic Data Integration and Information Exchange .................................... 6
   3.1 Ontology Building Recipe ................................................................................ 7
   3.2 Ontology Languages ....................................................................................... 10
4. Storing and Accessing Ontologies: the Where? .................................................... 11
5. Concluding Remarks ............................................................................................ 12
6. References ........................................................................................................... 13
1. Introduction to Ontologies: the What?

The word "ontology" in general refers to the branch of philosophy that deals with the study of being. When used in the context of knowledge representation, an ontology is a formal specification of a conceptualization [1]. Ontologies provide the shared vocabulary that enables professionals and scientists in different disciplines to communicate using a "common language" and to know with precision what the terms in their language mean. In biomedical informatics, "ontology" has become a convenient moniker which can mean everything or very little—artifacts created for representing biomedical entities, their terms and relations, often referred to as classifications, vocabularies, terminologies, and ontologies [2, 3]. Previous works [2, 4] have proposed definitions for these artifacts and also attempted to characterize distinctions among them [5], although in practice, these names are often used interchangeably. Hence, for the sake of simplicity, we refer to the knowledge artifacts as ontologies.

The development and standardization of ontologies, at least in biomedical informatics, is not recent. In the early 17th century, to list approximately 200 causes of death, health authorities in London developed a health statistic called Bills of Mortality that was eventually compiled into the International Classification of Diseases [6]. In addition to terms, scientists such as Linnaeus started formalizing the relations among biological entities, in order to represent and share their knowledge of the world [7]. In 1966, the National Library of Medicine added "Medical Nomenclature" to the Medical Subject Headings (MeSH – their list of terms used to index the biomedical literature), and articles indexed with that term jumped to around 500 a year and continued to increase at about 4.5% per year. Similarly, as time went by, the number of citations on ontologies and controlled vocabularies in the PubMed/MEDLINE database has grown by 600% to about 1200 per year. The Gene Ontology, which has as an objective to unify the representation of gene and gene product attributes across all species, is a remarkable example of an ontology that has captured the interest of entire biomedical community [8]. As a matter of fact, Gene Ontology has become the most cited ontology, with approximately 450 PubMed/MEDLINE citations per year [3]. This raises the question of what role ontologies play in representation, exchange and understanding of information. We describe a few in the following [9]:

- **Sharing a common understanding of structure of information among people or software services** is one of the most common goals for developing ontologies [10]. For example, if several different health-related websites and online databases use the same underlying ontology for human anatomy, then software agents can extract and aggregate information from these different sites (e.g., about anatomical sites) to answer user queries.
- **Enabling reuse of domain knowledge** is another driving force behind standardized, community-driven ontology development efforts. For example, different areas of biomedical research need to represent the notion of "gene" and its attributes (e.g.,
biological processes). If a group of researchers, such as the Gene Ontology consortium, develop such an ontology with enough detail and information, others can simply reuse it for their specific projects.

- **Making domain assumptions explicit** in an underlying an implementation makes it possible to change these assumptions easily if our knowledge about the domain changes. Hard-coding assumptions about the world, for example, in software implementation code, make these assumptions not only hard to find and understand but also hard to change, in particular for someone without programming expertise. In addition, explicit specifications of domain knowledge are useful for new users who must learn what terms in the domain mean.

- **Separating the domain knowledge from the operational knowledge** is another common use of ontologies. Developing an ontology is akin to defining a set of data and their structure for other programs to use. Problem-solving methods, domain-independent applications, and software agents use ontologies and knowledge bases built from ontologies as data.

Note that often developing a domain ontology is not a goal in itself. In particular, for the purposes of this paper, we will define an ontology to be a formal, explicit description of concepts in a domain of discourse (classes), and relationships between the classes described by properties that define various features and attributes of the classes, as well as role restrictions that define restrictions on the properties. Furthermore, in some cases, an ontology comprises a set of individual instances of the classes.

For example, a class of *Drug* represents all drugs that can be prescribed, and specific drugs are instances in this class. A particular class can be further refined or specialized by defining subclasses (e.g., antibiotics, antihistamines). Furthermore, properties can describe additional information as attributes for the classes and instances. For example, *Zyrtec* is-a *Antihistamine* and is-manufactured-by *Drug-Company*, indicating that the instances of the class *Zyrtec* will have the property describing its manufacturer.
Hence, in practical terms, modeling an ontology includes:

1.1. Defining the classes and their properties  
1.2. Defining additional sub-classes and arranging them in a hierarchy (super-class and sub-class relationships)  
1.3. Defining individual instances for the classes

In the next several sections, we discuss some of the use-cases for application of ontologies as well as more details about ontology modeling and knowledge engineering principles.

2. Ontologies in Action: the Why?

As illustrated above, the last decade has seen a marked increase in representation of "biomedical entities" in the form of ontologies. Arguably, this has led to their adoption in various aspects of biomedical informatics ranging from clinical decision support [11] to scientific knowledge management [12] to semantic data integration and exchange [13]. In the following (and as discussed in [3]), we discuss two such areas where ontologies are playing an important role in clinical and translational research activities.

2.1 Indexing and Data Annotation

The term "indexing" is typically associated with the notion of assignment of terms and entries from a controlled vocabulary or dictionary to documents, such as biomedical literature. A classic example in this space is the vast collection of articles that are indexed as part of the PubMed resource (http://pubmed.gov), which is composed of more than 19 million citations for biomedical articles from MEDLINE and other life science journals. Each article in PubMed is indexed using the MeSH (Medical Subjects Heading [14]) vocabulary, which contains a list of more than 25,000 "descriptors" arranged hierarchically that are specifically designed to be used for indexing. Even though most of this indexing is done manually, several automatic techniques
have been developed recently [15]. Furthermore, with the inclusion of MeSH in the UMLS Metathesaurus (Unified Medical Language System [16]), it has become possible to leverage the rich UMLS relationships for "co-indexing". For example, GoPubMed [17] uses both MeSH and Gene Ontology for indexing the biomedical literature.

In the clinical world, the indexing of clinical documents and patient records is called "coding". One of the most widely used "code sets" in this context is the International Classification of Diseases (ICD), which has been used for more than a century for coding morbidity, mortality, billing and reimbursement. Another standard that is getting widely adopted, at least within the European Union, is SNOMED-CT (Systematized Nomenclature of Medicine-Clinical Terms [18]). Providing a wide and comprehensive coverage for diseases, findings, procedures, pharmaceuticals etc., SNOMED-CT provides a consistent way to index, store, retrieve, and aggregate clinical data across specialties and sites of care. Similarly, the UMLS Metathesaurus has also been used in some cases for coding of clinical documents [19].

2.2 Semantic Data Integration and Information Exchange

Biomedical ontologies are a required ingredient in the recipe for achieving seamless semantic interoperability and information exchange, along with messaging standards and clinical information models [20]. Some of the notable efforts in this area include BRIDG [21], HL7 Clinical Document Architecture [22], and NCI caCORE [23] projects.

The BRIDG project is a collaborative initiative between the National Cancer Institute (NCI), Clinical Data Interchange Standards Consortium (CDISC), Regulated Clinical Research Information Management Technical Committee (RCRIM TC) of Health Level 7 (HL7), and the Food and Drug Administration (FDA) to develop a model of the shared understanding of the semantics of protocol-driven clinical research. Represented in UML (Unified Modeling Language), the BRIDG model exists as a collection of UML views (diagrams) that describe the declarative semantics (i.e., descriptions of objects, classes, and relationships between classes) and procedural semantics (i.e., descriptions of behaviors, work processes, and organizational processes) of clinical research. In particular, the declarative semantics of clinical research are represented using UML class diagrams that represent the concepts and relationships between those concepts, and the business process or procedural semantics are represented using UML activity and state diagrams. Semantic interoperability between clinical information systems is supported in BRIDG via semantic harmonization. While BRIDG does not directly address the issue of binding ontologies to information models, several recent efforts have proposed techniques for addressing it [24].

The HL7 Clinical Document Architecture (CDA) [22] is a richly expressive model for formal representation of clinical statements, including clinical observations and medication
administrations. In particular, the CDA Release 2 model associates the HL7 Reference Implementation Model (RIM) with ontologies such as SNOMED [18] and RxNorm [25] for representing the semantics of the clinical documents.

The NCI Common Ontologic Representation Environment (caCORE) [23] supports creating a syntactically and semantically interoperable framework for biomedical information services. Systems developed using the caCORE methodology adopt a rigorous approach for defining, registering and using data and representation standards to ensure interoperability. caCORE comprises several important components, one of which is the Enterprise Vocabulary Services [26, 27] that provides the underlying infrastructure for hosting NCI’s description, logic-based ontology—the NCI Thesaurus [28]. Focusing on vocabulary for clinical care, translational and basic research, and public information and administrative activities, the NCI Thesaurus comprises approximately 70,000 concepts organized hierarchically in 19 distinct domains and represents an important element of the caCORE infrastructure.

3. Building Ontologies: the How?

While there is no one-single prescription for building an ontology, which arguably is an iterative process, in the following sections we discuss a general knowledge engineering recipe and methodology for modeling ontologies.

3.1 Ontology Building Recipe

**Scope and domain of the ontology:** A fundamental aspect of building an ontology is to ensure that the concepts in the ontology are closely related to objects (physical or logical) and relationships in the domain of interest. This naturally ties to defining the scope of the ontology which, in many cases, is refined iteratively during the ontology development process. As proposed by Gruninger and Fox [29], one way of determining the scope of the ontology is specify a set of "competency questions" that encompass different aspects of the ontology to be developed. For example, in the context of drugs and medications, the following are the possible competency questions:

- Which side effects or adverse events should I consider when choosing Acetaminophen?
- Is Acetaminophen an antibiotic or a non-opioid analgesic?
- Can Acetaminophen be used to treat nausea?
- What are the different forms and dosage in which Acetaminophen is available (e.g., oral tablet, injection)?
The overall objective of such questions is to serve as litmus that can provide information later about drugs and medications.

**Ontology reuse:** Given that with the emergence of Semantic Web [30] ontologies are being developed, often collaboratively, in multiple domains, there is a fairly high probability that one or more ontologies already exist that covers various "competency questions" outlined above (although the degree of coverage might vary). Therefore, it is considered a best-practice to analyze existing ontological resources, and evaluate if they can be refined and/or extended for the particular domain and task. A key advantage of ontology reuse is that it can enable system-level interaction and interoperability with the applications that have already committed to particular ontologies. Additionally, it could save one from re-invention and duplicating work. In the biomedical domain, the BioPortal ontology library and its suite of services at the National Center for Biomedical Ontology [31] is a good resource to find existing ontologies.

**Creating Classes and Properties:** In theory, there are three main ways for modeling classes and a class hierarchy: (i) in a top-down process, the high-level classes are defined first, and are subsequently refined; (ii) the bottom-up process starts with the most specific classes (the "leaves" in the hierarchy) which are then grouped under high-level classes, and (iii) a hybrid process that is a combination of the top-down and bottom-up processes.

While the specific modeling approach taken is dependent on the ontology developer, it is always considered useful to enumerate the list of terms (we refer to them as "classes") and their properties that the developer is intending to model. For example, in the domain of drugs and medications, some of the things to consider might be different types of drug categories (e.g., antianginals), their physiological effects (e.g., cellular motion alteration) or pharmacokinetics (e.g., hepatic excretion), the strength of a given drug (e.g., 100 mg), or its form (e.g., oral tablet). This step is important to provide a preliminary idea about the scope of the ontology as well without taking into consideration the specific relationships between the ontology concepts and their properties.

**Creating Relationships:** Once the list of classes and their properties have been determined, the next step is to create relationships between the classes, the primary of which is the "is-a" relationship representing a class hierarchy transitively. Formally, the "is-a" relation is defined as follows: a class B is subClassOf class A if and only if every instance of class B is also an instance of class A. For example, Loop Diuretic is subClassOf Cardiovascular Medication. Note that where possible, one should avoid defining cyclical relationships in the class hierarchy. (Stating that class B is subClassOf class A and class A is subClassOf class B amounts to stating that classes A and B are equivalent). Furthermore, it is important to distinguish between a class and its name(s) [9]: a class represents a concept in a domain and not the words that denote the
concept. In other words, even if it is possible to create synonyms for a particular class name, the synonyms do not represent different classes themselves.

Depending on the ontology modeling requirements, it may also be required to define additional relationships between the classes (apart from the "is-a" hierarchy). Continuing the example from above, a drug Streptomycin could potentially have the following relationships defined (adapted from [32]):

- Streptomycin has Physiological Effect of Decreased Protein Synthesis
- Streptomycin has Mechanism of Action of Protein Synthesis Inhibitors
- Streptomycin may treat Mycobacterium and Streptococcal infections

such that the text in *italics* indicates the relationships between Streptomycin and other relevant classes. Note that is also possible (at least in a few formal ontology languages—see Section 2.2 for more details) to restrict the "type" of source and target for a relationship. For example, one could assert that the target for *may treat* relationship cannot be another drug.

**Creating Instances:** One of the last steps in creating an ontology is to define individual instances of the ontology classes. However, a commonly occurring dilemma is to determine whether a particular informational entity should be defined as a class or an instance. For example, should "Streptomycin 500 mg/ml Injectable Solution" be an instance of a class "Aminoglycosides", or should it be defined as a class such that the physical injection administered to you by your nurse practitioner becomes an instance of it?

Commonly, it is argued that the class vs. instance decision hinges on the potential application of the ontology under development (and hence, the level of granularity of representation required): if the goal is to maintain an inventory of all the drugs in a drug store, then individual Streptomycin solutions become individual instances in our knowledge base. But, if our goal is to only describe different pharmacological aspects of a drug, such that, Streptomycin is the most granular term we use, then it becomes an individual instance of the class Aminoglycosides.

Another way of addressing this confusion is to analyze whether the individuals form a natural concept hierarchy, and hence should be represented as classes. For example, Aminoglycosides cannot be an individual instance of Antimicrobials because it is possible to further classify Aminoglycosides in a classification hierarchy.
3.2 Ontology Languages

A key aspect for a recipe to successfully build ontologies is the underlying language used for representing different elements of the ontology. While a survey of all the existing ontology languages is beyond the scope of this document, here we primarily focus on four widely used formal ontology representation languages: Resource Description Framework Schema (RDFS [36]), Web Ontology Language (OWL [33]), Open Biomedical Ontologies (OBO [34]) and Ontylog [35].

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. It is based on the idea of identifying things using Web identifiers (called Uniform Resource Identifiers, or URIs), and describing resources in terms of simple properties and property values. This enables RDF to represent simple statements about resources as a graph of nodes and arcs representing the resources, properties and values. RDF, however, provides no mechanisms for describing these properties, nor does it provide any mechanisms for describing the relationships between these properties and other resources. That is the role of the RDF vocabulary description language, RDF Schema (RDFS). RDFS defines classes and properties that may be used to describe classes, properties and other resources. In other words, RDFS is a semantic extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources.

The Web Ontology Language (OWL) is a standard recommendation from the World Wide Web Consortium (W3C) for formally representing an ontology. Based on a logical model, OWL provides the ability to describe "simple" concepts as well as a rich set of operators (e.g., union, intersection, negation), which enables defining complex concepts using the set of simple concepts. Furthermore, the logical model allows the use of a reasoning engine that can check for logical consistency of the statements and definitions used to define the concepts, as well as create a “parent-child” hierarchy for the ontology concepts. OWL provides three "flavors" of the language, OWL-Lite, OWL-DL, and OWL-Full, which have increasing expressivity.

OBO is an ontology language that is often used to model ontologies in the life sciences domain. OBO is similar to description, logic-like languages, such as OWL, albeit simpler. An important aspect of OBO is to provide the ability to track large amounts of meta-data about the ontology and provide mechanisms for some basic history auditing. Examples of such metadata include database cross-reference (dbxref), definitions for entities, user-defined categories that can be used for placing/organizing an entity, and so on.

Onytylog is another description, logic-based language which has been primarily implemented by Apleon Inc. in their proprietary Terminology Development Environment [35]. Hence, similar to OWL and OBO, Onytylog's fundamental construct is a concept which can be defined
compositionally in terms of other concepts, thereby inheriting information. In particular, a concept is modeled in the semantics of Ontylog as a set of things, and binary relations (called roles) between these sets determine how a subsumption hierarchy is created. Another key feature of Ontylog is the support of "kinds" that allows a given ontology under certain conditions to be partitioned into smaller graphs for the purposes of classification and reasoning scalability.

4. Storing and Accessing Ontologies: the Where?

Once an ontology for a particular domain has been developed (potentially by a group of experts), several services and modes are available for its public dissemination. We discuss two such resources: BioPortal [31] and Enterprise Vocabulary Services (EVS [27]).

BioPortal (http://bioportal.bioontology.org/) is an open publicly available repository that provides access to numerous biomedical ontologies, which were developed using OWL, OBO and other languages, via Web services and user navigable browser interfaces. BioPortal users can browse and search the ontologies, submit new ontologies or updated new versions of the existing ontologies in the repository, provide comments and metadata about the ontology and/or its contents, make suggestions to ontology developers, and so on. Furthermore, users can perform mappings between two or more different ontologies—a feature that is highly relevant in harmonizing local ontologies to standardized ontologies. Each mapping has its own set of metadata that describes who created the mapping and when, which algorithm was used to produce the mapping, application context in which the mapping might be valid, the specific mapping relationship, and other properties. Users can add notes on existing mappings and carry out discussions about the mappings. BioPortal also enables integrated search of biomedical data resources such as the Gene Expression Omnibus (GEO), ClinicalTrials.gov, and ArrayExpress, through the annotation and indexing of these resources with ontologies in BioPortal.

Enterprise Vocabulary Services (https://cabig.nci.nih.gov/concepts/EVS/) is an on-going effort at the NCI for supporting its broad ranging needs for terminologies and metadata to support the caCORE and caBIG infrastructure. Similar to BioPortal, EVS has developed browsers and interactive user interfaces for ontology viewing and browsing (https://cabig-kc.nci.nih.gov/Vocab/KC/index.php/NCI_Term_Browser). Furthermore, EVS also developed tools for collaborative, community-based ontology development in the form of the BiomedGT wiki platform (https://cabig.nci.nih.gov/tools/BiomedGT_Wiki). The goal of the project is to build a vocabulary server accessed through a well-structured application programming interfaces (API) capable of accessing and distributing vocabularies as commodity resources. The server is to be built using standards-based and commodity technologies, and leverages Mayo Clinic’s LexGrid model for standard storage of controlled vocabularies and ontologies. The model defines how
vocabularies should be formatted and represented programmatically, and is intended to be flexible enough to accurately represent a wide variety of vocabularies and other lexically-based resources. The model also defines several different server storage mechanisms (e.g., relational database, LDAP) and an XML format. This model provides the core representation for all data managed and retrieved through the EVS system, and is now rich enough to represent vocabularies provided in numerous source formats including OWL and OBO. More information about this server, called LexEVS, is available at: https://cabig-kc.nci.nih.gov/Vocab/KC/index.php/LexBig_and_LexEVS

5. Concluding Remarks

Biomedical ontologies provide essential domain knowledge to drive data integration, information retrieval, data annotation, natural-language processing and decision support. Developing an ontology for a particular domain requires training and background in ontology development best practices. This document has provided a brief overview in some aspects of that process. We encourage readers to visit the Vocabulary Knowledge Center at https://cabig-kc.nci.nih.gov/Vocab/KC/index.php/Main_Page for further information on these topics.
References


