Basic considerations for improving interoperability between ontology-based biological information systems

Der Fakultät für Mathematik und Informatik
der Universität Leipzig
eingereichte
DISSERTATION
zur Erlangung des akademischen Grades
DOKTOR RERUM NATURALIUM
(Dr. rer. nat.)
im Fachgebiet
Informatik
vorgelegt

von Robert Höhndorf
geboren am 30.05.1980 in Leipzig

Leipzig, den 05.03.2009

For my parents

Abstract

Ontologies are used in biology for the description of multiple kinds of entities. Large ontologies provide categories and relations for the basic features found in databases of model organisms. They serve as the basic means to integrate the data that is generated and interpreted by multiple heterogeneous groups and stored in distributed biological databases throughout the world. The use of a common vocabulary and common formal descriptions of the vocabulary's terms permit the comparison, retrieval and analysis of the data stored in these databases. The ontologies that are used for this purpose are primarily isolated, single-domain ontologies that have little or no interconnections specified among them. Ontology communities such as the Open Biomedical Ontologies (OBO) and the OBO Foundry establish guidelines to maintain quality and reusability of ontologies, and to facilitate interoperability between ontologies that are included in these projects.

I identify several facets of interoperability between ontology-based information systems in biology which are not currently addressed satisfactorily. First, the knowledge representation languages used to represent ontologies must be sufficiently rich to express the distinctions made by the ontology designers, and required by the applications of the ontology. Second, the basic categories of the biological ontologies must be analyzed and integrated within a common conceptual framework to permit information to flow between the ontologies. Finally, to let information flow between domain ontologies, the acquisition of additional knowledge from domain experts is required.

Most biological ontologies are represented in the OBO Flatfile Format and the Web Ontology Language (OWL). I propose extensions to both forms of representing biological ontologies. The semantics of the OBO Flatfile Format is not explicit, and the current proposals for a semantics of the OBO Flatfile Format do not coincide with the way it is used in many ontologies, in particular in statements that use negation. Therefore, I propose a more flexible semantics through a translation to OWL. The decidable version of OWL is equivalent to an expressive description logic. However, it is based on classical logics and exhibits the property of *monotonicity*. When combining ontologies, it is beneficial to consider alternative, non-classical logics that permit *nonmonotonic* inferences. I propose a method for integrating biological ontologies which are formalized either in the OBO Flatfile Format or OWL using a default logic.

Core ontologies provide an ontological foundation for domain ontologies by extending top-level ontologies with domain-specific axioms. They can be used to integrate domain ontologies and as a starting point for the development of new ontologies within a domain. I introduce the biological core ontology *GFO-Bio*. GFO-Bio is implemented in OWL and first order logic, and is accompanied by axioms in default logic. I include several elaborated modules in GFO, such as a module for biological functions, disposition or biological sequences. Additionally, I illustrate how GFO-Bio can be used to integrate biological domain ontologies and facilitate information flow among them.

To integrate biological domain ontologies using GFO-Bio or any other top-level or core ontology, additional knowledge about the interrelations between domain categories must be acquired from domain experts. Due to the large number of categories in these ontologies, such an effort is time-consuming and expensive. Methods and software applications that permit a large number of domain experts to collaborate on this task would enable the rapid and cheap acquisition of ontological knowledge. For this purpose, I introduce the BOWiki, an ontology-based semantic wiki, and a social tagging system. In addition, I suggest several novel methods for automatically extracting data and knowledge from natural language texts. Automated extraction of biological knowledge can provide an alternative to manual curation of ontologies and their annotations, or serve as a starting point for manual efforts of knowledge acquisition.

The primary focus of this work is the development and discussion of novel methods for improving interoperability between biological domain ontologies. These are classified in three major categories, and the relations between them are analyzed. I show how their application leads to improved interoperability and increased usability of the ontologies.

Acknowledgements

This work would not have been possible without the help of many people. I am deeply grateful to my supervisors Prof. Dr. Heinrich Herre and Dr. Janet Kelso. Without their continued support, helpful advice and constant discussions, this work would not have been possible.

Many thanks go to my colleagues Frank Loebe, Patryk Burek and Hannes Michalek for the many discussions we had on all aspects of ontology, and to Michael Backhaus, Alexandr Uciteli and Joshua Bacher for their creative ideas, suggestions and, most of all, their excellent work on the BOWiki and the tagging software. I enjoyed the weekly text mining meetings with Michael Dannemann and Axel Ngonga very much. I thank Kay and Johann. Finally, I am grateful to my parents for their continued support.

Contents

List of Figures

List of Tables

1 Introduction

> The word “definition” has come to have a dangerously reassuring sound, owing no doubt to its frequent occurrence in logical and mathematical writings.
>
> Willard van Orman Quine

Progress in biology has brought about a rapid increase in data generated by scientists working in this field. In particular, the field of molecular biology produces large amounts of novel data and knowledge. This data is often stored in distributed, heterogeneous databases. Findings pertaining to this data are communicated in scientific publications and sometimes stored in these databases.

To compare and integrate the data stored in different database for further scientific analyses, common vocabularies were developed to provide descriptions for some of the data’s features. Biomedical ontologies are formal specifications of the conceptualizations underlying these vocabularies. They describe the *meaning* of the terms in the vocabulary. Many ontologies have been developed for different biomedical domains. Ontologies covering domains from molecular functions and processes to organism-specific anatomy, to ontologies for species, are available.

But “merely using ontologies [...] does not reduce heterogeneity: it just raises heterogeneity problems to a higher level” [Euzenat and Shvaiko, 2007]. Particularily in the biological and biomedical field, the number of ontologies has grown rapidly over the past ten years. They were developed to solve the problem of describing the features of data in a uniform and well-defined way, and through this description to facilitate queries across the model organism databases. The immediate problem of integrating the model organism databases lead to the development of ontologies that did not always meet quality standards necessary to facilitate interoperability.

Interoperability between ontologies has been researched in the area of computer science and knowledge represention for some time. In the field of biomedical ontologies, some investigations pertaining to interoperability are underway and several guidelines to achieve interoperability have been suggested [Smith et al., 2007]. These guidelines primarily establish social criteria.

An analysis of the problem of interoperability yields several dimensions of requirements for interoperability of which the social dimension is only one. Additional requirements for interoperability between ontology-based information systems pertain to logic and knowledge representation, to formal ontology and to knowledge acquisition. Central questions in each of these dimensions remain unanalyzed, and are not yet addressed in any established criteria.

The important question that must be answered in the realm of logic is how two ontologies that are represented as logical theories can be combined or connected in such a way that information can flow between them. The easiest form of connection is to combine both theories into one. But assuming that both ontologies are represented as consistent theories, it is by no means obvious whether the combined theory is consistent [Dimitrakos and Maibaum, 2000].

In addition, many biomedical ontologies are not, *per se*, represented as logical theories but must first be translated into a formal representation. This translation may be the source of inconsistencies or errors. Formalizing the ontologies in a formal language is a necessary requirement if the ontologies are intended to be used for drawing logical inferences or employing algorithms to verify their consistency.

Ontologies are specification of the basic concepts that govern a domain. They are used to explicitly specify the ontological commitment of a vocabulary [Guarino, 1998]. This ontological commitment is frequently not made explicit in ontologies of biology and biomedicine. This lack of an explicit specification leads to potentially incompatible ontological commitments and to formal inconsistencies when multiple ontologies are combined. It is mandatory to make the ontological commitment underlying these ontologies explicit to facilitate interoperability between information systems based on them.

Neither the ontological analysis, nor an adequate form of knowledge representation alone suffice for information to flow between information systems based on ontologies. Additional knowledge is required to specify the connections of domains in reality. Research in biology and biomedicine identifies these connections. Discovering the connection between genotypic information and phenotypic phenomena is one of the most prominent areas of research in genetics today. Its goal is to identify how genotypic and phenotypic phenomena are connected, and what relations exist between these domains and the levels of granularity that lie between them. Letting information flow between ontologies of these domains and levels of granularity necessitates the acquisition of the knowledge discovered in this research, because information flow between these ontologies must obey the relations that exist between the domains covered by the ontologies. The acquisition of this knowledge necessitates the development of new ontology-based software applications that facilitate the efficient acquisition of knowledge from domain experts.

The division into the three topics logic, ontology and knowledge acquisition, and how they contribute to the interoperability of ontology-based information systems, governs the remainder of this work. It is the result of an attempt to systematically account for conditions for interoperability between ontology-based information systems in biology and biomedicine.

The structure of this work is as follows: In chapter 2, I provide background information about ontologies in the life sciences. I discuss top-level ontologies in section 2.2. I use these ontologies as foundation for the remainder of my analysis. In section 2.3, I describe several biological domain ontologies. They were chosen either for their unique features or to serve as example for a whole group of ontologies. The discussion in chapter 3 provides a thorough analysis of the problem of interoperability between ontology-based information systems in biology and biomedicine. I give both a definition and a formal account of interoperability, outline the importance of achieving interoperability and discuss dimensions of requirements for interoperability. In this chapter, the outline of the remaining thesis is motivated. Chapters 4, 5 and 6 address the areas of requirements on knowledge representation, formal ontological analysis and knowledge acquisition, respectively. The discussion in these chapters follows the problem description provided in chapter 3. Chapter 7 summarizes the findings and contributions, presents conclusions and provides an outlook.

2 Background

> Metaphysics may be, after all, only the art of being sure of something that is not so, and logic only the art of going wrong with confidence.
>
> Joseph Wood Krutch

Modern biology generates large amounts of data that must be analyzed and interpreted. In modern molecular biology, a large amount of data is generated by sequencing and analyzing whole genomes, microarray experiments, etc. Sequencing entire genomes has become cheaper with the development of new technologies such as high-throughput sequencing [Illumina, 2007, Margulies et al., 2005], which increases the amount of generated data further.

A variety of model organisms[1] are studied, including mice [Bult et al., 2008], fruit flies [Flybase, 1999] and worms [Rogers et al., 2007]. Additionally, extinct organisms such as neanderthals [Green et al., 2006] can now be studied, as can the genomes present in environmental samples [Allen and Banfield, 2005]. Fundamental biological functions, processes and structures are often shared

[1] A model organism is an organism which is extensively studied in biology, due to its exemplary features. It is assumes that the investigation of the model organism yields insights that are valid for other organisms as well.

between different organisms due to their common evolutionary origins. With the large-scale analysis of biological data and the rapid increase of knowledge that is made possible by modern technologies, a communication problem exists within the biological community; similar or identically-formed biological entities were named differently by independent groups [Bada et al., 2004]. As a result, comparison of the results obtained by these groups became a difficult problem. For example, the terms “programmed cell death”, “cell death” and “apoptosis” may all refer to the same kind of process, or they may all refer to different processes. If they refer to different kinds of processes, they may have nothing in common, overlap in their intension, or stand in different, more complex relations towards each other.

Additionally, research communities have developed in different locations and developed their own vocabularies, their own databases and applications that are governed by different schemata. Although these databases and vocabularies often overlap significantly, exchanging information between them is not always trivial due to their different ways of describing data and knowledge, and the use of different platforms and database management systems.

One solution to this communication problem is the use of controlled vocabularies as a foundation for data exchange and communication within a community. One of the first controlled vocabularies, and the currently most sucessful, is the Gene Ontology (GO) [Ashburner et al., 2000] for terms related to biological processes, molecular functions and cellular components that are relevant for gene products. The use of the GO led to the standardization of the terminology used in the model organism databases, and was first adopted by the fruitfly, worm, mouse and *Arabidopsis thaliana* databases. In contrast to earlier efforts to create standard terminologies and knowledge bases based on methods from computer science, artificial intelligence and knowledge representation [Rector et al., 1993], the GO was light-weight, easy to understand and apply, and developed primarily by a community of biologists in order to solve one particular problem: “to provide a common vocabulary for describing gene products [...] for the primary purpose of consistently annotating entries in biological databases” [Bada et al., 2004].

While the GO did provide a common terminology for the standardization of biological databases, several problems emerged in the GO. With increasing size and wide and diverse applications, numerous errors and limitations within the GO were identified [Smith et al., 2003, 2005a, 2004a], some of which had been encountered before in knowledge representation [Guarino and Welty, 2004], but many of which required new methods and additional research for their solution.

Following the GO’s success, numerous controlled vocabularies have been developed for other areas of biology and biomedicine, covering a wide range of biological phenomena. While many of these ontologies were developed with the awareness of the other ontologies, their interoperability and integration continues to be a major area of research [Smith et al., 2007]. Multiple ontologies focus on different aspects of similar or identical entities or types of entities, but explicit interrelations between different controlled vocabularies and ontologies are rare.

The lack of a common ontological foundation led to multiple implicit conceptualizations for a domain and to logical and ontological errors in the representation of the ontologies. Automated verification of the ontologies is hindered by their lack of formalization. Meaningful data queries across multiple ontologies would permit insight into multiple aspects and dimensions of a biological entity. But these queries presuppose an ontological understanding of the connection between the kinds of entities described in different controlled vocabularies and ontologies.

Interoperability between multiple ontologies is not the only problem remaining in the area of biomedical ontologies. A single controlled vocabulary representing only one type of entity should do so in an agile way in order to benefit most users and use cases, without sacrificing ontological and logical accuracy. Some ontologies, however, do not make their ontological commitments explicit. This leads to problems not only in interoperability with other ontologies, but also within its own structure. Before the problem of interoperability between ontologies can be addressed, problems that exist within the ontologies must be solved and their ontological commitment made explicit.

Finally, knowledge aquisition in biology is currently a slow, time-consuming and expensive process. The annotation of gene products with categories from biological ontologies and the curation of these ontologies themselves is done manually by few experts. In particular the annotation of data with controlled vocabularies is a bottleneck that slows the progress and utilization of ontologies in biomedicine. Alternative curation and annotation models, such as those based on collaboration and annotation by a community or the extraction of information from natural language texts, may provide the means to overcome this bottleneck.

2.1 Biological and Biomedical Databases

A large number of biological databases exist. Some contain information on protein functions and sequences [Consortium, 2007], protein families [Mulder et al., 2005] or DNA sequences [Benson et al., 2005]. Several of them have developed into central resources for the biological research community, including those which provide organism specific information [Bult et al., 2008, Twigger et al., 2007, Flybase, 1999, Sprague et al., 2007]. These organism-specific databases are generally developed and maintained by research groups focusing on the study of these organisms.

Many of these databases are manually curated. Professional database curators manually analyze scientific publications and enter the relevant information in the databases. Additionally, they may review and verify information that was automatically generated. For example, the UniProt Knowledge Base contains two components: Swiss-Prot, a manually annotated knowledge base of protein information and TrEMBL, which is automatically generated [Boeckmann et al., 2003].

Organism-specific databases collect information pertaining to a single species. Often multiple kinds of data are collected in these databases. For example, the Mouse Genome Informatics database [Bult et al., 2008] contains among others information about genes, phenotypes, gene expression or organ functions of mice.

Different species are often similar in large parts of their genomes. Genes with similar sequences often share a common function on a molecular level [Ashburner et al., 2000]. On a larger scale, organ functions are shared among many mammals and fundamental biological processes like *glycolysis* occur in most organisms.

Comparative studies between species necessitate an integrated view on the genomic data [Chicurel, 2002, Ureta-Vidal et al., 2003]. In particular, a comparison of a gene's functions within different kinds of organisms requires an analysis of the data pertaining to this function across multiple species. Similarily, comparing other features such as gene expression or phenotypes requires the use of a common or at least compatible vocabulary for describing these features. The Gene Ontology and subsequently other domain-specific biomedical ontologies were developed for this purpose. Before I provide an analysis of these domain-specific ontologies, I introduce a common reference framework in the form of a top-level ontology which serves as the basis for further analyses.

2.2 Formal Ontology

2.2.1 Formal Ontology in Information Systems

Several definitions for an ontology have been presented [Guarino, 1998, Gruber, 1995, Herre et al., 2006, Smith, 2004]. For this work, I adopt the following definition due to [Guarino, 1998]:

> An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models.

A conceptualization is a system of categories that accounts for a certain perspective on reality. Conceptualizations are *intensional* accounts of the categories and relations that govern reality according to the perspective taken on reality in the conceptualization. An ontology, on the other hand, is an engineering artifact that depends on language. It consists of a vocabulary that is used to describe (a part of) reality and a set of explicit assumptions that specify the intended meaning of the vocabulary's elements [Guarino, 1998]. In this sense, I use the short form of an ontology as the "explicit specification of the conceptualization of a domain" [Gruber, 1995].

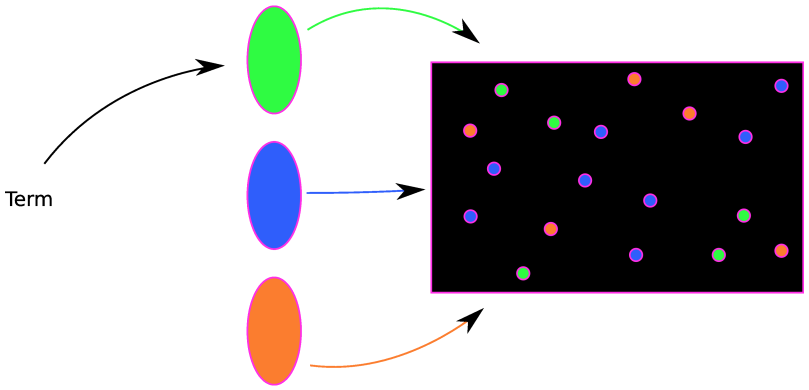


Figure 2.1: The figure illustrates a simplification of the relations between terms, conceptualization and reality. Terms relate to reality through the concepts in a conceptualization. A term is illustrated on the left side of the figure, reality on the right and the conceptualization in the center. Entities in reality that fall under a certain concept are illustrated in the same color as the concept in the conceptualization.

There are limitiations to the kind of information ontologies represent. Ontologies explicitly specify the meaning of terms in a language by formalizing *how* a term refers to reality. Terms in a language refer to reality through the categories in a conceptualization, as illustrated in figure 2.2.1. Depending on the conceptualization, terms can refer to reality in different ways. In minimal philosophical ontologies (conceptualizations) such as the General Process Theory (GPT) [Seibt, 2008] or Armstrong [1997], the conceptualization may consist of a single category. Therefore, terms of a language that is committed to one of these conceptualizations will always refer to one kind of entity in reality. On the other hand, conceptualizations that provide more categories such as those underlying most upper-level ontologies provide multiple categories with different properties. Different ontological commitments of languages can influence the way that theories in these languages are formulated.

Ontologies do not represent contingent knowledge, in particular not scientific theories. Scientific theories can be shown to be *false* through counter-examples and observations that contradict the predictions made by the theory [Popper, 1994]. A statement in an ontology cannot be false in this sense [Rector, 2008], because ontologies provide the foundation for making and reporting about observations in the world.

An ontology can be inconsistent, inapplicable for a specific purpose, or incomplete. It is inconsistent if it contains a contradiction. Some ontologies are developed for a specific purpose and cannot be applied to different use cases without modifications to the ontology. An ontology is incomplete if it does not completely cover the concepts governing one domain and needed in the intended application of the ontology.

For an ontology to be incorrect, i.e., to contain a false statement, it must be possible to find a counter-example, i.e., to show that the statement does not correspond to reality. Ontologies specify concepts, *meanings* of terms, and therefore provide the foundation on which true and false statements can be constructed.

There are two possibilities how a statement in an ontology can be considered incorrect. Either a term is not used in the meaning specified by the ontology (the ontology does not correspond to the intended meaning of a term), or a concept in the ontology does not refer to anything in reality (the ontological category has no instances). Usage of a term refers to usage in natural language. Therefore, an ontology may label its categories inadequately. As a formal theory, however, it specifies the meaning of terms in a formal language. Therefore, I do not consider it wrong to label ontological categories arbitrarily and different from their use in natural language[2]. I consider concept labels and their

[2] I take ontologies to represent concepts, not terms or their usage in natural language.

synonyms as being outside of the ontology. Alternatively, the formal representation of a category in the ontology may not correctly capture its intended meaning. I consider this as a case of incompleteness of the ontology, because the intended category is not yet included.

On the other hand, concept may not refer to anything in reality, i.e., there may be no instances of a category defined in an ontology. Again, I consider the question of whether a category specified in an ontology has instances in most cases as being outside the realm of ontology. An ontology specifies how a term refers to reality. Reality may be contingently structured in such a way that nothing falls under this meaning. This, however, does not invalidate the specification of this meaning in the ontology, as long as it is *possible* for a category to have instances. An example of such a category is *Unicorn*. Unicorns do not exist, but they could exist. Therefore, the *meaning* is valid and *Unicorn* can therefore be a category in an ontology. *Unicorn*'s lack of instances is a contingent fact. In my view on ontologies, contingent existence is outside of ontology. As a corollary, ontologies rarely contain existential statement. The only exceptions are existential statements for entities which necessarily exist, such as the empty set $\emptyset$ or the number 0.

2.2.2 General Formal Ontology

The General Formal Ontology (GFO) [Herre et al., 2006] is a formal foundational ontology developed by the Onto-Med Group at the University of Leipzig. It is the successor of the General Ontological Language (GOL) project [Heller and Herre, 2004]. The GFO is based on principles taken from computer science, logics and philosophy.

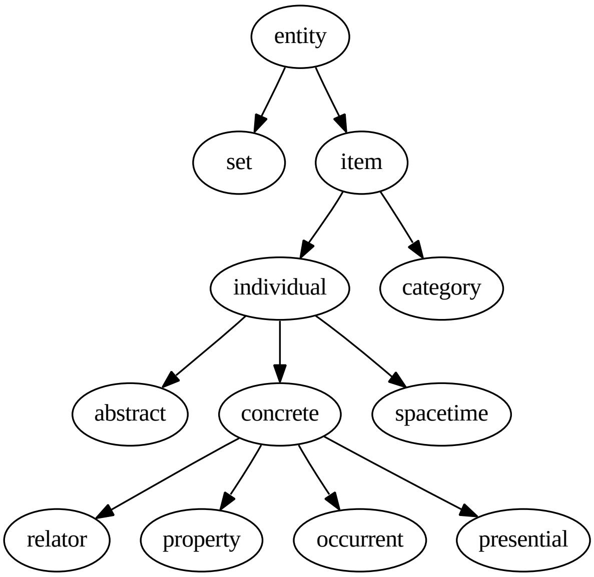


Figure 2.2: The taxonomic tree of the GFO.

Basic taxonomic structure

Categories

The basic taxonomic structure is illustrated in figure 2.2.2. The top-level distinction is between set and item. Sets are extensionally defined entities of set theory. They satisfy the axioms of classical set theory such as the axioms of ZFC [Zermelo, 1908]. All entities that are not sets are considered items. Items are further divided into categories and individuals. Categories are entities that are general in reality. They can be instantiated, i.e., predicated of things. Individuals are items that cannot be instantiated. Examples of categories are *Apple*, *House*, *Marathon*, the letter *A* and *Unicorn*. The GFO distinguishes several types of categories. One distinction pertaining to categories is made based

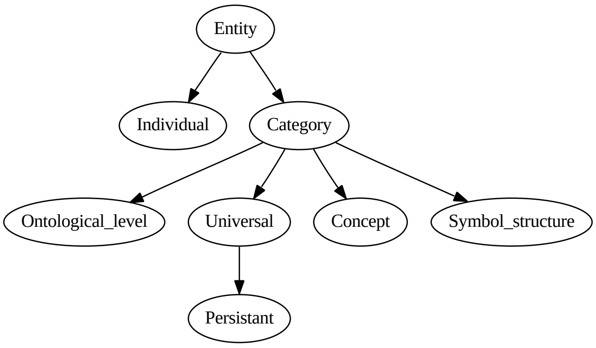


Figure 2.3: The categories of the GFO.

on the kind of instances categories have: there are categories of processes, of properties, of invididuals and of categories. Categories are not restricted to *first order categories* which have individuals as instances, but the GFO permits higher order categories with categories as instances. A further distinction can be drawn between the types of categories: universals, concepts, symbols and levels of reality. Universals are similar to Aristotelian universals in that they exist *in re*, concepts are mind-dependent entities, while symbols require conventions and possibly social facts for their existence [Gracia, 1999]. Figure 2.2.2 shows the category part of the GFO's taxonomic tree.

I will use the more general term “category” throughout this thesis, except when I address explicitly mind-dependent entities or symbols.

Levels of Reality

The GFO includes a theory of levels of reality, illustrated in figure 2.2.2. The first well-developed theory of levels of reality can be traced back to the philosopher Nicolai Hartmann [Hartmann, 1942], and has been continuously developed further [Poli, 2001, Gnoli and Poli, 2004]. Two principally different ways of defining levels of reality have been proposed, one based on interaction of objects or *individuals*, the other taking a categorical approach.

The first approach to the problem of levels of reality assumes a level of reality to be defined by objects of a specific kind and their interactions. For example, atoms and their interactions form one level, while molecules and their interactions define another one. The relationship between atoms and molecules is an inter-level relationship. As Poli [2001] points out, problems arise when levels are not ordered in a linear hierarchy, but non-linear relationships between levels are permitted.

Therefore, the second option defines levels of reality as a group (or system) of ontological categories, and this is the approach taken in the GFO. A level of reality is a system of interrelated categories, and the level itself is captured by means of a higher-order category of which the categories of the level and their interrelations are instances. Levels themselves may be interrelated in particular ways [Poli, 2001]. In particular, the categories of a higher level may depend[3] on the categories of a lower level.

Three major levels of reality, called *ontological strata*, can be distinguished: the *material stratum*, the mental or *psychological stratum* and the *social stratum* [Herre et al., 2006]. Each of these is further organized into sublevels, where scientific fields like physics, chemistry, or biology provide starting points for identifying such sublevels.

[3] The dependence here is not existential dependence, but another, as yet not further analyzed, form of dependence between categories. For example, the category $Molecule$ depends on the categories $Atom$ and $CovalentBond$.

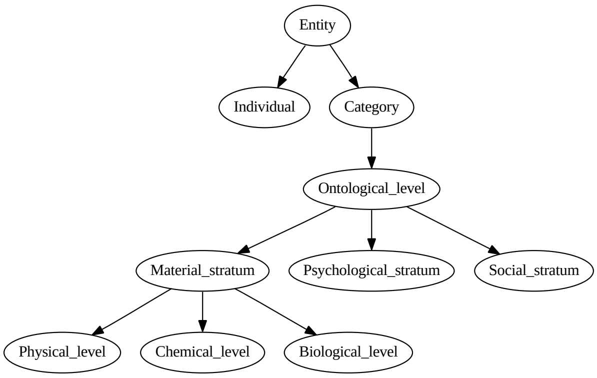


Figure 2.4: Levels of reality in the GFO.

Individuals

Individuals are entities that cannot be further instantiated. The GFO distinguishs three types of individuals: abstract individuals, concrete individuals and space-time individuals. The latter are the building blocks of space and time: time points and intervals, chunks of space and their boundaries. Concrete individuals are located in space and time, while abstract individuals are not. Abstract individuals are things like the number 0 or π. Examples for concrete individuals are the Ironman 2007 in Hawaii, the Eiffel tower or Napoleon.

An alternative way for dividing individuals is between dependent and independent individuals. Dependent individuals are *ontologically dependent* on some other entity [Correia, 2005], while independent individuals are not. The GFO considers substances and processes as independent, properties and some roles as dependent entities. Time-boundaries are dependent on an interval, and spatial boundaries are dependent on a chunk of space.

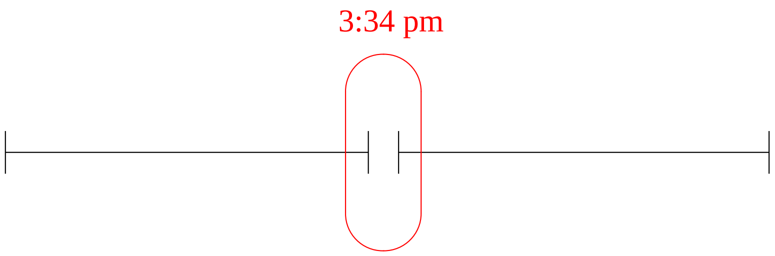


Figure 2.5: Two chronoids with coinciding time-boundaries in the GFO.

Space and Time

The model for space and time used in the GFO is based on the theories of the philosopher Brentano [Brentano, 1976]. Fundamental time entities in the GFO are called *chronoids*. A chronoid is a connected, temporally extended region of time. Every chronoid gives rise to two time boundaries, its left and right *time boundary*. Time boundaries are dependent on chronoids, and they are not temporally extended. Boundaries of different chronoids may **coincide**. Coinciding time boundaries are *at the same time*, but distinct. I will call two or more coinciding time boundaries a *time point*. Figure 2.5 illustrates the relations between two chronoids which **meet** [Allen and Hayes, 1989].

The theory of space in the GFO is also based on Brentano's works. *Topoids* are connected regions of space. Topoids have two-dimensional boundaries (areas), which have one-dimensional boundaries (lines), which in turn have zero-dimensional boundaries (points). Boundaries of the same dimension may coincide.

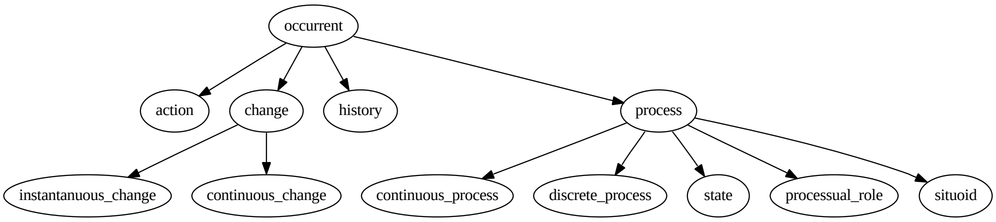


Figure 2.6: Basic classification of processes in GFO.

Presentials

Presentials are individuals that exist at exactly one time boundary. They are wholly present at the time at which they exist, they do not have temporal parts. The notion of a presential is rare in formal ontology, and corresponds to one aspect of endurants or continuants. It is, however, not equivalent to the notion of endurant. Presentials do not persist in time or change their properties. They exist at a single time boundaries and are not present at any other time. Persistance through time is analyzed by means of a special type of category, the persistant, and an abstract individual, the perpetuant (see section 2.2.2). An example of a presential is a specific apple at a specific time boundary.

Presentials depend for their existence on processes. In particular, a specific apple is not uniquely determined by a point in time (approximated by two coinciding time boundaries), but rather by a time boundary. The time boundary starts or ends a chronoid, and is existentially dependent on it. Since processes are framed by chronoids, presentials can be associated to processes. In a sense, processes are considered ontologically prior to presentials.

Processes and Occurrents

Every concrete individual that is not a presential, i.e., that does not exist at exactly one time boundary, is a *processual entity*. A *process* is a temporally extended independent individual. Processes have temporal parts. Processes are framed by a chronoid, they have a duration.

When considering two different time boundaries, the category of *change* can be defined. An *instantanuous change* is determined by two coinciding process boundaries, that differ in at least one category[4]. Figure 2.2.2 shows GFO's classification of processes.

Relations and Properties

Properties and relations are concrete individuals. Properties depend on a bearer. Relationships can be considered *n*-ary properties that inhere in multiple entities.

Properties **inhere in** their bearers. To analyze, for example, the property of an apple's being red, four entities are relevant: the $Apple$ category, the individual apple a, the *Red* category and the individual r. The individual a is an instance of $Apple$ and r is an instance of Red. The property r inheres in the apple a.

Relations are individuals: they are "the glue that holds things together, the primary constituents of the facts that go to make up reality". The relata of a relation participate in the relation in different ways. Therefore, relations are divided into relational roles. Relational roles are individuals, and are dependent

[4] In the sense that an instance of a category is present at one boundary but not at the other or *vice versa*.

on a player and a context: an entity *e* plays a role *r* within a relation *t* [Loebe, 2007].

Identity in GFO

Based on the GFO's model of time, presentials exist at exactly one time boundary. To analyze that some entity persists through time, that it maintains its identity, presentials are not sufficient. The GFO uses a kind of abstract individuals named *perpetuants* together with processes to analyze persistence through time. Perpetuants are abstract individuals that are abstractions of only the identity phenomenon. They **exemplify** presentials that are identical with respect to a perpetuant. In addition, a process that has as particiants only these exemplified presentials captures the dynamic aspect of the persistence of the object.

As example I use the famous Theseus' paradox. It is sketched in figure 2.2.2.

> The ship wherein Theseus and the youth of Athens returned from Crete had thirty oars, and was preserved by the Athenians down even to the time of Demetrius Phalereus, for they took away the old planks as they decayed, putting in new and stronger timber in their place, insomuch that this ship became a standing example among the philosophers, for the logical question of things that grow; one side holding that the ship remained the same, and the other contending that it was not the same.
>
> Plutarch, "Theseus".

Example 1. *The presential s_1 is the ship of Theseus at time boundary t_1. s_1 is exemplified by two perpetuants, S_1 and S_2. S_1 exemplifies the presentials s_2, s_4, s_6 and s_8, while S_2 exemplifies the presentials s_3, s_5, s_7 and s_9 at the time boundaries t_2, t_3, t_4 and t_5. All these presentials that are exemplified by S_1*

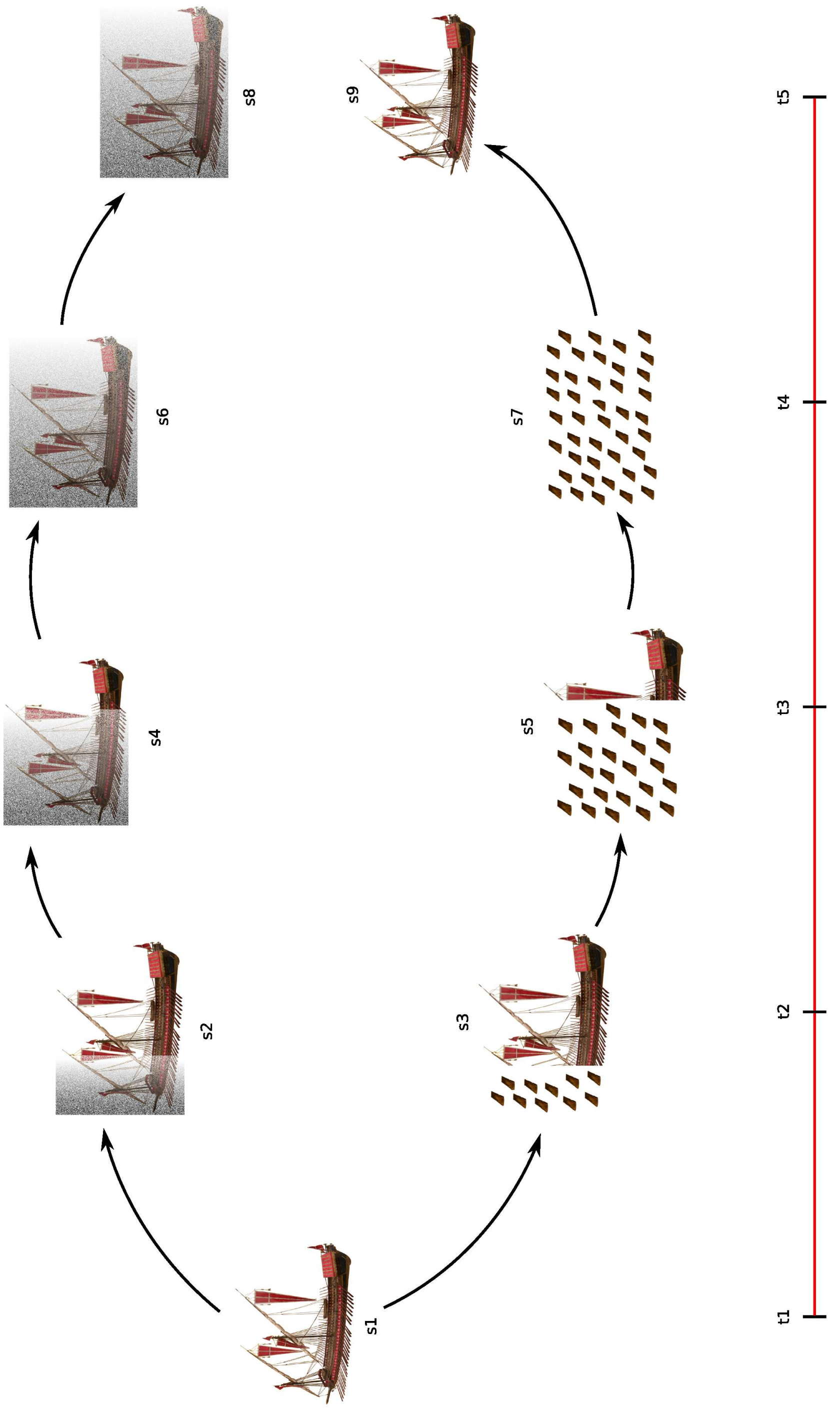


Figure 2.7: Illustration of modelling Theseus' paradox in the GFO.

instantiate the Ship category. The presentials exemplified by S_2 are collections of planks. They instantiate the Ship category only at t_1 and t_5 (the presentials s_1 and s_9).

In addition there are two processes, p_1 and p_2. p_1 has as its only participants at t_1, t_2, t_3, t_4 and t_5 the presentials s_1, s_2, s_4, s_6 and s_8 respectively, while the process p_2 has as its only participants at these time boundaries s_1, s_3, s_5, s_7 and s_9.

For the perpetuant S_2, the mereological theory of identity holds: the identity of the object depends on the identity of its parts. Perpetuant S_1 employs no such principle; although its parts change continuously, and due to this fact the presentials that are exemplified by S_1 change properties, all presentials exemplified by S_1 instantiate the Ship category.

The GFO explicitly includes identity criteria for objects. It used a combination of perpetuants and processes that connect all presentials that are exemplified by a perpetuant. This permits consistently modelling multiple views on an entity's identity within the same knowledge base.

2.2.3 Basic Formal Ontology

The Basic Formal Ontology (BFO) [Grenon, 2003a] (shown in figure 2.2.3) contains categories that always have as instances individuals. These categories are provided as a taxonomy with textual definitions. The BFO's primary distinction is between occurrents and continuants. Occurrents are entities that unfold in time and have temporal parts, while continuants are entities that are wholly present at each point in time at which they exist and persist through time.

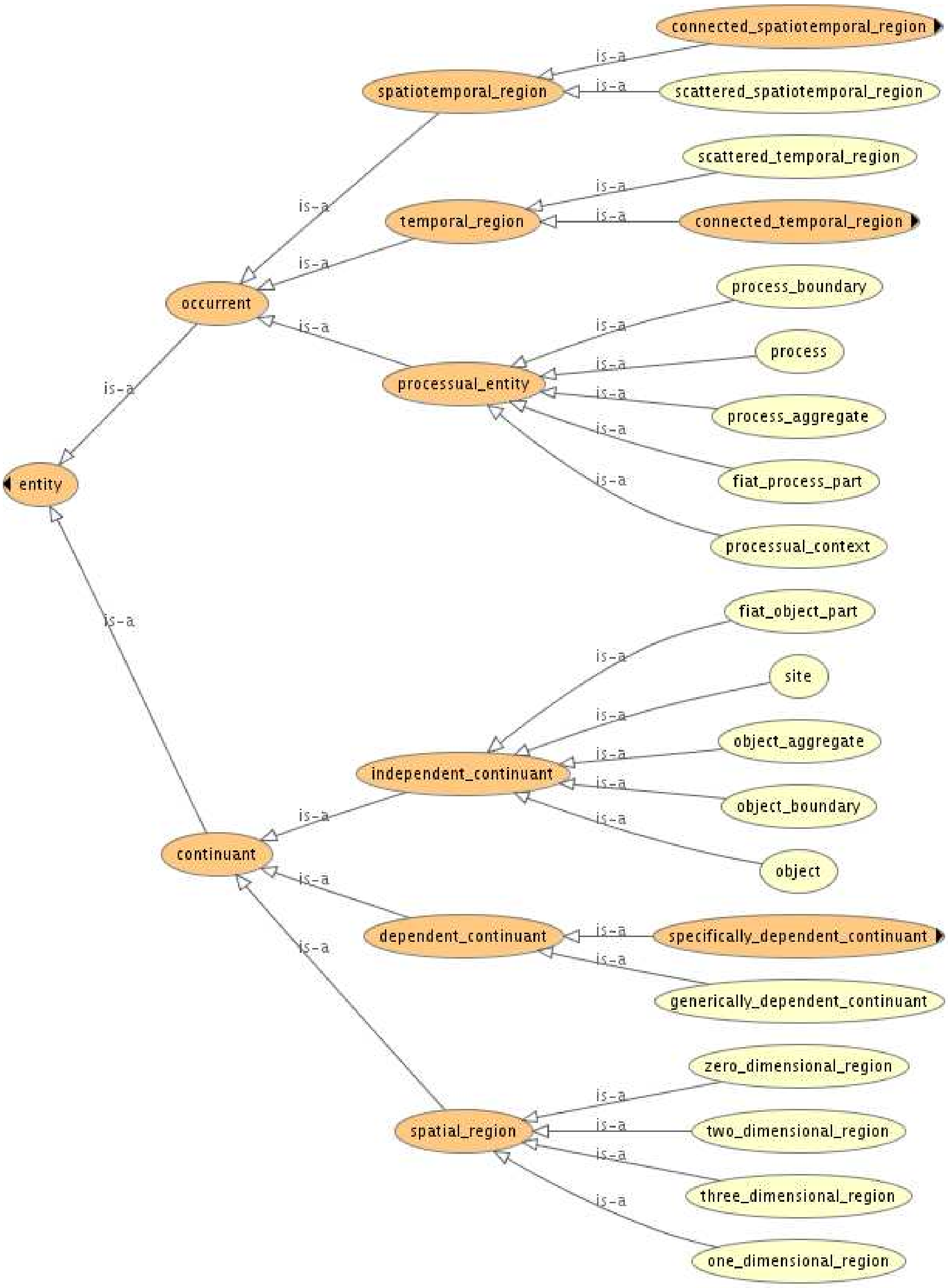


Figure 2.8: The taxonomic tree of the BFO.

Occurrents are further divided into processes, fiat process parts, process aggregates, process boundaries and processual contexts. Processes are spatiotemporally connected occurrents that have clearly delineated beginnings and endings. Fiat process parts are parts of processes that have no such *bona fide* beginnings and endings. Process aggregates are mereological sums of processes. Process boundaries are instantanuous boundaries of processes and the only kind of occurrents in the BFO that have no temporal duration.

Continuants are subdivided into dependent and independent continuants. Dependent continuants are existentially dependent [Correia, 2005] on another entity, while independent continuants are not. Independent continuants include objects, object aggregates, object boundaries, fiat object parts and sites. Objects are spatially extended and connected entities that possess internal unity and can be delineated from their surroundings. Fiat object parts are parts of objects that do not show physical discontinuities from their surroundings. Object aggregates are mereological sums of objects, object boundaries constitute the boundary of objects.

Dependent continuants include realizable entities and qualities. A quality **inheres in** some continuant entity. Realizable entities are either dispositions, functions or roles. A realization of a realizable entity is always a process.

2.3 Biomedical domain ontologies

2.3.1 Gene Ontology

The Gene Ontology Consortium designed the Gene Ontology (GO) [Ashburner et al., 2000] to address the problem of integrating data between the model organism databases. Initially, the fly [Flybase, 1999], yeast [Cherry et al., 1998] and mouse [Bult et al., 2008] genome databases participated in GO's construction and used the GO for annotating their data. Nowadays, most major genome and protein databases use the GO for annotating data.

At the time of GO's creation, the biological databases used different, non-standardized terminology to describe the features of a gene or a gene product [Bada et al., 2004]. Due to the large number of homologous[5] genes in different organisms, the gene products in different organisms share similar or identical functions, participate in the same kinds of processes and occur in the same parts of cells.

The GO consists of three ontologies that describe the biological processes in which a gene product or group of gene products may participate, the molecular functions it may have and the cellular components in which it may be active. These ontologies contain a set of categories and relations between them. Originally, two relations were used in the GO: **is-a** and **part-of**. Later, they were extended by a group of **regulates** relations.

The GO is used to annotate gene products or groups of gene products. The annotation of a gene product to a process, function or component category essentially means that the gene product can participate in the kind of processes, has the kind of function and can be located in the kind of cell components to which it has been annotated.

Molecular function

The GO website[6] describes the Molecular Function (MF) category as:

[5] Homology refers to a similarity due to a common evolutionary history.
[6] `http://www.geneontology.org/`

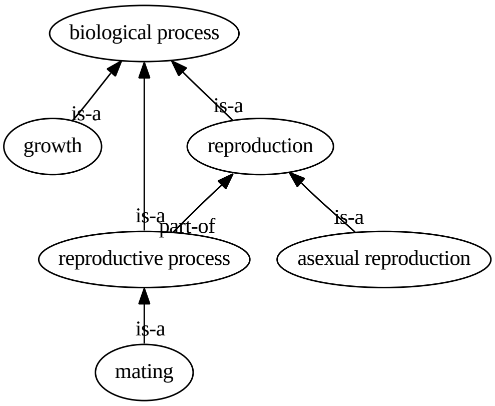


Figure 2.9: The top-level of the GO's Biological Process ontology.

> The functions of a gene product are the jobs that it does or the "abilities" that it has. These may include transporting things around, binding to things, holding things together and changing one thing into another. This is different from the biological processes the gene product is involved in, which involve more than one activity.

Therefore, molecular functions are often basic, "single-step" processes that cannot be further divided into sub-processes. As a consequence, the molecular function ontology uses only the **is-a** relation, but not the **part-of** relation.

Biological process

The Biological Process (BP) ontology (figure 2.3.1) of the GO classifies processes. A process is understood as a sequence of events or molecular functions. Processes are assumed to have a definite beginning and a definite end. Parts of processes can be distinguished. Therefore, the BP ontology uses both the **is-a** and the **part-of** relation.

Biological processes may **regulate** other biological processes, molecular functions or biological qualities. According to the GO, a category of biological processes *P* **regulates** another category of processes *R* iff every instance of *P* modulates the occurrence of instances of *R*. *P* **regulates** a category of properties *S* iff every instance of *P* modifies the *values* of some instances of *S*. As a result of this definition, two sub-relations of **regulates** are used: **positively regulates** and **negatively regulates**.

Cellular component

The Cellular Component (CC) ontology of the GO contains categories pertaining to parts of cells, including encapsulating structures external to a cell such as cell walls. On the lower end of the granularity scale, it contains complexes of gene products as components, but not individual gene products. Its purpose is to describe the locations at which gene products are active.

2.3.2 Anatomy and Development

The Common Anatomy Reference Ontology (CARO) [Haendel et al., 2007] is a species-independent ontology for the anatomy domain. It is based on the most general categories of the Foundational Model of Anatomy (FMA) [Rosse and Mejino, 2003], and is intended to be used as a common top-level for all biological anatomy ontologies and a template for the development of new anatomy ontologies.

The CARO's basic taxonomic structure is shown in figure 2.3.2. The top-level entity is *Anatomical entity*, with *Material anatomical entity* and *Immaterial*

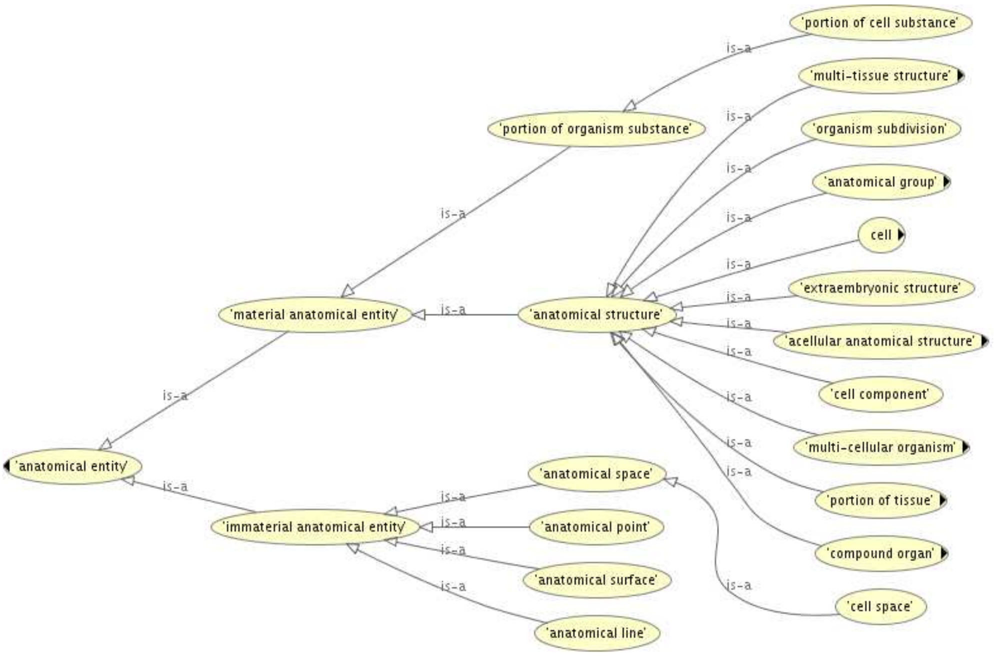


Figure 2.10: Top-level of the Common Anatomy Reference Ontology (CARO).

anatomical entity as sub-categories. Anatomical entities are either whole organisms or entities that structurally organize an organism. Immaterial anatomical entities have no mass. Examples for these are cavities or locations. Material anatomical entities are either anatomical structures or body substances. An example for a body substance is *Urine*. Among material anatomical entities are *Organ*s, *Cell*s, *Tissue*s or *Cell component*s. The anatomical entities in the CARO are related by **is-a** and **part-of** relations.

Domain-specific anatomy ontologies are embedded in the CARO by declaring their top-level categories to be sub-categories of CARO categories. Alternatively, more complex definitions or restrictions can be given for a domain anatomy ontology's top-level categories, and using the categories of the CARO. By providing a top-level structure for anatomical entities, the CARO can also serve as a template for the development of new anatomy ontologies.

Organism development is often included in anatomy ontologies such as the Plant Ontolology [The Plant Ontology Consortium, 2002]. In the description of an organism's development, the life cycle of an organism is divided into stages. The life of the organism is considered to be a process, and the stages are part of this process. Relations between developmental processes include **part-of** and temporal ordering relations among development stages.

A developmental anatomy combines development stages and anatomical parts present at these stages. Some anatomical entities exist only during some development stages, and change into other anatomical entities during continued development. These are related using the **develops-from** relation. Anatomical parts can **participate in** some development stages.

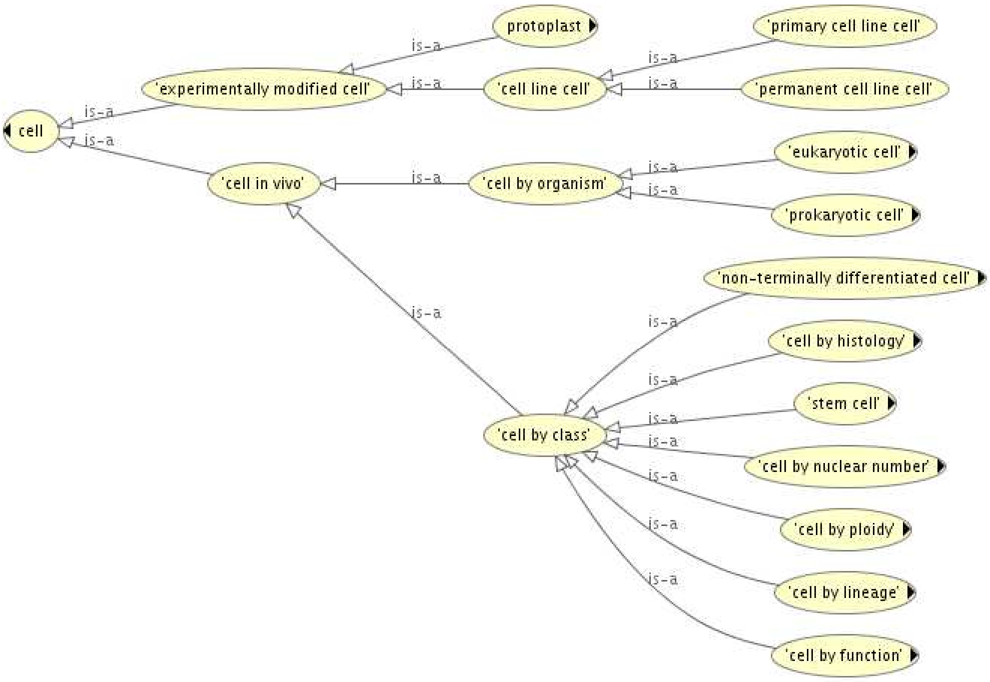


Figure 2.11: Top-level of the Celltype Ontology.

2.3.3 Classifications and Taxonomies

Celltype

The Celltype Ontology [Bard et al., 2005] provides a classification of celltypes, starting with the top-level class *Cell*. Cells are subdivided into cells occuring naturally in organisms (*Cell in vivo*) and *experimentally modified cell*s. Experimentally modified cells are either cells in a cell line or protoplasts[7].

[7] A protoplast is a cell after removing its cell wall.

The *Cell in vivo* category employs several structuring axes. One axis distinguishs cells by the kinds of organisms in which they occur, either prokaryotic or eukaryotic. Other axes include the function of a cell, the cell's histology, the number of nuclei in the cell, the cell's ploidy and the cell's lineage.

The top-level categories of the Celltype Ontology are illustrated in figure 2.3.3. Within the Celltype Ontology, the only kind of entity considered are cells. The various axes used to distinguish among kinds of cells are not explicitly defined, and the entities that are used in these axes not explicitly defined either. Some of these, such as cell functions, currently cannot be found in other biomedical ontologies, while some properties such as ploidy are included in other ontologies.

The structuring relations in the Celltype Ontology are **is-a** and **develops-from**. The **develops-from** relation is a relation between two types of cells where instances of one cell type always develop out of instances of the other cell type.

Organism Taxonomy

Multiple organism taxonomies are available, the largest being the NCBI taxonomy [Wheeler et al., 2004].The major difficulty in representing organismal taxonomy is representing the relations between categories on different categorization levels or *rank*s. Ranks are categories such as *Species*, *Genus*, *Family* or *Kingdom*. Several options for representing the classification of organism types in ontologies were proposed [Schulz et al., 2008].

The Teleost Taxonomy Ontology (TTO) uses a relation **has-rank** to relate *categories* directly to their rank. This relation asserts properties to categories, and is not reduced to a relation between individuals. Biological taxa are related by the **is-a** relation, and the different levels of these taxa asserted using the **has-rank** relation.

For example, gentoo penguins (species) belong to the family *Spheniscidae*, the class *Aves* in the kingdom *Animalia*. According to the schema used in the TTO, these relations are represented as follows:

$$\begin{aligned}
&isA(GentooPenguin, Spheniscidae)\\
&isA(Spheniscidae, Aves)\\
&isA(Aves, Animalia)\\
&hasRank(GentooPenguin, Species)\\
&hasRank(Spheniscidae, Family)\\
&hasRank(Aves, Class)\\
&hasRank(Animalia, Kingdom)
\end{aligned}$$

In addition, the taxonomic ranks are not part of a taxonomy ontology itself, but maintained separate (in a TaxonRank Ontology[8]). The relation between taxonomic ranks is the **rank order** relation. The relation **rank order** is transitive and antisymmetric.

2.3.4 Qualities, Properties and Phenotypes

There are two approaches to representing phenotypes. The first is implemented in the PATO ontology, which is an ontology of phenotypic qualities. PATO is an ontology of properties organized in a taxonomy. The main axes used in this classification are whether or not the properties **inhere in** an object or

[8] `https://www.nescent.org/phenoscape/Taxonomic_Ranks`

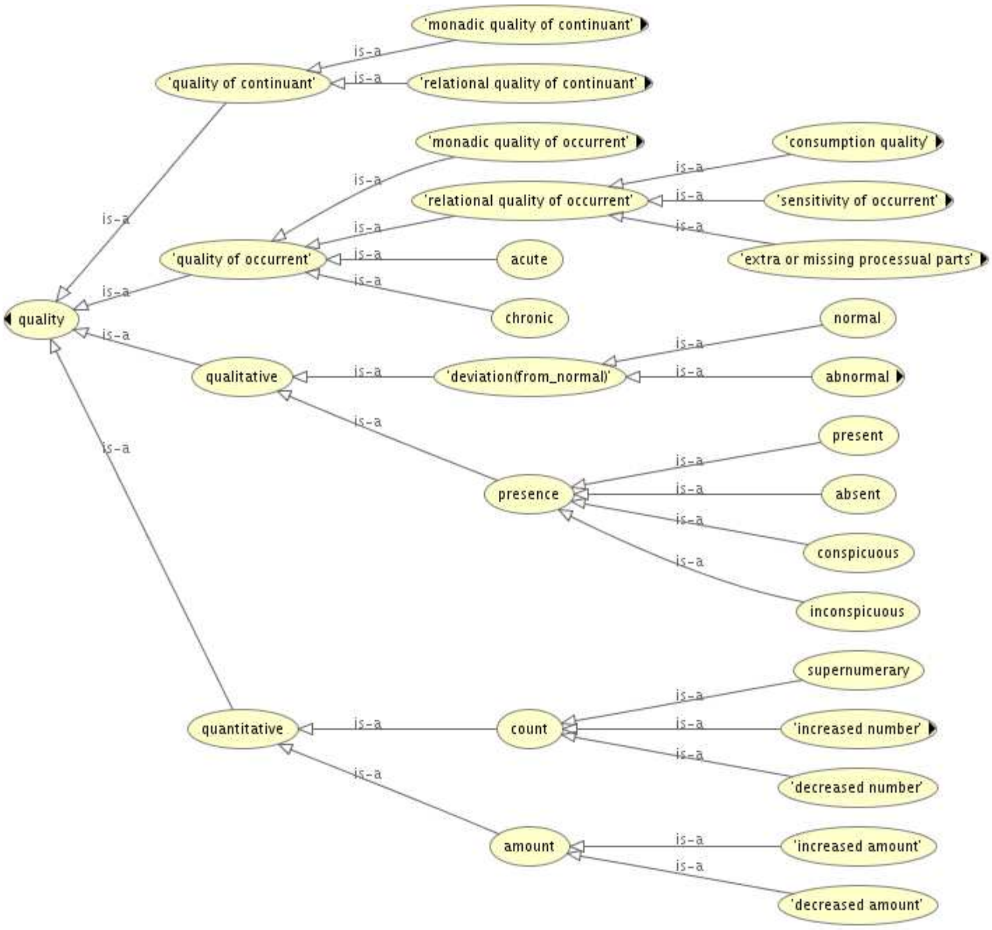


Figure 2.12: Top-level of the PATO ontology.

process, whether the quality is relational or monadic and whether the quality has quantitive values or qualitative ones.

Monadic qualities inhere in exactly one entity, while relational qualities inhere in multiple entities at the same time. An example of a monadic quality is *Color*, while a relational quality is *Flavor* which inheres in some entity and must be perceived by another.

PATO combines qualities with their values. Qualities such as *Color* and values such as *Red* are both included in the same taxonomy, and *Red* is a sub-category of *Color*. To distinguish between qualities and values of qualities, annotation properties in OWL or the OBO Flatfile Format are used.

The Cereal Plant Trait Ontology (TO) [The Plant Ontology Consortium, 2002] (figure 2.3.4) uses a different method to represent qualities. It contains a classification of traits or properties and rarely includes the property's values[9]. The TO also includes complex properties, i.e., properties that are **part of** other properties. For example, *Grain thickness* is a **part of** *Grain size* in the TO.

The Mammalian Phenotype Ontology (MP) [Smith et al., 2005b] (figure 2.3.4) is used for the description of mutant phenotypes of mice. It primarily contains categories for the description of abnormal phenotypes. These are described by reference to an anatomy ontology.

The categories of the MP are derived from reified relations. The instances of the MP categories are entities exhibiting a property or standing in relation to another entity. An example is the *Absent tail* category, which describes a mouse without a tail. Every instance of *Absent tail* has no instance of *Tail* as part.

As a corollary, three kinds of representing phenotypes can be distinguished: the first combines properties and their values in a single taxonomy and uses

[9] Examples of values that are included in the TO are *embryoless* and *vivipary*.

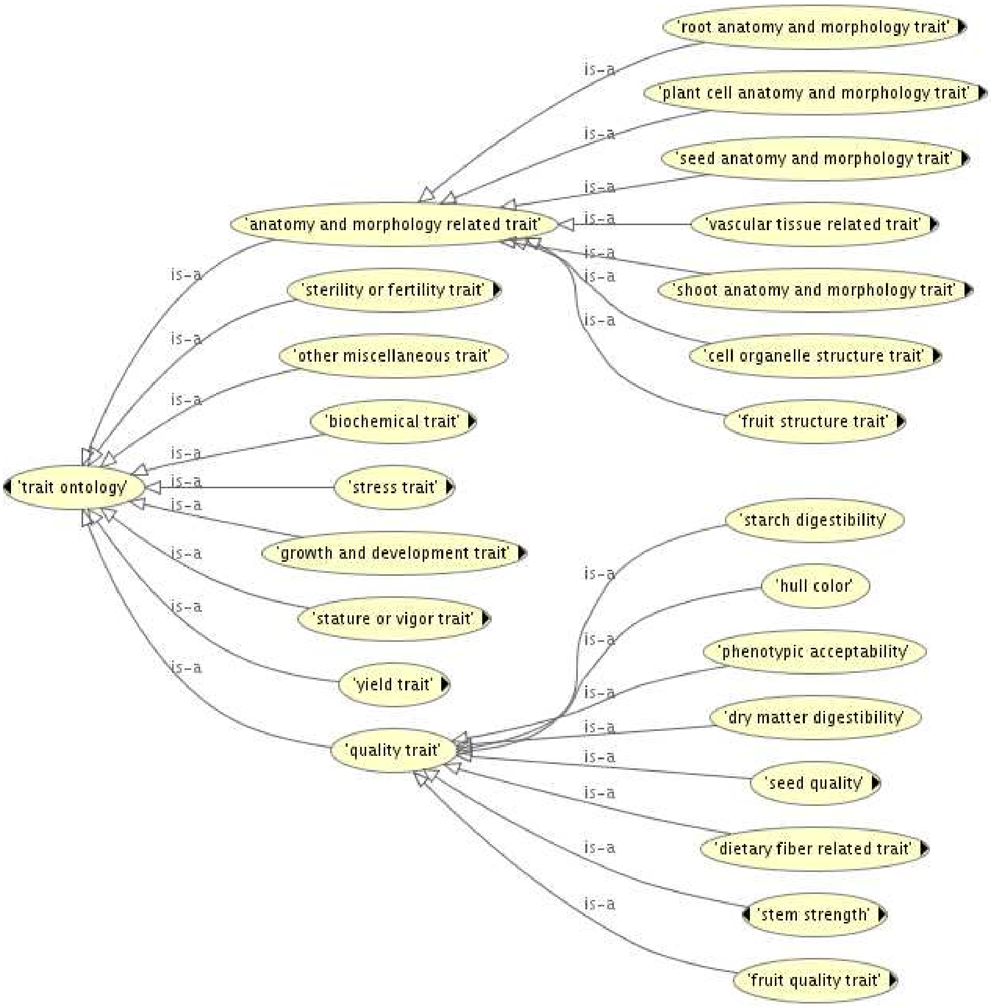

'trait ontology'
'anatomy and morphology related trait'
'sterility or fertility trait'
'other miscellaneous trait'
'biochemical trait'
'stress trait'
'growth and development trait'
'stature or vigor trait'
'yield trait'
'quality trait'
'root anatomy and morphology trait'
'plant cell anatomy and morphology trait'
'seed anatomy and morphology trait'
'vascular tissue related trait'
'shoot anatomy and morphology trait'
'cell organelle structure trait'
'fruit structure trait'
'starch digestibility'
'hull color'
'phenotypic acceptability'
'dry matter digestibility'
'seed quality'
'dietary fiber related trait'
'stem strength'
'fruit quality trait'
is-a

Figure 2.13: Top-level of the Cereal Plant Trait ontology.

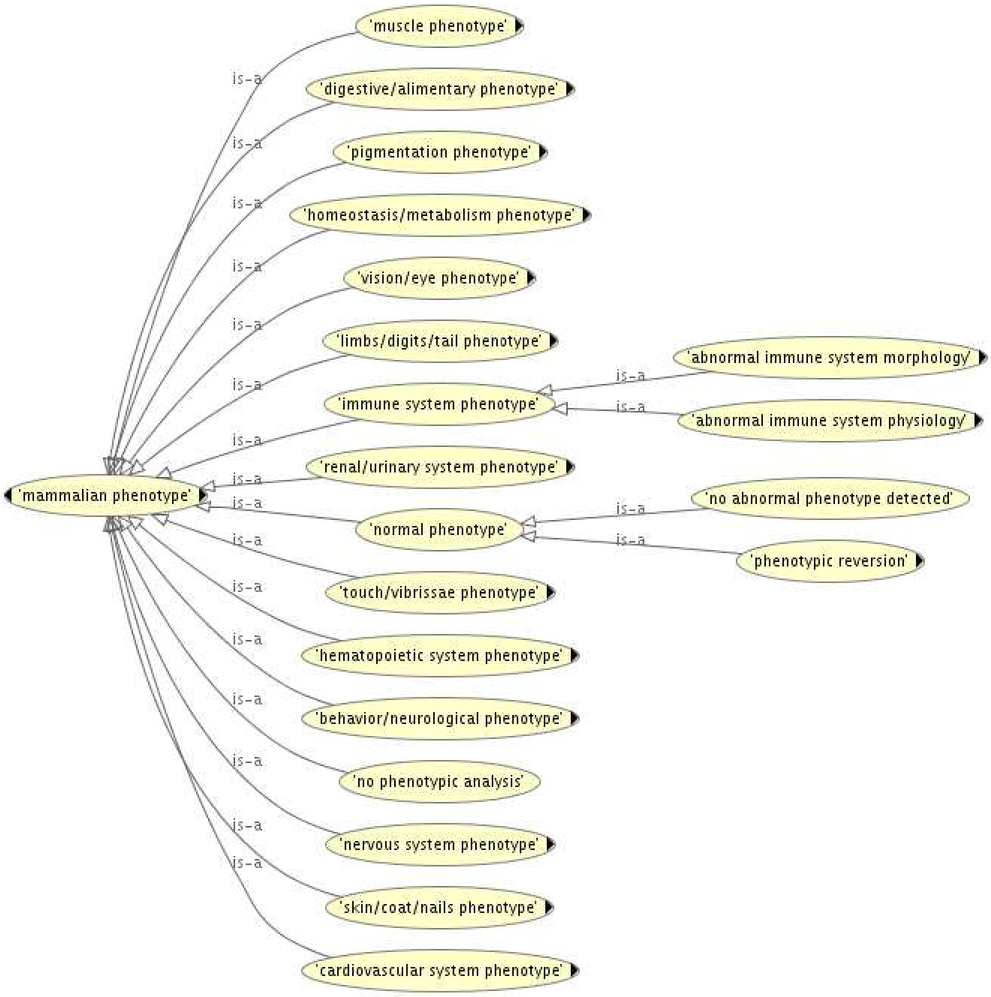

'muscle phenotype'
'digestive/alimentary phenotype'
'pigmentation phenotype'
'homeostasis/metabolism phenotype'
'vision/eye phenotype'
'limbs/digits/tail phenotype'
'abnormal immune system morphology'
'immune system phenotype'
'abnormal immune system physiology'
'renal/urinary system phenotype'
'mammalian phenotype'
'no abnormal phenotype detected'
'normal phenotype'
'phenotypic reversion'
'touch/vibrissae phenotype'
'hematopoietic system phenotype'
'behavior/neurological phenotype'
'no phenotypic analysis'
'nervous system phenotype'
'skin/coat/nails phenotype'
'cardiovascular system phenotype'
is-a

Figure 2.14: Top-level of the Mammalian Phenotype ontology.

the inherence relation to relate them to their bearers; the second classifies traits without their values, and describes the constitution of these traits by means of its parts, which are related to their bearers by the inherence relation; and finally, reified relations are used to form categories that are predicated of bearers of qualities.

2.3.5 Experiments

The Ontology of Biomedical Investigations (OBI) and the Ontology of Scientific Experiments (EXPO) are specifications of the domain of scientific experiments. The EXPO is based on the Suggested Upper Merged Ontology (SUMO) [Niles and Pease, 2001], while the OBI is based on the Basic Formal Ontology (BFO) [Grenon, 2003b].

The EXPO extends the SUMO by categories pertaining to actions, plans or hypotheses that are used within a scientific investigation. It is applied for describing experiments performed by the Robot Scientist [Soldatova et al., 2006]. The EXPO defines several sub-relations of the **has-part** and **has-attribute** relations and uses them to define its classes. The EXPO is developed in the OWL-DL language. It is not widely used and contains several logical inconsistencies[10].

The OBI is developed as a large collaborative project. It aims to provide a vocabulary for the description of all kinds of biomedical and clinical experiments and investigations. The OBI is developed in OWL-DL. Partial logical definitions are provided for some categories, but the majority is defined using natural language.

[10] The inconsistencies are present in version 2 of the EXPO ontology, last accessed on Dec 2, 2008 from `http://sourceforge.net/projects/expo`.

The axes of classification in the OBI are determined by its top-level ontology BFO. Therefore, the primary distinction is made between occurrents and continuants. The occurrent categories in the OBI are a number of process categories that occur as part of experiments. They include experimental actions such as the administration of substances into something or the immobilization of an entity. Other kinds of processes are data tranformations, interpretations of data, documenting or planning of processes.

One distinction between processes is between planned processes, objective-driven processes and spontaneous processes. An objective-driven process is initiated by an agent with a desired outcome, a goal which is to be achieved by the process. This does not entail a plan. A realization of a plan is a planned process. Spontanuous processes are processes which are not initiated by an investigator but which are external to the investigation.

Continuants are divided into dependent and independent continuants. Independent continuants in the OBI are material objects and include categories from other domain-specific ontologies. For example, anatomical structures and parts, chemical entities, organisms and cells and cell components are sub-categories of material entities in the OBI. The OBI includes a novel classification of instruments or other devices that participate in scientific investigations.

Dependent continuants are existentially dependent on another entity. They include qualities, realizable entities and information artifacts. The qualities included in the PATO ontology are used in the OBI. Realizable entities are divided into dispositions, functions and roles. The OBI uses only functions and roles.

Functions in the OBI are specifications of goals. Examples include *Connection function* or *Cool function* with the goal of connecting or cooling, respectively.

The functions included in the OBI depend on the intentions of the investigating agent, i.e., no *intrinsic* functions are included.

Roles are used to distinguish *how* entities participate in some context. A context is either a social context or a process. Example roles are *Drug* or the *Patient* role.

Information artifact is a sub-category of generically dependent continuants. Information artifacts are divided into realizable and non-realizable entities. Non-realizable information artifacts are either digital entities or information content entities. Digital entities are collections of bits that can be represented in multiple physical representations. An example is a specific implementation of an algorithm that is physically present in multiple files. An information content entity is a piece of data that can be represented digitally in multiple ways. An *Image* is an example of an information content entity.

2.3.6 Chemical substances

The Chemical Entities of Biological Interest (ChEBI) [Degtyarenko et al., 2007] ontology contains a classification of chemicals. Chemicals are material structures that have parts. The relations used in the ChEBI ontology are **part-of**, **has-functional-parent** and a number of chemistry-specific relations such as **is-enantiomer-of**.

2.3.7 Sequences

The Sequence Ontology (SO) [Eilbeck et al., 2005b] is an ontology of sequences and sequence features. The SO distinguishes between kinds of sequences, qualities of sequences, operations on sequences and sequence variants. A sequence feature is an extended or non-extended biological sequence. Extended sequences are genes, intergenic regions or sequences of polypeptides. Non-extended sequences are junctions, boundaries between two extended sequences.

Qualities of sequences include whether or not a sequence encodes a protein, whether a sequence acts enzymatically when transcribed, or whether the sequence is conserved. Properties that exist by virtue of a scientific investigation are included as well, e.g. *Validated* or *Invalidated* features of sequences.

2.3.8 Relations

Several relations are used in multiple biomedical domain ontologies. The OBO Relationship Ontology (RO) [Smith et al., 2005a] aims to provide common definitions for these relationships to ease interoperability between the domain ontologies using these relations. For this purpose, the RO provides a basic classification of entities into *Individuals* and *Categories* and further into categories of *Continuants* and *Processes*. Furthermore, it introduces basic time entities together with a linear order between time points.

The RO defines the relations that are asserted to hold directly between categories by using relations that are defined to hold between individuals. The common definition pattern[11] for a relation $R(C,D)$ between categories C and D is

$$\begin{aligned} R(C,D) \iff \forall c,t(instanceOf(c,C,t) \rightarrow \\ \exists d(instanceOf(d,D,t) \wedge R^i(c,d,t))) \end{aligned} \tag{2.1}$$

[11] This patterns is employed for the majority of relations in the RO, but not all. For example, the **transformation-of** relation is defined in terms of identity and instantiation, instead.

where the relation R^i is the counterpart of the relation R but holds between individuals. This pattern is used for relations between continuants. When categories of processes participate in the relation, the time parameters are chosen differently or omitted.

The definitions of the relations are given in first order logic. Basic axioms are included. The axioms that are given for the primitive relations between individuals pertain to either transitivity, symmetry or reflexivity.

2.4 Upper Domain Ontologies

The majority of biomedical ontologies are domain specific, covering domains as diverse as organism development, anatomy [Henrich et al., 2005], cell types [Bard et al., 2005], processes, functions [Ashburner et al., 2000], roles, pathways [Yamamoto et al., 2004], species [Phan et al., 2003], phenotypes [Smith et al., 2005b], among others.By contrast, little attention has been given to the development of upper domain and core ontologies for biology. An upper domain ontology or *core ontology* is an ontology that formally describes and defines the basic categories within a domain [Valente and Breuker, 1996]. Because a core ontology's categories are so general, they are similar to the categories found in foundational or top-level ontologies. A foundational ontology contains categories covering all domains of reality [Sowa, 2000, Herre et al., 2006].

One function of a core ontology is to specialize the concepts and relations of a foundational ontology to those concepts that exist in a domain. It then acts as an intermediate layer between a top-level ontology such as the GFO and domain ontologies that use the top-level ontology. Core ontologies can be used to ease the integration of domain ontologies under the top-level ontology by providing additional, domain-specific concepts, or by adding an additional layer of restrictions that are valid within one domain, but not another.

A small number of biomedical core ontologies have been developed [Schulz et al., 2006a, Rector et al., 2006b] and they are subject to different stengths and weaknesses. These weaknesses and strengths partially arise from the top-level ontology that these core ontologies use. I discuss only one such ontology, the BioTop ontology [Schulz et al., 2006a].

2.4.1 BioTOP

The BioTop Ontology [Schulz et al., 2006a] started as a further development of the GENIA upper ontology [Kim et al., 2003]. The GENIA upper ontology is intended for use in semantic annotation of texts in biological text mining. Several problems with GENIA's upper ontology have been identified [Schulz et al., 2006a], mainly related to a lack of formalization of the categories used in GENIA.

BioTop is an upper domain ontology for biology based on the top-level ontology BFO [Grenon, 2003b] and DOLCE [Masolo et al., 2003]. The relations used in BioTop are those used in the OBO Relationship Ontology, plus some additional relations pertaining to a distinction between collections and collectives [Rector et al., 2006a], like **has-grain** or **has-constituent**.

BioTop is, like GENIA's upper ontology, mainly an ontology of continuants: entities that are wholly present at each point in time at which they exist, and may preserve their identity through time. Axioms are given in OWL-DL for upper categories used in biomedical domain ontologies. For example, the category *Cell* is defined as having some *Cytoplasm* and no *Cell* as proper part, and having some *Cellular component* and some *Membrane* as component.

BioTop is considered to be applied as an upper level ontology for all ontologies listed under the OBO umbrella. By providing definitions for upper categories of these ontologies, it enforces ontological rigor and attempts to eliminate ambiguities in the use of categories. For example, when two ontologies include a *Cell* category, and both use BioTop for defining this *Cell* category, interoperability between both ontologies is made simpler.

However, several restrictions on BioTop's application remain, some introduced through the use of the BFO as top-level ontology. BFO does not include a means to model categories of higher order (i.e., categories that have categories as their instances) or to represent abstract entities like *Information* or a *Sequence of symbols*. Both of these are relevant in the biomedical domain: *Species* can be considered a category of higher order, which has types of organisms (like *Mus musculus*) as instances; in bioinformatics, many analyses and algorithms operate on abstract sequences. However, an ontological analysis of the representation of *Species* in BioTop has been performed [Schulz et al., 2008]. Several approaches were considered: *Species* as a category of higher order, with organism categories as instances; regarding *Species* as a super-category (via **is-a**) of organim categories; *Species* as collectives of organisms; *Species* as properties and *Species* represented as qualia [Masolo et al., 2003].

BioTop also contains a *Biomolecular sequence* and *Biomolecular sequence information* category. While the biomolecular sequence is seen as a concrete individual (a molecule), the sequence information is a kind of generically dependent continuant[12] which is dependent on a sequence (the molecule). It is therefore difficult to represent sequences which are not sequences of some molecule in BioTop, i.e., sequences as entities in their own right.

[12] *A* is generically dependent on *B* if, whenever some instance *x* of *A* exists, necessarily, there exists some instance *y* of *B*.

2.5 Ontological methods and principles in biomedicine

2.5.1 The annotation relation

Biomedical ontologies in the OBO are primarily used to annotate biological entities across multiple databases. The annotation relation establishs an association relation between a category from an ontology and a piece of data stored in a database. The data stored in databases such as the model organism databases or the UniProtKB, often refer to categories of genes or proteins in the sense that they do not denote individuals but classes or categories of proteins or genes.

Annotation is not a well-defined relation, but commonly establishes an association between two categories. However, this association is not arbitrary. The annotation of a protein category with a biological process category means that instances of the protein category can participate in instances of the process category. Annotation with a function category usually means that instances of the protein category have the function to which the protein category is annotated.

The annotation relation is accompanied with meta-information about how the particular instance of the annotation relation has been identified, where further information can be found, where the analysis has been published and which methods were used to identify the association.

Evidence codes were first used in the annotation of gene products with categories from the GO. They represent the *justifications* for including particular annotations of gene products with ontological categories. Evidence codes group justifications into experimental findings, findings through computational analyses, publicly stated facts from authors, inferences made by curators and automatically assigned annotations. The evidence codes provide different levels and measures of confidence in an annotation, and can be used to identify high-confidence annotations for inclusion in further analyses.

For analyses, the GO and other biological ontologies employ the True Path Rule as their only semantic rule pertaining to annotations. The True Path Rule states that an annotation is transitive over the **is-a** and **part-of** relations: if P is annotated with C and C **is-a** D or C **part-of** D, then P is annotated to D. The True-Path-Rule can be employed to support functional analyses of gene expressions or other features of annotated gene products [Prufer et al., 2007].

2.5.2 OBO and OBO Foundry criteria

To support interoperability between ontologies, the Open Biomedical Ontologies (OBO) [Smith et al., 2007] specifies a number of criteria that ontologies included in the OBO must satisfy. Most of the criteria are social criteria: openness and free accessibility of the ontologies, clearly delineated content and orthogonality of all ontologies included in the OBO, a heterogenuous userbase and collaborative development. Other criteria are technical criteria such as the use of a common syntax for the representation of the ontologies (either the OBO Flatfile Format or OWL), inclusion of definitions for each term, methods for identifying versions of the ontology, the use of a unique identifier within the OBO ontologies and the use of the OBO Relationship Ontology (RO) [Smith et al., 2005a].

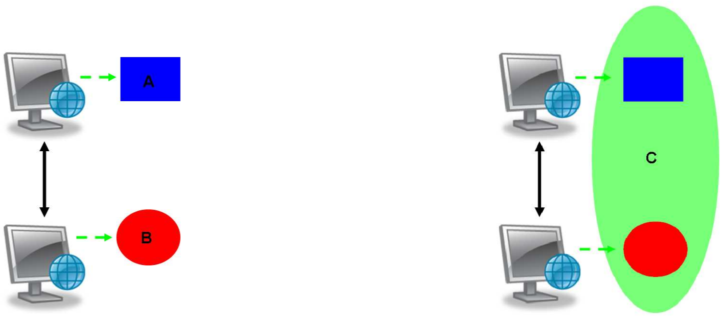
Only the use of the RO employs semantic and ontological criteria while the other criteria remain technical and primarily social. As such, they provide the foundations for interoperability. Without these criteria, it would be difficult to gain access to the ontologies and analyze, modify or use the ontologies. Without the requirement for different identifiers, it would be difficult to identify categories within the OBO because identifiers may overlap. The collaborative development and orthogonality criteria establish a basic consensus within each domain for which an ontology is developed and permits the combination of ontologies, as there should be no overlapping content between the included ontologies. It is an open question whether these criteria suffice to achieve interoperability between the biomedical ontologies that are included in the OBO or OBO Foundry, or if additional criteria must be employed.

3 The Issue of Interoperability between Ontology-Based Information Systems in Biology

> The knowable world is
> incomplete if seen from any one
> point of view, incoherent if seen
> from all points of view at once,
> and empty if seen from nowhere
> in particular.
>
> Richard Shweder

3.1 What is Interoperability?

Ontologies have been proposed as a solution to the problem of interoperability between information systems [Bodenreider, 2008, Noy, 2004]. The assumption is that two information systems that share the same ontology – and therefore the same conceptualization of parts of reality – will be able to interoperate (see figure 3.1). But "merely using ontologies [...] does not reduce heterogeneity: it just raises heterogeneity problems to a higher level" [Euzenat and Shvaiko,



(a) (b)

Figure 3.1: On the left-hand side of the figure, two information systems based on ontologies *A* and *B* are illustrated together with a communication channel between them. On the right-hand side of the figure, ontologies *A* and *B* are integrated into a new ontology *C* that is used by both information systems.

2007]. With the development of more and more ontologies, the difficulty of achieving interoperability between the ontologies themselves has increased.

The IEEE's definition of interoperability [Geraci, 1991] is

> the ability of two or more systems or components to exchange information and to use the information that has been exchanged.

This leaves a wide range of requirements for interoperability. To exchange information, a physical connection must be present between the two systems. Signals must be encoded in a certain way understandable to both systems, and signals of a certain type must have an interpretation shared by both systems. Ultimately, it can mean that it should be possible to consistently combine the conceptual schemata (conceptualizations) of both systems to obtain the highest degree of interoperability.

One approach of formalizing interoperability between information systems was

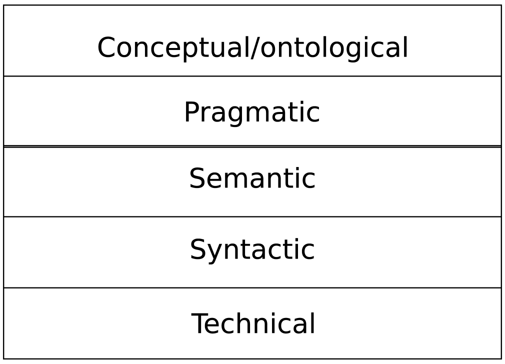


Figure 3.2: An overview of the Levels of Conceptual Interoperability Model.

developed as a layered model of interoperability. The Levels of Conceptual Interoperability Model (LCIM) [Tolk and Muguira, 2003, Dobrev et al., 2007] provides a layered approach to understanding interoperability between systems, as illustrated in figure 3.1.

- Level 1: Technical interoperability is based on a physical connection between systems. This connection is used as a communication infrastructure. The necessary network protocols are defined. An example can be TCP/IP over Ethernet.
- Level 2: The level of syntactic interoperability provides a common structure for exchanging data, such as a common data format or application programming interface. An example is XML.
- Level 3: Semantic interoperability presupposes a common information exchange reference model, a common method for describing the meaning of data. An example is OWL-DL, RDFS or Common Logic.
- Level 4: Pragmatic interoperability means that two systems are aware of how data is used and how data is processes in the other system. An example may be two ontology-based information systems that formally

share the same meaning for the category *Cell*, but one uses it only in the context of eucaryotic cells.

- Level 5: Conceptual interoperability is reached when two systems share the same conceptual schema, i.e., the specification of the abstractions from reality used in the software. This means that they assume the same ontological commitment of the information they process.

Interoperability between ontology-based information systems can be understood in – at least – these five ways. In the first level, data is transfered, usually in the form of bits, and a physical signal must be transformed to the representation of bits. The Internet provides this communication infrastructure for most ontology-based information systems.

In the second level, syntactic constructs must be recognized. This requires a syntax and a parser for that syntax. There are several languages in use by ontology-based information systems in biology. The most prominent are the XML, functional and Manchester syntax of OWL, the XML and N3 syntax of RDF, the OBO Flatfile Format, comma- or tab-separated value files and CycL. Translation between these languages on a syntactic level is not straight-forward. Syntactic translations are defined between the different OWL and RDF formats, as well as between the OBO Flatfile Format and OWL.

Third, the semantics of syntactic constructs must be recognized and represented adequately within each of the ontology-based information systems. The semantics of ontology representation languages is usually defined as a model-theoretic semantics. For example, a model-theoretic semantics for OWL and RDF is given, and through the syntactic mapping between the OBO Flatfile Format and OWL, also a model-theoretic semantics for OBO. In model theory, the semantics of a language is defined by recursively mapping syntactic structures to elements of a pre-defined mathematical domain. In first order logics, for example, terms (function, variable and constant symbols) are mapped to elements from a universe, and propositions to *true* or *false*.

Semantic interoperability must be distinguished further. Given a model structure $\mathcal{A}$ and formula F and a translation function tr, several options arise how interoperability can be understood. In the first scenario, the function tr translates both $\mathcal{A}$ and F such that:

$$\mathcal{A} \models F \iff tr(\mathcal{A}) \models tr(F) \tag{3.1}$$

A weaker form of interoperability that can be established using the function tr is based on equisatisfiability: if there is a structure $\mathcal{A}$ and an interpretation μ which satisfies the formula F, then there exists a structure $\mathcal{B}$ and interpretation μ' which satisfies $tr(F)$. I will introduce a precise formulation of semantic interoperability between two information systems later using the notion of an *infomorphism*.

At level four in the LCIM, the pragmatic interoperability level, the application and use of an ontological entity within an information system is considered. For example, the Foundational Model of Anatomy (FMA), Gene Ontology's (GO's) cellular component ontology and the Celltype Ontology include a *Cell* category. Some time ago, these categories were formally indistinguishable, but refered to *human cell*, *eucaryotic cell* and *any cell*, respectively, in the different ontologies, because these were, except for the Celltype Ontology, developed for use within a delimited domain. Information systems based on these ontologies interpreted them accordingly. Transmitting information between information systems based on two of these ontologies may fail when this pragmatic aspect is not taken into consideration.

Finally, conceptual interoperability requires that information systems use aligned conceptual models. A conceptual model or conceptualization is constituted by the fundamental assumptions, distinctions and constraints pertaining to the parts and aspects of reality that govern the structure and behaviour of the information system. A specification of the conceptual model – an ontology – makes the intension of the basic distinctions made by an information system explicit: it specifies how an element of a language, how a basic concept or relation refers to reality.

For the purpose of ontology-based information systems, I will make use of a more formal notion of interoperability that is applicable to the syntactic, semantic, pragmatic and conceptual level of the LCIM. It is more general, because it is based on a general notion of *classification* that can be applied to syntactic structures as well as to elements of reality as modelled by an ontology.

Schorlemmer and Kalfoglou [2003] recognize that "for two systems to be semantically interoperable (or semantically integrated) we need to align and map their respective ontologies such that *the information can flow*". Consequently, they use channel theory [Barwise and Seligman, 1997], a mathematical model of information flow based on situation theory [Devlin, 1991, Barwise, 1988], for describing semantic interoperability between ontologies.

Following Schorlemmer and Kalfoglou [2003], I represent an ontology-based information system *IS* by an abstract logic $\mathcal{L} = (L(\mathcal{L}), M(\mathcal{L}), \models_{\mathcal{L}})$. It consists of a set of types $L(\mathcal{L})$, a set of tokens $M(\mathcal{L})$, and a classification relation $\models_L \subseteq M(\mathcal{L}) \times L(\mathcal{L})$ which assigns tokens to types. In first order logic, $L(\mathcal{L})$ is a language over a signature Σ and $M(\mathcal{L})$ the set of Σ-structures[1]. The abstract logic $\mathcal{L}$ captures the syntax and semantics local to an information system *IS*: the syntactic expressions local to *IS* are the types of $\mathcal{L}$, and the meaning of these expressions is modelled by the way that tokens are classified to types.

[1] Worlds or interpretations.

A theory $T = (L(T), \vdash_T)$ is a set of types $L(T)$, and a relation $\vdash_T \subseteq L(T) \times L(T)$. A pair (Γ, Δ) of subsets of $L(T)$ is called a sequent. If $\Gamma \vdash_T \Delta$, then $\Gamma \vdash_T \Delta$ is called a constraint. A theory T is called regular if for all $\alpha \in L(T)$ and all Γ, Γ', Δ, Δ', $\Sigma \subseteq L(T)$:

1. Identity: $\alpha \vdash_T \alpha$
2. Weakening: If $\Gamma \vdash_T \Delta$, then $\Gamma, \Gamma' \vdash_T \Delta, \Delta'$.
3. Global cut: If $\Gamma, \Sigma_0 \vdash_T \Delta, \Sigma_1$ for each partition (Σ_0, Σ_1) of Σ, then $\Gamma \vdash_T \Delta$.

A local logic $\mathcal{L} = (M(\mathcal{L}), L(\mathcal{L}), \models_{\mathcal{L}}, \vdash_{\mathcal{L}}, N)$ consists of an abstract logic $\mathcal{S} = (M(\mathcal{L}), L(\mathcal{L}), \models_{\mathcal{L}})$, a regular theory $T = (L(\mathcal{L}), \vdash_{\mathcal{L}})$ and a subset $N \subseteq M(\mathcal{L})$ of normal tokens which satisfy all the constraints of T. A token β satisfies the constraint $\Gamma \vdash_{\mathcal{L}} \Delta$, if when β is of all types of Γ, β is of some type of Δ.

An informorphism $f = (f^{\rightarrow}, f^{\leftarrow}) : \mathcal{L}_1 \rightarrow \mathcal{L}_2$ from an abstract logic $\mathcal{L}_1$ to the abstract logic $\mathcal{L}_2$ is a pair of functions $f^{\rightarrow} : L(\mathcal{L}_1) \mapsto L((\mathcal{L}_2)$ and $f^{\leftarrow} : M(\mathcal{L}_2) \mapsto M(\mathcal{L}_1)$ satisfying, for each $a \in L(\mathcal{L}_1)$ and $b \in M(\mathcal{L}_2)$:

$$f^{\leftarrow}(b) \models_{\mathcal{L}_1} \alpha \text{ iff } b \models_{\mathcal{L}_2} f^{\rightarrow}(\alpha) \tag{3.2}$$

An information channel consists of two abstract logics $\mathcal{L}_1$ and $\mathcal{L}_2$ connected through a core logic $\mathcal{C}$ via two infomorphisms f_1 and f_2:

$$f_1^{\rightarrow} : \quad L((L)_1) \mapsto L(\mathcal{C}) \tag{3.3}$$

$$f_1^{\leftarrow} : \quad M((C)) \mapsto M(\mathcal{L}_1) \tag{3.4}$$

$$f_2^{\rightarrow} : \quad (L((L)_2 \mapsto L(\mathcal{C}) \tag{3.5}$$

$$f_2^{\leftarrow} : \quad M((C)) \mapsto M(\mathcal{L}_2) \tag{3.6}$$

This formalism was applied to the problem of ontology alignment in [Schorlemmer and Kalfoglou, 2003]. The notion of an infomorphism presents a formalization of interoperability that is independent of the used logic, or whether the interoperating information systems use any kind of logic-based formalism at all. In order to apply the notion of an infomorphism, it is sufficient to have two information systems that use some kind of classification scheme and apply it to token from some part of reality. Ontology-based information systems use ontological categories as types, and their instances as the tokens. In first order logic, formulas are types and models are the tokens.

I use infomorphisms as a model for interoperability between ontology based information systems in biomedical applications. In the biomedical domain, ontologies are often kept separate. Allowing for information to flow between these applications and between the ontology-based knowledge bases is highly desired for various reasons, which will be discussed in next section.

3.2 What makes interoperability desirable

To unify the description of a gene's features such as the processes in which it is involved, biomedical ontologies like the Gene Ontology (GO) [Ashburner et al., 2000] were developed. These ontologies are single-domain ontologies. They solely describe processes, functions, locations, types of cell, organism-specific anatomy or similar.

This description aims to allow an unambiguous description and analysis of biological data, whenever an information system is able to process and understand the ontology or ontologies used in this description. A simple example of such an information system is the web-based application using a database that lists all the cellular locations in which a given gene is expressed, and uses the GO annotations of genes for this purpose. However, additional information pertaining to the type of cell, e.g. a red blood cell, and the anatomy of specific cell types, e.g. red blood cell's lacking of a nucleus, could be used to make the query semantically richer: genes which are expressed in the spliceosome (a complex within the cell nucleus) should not be included as answer to a query of genes expressed in cells that are known to be red blood cells since red blood cells have no nucleus as part. In order to perform complex queries across multiple databases, information flow between these databases must first be established. When the data is described using different ontologies, the ontologies must interoperate.

Independently of answering more complex queries, combining the knowledge contained in multiple ontologies allows lifting the applications using the ontologies from data-driven and data-based applications to knowledge-based applications. While most ontology-based applications in biology utilize ontologies to analyze or describe biological data, knowledge-based applications can utilize the knowledge contained in the ontologies themselves to discover new knowledge, verify novel findings, or develop hypotheses.

Novel knowledge can be discovered using various kinds of logical inference, either deductive, abductive, inductive or analogical reasoning. Deductive reasoning infers true conclusions from true premises, abductive reasoning infers the most likely premise given a conclusion and a set of constraints, and inductive reasoning infers a general principle given a set of individual facts. Analogical reasoning identifies patterns that are similar to known patterns [Sowa and Majumdar, 2003]. These forms of reasoning are suited for different applications. But all share the property that they allow infering new, i.e. non-asserted and potentially previously unknown, knowledge from a set of given facts and constraints. This new knowledge can be utilized for various purposes, among others

- to test whether a hypothesis is sound, i.e. whether it contradicts ontological knowledge (and if it does, which parts of the knowledge),
- to generate novel hypotheses,
- to verify the consistency of findings (and which parts of the formalized knowledge a new finding contradicts), and
- to answer extended queries and searches.

Finally, molecular biology is making progress in understanding the relations between different domains and different levels of granularity. The relationship between a genotype and a phenotype involves several domains and levels of granularity within organisms, within habitats and within cells. One goal of current biological research is to understand these relationships, and ontologies that describe the results of this kind of research must be able to combine categories and relations of multiple different domains. As biology progresses as a science, its findings are naturally integrated into larger theories involving complex relationships and describing systems instead of individual components. The continuing application of ontologies to describe and communicate research results must, as a consequence, pay tribute to these more complex and integrated descriptions and permit the combination of categories and relationships of all domains of biology. As such, it is not only beneficial to achieve interoperability between domain ontologies, but mandatory in order to keep up with the scientific progress in this field.

Ontologies can not play the role they are intended to play in the unambiguous description of biomedical research findings if they remain isolated and restricted to a single-domain. Several issues still hinder the achievement of interoperability.The semantics of the representation languages in which these ontologies are formalized must be explicated where this is not yet the case. In several cases, the semantics of classical logics does not suffice, and a form of non-monotonic logic must be used to represent the ontology according to how it is used. Explication of the ontological commitment within domains is required to allow information flow between them. Finally, identifying the relationships between different domains is the result of scientific research, and must be captured to be used in formal semantic systems. The next section analyzes these issues for interoperability.

3.3 Problems with Interoperability

3.3.1 Logic and knowledge representation problems for interoperability

The choice of the logic and knowledge representation formalism influences whether and how interoperability can be achieved. As claimed in the previous sections, interoperability depends on the ability to use the knowledge encoded in multiple formalized ontologies for inferences. This presupposes that the underlying logics allow this use; they should be decidable (the set of universally valid formulas should be decidable) and inferences should be tractable.

However, most expressive logics are neither decidable nor tractable. For example, first order logics is undecidable [Church, 1936] and concept satisfiability in the description logics used in OWL-DL is NEXP-Time complete [Tobies, 2001]. Nevertheless, these logics are widely used. In practice, there are effective algorithms to decide problems in either logic, due to heuristics and optimization techniques [Tsarkov and Horrocks, 2006a, Riazanov and Voronkov, 1999].

While decidability is a general problem of logics, two problems are specific to the problem of interoperability. First, given a logic $\mathcal{L}$ and two theories T_1 and T_2, is it possible to construct a theory $T = T_1 \cup T_2$ such that for every ϕ with $T_1 \models \phi$ or $T_2 \models \phi$, $T \models \phi$? This question is the inverse of the modularization problem in which a theory T is divided in two theories T_1 and T_2 such that for every ϕ with $T \models \phi$, either $T_1 \models \phi$ or $T_2 \models \phi$. It is often (but not always, depending on the logic $\mathcal{L}$) desired that T is consistent.

An example of the first problem is combining two biomedical ontologies, such as the GO's Biological Process ontology, T_1, with the Celltype Ontology, T_2. Since both have disjoint vocabularies, i.e., their *types* are disjoint[2], combining T_1 and T_2 is trivial: $T = T_1 \cup T_2$. Then, for every ϕ_1 and ϕ_2 such that $T_1 \vdash \phi_2$ and $T_2 \vdash \phi_2$, $T \vdash \phi_1$ and $T \vdash \phi_2$. All OBO and OBO Foundry ontologies can be consistently combined in such a way, because one of the criteria for inclusion of an ontology in the OBO is the use of unique identifiers for the categories of the ontology.

Second, when two ontologies are formalized in different logics, $\mathcal{L}_1 = (L_1, M_1, \models_1)$ and $\mathcal{L}_2 = (L_2, M_2, \models_2)$, the problem of translating from one to the other arises. The translation tr should be an infomorphism, $tr = (tr^{\rightarrow}, tr^{\leftarrow})$ such that $tr^{\rightarrow} : L_1 \mapsto L_2$ and $tr^{\leftarrow} : M_2 \mapsto M_1$, and for each $b \in M_2$ and $\alpha \in L_1$:

$$f^{\leftarrow}(b) \models_1 \alpha \text{ iff } b \models_2 f^{\rightarrow}(\alpha) \tag{3.7}$$

For the purpose of establishing a flow of information between theories in different logic languages, the ontological commitment of these languages must be taken into consideration. An ontological semantics [Loebe and Herre, 2008] for these languages permits translations that maintain the languages' ontologi-

[2] Their signatures are not disjoint; they share symbols for the relations **is-a** and **part-of**. However, the arguments for these relations in each ontology are disjoint.

cal commitments.

A problem of this kind arises for example when translating the OBO Flatfile Format [Golbreich and Horrocks, 2007] to OWL-DL or *vice versa*. Here, the two logics have different expressivity, and finding an adequate translation is not straightforward. Two sub-problems can be identified here. The first presupposes that there is a well-defined semantics for each logic involved in the translation, and these semantics must be reflected in the translations. The other problem is finding the right semantics for a language when such a semantics does not exist or is insufficient. Not every knowledge representation language can express the same ontological and epistemic distinctions that are possible to express in some other languages, and not every semantics for a language is adequate with respect to how the language is used. Therefore, it is necessary to analyze whether a semantics for a language reflects the ontological and epistemic distinctions made by the users of the language.

In biology, the OBO Flatfile Format is the primary language used to specify ontologies. Historically, the OBO Flatfile Format specified a graph structure, but no formal semantics was defined. Multiple, sometimes conflicting semantics for this format were developed, each intended for different applications. The challenge is to find a semantics that resembles how the language is pragmatically used to describe biological knowledge in most or all cases.

3.3.2 Ontological issues for interoperability

Apart from logical challenges for interoperability, ontological issues arise. Interoperability between ontology-based information systems requires compatible ontological commitments between the interoperating information systems.

These commitments are represented in the conceptual schema of the information system. For information to flow, their commitments must be evaluated against the conceptual schemata in use by either information system.

Ontology integration

One possibility to achieve interoperability between two information systems based upon different conceptual schemata is to merge their ontologies into a single ontology. This is a strong form of interoperability known as ontology integration [Sowa, 2000], and usually requires extensive changes to all merged ontologies. Once the ontologies are integrated, however, the information systems then use the merged ontology, and therefore share the same ontological commitment (see figure 3.1). Information flow is realized as a flow of information between modules[3] of the merged ontology.

A special case of integrating ontologies is the ontological foundation of ontologies in a top-level ontology.In this case, the ontologies used by the information systems are analyzed with respect to a top-level ontology, and their relation to the top-level categories specified using a method of ontological mapping and reduction [Herre and Heller, 2006]. This leads to ontologies that are founded in a common top-level ontology. Information flows between the two or more ontologies via a basic core classification which is provided by the top-level ontology.

A refinement of this method is to use more specialized ontologies. These ontologies are high-level ontologies within a domain. They provide fundamental

[3] I use “module” here in its general, common sense without definition. One way for defining “module” is by reference to use within an application: the module consists of the theory that is used within the application. The flow of information between these modules is mediated by common theorems and the use of logical reasoning (deduction, induction or abduction).

types and relations pertaining to the domain. These are called *core ontologies* [Valente and Breuker, 1996].

The categories used in biological ontologies cover the whole range of categories found in top-level ontologies. While many biological domain ontologies were already analyzed with respect to their relation to top-level ontologies, several ontological issues remain open. Many of the open problems are related to controversial issues in the research field of formal ontologies, such as the notion of function [Searle, 1997, Wright, 1973, Millikan, 1988], of concepts [Smith, 2004] and sequences [Herre et al., 2006], of categories and instantiation [Loebe and Herre, 2008, Herre et al., 2006], granularity [Rector et al., 2006a], identity and persistence [Herre et al., 2006, Johansson and Althoff, 2005], principles of core and upper domain ontologies [Valente and Breuker, 1996] or representing normality and defaults [Kolovski et al., 2006, Rector, 2004, Hoehndorf et al., 2007].

Functions

Functions are an important concept in biology, and they are studied in the context of genetics as the functions of genes and gene products [Ashburner et al., 2000, Hieter and Boguski, 1997], of cells and celltypes [Mcneish, 2004], or anatomical parts and organ systems [Albin et al., 1910] or in the context of behaviour and social structure [Searle, 1997]. In particular, the notion of function is used in the Gene Ontology [Ashburner et al., 2000], the Celltype Ontology [Bard et al., 2005] and the ChEBI ontology [Degtyarenko et al., 2007]. Several authors investigated the notion of a biological function in philosophy of biology [Millikan, 1988, Wright, 1973, Searle, 1997, Hartmann, 1966]. The major divide is between the philosophers who regard functions as emergent from purely causal properties and interactions, and the philosophers who see functions as inherently social objects, which can only be understood in a social context. This influences how functions relate to other entities, such as processes and roles, and how things obtain a function. Additionally, things that are unable to perform their function, that are mal-functioning, are analyzed differently in the two alternative ontological views on functions. An ontological analysis of functions and the implications of the chosen theory of functionality on current biological ontologies should benefit information flow between ontology-based information systems that employ the notion of function.

Sequences

The ontology of sequences is another controversial issue for ontological analysis [Pearson, 2006]. Biological sequences play a major role in molecular biology, genetics and bioinformatics. They are related to different kinds of molecules (at least proteins, DNA and RNA molecules).An analysis of what kind of entities sequences are and how they relate to other entities would benefit the integration of large parts of data in genetics and permit information to flow between ontologies in this field. Ontological choices that must be explicitly stated include the existential dependency of sequences on other entities (like molecules), whether or not they can be considered categories, what and how they denote and relate to their referents and what relation they have to information.

This also tackles the problem of relating sequences and symbols to their tokens. One option is to treat the **token-of** relation as a special kind of instantiation relation. This implies that sequences are categories (and *Sequence* a higher-order category), and their tokens are instances. Other approaches may consider sequences to be properties or abstract individuals.

Higher-order categories

A related but different ontological issue in biology is the existence of higher-order categories. A higher-order category is a category that has as instances other categories, in contrast to a category that has as instances only individuals. An example of a category that can be considered *higher-order* is the category *Species*. The instantiation hierarchy for the penguin Tweety would contain: $Tweety :: Penguin$ and $Penguin :: Species$. Other examples for candidates of higher-order categories include sequences. Higher-order categories may include levels of reality, which are included in the GFO, and are an attempt to bridge levels of granularity.

Defaults and Exceptions

A further open issue pertaining to ontologies is the problem of addressing defaults and exceptions, idealizations and abnormalities. While it seems perfectly reasonable to state in an anatomy domain ontology for mice that mice have coat hair, a tail, two eyes, four legs, all these statements are false when interpreted as "all instance of *Mouse* have as part some instance of X"[4]. More correct would be to state that every *anatomically normal* mouse has coat hair, a tail, etc. It is, however, unclear what kind of ontological entity an anatomically normal mouse is, how it relates to the category *Mouse*, how it relates to instances of the *Mouse* category, whether it is existentially dependent on real mice, and similar.

The explicit incorporation of a model of normality or default knowledge in biological and biomedical ontologies is required to achieve interoperability. For

[4] More precisely, they are false when *Mouse* is intended to represent the category of all mice, understood in its usual way.

example, interoperability between anatomy and phenotype ontologies, or between phenotype and disease ontologies, requires an analysis of normality and abnormality. If no principled way for representing default knowledge is available, it may also happen that inconsistencies arise when ontologies describing phenomena and ontologies describing defaults are combined. This issue is closely related to the choice of the knowledge representation formalism. Default reasoning is inherently non-monotonic, and a knowledge representation language must be used that supports this.

Persistence

Modelling persistence and change of an object over time addresses a problem for which a solution was already proposed in the biomedical domain [Smith et al., 2005a]. Entities are divided into occurrants and endurants. Occurrants are entities with temporal parts, while endurants are wholly present at each point in time at which they exist. This distinction dates back to Lewis [2001], and is used in the DOLCE [Masolo et al., 2003] and BFO [Grenon, 2003b] top-level ontologies. Identity of endurants is often closely related to the processes in which they participate, and interrelating processes and objects often problematic. For example, consider a cell dividing into two cells. The process starts with one cell being present, and ends with two cells present. Whether one of the two cells at the process' end is identical to the cell at the process' beginning, and which of the two cells, cannot easily be answered. There are multiple choices, and no obvious way to prefer one over another.

A rigorous analysis of the kind of persistence and identity conditions employed in biological and biomedical theories must be performed and explicitly stated to avoid incompatible identity conditions for biological entities. While uniform identity conditions for many material objects are employed, identity conditions for non-material and abstract entities like sequences remain more difficult.

Core Ontologies

Finally, domains like biology or medicine exhibit their own ontological structure, that distinguishs them from other domains. Ontologies that specify these domain-specific upper-level concepts and constraints are called *core ontologies*. They can be used to structure and organize domain ontologies, support their development by providing principles for classifying domain entities and relating them to other domain entities. Additionally, they are useful to make the specific structure of a domain explicit. This is helpful in order to relate it to other, different domains. A biological core ontology could provide the means to situate biological domain ontologies within a wider context.

Even if all ontological difficulties were solved, there remains a gap that must be filled to establish links between ontologies of different domains and levels of granularity. High-level ontological analyses may provide a framework for representing knowledge about how entities in different domains and different levels of granularity can be related *in principle*. But there is domain-specific scientific knowledge *in* the links between different domains. Effective establishment of the relations between ontologies of different domains necessitates the acquisition of this knowledge.

3.3.3 Knowledge Acquisition

Letting information flow between ontologies that are developed disjointly requires additional information concerning how the categories of two ontologies are related. The OBO Foundry ontologies are intentionally *orthogonal* to each other. In particular, they do not overlap in their categories. They are, however, related. Identifying the kind of relationships between two categories of different ontologies is not a trivial task. Consider the GO's Biological Process ontology and the Celltype Ontology. Instances of a category of biological processes may always be **located in** cells of a certain type. For example, a *Leucocyte activation* is always located in *Leucocyte* cells. A process may always have cells of a specific type as participants, like *Oxygen transport* having *Red blood cells* as participants. Certain relations exclude others, while some relations require the presence of additional ones.

Formally, the need for additional knowledge can be analyzed as follows: let T_1 and T_2 be two ontologies that have no categories in common. For OBO ontologies, the combination $T = T_1 \cup T_2$ is consistent. To relate the categories used in T_1 and the categories used in T_2, addition theorems must be added that establish the relations between these categories. Formally, the additional knowledge S must be captured such that $T' = T_1 \cup T_2 \cup S$ and T' is consistent.

The task of identifying these relations is called ontology aligment or ontology matching. This alignment can be performed manually by domain experts. For each category in one ontology and a fixed relationship or a fixed set of relationships, the expert identifies the categories to which it stands in the relationship. In addition, the expert may also assert to which categories it does not stand in the relationship. Due to the size of the ontologies in biology and medicine, manual alignment by domain experts is both labour-intensive, expensive and error-prone.

One possible improvement is to utilize the collective power of the scientific community within a domain to create these alignments. While systems exist that permit individual, single users to create these relationships between ontologies, it is a challenge to provide community-based tools to support the task of ontology alignment.

Semantic Wikis

Wikis are web-based platforms that permit multiple users to collaborate on the acquisition of knowledge. Wikis are collaboratively maintained websites that permit easy modification and extension of their content [Leuf and Cunningham, 2001]. However, a wiki traditionally contains free-text content, i.e., non-formalized knowledge primarily intended for use by human users. In addition, quality of a wiki's content is only enforced through revisions and modifications performed by the users of the wiki. The ontologies, however, are structured representations of knowledge that are intended to be used not only by humans but also by machines for the automated analysis and retrieval of information.

One approach to bridge the gap between wikis and structured knowledge acquisition is the use of a semantic wiki. A semantic wiki is a collaborative website that can be edited and modified by anyone, and that has an underlying formal model of its content. Semantic wikis that utilize RDF and OWL as datamodels have been developed [Völkel et al., 2006, Schaffert et al., 2006]. These permit the representation of structured, formal knowledge in addition to the tradition free-text content, and the use of this formalized content for queries and further analyses.

On the other hand, due to this formal data model, additional difficulties in maintaining the quality the wiki's content arise. Due to the formal semantics of the data models, single mistakes may propagate throughout the knowledge base and lead to invalid content which result in invalid query results. For example, a single logical contradiction causes every formula to be derivable from the knowledge base when deduction is used as the form of logical inference. Therefore, any approach to collectively construct knowledge bases must prevent the inclusion of inconsistencies, or provide other means for maintaining consistency. One option to perform this automatic detection of inconsistencies is through the use of automated reasoners [Sirin and Parsia, 2004].

On top of logical inconsistencies, incorrect knowledge could be captured because users of such a collaborative platform conceptualize a domain in different ways, i.e., they do not commit to a common ontology. Formalized ontologies explicitly specify the conceptualization underlying a certain vocabulary, and it would benefit the quality of a knowledge base that is created and maintained by multiple users, if the captured knowledge is consistent with a common, formal ontology as the foundation of such a semantic wiki. In conjunction with automated reasoners, a formal ontology can be used to enforce a common conceptualization for the knowledge captured within such a wiki. In addition, providing easy access to inferences of these reasoners can help in maintaining not only a consistent, but also a correct knowledge base.

Social Tagging

A less powerful but simpler and therefore more easily adopted approach to collaboratively acquire knowledge from domain experts is the use of a collaborative tagging system in order to harvest information from domain experts. Tagging refers to the association of free-text keywords to a resource, and is often used by agents for organizing information according a their preferred vocabulary and conceptualization of a domain. Neither the free text keyword nor the association relation bear any kind of explicit, pre-defined semantics; the interpretation is left to the tagging agent. Nevertheless, sets of tags can be analyzed to reveal parts of the meaning that taggers associate with a tag and may shed light on the relation between the tagged object and the object denoted by the tag.

Text Mining

Alternatively, completely automated approaches can be used. Due to the large volume of biomedical literature, a promising method is the use of data or text mining to extract meaningful biological facts that can be used to align ontologies. Data mining applies (often statistical) algorithms to databases or otherwise available data sources to extract meaningful patterns. These can, together with an interpretation that situates these patterns against the used algorithm and a hypothesis, form the foundation of a knowledge base. Text mining is a sub-discipline of data mining, and uses primarily natural language texts as data for analysis. Due to the large amount of published literature in the biological and biomedical domains, it would be beneficial if results can be automatically extracted from these texts.

Text and data mining have already been used to identify partial alignments of ontologies [Ogren et al., 2004]. In general, however, ontology alignment through the analysis of texts is not a solved problem. Also, specific subproblems arise for analyzing texts in biomedicine and biology. For example, identifying gene names or the names of proteins is hard because no unique nomenclature exists for these types of entities. Developing algorithms to extract meaningful biological information from texts would benefit the alignment of ontologies in biology, and therefore the interoperability between information systems based on these. Other problems in biology and biological knowledge representation could be automated as well using these methods, in particular the annotation of genes and gene products with their functions.

4 Knowledge Representation

Logic is the beginning of wisdom,
not the end.

Lieutenant Commander Spock

4.1 Relationships and DAG semantics

The GO was initially represented as a directed acyclic graph (DAG), with edges labeled either **is-a** or **part-of**. Several idiosyncracies were discovered in the representation of the GO [Smith et al., 2003]. Attempts have therefore been made to represent these ontologies in formal languages [Wroe et al., 2003]. Smith et al. [2005a] provides a translation of the OBO DAGs into first-order logic. Golbreich and Horrocks [2007] give a semantics of the OBO flatfile format through a translation to OWL.

The basic intuition in [Smith et al., 2005a] is that the nodes of a DAG represent ontological categories, while the edges represent ontological relations between these categories. The categories can have instances, and the relations between the categories express facts about the relations between the instances of these categories. The relations between categories are explicitly defined using relations that hold between individuals. For example, the **is-a** relation between categories is defined as

$$\textbf{is} - \textbf{a}(A,B) \iff \forall t,x(instanceOf(x,A,t) \rightarrow instanceOf(x,B,t) \tag{4.1}$$

where t ranges over time points, A and B over categories and x over instances. The **part-of** relation is defined in a similar way as

$$\begin{aligned}\textbf{partOf}(A,B) \iff \forall t,x(instanceOf(x,A,t) \rightarrow \\ \exists y(instanceOf(y,B,t) \wedge partOf^{I}(x,y,t)))\end{aligned} \tag{4.2}$$

Here, $partOf^{I}$ is a relation that holds between *instances* (and in this particular case also between individuals). The OBO Relationship Ontology (RO) provides these definitions for a set of relationships. In addition, it gives a number of basic axioms for these relationships, such as transitivity and reflexitivity for **part-of**.

A problem with the approach taken by the RO is that it introduces a number of ontological distinctions on top of the definitions of the relationships. Therefore, it does more than just providing a clear semantics for the relations used in biomedical ontologies, it gives an ontological interpretation of these ontologies: it analyzes the DAG structures used in biomedical ontologies using an ontology. It is therefore not neutral with regard to how ontology creators conceptualize the world, but enforces the use of one pre-determined conceptualization of the world. In the case of the RO, the conceptualization is fixed by the top-level ontology BFO. As previously illustrated, biomedical domain ontologies use diverse conceptualizations that may not always be compatible with this top-level ontology.

The second kind of semantics that has been applied to provide a formal account of the DAGs that are used to represent biomedical ontologies is due to

Golbreich and Horrocks [2007]. Golbreich and Horrocks [2007] specifies both a syntax and a model-theoretic semantics for the OBO Flatfile Format that is commonly used to specify biomedical ontologies, and the DAGs that are represented using this format. The semantics is given by translating the OBO Flatfile Format to OWL-DL. Since OWL has a well-defined model-theoretic semantic, the translation of the OBO Flatfile Format to OWL yields a semantics for the OBO format. **is-a**-labelled edges between the nodes C and D are translated as

$$isA(C,D) \iff C \sqsubseteq D \tag{4.3}$$

while all edges between C and D labelled R (and not **is-a**) are translated as

$$\mathbf{R}(C,D) \iff C \sqsubseteq \exists R.D \tag{4.4}$$

Together with the OWL semantics, this provides a model-theoretic semantics for the language of the OBO Flatfile Format. However, due to the uniform interpretation of the relations between categories as existentially quantified description logic statements, it fails to capture the intuition of the ontology developers in several cases. The most obvious example is the relation **lacks-part** which relates categories whose instances are not part of each other.

Although some biomedical ontologies are now developed using OWL, the DAG representation of ontologies remains dominant in the biomedical domain due to its widespread use, simplicity, and because it suffices for many applications for which these ontologies are used. However, for interoperability between these applications, a semantic interpretation that closely reflects the intuitions of the ontology builders must be given. As I have argued, neither Golbreich and Horrocks [2007] nor Smith et al. [2005a] achieve this goal; the first gives a translation to OWL-DL, which does not necessarily reflect the intuitions of the ontology builders, while the second fixes a particular ontological commitment which hinders interoperability between ontologies built with a different conceptualization in mind.

I propose an intermediate solution, that provides for making the ontological commitment of the developers of biomedical ontologies explicit, without determining it in advance. I propose to let the ontology developers that use the OBO Flatfile Format make their intension explicit whenever they use a type of relation. This can be achieved by either giving an explicit definition of a relation, or by axioms that describe the meaning of the relation. These axioms can be included with the ontology, or kept in a separate ontology like the OBO Relationship Ontology.

The translation to OWL's abstract syntax for an OBO Flatfile relationship statement

```
relationship: relationship-id term-id
```

is according to [Golbreich and Horrocks, 2007]

```
restriction(relationship-id someValuesFrom(term-id))
```

This fixes a particular interpretation of what a relation between two terms in the OBO Flatfiles designates. Although the intension of the OWL relation `relationship-id` is not specified, the relationship represented in the OBO flatfile, as a relationship between two terms (which represent categories), is defined using a new relationship between the instances of these categories; and this new relationship is used in an existential statement.

A minimal extension of the current OBO flatfile semantic, that still permits the translation to the decidable logics OWL-DL or OWL 1.1, is to include the OWL translation of a relationship in the `typedef` stanza of the OBO flatfile. This leads to a modified `typedef` stanza:

```
typedef-stanza :=
   ’[Typedef]’
   typedef-TVP
   ’name:’<string>
   [  ]
   [ <namespace> ]
   { <alt_id> }
   [ <def> ]
   [ <comment> ]
   { <subset> }
   { <synonym> }
   { <xref> }
   [ meta-property-TVP-modified ]
   [ ’is_metadata_tag:true’ | ’is_metadata_tag:false’ ]
   [ <is_obsolete> ]
   [ <replaced_by> ]
   { <consider> }
```

I define `meta-property-TVP-modified` and `meta-property-old` as

```
meta-property-TVP-modified :=
   meta-property-old | relationship-definition

meta-property-old :=
   [ domain-TVP ]
   [ range-TVP ]
   { meta-property-TVP }
   { r-isa-TVP }
   [ inverse-TVP ]
   [ transover-TVP ]
   { relationship-TVP }
```

One difficulty is deciding on a syntax of the relationship definition. The relationship that is to be defined holds between two categories, *C* and *D*. The translation of this relation must yield a valid OWL expression. In this OWL expression, *C* and *D* are variables that are filled by the actual participants of a use of the defined relation. Since the OBO Flatfile Format is intended to be read both by machines and read by humans, I chose the Manchester OWL Syntax [Horridge et al., 2006] to represent the OWL statement. To represent both categories as variables, I extend the Manchester OWL Syntax as well by `?X` and `?Y` constructs.

```
relationship-definition :=
   owldef: " manchester-owl-statement "
```

`manchester-owl-statement` is an OWL axiom in Manchester syntax, where `?X` and `?Y` stand for *classIDs*.

The translation to OWL presented in [Golbreich and Horrocks, 2007] must be adapted to translate this new type of statement. However, the newly introduced rule is part of this translation itself: every occurrence of the defined relation is translated by its definition. This cannot be represented using a non-conditional replacement function.

If a relationship is not defined by a `relationship-definition` (but only a `meta-property-old`), the translation function is not changed. Otherwise, the following translation is used. The `relationship-TVP` that occur in a `[Term]` stanza with `term-id` as ID are currently translated as `SubClassOf(term-id T(relationship-TVP))`, and the restriction in the `relationship-TVP` as

```
 restriction( relationship-id someValuesFrom(term-id))
```

This must be changed to another translation. Every occurrence of a

```
 relationship: relationship-id term-id
```

in a `[Term]` stanza with `term-id-2` as its ID must be replaced with

$$MT(Subst(manchester\text{-}owl\text{-}statement, term\text{-}id, term\text{-}id\text{-}2)).$$

$Subst(S,X,Y)$ is a function that substitutes every occurrence of `?X` in S with X and every occurrence of `?Y` in S with Y. $MT(S)$ translates the Manchester OWL Syntax to OWL Abstract Syntax[1].

As example, the relation **lacks-part** will be defined as follows:

```
[Typedef]
id: lacksPart
owldef: "Class: ?X SubClassOf: not hasPart some ?Y"
```

Then, a definition of the category *Mouse with absent tail* is

```
[Term]
id: MouseWithAbsentTail
name: Mouse with absent tail
relationship: lacksPart tail
```

The translation function will yield the following OWL Abstract Syntax for this statement:

[1] This translation must be performed to be compatible with the translation function defined in [Golbreich and Horrocks, 2007].

```
Class(MouseWithAbsentTail
  complementOf(restriction(hasPart someValuesFrom(tail))))
```

The definition of the **lacks-part** relation can be refined by defining the **has-part** relation using a `meta-property-old` element, i.e., defining **has-part** as transitive and symmetric, which influences not only the interpretation of **has-part** but of **lacks-part** as well when the above definition is used.

To specify the intensions of relations used in the OBO Flatfile Format, I have extended the syntax of OWL by the variables `?X` and `?Y`. Both are variable symbols that are intended to represent *concepts*. To generalize the approach of defining relations between concepts using this extension of OWL, the OWL semantics must be extended to include an interpretation of these concept variable symbols. The semantics of OWL is given in [Patel-Schneider et al., 2004, Baader, 2003].

The semantics of a description logic theory over a signature $\Sigma = (C, R, A)$, with C a set of concept symbols (including $\top$ and $\bot$), R a set of relation symbols and A a set of individual symbols, is given by an interpretation $\mathcal{I}$. The interpretation $\mathcal{I}$ consists of a non-empty set $U^{\mathcal{I}}$ and an interpretation function δ, such that for every $C_i \in C$, $\delta(C_i) \subseteq U^{\mathcal{I}}$, $\delta(R_i) \subseteq U^{\mathcal{I}} \times U^{\mathcal{I}}$ for every $R_i \in R$ and $\delta(a) \in U^{\mathcal{I}}$ for every $a \in A$. The interpretation function is inductively extended in the usual way. Using standard description logic notation [Baader, 2003], examples of these inductive definitions include:

$$\top^{\delta} = U^{\mathcal{I}}$$

$$\bot^{\delta} = \emptyset$$

$$(\neg A)^{\delta} = U^{\mathcal{I}} \backslash A^{\delta}$$

$$(C \sqcap D)^{\delta} = C^{\delta} \cap D^{\delta}$$
$$(\forall R.C)^{\delta} = \{a \in U^{I} | \forall b.(a,b) \in R^{\delta} \rightarrow b \in C^{\delta}\}$$
$$(\exists R.C)^{\delta} = \{a \in U^{I} | \exists b.(a,b) \in R^{\delta} \wedge b \in C^{\delta}\}$$

Using a higher-order logic, the interpretation will map free concept variables such as ?X and ?Y to a subset of the powerset of U^{I}:

$$\delta(?X) \in \mathcal{P}(U^{I})$$

$$\delta(?Y) \in \mathcal{P}(U^{I})$$

Universal quantification over these free variables would then range over the full powerset of U^{I}. In particular, satisfiability of terminological axioms[2] that contain concept expressions involving ?X or ?Y must consider the powerset of U^{I}. The use of the powerset in the interpretation yields undecidability.

For defining relations in the OBO Flatfile Format using the extended OWL statements that I introduced, it is not necessary to use the full powerset in the interpretations of the two concept variables. Instead, the variable symbols ?X and ?Y can be interpreted with an extension of one of the atomar concepts from the signature Σ. If Σ is finite, then satisfiability of terminological axioms in OWL extended with ?X and ?Y will be decidable.

Formally, let T be a description logic theory over the signature $\Sigma = (C \cup \{?X, ?Y\}, R, A)$ and I be an interpretation with the interpretation function δ and a domain U^{I}, and $P^{-}(U^{I}) = \{C_i^{\delta} | C_i \in C\}$. Then $\delta(?X) \in P^{-}(U^{I})$ and $\delta(?Y) \in P^{-}(U^{I})$.

[2] Terminological axioms in description logics are of the form $C \sqsubseteq D$ or $C \equiv D$ with C and D being concept expressions, or $R \equiv S$ with R and S being relationship (role) expressions.

This restriction leads to decidability of the satisfiability problem for terminological axioms, as long as the signature Σ is finite: satisfiability of a terminological axiom involving `?X` or `?Y` can be decided by verifying the satisfiability of the terminological axioms that arise through substituting `?X` and `?Y` with all atomar concept symbols in Σ. Since the signature $\Sigma = (C, R, A)$ is finite, $|C|^2$ terminological axioms must be verified for satisfiability to decide the satisfiability of one axiom involving `?X` and `?Y`.

Due to the decidability of satisfiability of terminological axioms, the definition schema for relations in the OBO Flatfile Format can be employed in the inverse direction. I described how relations can be defined and be translated to OWL according to this definition. Based on these definitions, new relations between categories can be extracted from an OWL knowledge base. Therefore, these definitions can also serve as a method for an extended form of reasoning using the OBO Flatfile Format.

4.2 Semantics for Frame-Based Ontologies

Some ontologies like the Foundational Model of Anatomy (FMA) are being developed using a frame-based system [Minsky, 1977]. Considerable research has been done to provide formal semantics for frame-based systems [Lassila and McGuinness, 2001, Fikes and Kehler, 1985, Brachman and Schmolze, 1985, Borgida et al., 1989], and for the FMA in particular [Dameron et al., 2005].

The FMA consists of a set of categories and relations between them. The relations that relate categories have inverse relations defined. These are inverse relations between categories: for two categories C and D, when R^{-1} is the inverse of R and $R(C,D)$, then $R^{-1}(D,C)$. These inverses do not translate uniformly to instances of the categories, and the definition schemata used in the OBO Relationship Ontology cannot be applied to these. The assertion that $partof(Appendix,Human)$ is, according to the RO, an assertion that all instances of *Appendix* are part of some instance of *Human*. This does not logically entail $haspart(Human,Appendix)$, i.e., that all instances of *Human* have as part some instance of *Appendix*. In fact, both are very different statements when the definition schema for relationships used by the RO is employed. This is particularily important in the case of gender-specific statements such as *partof(Uterus, Human)*.

Additionally, a similar problem as in many OBO ontologies arises in the case of default statements. Many statements in the FMA are not universally true. It is not the case that every instance of a *Human* has as part an *Appendix*. These statements must be interpreted as *defaults*, and this must be reflected in their semantics. I defer this discussion to section 5.3, which discusses the role of defaults in representing ontologies such as the FMA.

4.3 Annotation relation and semantics

Most biomedical ontologies are developed and applied for the *annotation* of biological entities. The annotation relation is usually not part of the ontologies, but an extension of the theories in which they are expressed. The use of the annotation relation therefore becomes a problem of extending the theory representing the ontology by additional facts.

In this section, I consider each entity that can be used in an annotation relation to be a logical individual (but not an ontological individual), i.e., represented by a constant symbol. Then, given a theory T, an entity d can be annotated to a category C, $T \models ann(d,C)$, it can be provably not annotated to C, $T \models \neg ann(d,C)$, or it can be unknown whether it is annotated to C or not, $T \not\models ann(d,C)$ and $T \not\models \neg ann(d,C)$. As a result, it is possible to define the logical incompleteness of the annotation as the cardinality of the set $\{ann(x,Y)|T \not\models ann(x,Y)$ and $T \not\models \neg ann(x,y)\}$.

The underspecification of the annotation relation does not entail that no axioms can be developed for it. The most prominent example of an axiom involving the annotation relation is the True Path Rule [Ashburner et al., 2000] which states that annotation is transitive over both **is-a** and **part-of**.

However, the annotation relation can be ontologically analyzed and the relation between the datum annotated and the category to which it is annotated analyzed and explicated. The next chapter will perform this analysis for ontological categories and relations as well as, in parts, for the annotation relation.

5 Ontological requirements for interoperability

> The genius of culture is to create an ontological system so compelling that what is inside and outside of a person are viewed as of a piece, no seams and patches noticable.
>
> Richard Shweder

The second major component for permitting information flow between ontologies is the basic conceptualization that is used in the ontologies between which information flow is to be established. Ontologies are specifications of the meaning of a domain conceptualization. These conceptualizations may be conflicting in such a way that it is not obvious how a statement made using one ontology can be expressed in the other ontology. Two ontologies may refer to the same or similar parts of reality in very different ways by using different conceptualizations. Statements using one conceptualization may carry a lot of additional information when correctly interpreted within another conceptualization.

Consider an example of a protein with a function – e.g., to transport sugar – described using an ontology containing only categories of functions. The information about this protein is rather limited. However, having this function may carry the information about processes – sugar transport processes – and the participants and roles in these processes. They may carry information about the physical structure of the protein – having at least a binding site for sugar molecules. Within a systems approach, it carries information about pre- and post-conditions of pathways or other complex interactions. But for a statement to carry this information, an information flow must be established between ontologies of functions and ontologies of processes, structures or systems. Several questions must be answered to establish this flow of information: how do and how can functions relate to processes, to physical structures, to other functions or to systems? These are ontological questions and they must be answered within a general ontological framework.

One solution to achieving a conceptual homogeneity within a domain is the development of a top-level ontology for the domain, a *core ontology*. Core ontologies are more specific than top-level ontologies, but more general than domain ontologies. They provide a conceptual framework for the entire domain. Core ontologies can, therefore, be used to make the ontological commitment of domain ontologies explicit, and integrate them in a top-level framework.

We have developed GFO-Bio to play this role for the biological domain [Hoehndorf et al., 2008a]. GFO-Bio is a core ontology for biology that is intended for analysis and specification of the ontological commitment of biological domain ontologies. It contains several basic categories, relations and axioms that are formalized in OWL and first order logic. However, GFO-Bio contains two components that go beyond an implementation in OWL or first order logic, and require more elaborate discussion. The ontology of functions and the ontology of reference models combined with abnormalities are therefore discussed separately. This chapter starts with an extended discussion of GFO-Bio and a presentation of its category system together with its axioms. Then the notion of a *function* in biology is analyzed, and finally I analyze the role of domain ontologies that form reference models within their domain, and analyze how they can be combined with other kinds of ontologies in the framework of GFO-Bio.

5.1 GFO-Bio: A biological upper domain ontology

One approach to achieving interoperability between ontologies is through top-level ontologies. The top-level ontology can be used to make the ontological commitment of the domain ontologies explicit. In addition, a top-level ontology provides a common conceptualization of the most general kinds of entities in reality.

Categories within a domain ontology can be restricted by means of axioms using the categories from top-level ontologies. The simplest form of such an axiom is the assertion of an **is-a** relation between a domain category and a category of a top-level ontology. For example, a domain ontology may contain the category *Apoptosis* (controlled cell death), a top-level ontology the category *Process*. The statement that *Apoptosis* **is-a** *Process* establishes a relation between both, and enforces the axioms of the top-level category *Process* for the domain category *Apoptosis*. It is then possible to conclude, for example, that Apoptosis has a temporal extension, at least one participant, etc.

There are usually multiple categories in a domain ontology, which are themselves related in particular ways, such as forming a taxonomy or partonomy. It is therefore beneficial to choose the most general categories of the domain ontology and give axioms for these. The axioms are then inherited along a taxonomy, and can be inherited along other kinds of relations as well. This is made easier by a guideline within the OBO ontologies that ontologies should be **is-a**-complete, i.e., include explicit **is-a** relations for each category to a common super-category or a small set of super-categories [Smith et al., 2007].

I call an ontology which contains the most general categories (with respect to a taxonomy) within a domain an *upper domain ontology*. The categories in the upper domain ontology are restricted by axioms, often in the form of a specialisation of top-level concepts and additional restrictions. These restrictions can be given as explicit definitions [Barwise, 1985], or in the form of axioms.

For example, *Biological Process* can be introduced using the statement

$$IsA(BiologicalProcess, Process) \tag{5.1}$$

Then, a domain ontology which uses *Biological Process* as its most general category can be integrated with the upper domain ontology by *defining* the domain ontology's *Biological Process* to be equivalent with the *Biological Process* category of the upper domain ontology. More information is added when *Biological Process* is defined as a process which has as participant some organism or part of some organism[1], because it permits the derivation of additional information from the definition.

These principles do not yield a general limiting principle for the categories that must be included in the upper domain ontology; it remains a matter for the ontology designer to decide which categories are considered to be general enough for inclusion in the upper domain ontology. This decision will depend on the intended use of the ontology.

One use of an upper domain ontology is the integration of several domain on-

[1] I do not want to make the claim here that this is a good definition for the *Biological Process* category, but only use it to illustrate the example.

tologies. Ontology integration is the “process of finding commonalities between two different ontologies *A* and *B* and deriving a new ontology *C* that facilitates interoperability between computer systems that are based on the *A* and *B* ontologies” [Sowa, 2000].

The integration of domain ontologies by constructing an upper domain ontology can be performed in several steps: first, the most general domain categories used in each of the domain ontologies are identified[2]; second, partial definitions for these categories are given using the categories of a top-level ontology. The third step consists of establishing axioms for the categories introduced in step one. For example, biological processes may be required to have at least one biological material object as participant. The result of steps one to three is the upper domain ontology. The final step in the integration of domain ontologies is the definition of the most general concepts of the domain ontologies (from step one) using relations and concepts from the upper domain ontology. This results in a combined theory consisting of three kinds of modules: one top-level ontology, one upper domain ontology and several domain ontologies.

The upper domain ontology, which can be a product of the integration of several domain ontologies, can be further used to guide the construction of new ontologies within that domain. It can serve as a starting point for the description of further, more specific categories as sub-categories of the categories in the upper domain ontology.

We developed the biological upper domain ontology GFO-Bio. It is intended for use within the Semantic Web [Berners-Lee et al., 2001], and therefore is formalized primarily in OWL. GFO-Bio extends the top-level ontology GFO

[2] This is no easy task in itself, and to my knowledge, no principled method exists to achieve this goal. Within the OBO, every ontology must have a single root category (via **is-a**). The most general categories can therefore be identified using the ontologies’ taxonomic structure.

by categories and relations pertaining to the biological domain. In the following sections I describe the structure and axioms of GFO-Bio, following the major distinctions made in the GFO (presentials, occurrents and categories). I conclude the description of GFO-Bio with an analysis of how it can be applied to the integration of domain ontologies.

5.1.1 Biological Presentials

Most presentials that are considered in biology are sub-categories of *Material object*s in the GFO. Important biological material objects are *Organism* and *Cell*. Both cells and organisms exhibit the property of *autopoiesis* [Varela et al., 1974], which some philosophers suggest as a defining, emergent property of the domain of biology. We included other categories in GFO-Bio because of their relation to cells and organism. Derived from the *Cell* and *Organism* categories are populations of organisms, tissue, cell components and macro-molecules.

In GFO-Bio, *Cell* and *Organism* are not explicitly defined or axiomatized. They are understood as autopoietic systems, systems which are organized as a network of processes which cause themselves. We developed no formal theory of autopoiesis, but use the theory provided by Maturana and Varela [1980].

The categories of presentials that are elaborated in GFO-Bio are the categories *Population*, *Tissue* and *Cell component*. I consider these, together with *Cell* and *Organism*, the most important categories for integrating domain ontologies.

I consider populations to be homogenuous groups of organisms: they contain as members only organisms of a single species. They are homogenuous in the sense that all members of a population share a common property, which provides an identity criterion for the population as well as a criterion for membership in the population. Often, this property is responsible for limiting the gene flow between members of the population and individuals outside the population: due to this property shared by the members of the population, they interbreed more often with other members of the population than with organisms that do not have this property. The kind of property may vary, and include the membership in an ecological niche, the geographic location, a specific genetic trait or polymorphism or behavior.

$$hasMember(x,y) \leftrightarrow hasPart(x,y) \wedge y :: Organism \wedge x :: PopulationPres \tag{5.2}$$

$$x :: PopulationPres \rightarrow \exists S(S :: Species \wedge \forall y(hasMember(x,y) \rightarrow y :: S)) \tag{5.3}$$

$$\begin{split} x :: PopulationPerp \rightarrow & \exists P(isa(P, Property) \wedge \forall y,z(exhibits(x,y) \wedge \\ & (hasMember(y,z) \leftrightarrow \exists p(p :: P \wedge inheresIn(p,z))))) \end{split} \tag{5.4}$$

The category *Tissue* is similar in some aspects to the category *Population* in GFO-Bio. A tissue consists of a group of cells within an organism that share a common function. Although they are often part of an organ, this is not necessarily the case, as for the tissue *Blood*.

$$x :: Tissue \rightarrow \exists y(y :: Organism \wedge partOf(x,y)) \tag{5.5}$$

$$\begin{split} x :: Tissue \rightarrow & \exists F \forall y(partOf(y,x) \wedge y :: Cell \rightarrow \\ & \exists f(f :: F \wedge hasFunction(y,f))) \end{split} \tag{5.6}$$

In GFO-Bio, I define cell components as parts of cells that have at least one molecule as proper part. Therefore, they lie between cell and molecule in a partonomy. It is tempting to require that each component must have some function, so that the components of a cell are defined not as arbitrary parts of cells, but as functional sub-units of a cell. In this sense, these components could be seen as the components that constitute the cell as an autopoietic system. However, it may well be that parts of a cell are identified as cell components while not having a function, or of which an ontology designer does not want to assert a function. It is possible to define the category of a functional cell component as a cell component which has some function, but requiring that each cell component has a function is an axiom I find too strong for addition in GFO-Bio, and it should be included in a domain ontology.

$$\begin{aligned} x :: CellComponent \rightarrow & \exists y,z(y :: Cell \wedge z :: Molecule \wedge \\ & partOf(x,y) \wedge properPartOf(z,x)) \end{aligned} \tag{5.7}$$

This condition differs slightly from the use of *Cell component* in the GO. The GO employs the following definition for the *Cell component* category:

> The part of a cell or its extracellular environment in which a gene product is located. A gene product may be located in one or more parts of a cell and its location may be as specific as a particular macromolecular complex, that is, a stable, persistent association of macromolecules that function together.

GO's definition includes entities that lie outside a cell, and is more general than the restriction I propose here. My proposal corresponds to GO's *Cell part* (`GO:0044464`) category.

Molecules such as proteins, amino acids, nucleotides, DNA or RNA molecules are included in GFO-Bio, too. Their inclusion in a biological ontology is less well motivated than the inclusion of many of the other categories. However, modern molecular biology and genetics depend heavily on the study of the function and structure of these molecules and their interactions. Understanding the relation between the information encoded in chains of molecules and the phenomena that can be observed on a macroscopic scale poses one of the most challenging problems of biology today. It is clear, however, that many biological phenomena can only be understood when information on the molecular scale is taken into consideration.

GFO-Bio distinguishes amino acids, nucleotides, proteins, protein domain, and the polynucleotide molecules RNA and DNA. The axioms that distinguish them pertain mostly to the parts they have: proteins consist of multiple amino acids, polynucleotides of multiple nucleotides.

Further presentials that are provided by GFO-Bio are *Material boundary* and *Amount of substrate*. These come from the GFO, and are not further extended in GFO-Bio.

5.1.2 Biological Occurrent

Processual entities complement presentials in the GFO. In the GFO, temporally extended entities are divided in processes and occurents. Occurents are changes, events and histories, entities that are not genuinely temporally extended, but are abstracted from a series of time boundaries. Processes, on the other hand, unfold in time and cannot be reduced to a series of time boundaries. Examples of occurrent categories in GFO-Bio include experiments, experiment actions, chemical reactions and pathways, organism development, the development of anatomical parts and development stages.

When modelling a domain, it is often a matter of granularity whether a kind of temporally extended entity is a process or an event that occurs at an instant. For example, consider the category *Chemical reaction*. It is possible to classify chemical reactions as sub-categories of either *Process* or *Instantanuous change* in the GFO, and neither is *a priori* preferable to the other. For the purpose of a biological upper domain ontology, which is intended to be independent of granularity, this poses a problem. Therefore, whenever both options are available for a user of GFO-Bio, I declare the corresponding category to be a sub-category of *Processual structure* instead of one of the more specific sub-categories *Process* or *Occurrent*. Then, both options are available when extending one of GFO-Bio's categories with sub-categories.

Experiments

The first, experiments and experiment actions, are phenomena not only of the biological, but also of the mental and social world. An action is a directed (goal-oriented), causal process. The presential that causes the process throughout its existence is called the agent. Experiments are actions that may consist of a series of sub-actions. These categories are not further elaborated in GFO-Bio. An elaborated theory of experiments and their relation to actions, goals and objectives will be part of the Ontology of Biomedical Investigations [Whetzel et al., 2006, Smith et al., 2007].

Chemical reactions

*Chemical reaction*s are processual structures that can be conceptualized as either instantanuous changes or as processes[3]. They have at least two chemical substances as participant, which play the roles of *Reactant* and *Product*. The chemicals playing the role of the reactant are transformed to chemicals playing the role of product by the chemical reaction. In general, the kind of chemicals playing the reactant and product roles differ because they underwent a chemical change in the course of the reaction. Therefore, a chemical reaction is determined by at least two distinct time boundaries: at the first, a number of chemicals playing the reactant role are present, at the second the chemicals playing the role of the product of the reaction; all the parts of the chemicals playing the reactant role that are atoms or electrons are also present at the second time boundary, as parts of the chemicals that play the product role.

$$x :: Reactant \rightarrow x :: Role \tag{5.8}$$

$$x :: Product \rightarrow x :: Role \tag{5.9}$$

$$\begin{aligned} hasReactant(x,y) \rightarrow & x :: ChemicalReaction \wedge y :: Presential \wedge \\ & \exists z (z :: Reactant \wedge roleOf(z,x) \wedge plays(y,z)) \end{aligned} \tag{5.10}$$

$$\begin{aligned} hasProduct(x,y) \rightarrow & x :: ChemicalReaction \wedge y :: Presential \wedge \\ & \exists z (z :: Product \wedge roleOf(z,x) \wedge plays(y,z)) \end{aligned} \tag{5.11}$$

$$x :: ChemicalReaction \rightarrow x :: ProcessualStructure \tag{5.12}$$

[3] It is better to say that the *Chemical reaction* category in GFO-Bio is a template for domain-specific categories of chemical reactions. The category for chemical reactions will probably be conceptualized as either processes or instantanuous changes in any application of GFO-Bio.

$$\begin{aligned} x :: ChemicalReaction \rightarrow & \exists t_1, t_2 (tb(t_1) \wedge tb(t_2) \wedge \\ & \forall y, z (hasReactant(x,y) \wedge hasProduct(x,z) \rightarrow \\ & at(y,t_1) \wedge at(z,t_2))) \end{aligned} \tag{5.13}$$

$$\begin{aligned} x :: ChemicalReaction \wedge hasReactant(x,y) \wedge hasProduct(x,z) \rightarrow \\ & \forall a (partOf(a,y) \wedge \\ & (a :: Electron \vee a :: Atom) \rightarrow \\ & \exists p, t_1 (p :: Perpetuant \wedge exhibits(p,a,t_1) \wedge \\ & \exists b, t_2 (exhibits(p,b,t_2) \wedge partOf(b,z)))) \end{aligned} \tag{5.14}$$

I introduce one further category that is required for modelling biological sequences, chemical bonds. I refrain from a detailed analysis of a chemical bond at this point, as it belongs more to the chemical domain. Further, I do not include the dynamic properties of chemical bonds.

$$x :: ChemicalBond \rightarrow x :: Relator \tag{5.15}$$

$$\begin{aligned} x :: & ChemicalBond \rightarrow \exists r_1, r_2, a, b (r_1 \neq r_2 \wedge \\ & r_1 :: RelationalRole \wedge r_2 :: RelationalRole \wedge \\ & (a :: Atom \vee a :: Molecule) \wedge (b :: Atom \vee b :: Molecule) \wedge \\ & roleOf(r_1,x) \wedge roleOf(r_2,x) \wedge plays(a,r_1) \wedge plays(b,r_2)) \end{aligned} \tag{5.16}$$

$$bound(a,b) \rightarrow \exists x, r_1, r_2 (x :: ChemicalBond \wedge \\ roleOf(r_1,x) \wedge roleOf(r_2,x) \wedge plays(a,r_1) \wedge plays(b,r_2)) \tag{5.17}$$

Organism development

Organism development and the development of anatomical parts are domains for which a multitude of ontologies have been developed [Haendel et al., 2007, Hayamizu et al., 2005]. They describe how organisms of a certain species normally develop, usually in terms of the parts they have at each development stage. In GFO, the development of an organism can be analyzed as the process that is associated to the organism's *Perpetuant*. At each process boundary of the organism's development, the presential that participates in the process has parts. The sub-process of the organisms development in which it has a certain collection of parts or some other features is a development stage with respect to this collection of parts.

5.1.3 Biological categories

A feature that distinguishes the GFO from many other top-level ontologies is its inclusion of higher-order categories, i.e., categories that have categories as their instances. The use for categories in a biological upper domain ontology is twofold. GFO-Bio includes *Species* as a sub-category of *Category*, and it includes *Biological sequence* as a sub-category of *Symbol structure*.

Species

Species and other biological taxa are higher-order categories in GFO-Bio: their instances are categories of organisms. The instance of *Species* is, among others, the category *Dog*, which has individual dogs as instances. Instances of *Species* have as instances only instances of *Organism*.

$$x :: Species \rightarrow x :: Category \tag{5.18}$$

$$x :: Species \rightarrow \forall y(y :: x \rightarrow y :: Organism) \tag{5.19}$$

Other biological taxa can be represented similarily, resulting in an instantiation hierarchy of biological taxa. The problem with such an approach is, that the relation between an individual such as the dog "Nero" and its species, family, genus, kingdom or domain becomes blurred with every further instantiation relation. Also, a query for a list of all biological taxa that Nero belongs to is complicated. The reason for this difficulty is the anti-transitivity of the instantiation-relation. For this purpose, I introduce a new relation $::^*$ which represents the transitive closure of the **instance-of** relation:

$$x :: y \rightarrow x ::^* y \tag{5.20}$$

$$x ::^* y \wedge y ::^* z \rightarrow x ::^* z \tag{5.21}$$

$$\begin{gathered}\forall R[\forall x,y,z((x :: y \rightarrow R(x,y)) \wedge (R(x,y) \wedge R(y,z) \rightarrow R(x,z))) \rightarrow \\ \forall x,y(x ::^* y \rightarrow R(x,y))]\end{gathered} \tag{5.22}$$

Then, the dog Nero is not an instance of *Species* or the kingdom *Animal*, but stands in the relation $::^*$ to both.

Other forms of representing species are compatible with GFO-Bio, as well. In particular the methods discussed in the comprehensive survey of representing biological taxa in ontologies performed by Schulz et al. [2008] can be applied within GFO-Bio. The use of the instantiation relation for the representation of biological taxa can express the same distinctions as the other methods in [Schulz et al., 2008]. In GFO, instantiation is an explicitly introduced relation that is not equivalent to predication in logics. Therefore, use of categories of higher order does not necessitate the use of a higher order logic.

Symbols and Symbol Sequences

The second kind of higher-order categories in GFO-Bio are biological symbols and sequences. The primitive biological symbols included in GFO-Bio are either symbols standing for the nucleotides *Adenine*, *Guanine*, *Thymine*, *Cytosine* and *Uracil*, or symbols standing for the 20 amino acids that can be found in proteins. The tokens of these symbols are particular molecules. These symbols are primitive; they do not have an internal structure.

The theory of biological symbols and sequences that I propose here is intended to be compatible with the Sequence Ontology (SO) [Eilbeck et al., 2005a]. The sequence ontology uses two basic categories in the characterization of sequences, *Sequence* and *Junction*. Both can have *attributes*, i.e., properties. For example, a sequence may be a *gene* or a *base*, a junction an *insertion site* and a sequence attribute *enzymatic*.

In addition to these basic categories, the SO introduces collections of sequences such as a *genome*, operations on sequences such as *delete* and *insert*, and events that change sequences (*mutations*).

The basic relations used in the SO are **partOf** and **derivesFrom**, as well as a group of similarity relations between sequences: **homologousTo**, **orthologousTo**, **paralogousTo** and **nonfunctionalHomologOf**.

Hidden in the definitions of the categories used in the SO are multiple categories that are not explicated, most notably the notion of *Function* which is used in the definition of several categories such as *Pseudogene* and, indirectly, in the definition of *gene*.

I provide a characterization of *Sequence* and *Junction* in the framework of GFO-Bio, together with a mereological system that is applicable to sequences. The theory proposed here assumes that *Sequence* and *Junction* are primitive categories. In particular, they are not defined, but characterized axiomatically.

Sequences are linear entities and can come in two facets. Sequences can either have a start and an end point (see figure 5.1.3), or form circles (see figure 5.1.3 and 5.1.3). There are sequence atoms, which I call primitive biological symbols. Primitive biological symbols have no proper sequence parts.

Sequences can have boundaries, but not necessarily directionality. The boundaries cannot always be divided into a start and an end. For DNA sequences, a direction can be established, from the *Five prime untranslated region* to the *Three prime untranslated region*. However, the general theory of sequences I propose here uses no such directionality, and it should be introduced at a later stage as extension of the theory.

The theory is based on these primitives: the categories *Seq* of biological sequences, *Jun* of junctions, *Mol* of molecules, and the relations **sPO** (sequence-part-of), **PO** (part-of), **binds**, **::** (instantiation), **between**, **end** and **conn**.

The first part consists of axioms that ground sequences and molecules in the GFO, and restricts the arguments of the relations. Additionally, an axiom re-

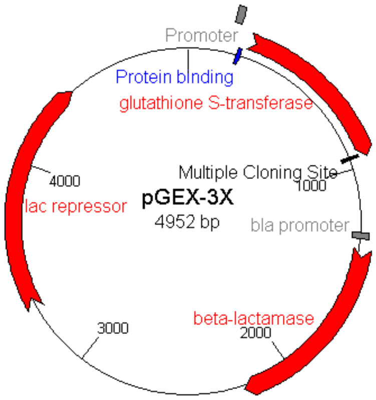


Figure 5.1: The pGEX-3x plasmid cloning vector is an example of an entity which exhibits a circular sequence.

quiring all sequences to have only molecules as instances is introduced.

$$Seq(x) \rightarrow x :: SymbolStructure \tag{5.23}$$

$$Jun(x) \rightarrow x :: Abstract \tag{5.24}$$

$$Mol(x) \rightarrow x :: Presential \tag{5.25}$$

$$sPO(x,y) \rightarrow Seq(x) \wedge Seq(y) \tag{5.26}$$

$$PO(x,y) \rightarrow Mol(x) \wedge Mol(y) \tag{5.27}$$

$$binds(x,y) \rightarrow Mol(x) \wedge Mol(y) \tag{5.28}$$

$$Seq(x) \rightarrow \forall y (y :: x \rightarrow Mol(y)) \tag{5.29}$$

As a corollary of these axioms, sequences, junctions and molecules are disjoint, because symbol structures, abstract individuals and presentials are disjoint in the GFO.

The relation **sPO** is a parthood relation that holds for sequences when one sequence contains the other as a sequence motif. It satisfies reflexivity, transi-

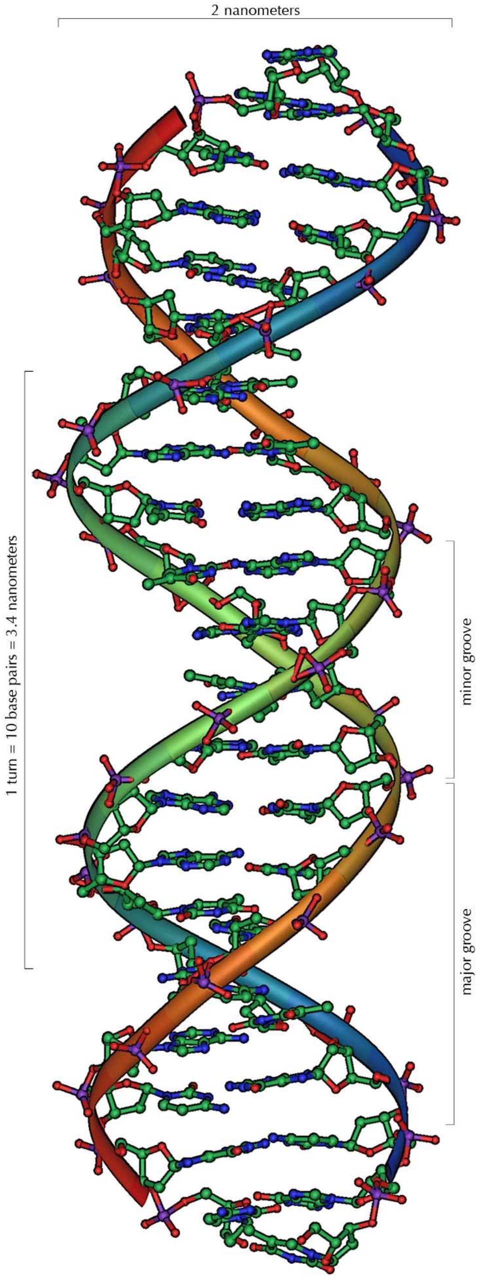
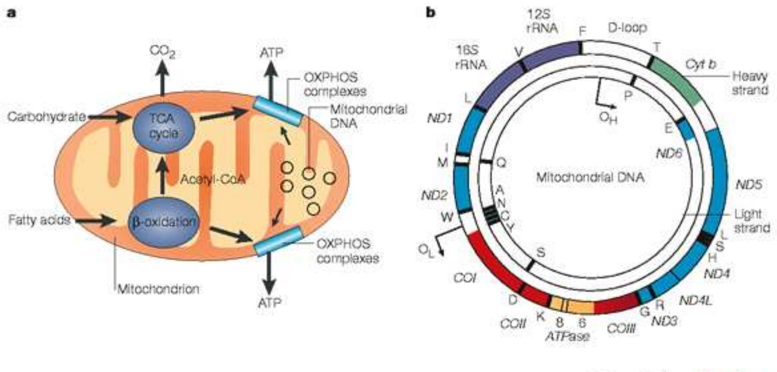


Figure 5.2: An illustration of the structure of DNA molecules. Attached to the backbone are nucleotides, which form a linear sequence.



Nature Reviews | Genetics

Figure 5.3: An illustration of the structure of a mitochondrion genome (from [Taylor and Turnbull, 2005]). The genomes of mitochondria are examples of circular sequences.

tivity and antisymmetry, and therefore forms a partial order.

$$sPO(x,y) \land sPO(y,z) \rightarrow sPO(x,z) \tag{5.30}$$

$$Seq(x) \rightarrow sPO(x,x) \tag{5.31}$$

$$sPO(x,y) \land sPO(y,x) \rightarrow x = y \tag{5.32}$$

Next I define the relation **sPPO** (proper sequence part) and the category of primitive biological symbols (*PBS*) as well as the **soverlap** and **sdisjoint** relations:

$$sPPO(x,y) \leftrightarrow sPO(x,y) \land x \neq y \tag{5.33}$$

$$PBS(x) \leftrightarrow Seq(x) \land \neg\exists y(sPPO(y,x)) \tag{5.34}$$

$$soverlap(x,y) \leftrightarrow \exists z(sPO(z,x) \land sPO(z,y)) \tag{5.35}$$

$$sdisjoint(x,y) \leftrightarrow \neg soverlap(x,y) \tag{5.36}$$

Sequences consist entirely of atoms with respect to the relation **sPO**. The following two axioms require that all sequences have atoms as part, and that they are constituted of only atoms.

$$Seq(x) \rightarrow \exists y(PBS(y) \land sPO(y,x)) \tag{5.37}$$

$$Seq(x) \rightarrow \neg\exists y(sPPO(y,x) \land \forall u(sPPO(u,x) \land PBS(u) \rightarrow sPO(u,y))) \tag{5.38}$$

The relation **sPO** satisfies both the weak and the strong supplementation principles [Guizzardi, 2005].

$$sPPO(x,y) \rightarrow \exists z(sPO(z,y) \land sdisjoint(z,x)) \tag{5.39}$$

$$\neg sPO(x,y) \rightarrow \exists z(sPO(z,x) \land sdisjoint(z,y)) \tag{5.40}$$

Next, I restrict the arguments for the **between** and **end** relation, and introduce the relation **in** through an explicit definition.

$$between(j,p_1,p_2,s) \rightarrow Jun(j) \land PBS(p_1) \land PBS(p_2) \land Seq(s) \tag{5.41}$$

$$end(j,p,s) \rightarrow Jun(j) \land PBS(p) \land Seq(s) \tag{5.42}$$

$$conn(j_1,j_2) \rightarrow Jun(j_1) \land Jun(j_2) \tag{5.43}$$

$$in(j,s) \leftrightarrow \exists p_1,p_2(between(j,p_1,p_2,s)) \lor \exists p(end(j,p,s)) \tag{5.44}$$

The following set of axioms pertains to the **conn** relation of connectedness between junctions. It is used to represent the order of the sequence through an order of junctions. The following axioms hold for the **conn** relation:

$$conn(j_1, j_2) \rightarrow conn(j_2, j_1) \tag{5.45}$$

$$conn(j_1, j_2) \rightarrow j_1 \neq j_2 \tag{5.46}$$

$$in(j_1, s_1) \wedge in(j_2, s_2) \wedge \neg soverlap(s_1, s_2) \rightarrow \neg conn(j_1, j_2) \tag{5.47}$$

$$conn(j_1, j_2) \wedge in(j_1, s) \rightarrow in(j_2, s) \tag{5.48}$$

The axioms presented here are mostly first-order axioms and do not suffice to require connectedness of sequences. Instead, a second-order axiom is require to express the fact that sequences must be connected:

$$\begin{gathered}\forall s \forall P(\forall x(P(x) \leftrightarrow in(x,s)) \wedge \forall Q(\exists a Q(a) \wedge \forall x(Q(x) \rightarrow P(x)) \wedge \\ \forall u,v(Q(u) \wedge conn(u,v) \rightarrow Q(v)) \rightarrow \forall x(P(x) \rightarrow Q(x))))\end{gathered} \tag{5.49}$$

The following axioms pertain to between and end, and entail that junctions belong to exactly one sequence.

$$between(j, p_1, p_2, s) \rightarrow between(j, p_2, p_1, s) \tag{5.50}$$

$$\begin{aligned} between(j, p_1, p_2, s) \wedge between(j, p'_1, p'_2, s') \rightarrow & \\ & ((p_1 = p'_1 \wedge p_2 = p'_2) \vee (p_1 = p'_2 \wedge p_2 = p'_1)) \\ & \wedge soverlap(s, s') \end{aligned} \tag{5.51}$$

$$end(j, p, s) \wedge end(j, p', s') \rightarrow p = p' \wedge soverlap(s, s') \tag{5.52}$$

$$end(j_1,p,p) \wedge end(j_2,p,p) \wedge end(j_3,p,p) \wedge j_1 \neq j_2 \rightarrow j_3 = j_1 \vee j_3 = j_2 \tag{5.53}$$

A model of the theory proposed so far is illustrated in figure 5.1.3. After a discussion of the sequences, which are abstract entities, i.e., outside of space and time, it is important to consider the token level, i.e., molecules that exhibit sequential structure. The following axioms relate tokens to sequences. There are many sequences for which no token exists, and this fact is represented in the following axioms.

The first axioms belonging to tokens of sequences pertain to the **PO** (part-of) relation between molecules. The relation **PO** satisfies the axioms for a partial order, reflexivity, transitivity and antisymmetry.

$$PO(x,y) \wedge PO(y,z) \rightarrow PO(x,z) \tag{5.54}$$

$$Mol(x) \rightarrow PO(x,x) \tag{5.55}$$

$$PO(x,y) \wedge PO(y,x) \rightarrow x = y \tag{5.56}$$

Next I define the relation **PPO** (proper part of), **overlap**, **disjoint** and the cate-

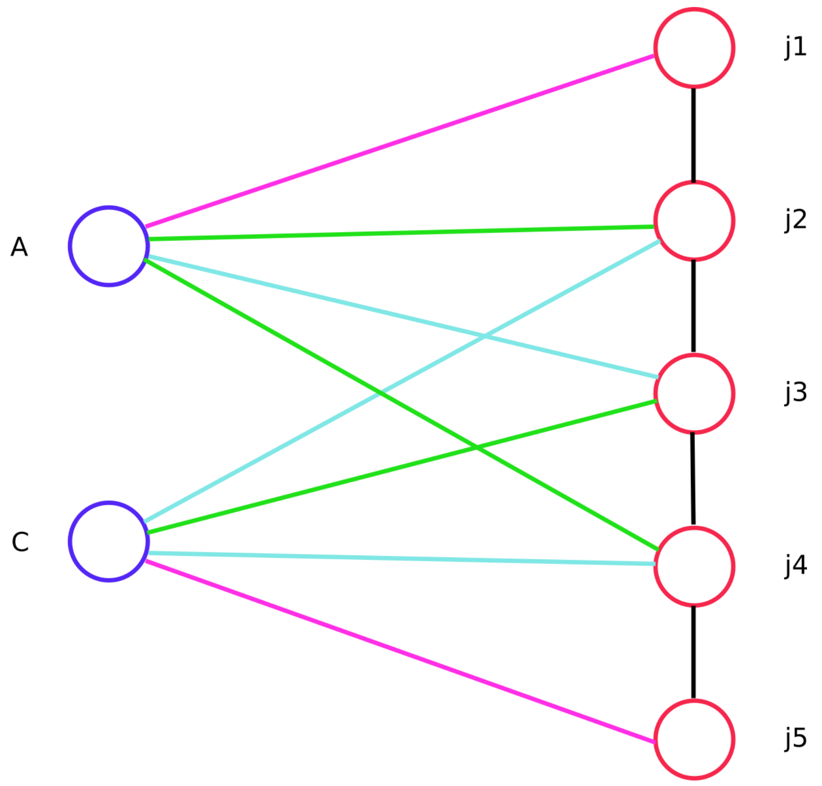


Figure 5.4: The representation of the sequence *ACAC* using the sequence module of GFO-Bio. Nodes in blue color represent primitive biological symbols (*PBS*), nodes in red color represent *Junction* categories. Black edges between junction-nodes denote the **conn** relation. A purple edge between a junction and a *PBS* node stands for an **end** relation between the junction, the PBS and some sequence. A green or cyan edge stands for a **between** relation, where the junction occurrs as second (green) or third (cyan) argument.

gory *Atom*[4].

$$PPO(x,y) \leftrightarrow PO(x,y) \wedge \neg PO(y,x) \tag{5.57}$$

$$overlap(x,y) \leftrightarrow \exists z(PO(z,x) \wedge PO(z,y)) \tag{5.58}$$

$$disjoint(x,y) \leftrightarrow \neg overlap(x,y) \tag{5.59}$$

$$At(x) \leftrightarrow Mol(x) \wedge \neg \exists y(PPO(y,x)) \tag{5.60}$$

The token of sequences consist entirely of atoms, and every sequence has an atom as part.

$$Mol(x) \rightarrow \exists y(At(y) \wedge PO(y,x)) \tag{5.61}$$

$$Mol(x) \rightarrow \neg \exists y(PPO(y,x) \wedge \forall u(PO(u,x) \wedge At(u) \rightarrow PO(u,y))) \tag{5.62}$$

For molecules, the tokens of sequences, both the weak and the strong supplementation principle holds. Axiom 5.63 states the weak supplementation principle, 5.64 the strong. Axiom 5.63 is a consequence of 5.64.

$$PPO(x,y) \rightarrow \exists z(PO(z,y) \wedge disjoint(z,x)) \tag{5.63}$$

$$\neg PO(x,y) \rightarrow \exists z(PO(z,x) \wedge disjoint(z,y)) \tag{5.64}$$

The correspondence between sequence atoms and token atoms is stated in the following axioms. Every token atom is instance of exactly one primitive biological symbol, and every instance of a primitive biological symbol is a token

[4] An atom is a primitive token of a sequence, and must not be identified with an atom in the physical sense. The category represents mereological atoms with respect to the relation **PO**. Therefore, every (mereological) atom is a molecule.

atom.

$$Seq(a) \wedge x :: a \wedge At(y) \wedge PO(y,x) \rightarrow \exists(=1,b)(sPO(b,a) \wedge PBS(b) \wedge y :: b) \tag{5.65}$$

$$PBS(x) \wedge a :: x \rightarrow At(a) \tag{5.66}$$

The next axioms restrict the **binds** relation, which is a relation between token atoms and it resembles the **conn** relation on the token level. **binds** satisfies anti-reflexivity[5] and symmetry.

$$binds(x,y) \rightarrow At(x) \wedge At(y) \tag{5.67}$$

$$binds(x,y) \rightarrow \exists u,v(PBS(u) \wedge PBS(v) \wedge x :: u \wedge y :: v) \tag{5.68}$$

$$\neg binds(x,x) \tag{5.69}$$

$$binds(x,y) \rightarrow binds(y,x) \tag{5.70}$$

The following set of axioms enforce that sequences are linear. Every token atom in a non-primitive sequence binds to at least one (5.72) and at most two (5.71) other token atom. Further, at most two token atoms bind to exactly one other token atom (5.73). Finally, in every proper sequence (i.e., not a primitive biological symbol), either all token atoms bind to exactly two other token atoms, or exactly two atoms bind to exactly one and the rest to exactly two token atoms (5.80).

$$\begin{split} Seq(x) \wedge PBS(y) \wedge sPO(y,x) \rightarrow & \forall a,b(a :: x \wedge b :: y \wedge PO(b,a) \rightarrow \\ & \exists(\leq 2,u)(binds(u,b))) \end{split} \tag{5.71}$$

[5] This axiom excludes circles of size 1 on the token-level.

$$Seq(x) \wedge \neg PBS(x) \wedge a :: x \rightarrow \forall b(PO(b,a) \wedge At(b) \rightarrow \exists c(binds(b,c))) \quad (5.72)$$

$$Seq(x) \wedge a :: x \rightarrow \exists(\leq 2, u)(partOf(u,a) \wedge \exists(=1,v)(binds(u,v))) \quad (5.73)$$

Sequences are either linear or circular. First, I define the two categories *CSeq* and *LSeq*, and axiom 5.80 states that sequences are either circular or linear.

$$\begin{aligned} CSeq(x) \leftrightarrow & Seq(x) \wedge \neg PBS(x) \wedge \forall a,b(a :: x \wedge \\ & PO(b,a) \wedge At(b) \rightarrow \exists(=2,c)(binds(b,c))) \end{aligned} \quad (5.74)$$

$$CSeq(x) \rightarrow \neg \exists j,p(in(j,x) \wedge end(j,p,x)) \quad (5.75)$$

$$CSeq(x) \wedge in(j,x) \rightarrow \exists p_1, p_2(between(j,p_1,p_2,x)) \quad (5.76)$$

$$\begin{aligned} LSeq(x) \leftrightarrow & PBS(x) \vee (Seq(x) \wedge \forall a(a :: x \rightarrow \exists(=2,b)(PO(b,a) \wedge \\ & At(b) \wedge \exists(=1,c)(binds(b,c) \wedge \forall d(d \neq b \wedge PO(d,a) \wedge \\ & At(d) \rightarrow \exists(=2,e)(binds(d,e))))))) \end{aligned} \quad (5.77)$$

$$LSeq(x) \rightarrow \exists(=2, j)\exists p(end(j, p, x)) \tag{5.78}$$

$$LSeq(x) \wedge in(j, x) \wedge \neg\exists p(end(j, p, x)) \rightarrow \exists p_1, p_2(between(j, p_1, p_2, x)) \tag{5.79}$$

$$Seq(x) \rightarrow CSeq(x) \vee LSeq(x) \tag{5.80}$$

The following axioms states the existence of single tokens, i.e., tokens that bind to no other entity.

$$\exists x(Mol(x) \wedge \neg\exists y(binds(x, y))) \tag{5.81}$$

The relation between connectedness of junctions and the **between** and **end** relation is expressed in these axioms:

$$end(j, p, s) \rightarrow \exists(=1, j')(conn(j, j')) \tag{5.82}$$

$$between(j, p_1, p_2, s) \rightarrow \exists(=2, j')(conn(j, j')) \tag{5.83}$$

The last axioms establishes a relation between the connectedness within sequences, i.e., connectedness between junctions, and the **binds** relation on the token level.

$$between(j, p_1, p_2, s) \wedge m :: s \rightarrow \exists a_1, a_2 (At(a_1) \wedge At(a_2) \wedge PPO(a_1, m) \wedge PPO(a_2, m) \wedge binds(a_1, a_2)) \tag{5.84}$$

The axioms presented here specify two of the basic categories used in the Sequence Ontology, *Junction* and *Sequence* (or sequence region). To analyze the remaining basic categories, *Sequence operation* and *Sequence collection*, additional relations and categories must be introduced. Collections can be analyzed using set theory, i.e., as sets of sequences. Operations can be defined using the two parthood relations introduced in this theory. In the next section, I will show how to use the theory proposed here as a foundation for the sequence ontology.

I implemented the axiom system using the SPASS first-order theorem prover [Weidenbach et al., 2002]. The implementation can be found in appendix A and on the project webpage [Hoehndorf, 2009]. Due to the restriction of SPASS to first-order logic, I could not implement the axiom requiring connectedness of sequences. This axioms necessitates the use of monadic second-order logics. Furthermore, a condition that sequences must be finite could not be implemented due to the restrictions of first order logic.

I employed the SPASS theorem prover on the sequence axioms and attempted to prove the proposition $\phi \wedge \neg\phi$. If this logical contradiction can be derived from the axioms, the axioms would be inconsistent. On the other hand, if the axioms are consistent, SPASS should never terminate, because, in the general case, an automated consistency proof for first-order theories is impossible [Church, 1936]. The SPASS theorem prover could not find a proof for the contradictory statement $\phi \wedge \neg\phi$ in three weeks time. While this is merely an indication for consistency, it indicates at least the absence of trivial inconsistencies and permits the use of the axiom system for inferences.

5.1.4 Integrating domain ontologies with GFO-Bio

There are two aspects of integrating biomedical domain ontologies using GFO-Bio that I address here. The first aspects is technical and show how to use the OWL version of GFO-Bio for integrating the OBO Flatfile or OWL versions of the domain ontologies. The second aspect shows how to use the axioms presented here to analyze the domain ontologies and provide a foundation for them.

Technical aspects of ontology integration using GFO-Bio

Integrating biological ontologies using GFO-Bio involves several steps. First, an OWL version of each ontology must be aquired or produced. OWL-DL is a sufficiently expressive language because negation is available and logical inconsistencies can be formally detected in the OWL-DL framework. For the purpose of this conversion, we provide a tool [Hoehndorf et al., 2008a] that converts OBO Flatfile Format files [Golbreich and Horrocks, 2007] into OWL-DL. The generated OWL-DL file must then be imported by GFO-Bio. Each top-level class of the imported ontology is then defined, at least partially, using categories from GFO-Bio's *Individual* tree. For example, the *Cell* category of the Celltype Ontology [Bard et al., 2005] must be declared a sub-category of (or an equivalent of) GFO-Bio's *Cell* category.

Further, a second OWL-DL file can be produced for each integrated ontology containing the ontology's categories as instances of GFO-Bio's category branch. We also provide a tool for performing this conversion for OBO files [Hoehndorf et al., 2008a]. This file must be imported by GFO-Bio as well. In this file, relationships between categories, as directly expressed in the OBO-style directed acyclic graphs (DAGs), are modelled as relationships between OWL instances.

For example, the relationship expressed in the DAG of the Gene Ontology's cellular component ontology, *Membrane* **part-of** *Cell*, is represented twice in GFO-Bio: First, *Membrane* and *Cell* are created as classes in OWL, and the following restriction created (in line with [Golbreich and Horrocks, 2007]):

```
 SubClassOf(Membrane restriction(II-part-of
                       someValuesFrom(Cell)))
```

In addition, the Gene Ontology's *Cell* category is declared equivalent to GFO-Bio's *Cell* category. Second, *Membrane* and *Cell* are treated as instances of GFO-Bio's *Category* class, and a relation **CC-part-of** ('CC' indicating the category–category reading of the relation) between *Membrane* and *Cell* is asserted:

```
Individual(Membrane value(CC-part-of Cell))
```

While neither the first nor the second step alone require more than the description logic fragment of OWL, in conjunction they result in an OWL-Full [Mcguinness and van Harmelen, 2004] ontology.

Ontological analysis of domain ontologies

The OBO and OBO Foundry ontologies satisfy the criterion of orthogonality. In addition, the OBO Foundry ontologies satisfy **is-a**-completeness, i.e., they contain one or only very few top-level categories, and all other categories are sub-categories of these. As a consequence, only these top-level categories have to be considered and ontologically analyzed.

GFO-Bio provides categories that can be used to *define* the top-level categories of biomedical domain ontologies. In many cases, a category that corresponds directly to the top-level category of a domain ontology is already defined in GFO-Bio. For example, the Sequence Ontology (SO) contains four top-level categories, *Sequence*, *Junction*, *Sequence collection* and *Sequence operation*. The first two are already contained in GFO-Bio, and are equivalent to the categories in the SO. *Sequence collection* can be defined using GFO-Bio's and GFO's categories:

$$x :: SeqColl \leftrightarrow x :: Set \wedge \forall y (hasMember(x,y) \rightarrow y :: Seq) \tag{5.85}$$

The category of *Sequence operation*s can be defined using GFO's *Relator* category. Sequence operations are relators with four arguments: a sequence on which the operation is performed, the sequence which is inserted or deleted, the junction at which the insertion takes place or at which the deletion took place, and the sequence that results from the operation. Additional axioms based on the relations for sequences introduced in GFO-Bio must be given to formalize the specific operations, i.e., deletion, insertion and modification.

Finally, not all domain categories can be defined using categories from GFO or GFO-Bio. For example, the category *Mouse* is not included in GFO-Bio. However, GFO-Bio contains *Organism*. An axiom such as 5.86 can be used to partially constrain the *Mouse* domain category.

$$x :: Mouse \rightarrow x :: Organism \tag{5.86}$$

5.1.5 Comparison with other biological upper domain ontologies

There are at least two upper biological ontologies aside from GFO-Bio, and several other projects that overlap in part with GFO-Bio. The ontologies BioTop [Schulz et al., 2006b] and the Simple Bio Upper Ontology (SBUO) [Rector et al., 2006b] are upper ontologies for the biological and biomedical domains. They provide well-defined categories that can be used to classify individuals. The main differences between GFO-Bio and alternative approaches are in a large part due to the properties of GFO-Bio's top-level ontology GFO: including higher-order categories, treating semiotic information, bridging levels of granularity and integrating objects and processes.

BioTop and the SBUO are biological core ontologies based on, or inspired by, the foundational ontologies BFO [Grenon, 2003b] and DOLCE [Masolo et al., 2003]. Neither, therefore, includes higher-order categories. Higher-order categories are used in GFO-Bio to model biological taxa, symbols and sequences, model persistence through time and explicate the intension of the relations used in biomedical ontologies.

BioTop contains explicit categories for biological sequences and symbols. In GFO-Bio, sequences are categories that can have instances (the tokens). They are entities *sui generis* and do not depend on any other entity, whereas in the BioTop ontology, they are generically dependent[6] continuants that depend on the existence of a molecule. For example, the instance of a DNA sequence in BioTop requires the existence of some DNA molecule that exhibits this sequential structure. However, the sequences used in biological research are not

[6] A category *C* is generically dependent on the category *D*, if, necessarily, whenever an instance *c* of *C* exists, then some instance *d* of *D* exists.

always the sequence of some molecule. It is unlikely that the *canonical* sequence of human chromosome 5 is exhibited by any DNA molecule, due to sequencing errors, the presence of mutations, variations or similar. It is not clear how sequencing errors, variations or mutations are represented in BioTop. The same holds for randomly or artificially created sequences that are studied as entities in their own right. The theory of biological sequences contained in GFO-Bio is capable of representing such sequences.

Biological taxa are higher-order categories in GFO-Bio and are related *via* the **instance-of** relation. This is due to the inclusion of higher-order categories in the GFO, and permits the representation of information pertaining to taxa without introducing an additional relation. In BioTop, several approaches to representing biological taxa are considered [Schulz et al., 2008]. Most of these can be adapted to GFO-Bio as well. The form of representing taxa in GFO-Bio is similar to the representation of taxa as meta-properties as discussed in [Schulz et al., 2008]. There, this form of representation was rejected, because it leads to undecidability when meta-properties are introduced via predication, i.e., as genuine higher-order predicates. Due to the inclusion of **instance-of** as a relation in the GFO, in contrast to the identification of instantiation with predication, this problem does not arise in GFO-Bio.

5.2 Representation of functional knowledge

One complex module of GFO-Bio is an ontological model of functions. In this chapter I will discuss what a biological function is, and how the approach taken by the GFO fits in this discussion.

The seminal paper that proposed a solution to the problem of functions in biology is Larry Wright's article on functions [Wright, 1973]. Although many extensions and alternatives have been proposed, it remains the central article when the concept of biological function is discussed. I will first give a summary of Wright's article, then discuss extensions of his work, most notably by Ruth Millikan [Millikan, 1988], and present an alternative view due to John Searle [Searle, 1997], before I propose my own account.

5.2.1 Wright on functions

The basic definition that Larry Wright gave on functions is:

Definition 1. *The function of X is Z means*

1. *X is there because it does Z,*
2. *Z is a consequence (or result) of X's being there.*

For example, "the function of the heart is to pump blood" means that (1) the heart is present (now) because it pumps blood (and pumped blood in the past), and (2) that the pumping of blood is a consequence of the presence of the heart. The first part of this definition explains why hearts are present now (because they pump blood, and pumped blood in the past). The second part explains the causal relation between hearts and the pumping of blood (hearts cause blood to be pumped, in the right circumstances). This definition allows the answer of two questions: *why* are hearts there, and *why* is blood being pumped. Wright emphasizes this explanatory power of statements like X has the function to Z.

Ruth Millikan extended Wright's function definition [Millikan, 1988]. Her definition is a biological one, borrowing many terms from evolutionary theory. Her definition of a proper function is as follows:

Definition 2. *Where m is a member of a reproductively established family R and R has the reproductively established or Normal character C, m has the function F as a direct proper function iff:*

1. *Certain ancestors of m performed F.*
2. *In part because there existed a direct causal connection between having the character C and performance of the function F in the case of these ancestors of m, C correlated positively with F over a certain set of items S which included these ancestors and other things not having C.*
3. *One among the legitimate explanations that can be given of the fact that m exists makes reference to the fact that C correlated positively with F over S, either directly causing reproduction of m or explaining why R was proliferated and hence why m exists.*

This restricts the definition of Wright in an important sense. Consider the following example (taken from Boorse [1976], Smith [1993]): A small rock holds a large rock in the river. If the small rock would not hold the large rock, it would be washed away. Therefore, the small rock is there because it holds the large rock, and holding the large rock is a result of the small rock's being there. According to Wright's definition, holding the large rock would be the small rock's function. In Millikan's approach this is immediately avoided due to the requirement that the rock belongs to a lineage of entities that is created by reproduction and replication. Therefore, Millikan's account on function is closer to biology in the sense that it is defined using central biological notions.

5.2.2 Searle on Functions

In contract to causal explanations of function, John Searle defends an inherently social view of functions. In [Searle, 1997], John Searle describes an account of functions that differs fundamentally from the account given by Wright or Millikan. Accordings to Searle, functions are never intrinsic of any entity, but are ascribed to entities from the outside by a conscious observer. Functions are therefore always *observer-relative*. When the function of the heart ("to pump blood") was discovered, it was in fact the discovery of a causal process in which the heart played a specific role (the *brute fact* underlying social ascription). This process is then situated against a system of values, intentions and beliefs of the observer, and by this it is assigned a teleology. While there are many causal processes the heart is involved in (e.g., creating thumping noises), assignment of function selects one or some of these causal processes and situates it against a system of background values and intentions: pumping of blood contributes to survival, and survival of an organism is *good* with respect to the values held by the observer; pumping of blood *explains* the development and presence of the heart best with respect to current scientific knowledge and theories. While causal facts are observer-independent (*brute*), functional facts are always dependent on an observer.

However, this does not imply that functions do not have a causal component or causal implications. Social ascriptions are not arbitrarily made. This is especially the case in a field like biology, where functions play a central role in scientific theories and have specific meanings. They are used in causal explanations and a statement about functions conveys information about material, observer-independent phenomena.

The Ontology of Functions (OF) [Burek, 2006] is compatible with the theories of function presented by [Wright, 1973, Millikan, 1988, Searle, 1997]. I add several axioms to the OF that relate functions to structures and processes by means of causation. These should be considered minimal conditions on function. The theory of function presented here can be extended by a a more specific theory if desired.

5.2.3 Ontology of Functions

My colleague Patryk Burek wrote a thesis about the Ontology of Functions (OF) [Burek, 2006]. He provided an account of how to represent functions in the top-level ontology GFO, and how to represent their relations to other entities within GFO. An overview of the basic concepts introduced in the OF is presented in figure 5.5. His basic assumption is that functional knowledge can be represented and described independently of the realization of function. In the OF, a function structure is described by a label, requirements, a goal and a functional item. The label is a non-formal name or description of the function. The requirement is a situation type T_{req} that must be realized for every realization of the function. The goal is a situation type T_{goal} that describes the state of the world that the function is supposed cause or otherwise bring about. The functional item is a *view* on the entities that can have the functions.

For example, the function F to transport goods G from A to B is described as follows:

- $label(F) =$ "to transport goods G from A to B"
- $T_{req}(F) = \{s|s \models \{< located - at, G, A; 1 >\}\}$
- $T_{goal}(F) = \{s|s \models \{< located - at, G, B; 1 >\}\}$
- $FI(F) =$ *transporter*, where *transporter* is a category that has as instances all entities capable of *transporting* things, i.e., a *view* on all transporters specifying all and only the necessary properties of an entity that enables it to transport things.

In contrast to Burek [2006], I use notation from situation theory [Barwise and Perry, 1983] for the description of functions. Situation theory provides a formalism for modelling situations as "parts of the world that can be comprehended as a whole" [Devlin, 1991]. Situation theory is a novel but well-studied

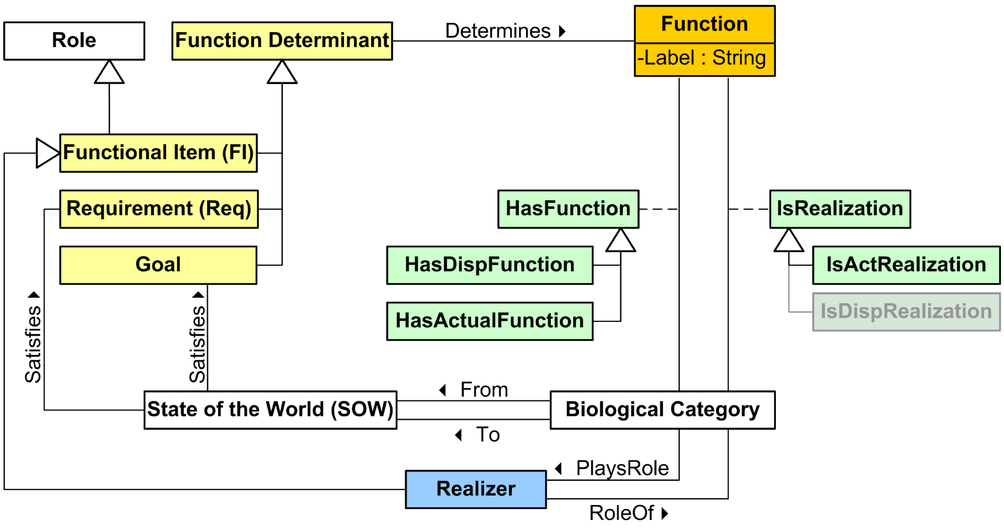


Figure 5.5: A schematic representation of the concepts used and introduced by the OF (using the Unified Modeling Language [OMG, 2006]). Unlabelled relations indicate generalizations, where large arrowheads point at the more general concept. Functions (the orange box) are determined by entities indicated in yellow: a goal, requirements, and a functional item. A biological category may be related to a function in two ways (cf. the green boxes which provide labels for those relations connected to them by a dashed line): its instances may **realize** the function or they may **have** the function. A biological entity (such as a process) is a realization of a function if it mediates between two states of the world, one satisfying the requirements, the other satisfying the goal. A realizer in the OF, presented in blue, is the role played by an entity in a realization. In the function this role is determined by the functional item, hence realizer is generalized by functional item. Biological categories whose instances can play the role defined by the functional item have the function. The **has-function** relation relates biological categories with functions if every instance of this category has the actual or dispositional function.

form of logic. Addtionally, situation theory is compatible with the GFO [Herre and Heller, 2005, Herre et al., 2006, Hoehndorf, 2005]. The notion of situation is relevant in the OF [Burek, 2006] and permits an ontological understanding of *goals* and *requirements*.

I model the requirements and the goal of the function as situation types and their realizations (instances) as situations. The advantage of using situation theory over the approach in the originial OF is that situation theory provides a specification language for requirements and goals. Additionally, I analyze the functional item differently and I consider functions as a special kind of properties. This entails that functions are individuals that inhere in their bearers, and it becomes important to distinguish between an individual function and a function category.

$$isa(Function, Property) \tag{5.87}$$

Next I extend the notion of requirement, goal and functional item to individual functions. These definitions may appear awkward, because they relate a concrete individual – the function individual – to three categories (and not other individuals). The reason for this is that a function is in many respects a kind of *disposition*, the possibility to bring another entity into being without actually doing it. Functions can also have *realization*, i.e., make some entities *real*. For example, the function “to transport sugar” will be realized by sugar transport processes.

I will express realization in terms of instantiation. Also, functions can be realized multiple times, and what makes a process a realization of a particular function is that is starts and ends with situations of specific kinds. Therefore, an individual function contains as requirements, goal and functional item categories. These categories are then instantiated in every realization of the function.

$$\begin{aligned} f :: F \wedge isa(F, Function) \rightarrow & \exists t_1 (t_1 = T_{req}(F) \wedge isa(t_1, Category) \wedge \\ & \forall x (x :: t_1 \rightarrow x :: Situation)) \end{aligned} \tag{5.88}$$

$$\begin{aligned} f :: F \wedge isa(F, Function) \rightarrow & \exists t_2 (t_2 = T_{goal}(F) \wedge isa(t_2, Category) \wedge \\ & \forall x (x :: t_2 \rightarrow x :: Situation)) \end{aligned} \tag{5.89}$$

$$f :: F \wedge isa(F, Function) \rightarrow \exists i (i = FI(F) \wedge isa(i, Category)) \tag{5.90}$$

A realization of the function $f :: F$ is a transition from a situation $s :: T_{req}(F)$ to a situation $t :: T_{goal}(F)$. Commonly, this transition is a process. This process starts with a situation of type $T_{req}(F)$, and ends with a situation of type $T_{goal}(F)$, formally:

$$realizes(x, f) \rightarrow x :: ProcessualStructure \wedge f :: Function \tag{5.91}$$

$$\begin{aligned} realizes(x, f) \wedge f :: F \wedge req(F) = T_{req} \wedge goal(F) = T_{goal} \rightarrow \\ \exists s_1, s_2 (starts(s_1, x) \wedge ends(s_2, x) \wedge \\ s_1 :: T_{req} \wedge s_2 :: T_{goal}) \end{aligned} \tag{5.92}$$

This is a necessary, but not a sufficient condition for a process' being the realization of a function. For a process to be a realization of a function, the function bearer must be active in the realization in a particular way: the function bearer must play the role of the functional item of the function.

The *realizer* of the function is the (individual) role that is played by the entity that causes the goal of the function. The realizer is an instance of the functional item. The following axiom establishes a relation between function realization and causation: the entity that plays the realizer in a function realization causes the goal of the function.

$$\begin{gathered} realizerOf(x,F) \rightarrow x :: Role \wedge x :: FI(F) \wedge \exists z(realizes(z,F) \wedge \\ roleOf(x,z)) \wedge \forall y(plays(y,x) \rightarrow \exists s(s :: T_{goal} \wedge causes(y,s))) \end{gathered} \tag{5.93}$$

The use of causality in this context is controversial, in particular as realizations of functions may involve intentional acts that cannot be reduced to causality. In biology, functions may be realized by behaviour and therefore not by purely causal processes. For the purpose of a general theory of functions, the use of causality here must be carefully examined. Additionally, the notion of causality that is used must be formally analyzed. Here I do not, however, analyze the notion of causality further. For a treatise of causality in the GFO, see [Michalek, 2009].

A processual structure p is a realization of the function $f :: F$, if the realizer role is played by the bearer of f in p.

$$\begin{gathered} realizerOf(x,f) \wedge plays(y,x) \wedge hasFunction(y,F) \wedge roleOf(x,z) \rightarrow \\ realizes(z,f) \end{gathered} \tag{5.94}$$

The dependent nature of function can be analyzed in an axiom requiring functions to have at least one bearer[7]:

$$x :: Function \rightarrow \exists y(hasFunction(y,x)) \tag{5.95}$$

However, since functions are a special kind of property, an additional axiom is the following, declaring the *hasFunction* relation as a subrelation of the *has*-

[7] This axiom is redundant, because functions are considered properties, which are already dependent on their bearer. I include this axiom for comprehensibility.

$Property$ relation:

$$hasFunction(x, f) \rightarrow hasProperty(x, f) \tag{5.96}$$

Theories on function differ in how they analyze the **has-function** relation. Burek [2006] follows [Searle, 1997] in assuming that functions are socially ascribed, i.e., come into being through conventions within a social context. Following this theory, functions are always observer-dependent: they come into being through a relation between an object and a concious observer. In an important sense, both [Burek, 2006] and [Searle, 1997] end their analysis with this observation: after the rejection of the causal theories of function [Wright, 1973, Millikan, 1988], they do not analyze whether some aspects of these theories remain valid within their analysis of function. Social ascription is both a necessary and sufficient condition in [Burek, 2006, Searle, 1997].

I am not convinced that this is the case. Functions are not arbitrarily ascribed to entities. I believe that there are causal properties that must necessarily be exhibited before a function is ascribed to some entity by an agent. In particular, the entity to which the function is ascribed must *normally* be able to **cause** the goal of the function given the requirements of the function.

Inspired by Hartmann [1966], I analyze three conditions that are necessary for some entity to obtain the function F with the goal T_{goal}:

1. An agent establishs a goal T_{goal} which lies in the future. This first step requires free movement in the *Anschauungszeit*[8], because it establishs a future goal. Setting this goal belongs to the mental stratum of reality, where free movement in time is possible. In particular, this step cannot

[8] *Anschauungszeit* is literally translated as “viewed-upon time”. It is time as perceived by a mind. In particular, in *Anschauungszeit*, i.e., for a mind, it is possible to move freely forward and backward in this time.

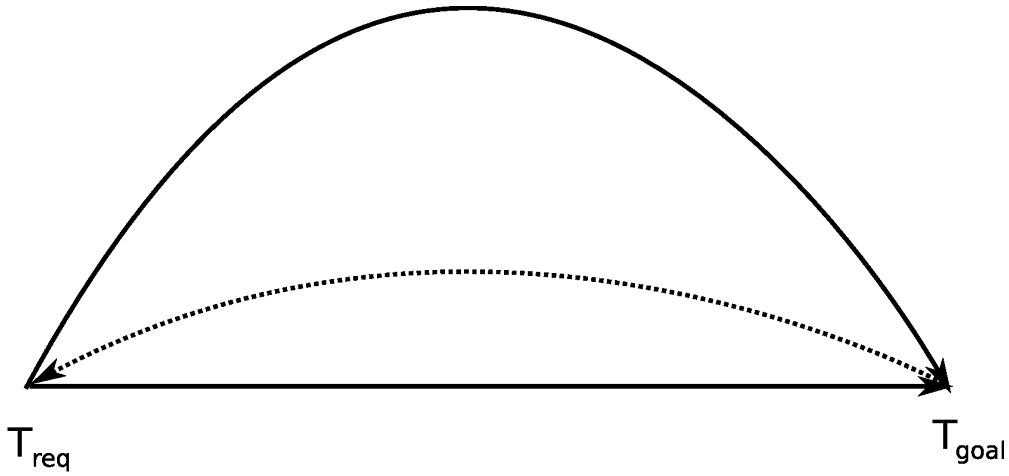


Figure 5.6: Three conditions for function ascription. First, a goal is established *in the future*. Second, the means for achieving the goal are selected or created. Finally, the goal is realized by causal means.

be performed in material reality alone. This establishment of the goal by a mind is also the source of the intensionality[9] and referential opacity[10] of statements pertaining to functions [Searle, 1997].

2. The agent generates a plan on how to achieve the goal. For this purpose, the agent goes backward in *Anschauungszeit* starting at the time of the goal and ending at the present time. This planning or design process is directed backward in time. It is this "going backward in time" from the goal to the present which determines a process – the realization of the function, once it is established – from its end, and therefore making them telologic in nature. The result of the second step, if successful, is the establishment of a structure or situation that is able to **cause** the goal established in the first step.

[9] Intensionality is the opposite of extensionality.

[10] A term *t* is referentially opaque in a statement *C*, if *t* cannot be replaced with a co-referential term *s* in *C* without changing the truth-value of *C*.

3. The final component is the causal process which starts at the present and ends with the achievement of the goal (i.e., reaching a situation which instantiates T_{goal}). Because the third condition requires the causal determination of the process, it is impossible to see on this process alone whether it is only causally determined or determined both causally and finally.

I believe these three conditions are required for any function ascription. The second condition may be replaced by convention, when entities of a certain kind are generally known to being able to cause a goal. Then an agent may establish a goal and use some object (causally) to achieve this goal, without constructing the object that is supposed to cause the goal. There is still a planning involved, however, as an object must be chosen to bring about the goal. The second condition explains why an object that has a function F with a goal T_{goal} has a specific structure: it must be able to cause the goal, and its structure may be a result of the planning process with the aim to cause the goal established as the first condition.

These observations shed light on *malfunctioning*s as well. An entity is malfunctioning when it is intended to cause a processes that ends in a goal of the kind T_{goal}, but cannot cause this process. When an entity is malfunctioning, the first and second conditions remain valid, but the third fails.

To formalize these observations, I introduce an additional entity in the ontology of functions. I call this a *disposition*. An individual e has the disposition d to cause or achieve T_{goal} iff e causes a situation $s :: T_{goal}$ to become realized whenever e is placed *in the right circumstances*. I model "being placed in the right circumstances" using a situational role[11] [Loebe, 2007] and an additional

[11] A situational role is the role that an entity plays in a complex situation. If situations are not permitted in the ontology, they can be considered a complex state of affairs, i.e., a complex of instances of relations (complex relator). In this view, a situational role is a (complex)

universal. This universal identifies the structural features of the entity with the disposition that are necessary to realize the disposition.

The terminology to describe a disposition category D is:

- $T_{req}(D)$ is the requirement of D,
- $T_{goal}(D)$ is the goal of D,
- $R(D)$ is a situation role and
- $U(D)$ is a category of material objects (e.g., a category with objects as instances, defined by its parts and structural connections between these parts).

The following axioms hold for dispositions:

$$isa(Disposition, Property) \tag{5.97}$$

$$isa(T_{req}(D), Situation) \tag{5.98}$$

$$isa(T_{goal}(D), Situation) \tag{5.99}$$

$$isa(R(D), SituationalRole) \tag{5.100}$$

$$isa(U(D), MaterialObject) \tag{5.101}$$

$$d :: D \wedge isa(D, Disposition) \wedge inheresIn(d,e) \leftrightarrow e :: U(D) \tag{5.102}$$

$$\begin{aligned} & d :: D \wedge isa(D, Disposition) \wedge inheresIn(d,e) \rightarrow \\ & \quad (plays(e,r) \wedge r :: R(D) \wedge roleOf(r,s) \wedge s :: T_{req} \rightarrow \\ & \quad \exists p,t (p :: Process \wedge t :: T_{goal}) \wedge \\ & \quad starts(s,p) \wedge ends(t,p) \wedge causes(s,p)) \end{aligned} \tag{5.103}$$

$$d :: D \wedge isa(D, Disposition) \wedge e :: U(D) \rightarrow inheresIn(d,e) \tag{5.104}$$

relational role.

Finally, a relation must be established between functions and dispositions, so that assertions about functions permit inferences about causal relations. I consider two possibilities to create such a relation. The first is to require that functions are subclasses of dispositions, and the requirement and goal of the function are the requirement and goal of the disposition. Then, every function $f :: F$ with goal $T_{goal}(F)$ and requirement $T_{req}(F)$ is a disposition with goal $T_{goal}(F)$ and requirement $T_{req}(F)$:

$$isa(Function, Disposition) \tag{5.105}$$

The difficulty with this approach lies in the treatment of malfunctionings. I consider an entity to be malfunctioning when it has a function $f :: F$ but does not have the disposition to cause $T_{goal}(F)$, i.e., is unable to realize the function. If functions are sub-categories of dispositions, it is not possible to assert that an entity has a function, but is malfunctioning. Instead, it must be denied that the entity has the function if it cannot cause the goal of the function.

Therefore, the second approach I suggest treats functions and dispositions as ontologically different entities, i.e., as disjoint categories, and establishes a relation between them explicitly. I suggest that every entity which has a function $f :: F$ with requirement $T_{req}(F)$ and $T_{goal}(F)$ *normally* has a disposition $d :: D$ with $T_{req}(D) = T_{req}(F)$ and $T_{goal}(D) = T_{goal}(F)$. Formally, I first define the formula $A(malfunctioning)$:

$$\begin{aligned} A(malfunctioning) = & hasFunction(e, f) \wedge f :: F \wedge \neg malfunctioning(e) \rightarrow \\ & \exists d(d :: D \wedge isa(D, Disposition) \wedge inheresIn(d, e) \wedge \\ & T_{req}(F) = T_{req}(D) \wedge T_{goal}(F) = T_{goal}(D)) \end{aligned} \tag{5.106}$$

The predicate $malfunctioning$ is an abnormality predicate. In order to treat $A(malfunctioning)$ in formula 5.106 as a default, the extension of the *malfunctioning* predicate must be minimized in every model. This is achieved by circumscribing [Mccarthy, 1986] $malfunctioning$ in *A(malfunctioning)* using *predicate circumscription*. The circumscription of $malfunctioning$ in *A(malfunctioning)* is the second-order formula:

$$A(malfunctioning) \wedge \forall P((A(P) \wedge \forall x(P(x) \rightarrow malfunctioning(x))) \rightarrow (\forall x(P(x) \leftrightarrow malfunctioning(x)))) \tag{5.107}$$

Alternatively, a formula in default logic [Reiter, 1980] can be chosen to replace axiom 5.107. For this purpose, I first define $A'(F,e)$:

$$\begin{aligned} A'(F,e) = & \exists d(d :: D \wedge isa(D, Disposition) \wedge inheresIn(d,e) \wedge \\ & T_{req}(F) = T_{req}(D) \wedge T_{goal}(F) = T_{goal}(D)) \end{aligned} \tag{5.108}$$

Then, the following default holds as axiom:

$$\frac{hasFunction(e,f) \wedge f :: F / A'(F,e)}{A'(F,e)} \tag{5.109}$$

Either 5.107 or 5.109 are chosen as axioms for this theory of functions and dispositions.

The $malfunctioning$ predicate can be extended to a binary predicate that includes an additional function argument. Then, $malfunctioning(e,f)$ would have to be interpreted as “entity e is malfunctioning with respect to function f”.

The use of nonmonotonic reasoning in a top-level framework requires a justification. Ontologies are supposed to specify the *meaning* of terms in a vocabulary. Nonmonotonic logics and forms of reasoning like circumscription and default logic, on the other hand, are used to formalized knowledge that is *true by default*. These formalisms do not appear to be a suitable formalism to specify the meanings of terms. However, if functions are considered to be (socially or mentally) ascribed entities, and related to expectations and intentions, default knowledge can be used to approximate this state. In the case of ascription of functions, the meaning of an entity's having a function *is* that this entity *normally* brings about a goal in certain circumstances. The need for nonmonotonic reasoning in representing functioning and malfunctioning was already recognized by Mccarthy [1986].

As a corollary from the axioms presented here, malfunctioning entities either do not have a function anymore (5.105) or continue to have a function but not a corresponding disposition (5.107 or 5.109). Both proposals suggested as components of this theory permit infering causal relationships from assertions about functions. I believe this to be useful particularily in the biological and medical domain, where function ascription is commonly used to describe and infer causal relations.

Finally, I believe that a generic framework of functions comes to its end, and a more fine-grained, more restricted theory of functions must be embraced to add further constraints and permit further inferences. The theory by John Searle appears to be the least restricted theory, as it assumes that functions are observer-relative, i.e., they are ascribed externally to some entity, without additional restrictions. Adoption Searle's theory of function does not permit adding additional statements about functions and their behaviour, at least when restricting the statements to ones pertaining exclusively to the material stratum of reality. Adopting axioms for Larry Wright's or Ruth Millikan's function theories allows deriving more knowledge, but is vulnerable to all the counter examples that were developed for their theories.

Differences to original OF

There are several differences to, and extensions of, the original OF I propose here. Most apply because the account here is specific to the domain of biology, and not a generic top-level framework for representing functions.

First, I do not use the notion of a *trigger*. In Burek [2006], a function must be triggered to be realized, and the trigger is external to the function description. I assume the trigger to be part of the requirement situation. It may sometimes be desirable to make the trigger of a function explicit. In this case, a trigger can easily be defined as a constituent part of the requirement situation. The requirement situation that I use for the notion of function is therefore richer than the original one proposed in the OF.

I use a different notion of a functional item. Here, a functional item is a processual role that the bearer of the function $f :: F$ must play in a realization of its function f. In the OF, the functional item is a universal that contains all the essential features of an object that are necessary for it to cause a realization of the function. This kind of entity is included in the analysis of dispositions presented here. Let $d :: D$ be a disposition of e. Whenever an instance of the player universal $U(D)$ plays the situational role $R(D)$ within a requirement situation $s :: T_{req}(D)$, the disposition d is realized.

I exclude functions that are realized instantaneously, i.e., where the requirement and goal situation are present at the same *time boundary*[12]. The difficulty I have with such a strong form of instantaneous function realizations is to understand what kind of force would bring the realization about besides the force

[12] I still permit functions to be realized instantaneously, i.e., by an *Instantaneous change*.

of logical inference. An example of such an instantaneous function is the function of the color of a moth, the function "to camouflage". The original analysis in [Burek, 2006] would be to analyze it as a function which is realized by the color itself, that forces a transition from a state of the world where the moth is not camouflaged to a state of the world where the moth is camouflaged, without any time elapsing between these. The only force that could bring such a transition about is the force of logical inference; causation cannot be at work here, as it would require the passing of time – however small[13] [Michalek, 2009]. I see a difficulty with functions that can be realized by logical inference: it is not clear whether the requirement situation should contain a fact about the "camouflagedness" of the moth, or whether it should contain the fact that the moth is *not* camouflaged. In the latter case, a genuine contradicting difference exists simultaneously between the requirement and goal. Two situations that exist at the same time boundary and have the same constituents (at least a color and a moth) but contradicting properties will lead to a formal inconsistency in the knowledgebase, and should therefore be excluded. Therefore, the fact about the camouflagedness of the moth is simply not contained in the requirement situation but consistently added in the goal situation by means of a logical inference. In this case, camouflagedness simply *means* having a particular color. I find it in contradiction to my intuition about functions and their realization to include this kind of transition as a possible function realization.

Application to OBO's Ontologies

Functions are used in a number of the OBO ontologies. The Gene Ontotology (GO) [Ashburner et al., 2000] contains an ontology of biological functions, the

[13] Time does not even have to pass in the sense of having a duration. It is necessary, however, to consider entities that are present at different time entities, e.g., time boundaries.

Celltype Ontology [Bard et al., 2005] and the ChEBI Ontology [Degtyarenko et al., 2007] use functions to distinguish between types of cells and types of chemical entities. These ontologies would benefit from a common formal theory of function, as it enables the derivation of additional facts from facts about functions. Here, I outline the work presented in [Burek et al., 2006].

The OF permits the formal specification of the structure of a function. In the ontologies that use functions now, this structure is hidden in textual definitions and explanations.

The first kind of application of the ontology of functions is the identification and explanation of relations between processes and functions. The Gene Ontology [Ashburner et al., 2000] provides a prime example in this respect as it contains both an ontology of molecular functions and an ontology of biological processes. There has been some controversy and discussion about whether the Molecular Function ontology of the Gene Ontology describes functions or activities, and how functions are related to processes [Smith et al., 2003]. Functions and activities are usually considered different entities, and actions or activities may realize certain functions. Therefore, while the function of an enzyme may be “to catalyze” a reaction, the activity performed by the enzyme is the catalysis itself, which may be embedded in another process.

We assume that at least parts of the Molecular Function taxonomy refer to genuine functions in the sense of the OF, and the annotation relation for some of the gene products annotated to these terms corresponds to the **has-function** relation.

A general example is `GO:0005215` (*Transporter activity*), which we understand as referring to the function “to transport”. A more specific example is the category `GO:0051119` (*Sugar transporter activity*), which can be understood as the function “to transport sugar”.

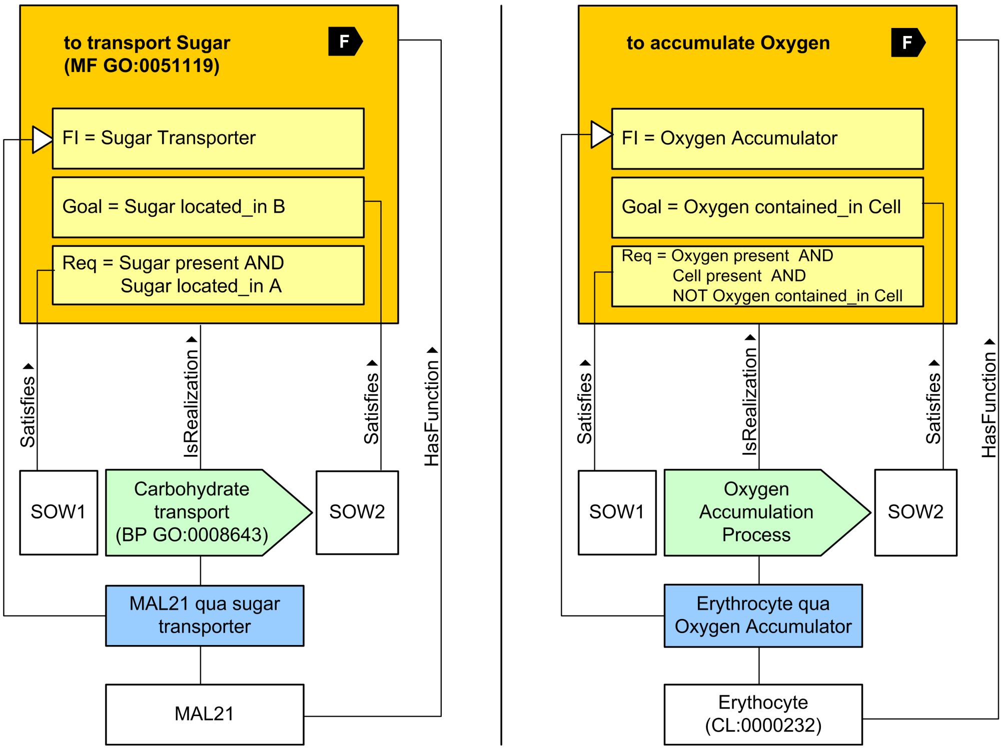


Figure 5.7: Two exemplary models employing OF, instantiating the general model in figure 5.5 (correspondences indicated by the coloring). On the left-hand side, a schematic version of the function “to transport sugar” together with its realization is shown. Processes of type *carbohydrate transport* realize this function, and an entity, in this case *MAL21*, has the function “to transport sugar”. Whenever applicable, the identifiers from the GO are used (for the function and process). *MAL21* is currently **annotated** to the function and the process in the GO. In this model, the annotation relation is replaced by the **has-function** relation. On the right-hand side, the function “to accumulate oxygen” is modelled. This is a function taken from the Celltype Ontology. Except for *Erythrocyte*, the entities involved in this model are not present in any of the OBO ontologies.

In the framework of the OF, the function “to transport sugar” can be formally represented:

- The requirements of the function is of the situation type where one sugar molecule (CHEBI:25407 or CHEBI:25679) is located at some location l_1 at time t_1: $T_{req} = \{s|s \models < \textit{located-at, mol}, l_1, t_1; 1 > \wedge < \textit{instance-of, mol}, \textit{CHEBI:25407}, t_1; 1 >\}$.
- The goal is the type of situation where the sugar molecule is located at a different location: $T_{goal} = \{s|s \models < \textit{located-at, mol}\ , l_2, t_2; 1 > \wedge <= , l_1, l_2; 0 >\}$
- The functional item is a role called *Sugar transporter*.

It can be observed that many gene products annotated with the *Sugar transporter activity* in GO’s Molecular Function Ontology are also annotated with some sub-category of the *carbohydrate transport* category in GO’s Biological Process taxonomy. With the help of the OF we can make these relations explicit: processes of the type *Carbohydrate transport* are realizations of the function “to transport sugar”; some of the gene products that are annoted to *Carbohydrate transport* stand in the **has-function** relation to “to transport sugar”.

The left-hand side of figure 5.7 demonstrates the full interconnections of this example by means of OF. In terms of the relations we introduced, these are captured by $Realizes($MAL21, GO:0051119, GO:0008643$)$. What could be directly added to the GO are links of **is-realization** and **has-function**: *IsRealization(*GO:0008643, GO:0051119*)* and *HasFunction(*MAL21, GO:0051119*)*.

The second application of the OF is in the identification of functions are processes that are only implicitly used. This kind of use of the concept of function occurs in the Celltype Ontology [Bard et al., 2005] (CL) and the Ontology of Chemical Entities of Biological Interest [Degtyarenko et al., 2007] (ChEBI).

CL uses the term function in the subtree under the *Cell by function* category which classifies cell types by the functions which they perform. A general example is *Stuff accumulating cell* (CL:0000325), and more specifically *oxygen accumulating cell* (CL:0000329), of which a red blood cell or *Erythrocyte* (CL:0000232) is a sub-category. The function "to accumulate oxygen (by a cell)" would be modelled as shown in the right-hand side of figure 5.7:

- The presence of *Oxygen* (ChEBI:25805) outside of a *Cell* (CL:0000000) is the requirement of the function: $T_{req} = \{s | s \models < \textit{contained-in,o,c}, t_1; 0 > \wedge < \textit{instance-of,o,CHEBI:25805}, t_1; 1 > \wedge < \textit{instance-of,c,CL:0000000}, t_1; 1 >\}$.
- The goal of the function is the cell's accumulation of oxygen, the oxygen being contained in the cell: $T_{goal} = \{s | s \models < \textit{contained-in,o,c}, t_2; 1 >\}$.
- The functional item is called *Oxygen accumulator*.

The subsumption of erythrocyte under oxygen accumulating cell in CL reflects the fact that erythrocytes have the function "to accumulate oxygen", *HasFunction(*CL:0000232, "to accumulate oxygen"*)*. Further, they may *act as* oxygen accumulators, a new category for CL, in the process of *Oxygen accumulation*, $IsRealization$("oxygen accumulation", "to accumulate oxygen"). Again, the **realizes** relation captures all these new relations appropriately: *Realizes(*CL: 0000232, "to accumulate oxygen", "oxygen accumulation"*)*.

5.3 Default and canonical knowledge

One particular difficulty in making biomedical ontologies interoperable results from the existence of two distinct types of biomedical ontologies. The first group describes a *canonical* or idealized view on a domain, such as ontologies of canonical anatomy. The other group describes *phenotypes*, properties or phenomena, that – when exemplified by individuals – may contradict knowledge represented in the first group. I call the former group *canonical ontologies* and the latter *phenotype ontologies*.

Many ontologies describing structure, such as cell structure, histology or anatomy, are canonical in this sense. On the other hand, a phenotype ontology describes phenomena whose exemplification by individuals may lead to deviations from this idealized structure.

5.3.1 Canonical facts and canonical ontologies

An example of a canonical ontology of anatomy is the Foundational Model of Anatomy [Rosse and Mejino, 2003] (FMA), which describes an idealized domain, i.e., it describes a prototypical, idealized human anatomy. For example, it contains statements such as:

Every instance of a *Human body* has as **part** some instance of *Appendix*. (5.110)

This does not necessarily apply to every real human body: an individual human body may **lack** an appendix as part. Statement 5.110 describes an idealized or *canonical* human.

The inverse of this statement, that every (human) appendix is part of some human body, is included in the FMA as well. It does not seem universally valid, either. After an appendix' removal, it may still be required to state that the appendix exists *as an appendix*, but is no longer a part of the human of which it was removed. For example, in a clinical application, an appendix could be sent to a pathology laboratory to obtain a pathological report. Inclusion of this information in the model using the anatomy ontology will require some form of refering to the appendix, and its relation to the human body in which it originated. In this example, the appendix should not be a part of a human body anymore.

Similar difficulties occur throughout all anatomy ontologies of which I am aware. However, most of these ontologies were developed in the OBO Flatfile Format [Golbreich and Horrocks, 2007] or using a frame-based system [Minsky, 1977]. In these languages, relations are asserted directly between categories. A formal semantics that reduces these relations between categories to relations between individuals has been introduced [Smith et al., 2005a, Golbreich and Horrocks, 2007], but only after many of the anatomy ontologies were already developed. These semantics have in common that their interpretation of

$$\textit{Human body } \textbf{has-part} \textit{ Appendix} \tag{5.111}$$

is the statement in equation 5.110 (sometimes with an additional time index). With this kind of interpretation, both statement 5.111 and

$$\textit{Appendix } \textbf{part-of} \textit{ Human body} \tag{5.112}$$

are false when *Appendix* and *Human body* are understood in the intuitive way, due to the universal quantification in 5.110. Nevertheless, the ontologies are being used in several applications. These applications are tailored to the ontologies and their application. They can interpret the ontologies pragmatically in a way that differs from the explicit, formal semantics of the ontologies' representations.

In order to make them interoperable with other ontologies, the currently implicit intension of the ontologies' statements must be made explicit. The semantics that were developed for these ontologies do not achieve this goal. Therefore, many of these ontologies are refered to as *canonical* ontologies, to set them apart from other ontologies.

A canonical ontology describes an idealized, prototypical domain. It contains categories that do not have instances in reality, but rather are the result of abstractions and expectations made by the ontology's creators, or by scientists within the domain that is modelled using a canonical ontology. These idealizations are often developed as *reference models* for communication between scientists. Human anatomy, as found in anatomy text books [Netter, 1997], is an example for such a reference model. It is used by biological and medical experts as a common reference for communicating their findings (e.g., about diseases or disabilities, signs and symptoms).

This kind of description is not unique to the anatomy of organisms. The need to establish reference models for other biological structures such as cells, pathways or functions lead to the development of a number of *canonical* ontologies, whose primary purpose is to provide a reference model for a domain. The Gene Ontology (GO) [Ashburner et al., 2000] is such a canonical ontology for cell components, biological processes and molecular functions.

An information system using these ontologies must be able to access the intension of statements in these ontologies and how they correspond to reality, in order to embed them in a wider context. The idealizations that are the basis of the canonical ontologies often correspond to a perceived *normality*: most humans have an appendix as part; most human arms are part of some human body; most cellular nuclei are part of a cell; most human hands have five fingers as part. This is one possible form in which a canonical fact can arise: because *most* instances of a particular category have a certain property (such as having certain parts), all *canonical* instances of this category have this property; and within the context of canonicity, as assumed in a canonical ontology, *all* instances have the property.

Other sources of canonical facts come from understanding the functioning of biological systems, and the role that structures play in these systems. A certain structure may be required to realize a function, or the structure developed throughout evolution in a certain way in order to realize a function. Even if this function cannot be realized in most systems that are investigated in reality, having the function, and a corresponding physical structure capable of realizing the function, may become a canonical fact. One example is osteoporosis[14] in human women. Although the majority of women develop osteoporosis at a certain age, it is considered a disease, and usually not included in a canonical human anatomy ontology. Even if all or most women at a certain age develop osteoporosis, and it would therefore be *normal* for women to develop it, it may not be included in a corresponding canonical ontology.

Additional sources of canonical facts depend on history, ethics or scientific state of the art. For example, historically developed concepts such as the notion of a *species* remain in use today. Whether behavioural facts such as sexual preferences are considered as canonical facts may depend on the currently accepted ethics within a society.

5.3.2 Phenotypic facts

Another kind of ontology describes phenotypic facts. A phenotypic fact is a fact that is observable in reality as a phenomenon. Examples of phenotypic facts include *being blue*, *being more than 5 meters in length*, *having a finger as part* or *lacking a tail as part*. These ontologies may contain statements that

[14] Osteoporosis is a disease of the bone in which the bone loses mineral density and the risk of fracture of the bone increases. Many women develop osteoporosis after menopause.

establish relations between these classes; for example, lacking a tail entails lacking a tail tip.

The important difference between these *phenotype ontologies* and canonical ontologies is that phenotype ontologies do not describe idealizations. They specify the meaning of a vocabulary that is used to describe *observations*. However, from a formal perspective, this criterion is insufficient to distinguish phenotype ontologies from canonical ontologies. The canonical ontologies also describe categories and the relations between them. These may not be applicable to the same entities in reality as the phenotype ontologies, or the canonical ontologies may have no referent in reality at all. Nevertheless, when considered in isolation and based only on their formal structure, it is difficult to make a distinction between canonical and phenotype ontologies.

Distinguishing between canonical and phenotype ontologies requires analyzing the different roles they play when they are used together. Difficulties arise when the two types of ontologies are combined and information flows between them.

5.3.3 Integration problem

Combining canonical and phenotype ontologies pertaining to the same domain (i.e., they contain overlapping categories) and applying both to individuals, may lead to formal inconsistencies. These inconsistencies arise when a phenotypic fact – an observation – contradicts an idealized fact that is part of the canonical ontology. I will use two examples throughout this section, the first taken from the Mouse Anatomy ontology [Hayamizu et al., 2005] and the Mammalian Phenotype Ontology [Smith et al., 2004b], the second from the

FMA and the International Classification of Diseases (ICD-9) [World Health Organization, 2001].

Mouse anatomy contains the statement that *Tail* is **part of** *Mouse body*:

$$partOf(tail, mouseBody) \tag{5.113}$$

According to the semantics currently in use for these statements, this can be translated to the fact that every instance of a (mouse) *Tail* is **part of** some instance of a *Mouse body*. I analyze the following example.

In an experiment (e.g. a tail transplant), a mouse tail is present that is not part of a mouse. This tail instantiates the *Detached tail* category. The mouse which originally owned the tail is also part of the experiment. It instantiates the category *Absent tail*[15], which is part of the Mammalian Phenotype Ontology. The definitions of these categories are:

$$x :: detachedTail \iff x :: Tail \land \neg\exists y(y :: MouseBody \land partOf(x,y)) \tag{5.114}$$

$$x :: absentTail \iff \neg\exists y(y :: Tail \land partOf(y,x)) \tag{5.115}$$

For this example I make several assumptions. The first category, *Detached tail,* is not contained in a published biomedical ontology, but can easily be defined. It is introduced and defined by me to illustrate the example. It is conceivable that such a category would be included in an ontology such as the Mammalian

[15] The category *Absent tail* is the name given to a category in the Mammalian Phenotype Ontology which is used to describe phenotypes in mice. As such, it is applied to mice and not to tails. In particular, an absent tail is not a sub-category of the *Tail* category; it is the *reification* of a negated **has-part** relationship, as in equation 5.115. A more appropriate name of this category would be *Absence of tail*, *Mouse without tail* or *Entity without tail*. Using this name has the additional advantage of making the intension of the category as a description of a phenomenon explicit.

Phenotype Ontology, if there was a need for it within the biomedical community. The second assumption I make is more controversial, and I will justify it later in more detail. I assume that the statement “*Tail* **part of** *Mouse body*” from the Mouse Anatomy Ontology has an inverse, the statement “*Mouse body* **has part** *Tail*”. This inverse cannot be formally proven using the semantics employed in the formalization of the Mouse Anatomy Ontology. However, I derive this statement from observing the applications of the Mouse Anatomy Ontology and anatomy models in general.

The second example combines the FMA and the ICD-9. The FMA contains a statement:

$$\textit{Human}\ \textbf{has-part}\ \textit{Nose} \tag{5.116}$$

The ICD-9 contains under 748.1 (other anomalies of nose) the class *Absent nose*. The ICD-9 does not contain formal definitions of its terms; however, I define the category from the ICD-9 *Absent nose* as:

$$x :: absentNose \iff x :: Human \wedge \neg\exists y(y :: Nose \wedge partOf(y,x)) \tag{5.117}$$

This definition is different from the similar category *Absent tail* defined in equation 5.115. The reason here is that the ICD-9 is explicitly a description of human diseases, while the MP describes phenotypes or phenomena, that are applicable to many domains[16]. In the example, the human Mary is classified both as a human (according to the FMA) and as a human without nose (according to ICD-9 748.1).

Both examples lead to formal inconsistencies. The combination of the two ontologies (or classifications, as for the ICD-9) together with their application to a domain (i.e., the introduction of individuals instantiating categories from

[16] As before, the label of the category does not reflect its intension appropriately. Here, *Human without nose* would be a more suitable label.

both ontologies) results in the derivation of contradictions. For the human Mary, the contradiction can be derived as follows:

$$x :: Human \rightarrow \exists y (y :: Nose \wedge partOf(y,x)) \tag{5.118}$$

$$x :: absentNose \iff x :: Human \wedge \neg\exists y (y :: Nose \wedge partOf(y,x)) \tag{5.119}$$

$$Mary :: Human \tag{5.120}$$

$$Mary :: absentNose \tag{5.121}$$

$$\exists y (y :: Nose \wedge partOf(y, Mary)) \qquad \textbf{(Subst+MP } (5.118)\textbf{,}(5.120)\textbf{)} \tag{5.122}$$

$$Mary :: Human \wedge \neg\exists y (y :: Nose \wedge partOf(y, Mary)) \quad \textbf{(derived from } (5.119)\textbf{,}(5.121)\textbf{)} \tag{5.123}$$

$$\neg\exists y (y :: Nose \wedge partOf(y, Mary)) \qquad \textbf{(clash with } (5.122)\textbf{)} \tag{5.124}$$

According to the FMA, all humans have as part a nose. The ICD-9, on the other hand, has a description of humans that lack a nose. These humans have a disease (according to the ICD), but are human nevertheless. Instantiating the corresponding categories from both ontologies leads to the inconsistency. The same kind of inconsistencies can be derived for the statements from the MA and the MP.

The cause of the inconsistency lies in the different uses of these ontologies. Canonical ontologies are used as reference models. They establish a basis for describing instances of a domain. The phenotype ontologies are used to describe deviations from this reference model. They do not contain the facts that are already contained in or derivable from the canonical ontologies.

Canonical and phenotype ontologies are frequently used together. The categories used in the MP are defined using categories taken from the MA, among others. However, due to inconsistencies that arise, neither can be consistently instantiated. This hinders the information flow that is possible between these ontologies. I believe that combining canonical and phenotype ontologies requires the use of a different semantics than the one currently employed. This alternative semantics must allow for the consistent combination of both types of ontologies, and make the nature of the canonical ontologies as *reference models* explicit. Applications that include both types of ontologies must already employ such a semantics on a pragmatic level. However, it would benefit the application and integration of biomedical ontologies, if a semantics for these ontologies could be provided that makes their nature explicit and still allows for a consistent integration of both. We have made this possible within the biological core ontology GFO-Bio [Hoehndorf et al., 2007].

5.3.4 Default rules and default logic

Using GFO-Bio as a framework for integrating biomedical ontologies, we address the problem of accurately representing canonical and phenotype ontologies. The core assumption is that canonical ontologies such as the FMA establish rules that do not necessarily apply to every instance: an individual human body may **lack** an appendix as part or mice may **lack** a tail. Instead, the rules describe an idealized or *canonical* domain. Phenotype ontologies describe phenomena, whose exemplification by individuals can be *deviations* from these idealizations. For example, an individual may be both an instance of a prototypical human body as described in the FMA (which implies an appendix as part) and an instance of the category *Human body with absent appendix*. In a classical logical framework, such as those commonly used in biomedical knowledge representation, e.g. in the form of OWL [Mcguinness and van Harmelen, 2004], a formalization of the conjunction of these two statements would lead to an inconsistency. A human body in the former case has an appendix as a part, while in the latter case it does not. Instantiating both categories creates the inconsistency. A logical inconsistency in the formal sense can only arise when the logical functor of negation is used. This functor is hidden in categories such as *Absent X*, as used in the Mammalian Phenotype Ontology [Smith et al., 2004b]. The formal detection of logical inconsistencies by inferences needs an explication of negation.

In order to avoid terms such as *Absent X* and make the negation explicit, we adopt a modified form of the **lacks** relation [Ceusters et al., 2006], which we explicitly define as:

> Individual p **lacks** category C with respect to relation **R**, if and only if there does not exist an x such that: $p\mathbf{R}x$ and x is an instance of C.

We use binary relations of the kind x **lacks-R** C instead of x **lacks** C with respect to **R**. For example, the fact that some individual x **lacks** a category C with respect to the relation **has-part** will be denoted as x **lacks-part** C. The **lacks-part** relation can also defined in the extension to the OBO Flatfile Format that I outlined in section 4.1.

Using the **lacks** relation may cause an inconsistency when a canonical ontology and a corresponding phenotype ontology are used in a classical logic formalism, such as first order logic [Hilbert et al., 1999] or description logic [Baader, 2003]. The reason is that classical formalisms enforce very strict interpretations, e.g. of quantifications like “every human”, which results in *monotonicity* of these formalisms: the inferences drawn from a classical logical theory T remain true in every extension of T with additional facts.

In order to prevent inconsistencies, while at the same time preserving the intuition behind statements such as “a human has an appendix as part”, the interpretation of such statements in the canonical ontology must be modified. In GFO-Bio, we use a *nonmonotonic* logic that treats the statements provided in a canonical ontology as true by default. Adding further knowledge, e.g. by referring to a phenotype ontology or using a statement involving the **lacks** relation (and therefore negation), may invalidate previously drawn conclusions.

Several ways of treating default rules and exceptions in logics have been proposed. The most popular among these proposals are default logic [Reiter, 1980], circumscription [Mccarthy, 1980, 1986] and autoepistemic logic [Konolige, 1988, Gabbay et al., 1994]. We use default logic for our application, because it admits a transparent representation, and allows a semantically correct translation to a form of nonmonotonic, declarative logic programs called answer set programs [Lifschitz, 2002].

In default logic, a *default rule* has the following form:

$$\frac{A(\overline{x}) : B(\overline{x})}{C(\overline{x})} \tag{5.125}$$

This means that if $A(\overline{x})$ is true (prerequisite), and *it is consistent to assume* that $B(\overline{x})$, then $C(\overline{x})$ can be derived. In order to formalize our example of humans normally having an appendix as part, we would use the following default rule:

$$\frac{Human(x) : x\ \mathbf{IC\text{-}has\text{-}part}\ Appendix}{x\ \mathbf{IC\text{-}has\text{-}part}\ Appendix} \tag{5.126}$$

Here, the precondition is $Human(x)$, the fact that x is a human. Then, if it is consistent to assume that x has as part an instance of $Appendix$, it is concluded that x has as part an instance of $Appendix$. The definition of the relation **IC-has-part** follows the schema in table 5.1 and is inspired by the interactions between the two modules of GFO-Bio that I described in section 5.1.4.

Nonmonotonicity arises from “it is consistent to assume that *x* **IC-has-part** *Appendix*”, which means that if *x* **IC-has-part** *Appendix* cannot be proven false from the given facts, its addition to the knowledge base does not lead to a contradiction. Adding the statement that *x* does not have an appendix as part (*x* **IC-lacks-part** *Appendix*) would lead to an inconsistency with *x* **IC-has-part** *Appendix*; therefore, this rule could no longer be used to derive that *x* has an appendix as part.

Answer-set programming, the formalism we use for our implementation, can mimic default rules. It uses two kinds of negation, called *strong* and *weak negation*. Strong negation is the classical (monotonic) negation, as used in the definition of the **lacks** relation. Weak negation, often denoted as `not A`, corresponds to the above statements “it cannot be proven that A is true”, or “it is consistent to assume that A is false”.

5.3.5 Formalizing defaults using relations

In a canonical ontology, relationships between its categories can be interpreted as *default* relations. By default, a human has some appendix as part. However, an instance of a human, such as *John*, may **lack** an appendix as a part; therefore, *John* is an instance of both *Human* and *Human without an appendix* (or *Absent appendix*). In order to include canonical relationships between two categories, new relations must be introduced, such as **CC-canonical-has-part**. Then, the relationship between *Human* and *Appendix* becomes “*Human* **CC-canonical-**

has-part *Appendix*". Further, this relationship corresponds to a *default rule*:

forall x, C_1, C_2:
if C_1 **CC-canonical-has-part** C_2 and x **IC-instance-of** C_1, then
by default:
there exists a y: y **IC-instance-of** C_2 and x **II-has-part** y (5.127)

Using a class of **lacks** relationships as introduced by [Ceusters et al., 2006], we formalize the default operator in the rule above as:

forall x, C_1, C_2:
if C_1 **CC-canonical-has-part** C_2 and x **IC-instance-of** C_1 and
it cannot be proven that x **IC-lacks-part** C_2, then
there exists a y: y **IC-instance-of** C_2 and x **II-has-part** y (5.128)

In general, for each relation **R** between the categories in an ontology, we create several new relations: **CC-R** for the monotonic relationship between the categories, **CC-canonical-R** for the nonmonotonic default relationship between categories, **IC-R** for the monotonic relationship between an individual and a category, such as "John **IC-has-part** *Appendix*", meaning that John has some appendix as part, and **II-R** for the monotonic relationship between individuals. In addition, we introduce a class of **lacks** relationships. A schematic view of the new relationships introduced is shown in table 5.1. The schema is somewhat incomplete, because the introduction of canonical relations can be extended to the class of **lacks** relation, in the sense that some category may canonically lack some other category with respect to a relation **R**. In this case, the relation **R** must be replaced by **lacks-R**. This allows the treatment of exceptions between categories. For example, the category *Mouse with absent tail* can be defined as a mouse which lacks a tail as part.

Implementation

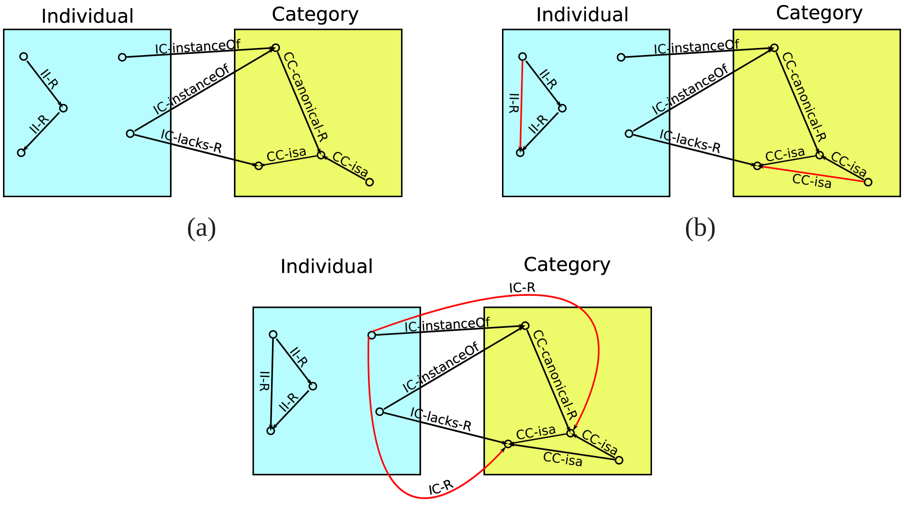
We have used a technique known as DL-programs [Eiter et al., 2005] to implement rules together with the OWL version of GFO-Bio. The system DLVHEX allows for a bidirectional flow of information between an answer-set program and a description logic knowledge base or ontology; thus, it is well suited for our purposes. DLVHEX is based on the well-established datalog system DLV [Leone et al., 2006] that uses answer set semantics.

Relationships that are used in GFO-Bio are made available in the DLVHEX system. It then becomes possible to express the necessary axioms for relations of the kind **CC-canonical-R**. For example, for the relationship **CC-canonical-has-part**, the following axiom is added, corresponding to formula (5.128) in DLVHEX:

```
IC-has-part(X,Y) :- ind(X),class(Y),class(Z),inst(X,Z),
                    CC-canonical-has-part(Z,Y),
                    not IC-lacks-part(X,Y).
```

This means that if two categories Z and Y stand in the relation **CC-canonical-has-part**, and *it cannot be proven that* X **IC-lacks-part** Y (`not IC-lacks-Part(X,Y)`), then it is concluded that an individual X, which is an instance of Z, stands in the relation **IC-has-part** to the category Y. An example illustrating this reasoning is shown in figure 5.8.

For an adequate integration of canonical and phenotype ontologies, nonmonotonically treated formulas must be added for each relation that is used in statements that are true by default. This requires the addition of an answer set pro-



(a)

(b)

(c)

Figure 5.8: In figure (a), the left side shows five individuals (instances of GFO-Bio's *Individual* category) and the right side contains four categories (instances of GFO-Bio's *Category* category). In addition, a number of relations are illustrated between the individuals, between the categories, and between individuals and categories. The relation **R**, denoted as **II-R**, is transitive. Figure (a) and the transitivity of **II-R** should be seen as the input ontology. In figure (b), the result of a classification using a description logic reasoner is illustrated. Here, the transitivity of the **CC-isa** relation and the relation **II-R** is resolved, reflected by the additional links. Figure (c) shows the result from applying the answer set rules formulated in DLVHEX. In this step, the default relationship between two categories, denoted by **CC-canonical-R**, is resolved. Two additional **IC-R** links are created for one individual. For the other individual, which instantiates the same category, these links are not created, because the **IC-lacks-R** relation blocks them.

gram for each relation **CC-canonical-R** and the corresponding relations **IC-R** and **IC-lacks-R**:

```
IC-R(X,Y) :- ind(X),class(Y),class(Z),inst(X,Z),
             CC-canonical-R(Z,Y),
             not IC-lacks-R(X,Y).
```

Relation to the OBO Relationship Ontology

The OBO Relationship Ontology [Smith et al., 2005a] requires several additions for our proposal to succeed. First, the classes of **lacks** relations, as described in table 5.1, must be added. This will allow absent body parts to be defined in ontologies such as the Mammalian Phenotype Ontology [Smith et al., 2004b]. This addition is already underway.

In the description logic variant of the Web Ontology Language [Mcguinness and van Harmelen, 2004, Baader, 2003] (OWL-DL), **lacks** relations can be expressed using negated statements. However, **lacks** relations are reduced to relations between individuals in a different way compared to what is done for most other relations in the OBO Relationship Ontology (cf. table 5.1). Ontologies developed directly in OWL-DL could use negation to avoid reference to **lacks** relations at all.

Second, **canonical-R** relations must be included as relations between categories, using the semantics introduced here. In particular, the **canonical-R** relations require a nonmonotonic knowledge representation formalism, and cannot be formalized using any form of classical logic. We presented one possible implementation using answer set semantics, but there are other alternatives. At its core, however, the definition of the **canonical-R** relations remains the same

Schema of introduced relations

Relation	**Domain:Range**	**Definition**
x **II-R** *y*	Individual:Individual	The individuals *x* and *y* stand in the relationship **II-R**.
x **IC-R** *y*	Individual:Category	There exists an individual *z*, such that: *z* **IC-instance-of** *y* and *x* **II-R** *z*.
x **CC-R** *y*	Category:Category	For all individuals *a* such that: *a* **IC-instance-of** *x*, *a* **IC-R** *y*.
x **CC-canonical-R** *y*	Category:Category	For all individuals *a* such that: *a* **IC-instance-of** *x*, by default, *a* **IC-R** *y*.
x **II-lacks-R** *y*	Individual:Individual	The individuals *x* and *y* do not stand in the relationship **II-R**.
x **IC-lacks-R** *y*	Individual:Category	The individual *x* does not stand in the relationship **IC-R** to *y*.
x **CC-lacks-R** *y*	Category:Category	For all individuals *a* such that: *a* **IC-instance-of** *x*, *a* **IC-lacks-R** *y*.

Table 5.1: For each relation used in an imported ontology, a number of relations between categories, individuals and between individuals and categories can be created. The **CC-canonical-R** relationship is a *default* relation that is accompanied by axioms in an answer set program to describe its semantics as a default.

in all possible formalisms dealing with defaults: *if it is consistent to assume that* some relation holds, this relation holds.

The method we propose can be used in conjunction with existing tools and ontologies. Little effort is required to modify current ontologies to fit within our proposed methodology. In [Hoehndorf et al., 2007], we have applied this method to the integration of the Adult Mouse Anatomy Ontology [Hayamizu et al., 2005] and the Mammalian Phenotype Ontology [Smith et al., 2004b] (MP), and show how its application leads to more expressive queries and a consistent integration of these ontologies.

5.3.6 Discussion

Meaningful integration of the numerous biomedical ontologies is a major task with many challenges. Currently, the infrastructure for such integration is developed in the form of top-level ontologies, biomedical core ontologies and logic-based inference systems. We propose the addition of another knowledge representation formalism based on a non-monotonic form of reasoning. The application of our method requires only few changes to existing ontologies, and we believe that the benefits of its application justify the work that is necessary to adapt ontologies to the method.

Concept conversion

The formalism we introduced requires reformulating the definitions for the categories expressed in phenotype ontologies. Categories in the form *Absent X* should be defined by, e.g., **CC-lacks-part** X, where X is a category in some canonical ontology. In some cases, this conversion can be done automatically using simple pattern matches. The Mammalian Phenotype Ontology [Smith et al., 2004b] contains 395 categories of the type *absent X*, which indicate a **CC-lacks-part** relationship. However, it is likely that an amount of manual curation will be required to convert relevant concepts into the required form. We believe that the advantages gained by having a common framework for integrating a large number of biomedical ontologies justifies this effort, in particular since it also allows for a semantically richer definition of terms.

Defaults and canonical knowledge

Not all facts in canonical ontologies refer to *default* knowledge, as discussed in section 5.3.1. However, we expect that a significant number of facts can be translated to the formalisms we propose, thereby making the nature of the fact as a default explicit. We believe that the framework of default logic, compared with other systems, provides the most adequate interpretation for canonical knowledge. This is due to the fact that most of the facts that are included in the canonical ontologies are derived from abstractions of what is true in *most* entities covered by the canonical ontology.

Comparison with other approaches

The important role of accommodating exceptions and defaults in biomedical knowledge representation has been recognized previously [Rector, 2004], where patterns to deal with a variety of cases were introduced and discussed. These cases are based on the description logic fragment of OWL [Mcguinness and van Harmelen, 2004], and therefore monotonic logic. In [Rector, 2004], three types of exceptions that occur in biomedical knowledge bases are distinguished:

1. Single exceptions: “Arteries carry oxygenated blood” except for the pulmonary artery. In [Rector, 2004], it is proposed to reformulate this statement to “Arteries except the pulmonary artery carry oxygenated blood”.
2. Exceptions due to context: “The normal human manus has five digits”, with “human” and “normal” being treated as explicit contexts.
3. Unpredictable number of exceptions, exceptions from exceptions, such as drug uses, contraindications and interactions.

We offer a method for representing these types of exceptions using a nonmonotonic knowledge representation formalism. We use answer set programs to provide the semantics for treating knowledge in OWL as default knowledge with additional exceptions. This does not exclude the possibility to treat these types of exceptions exclusively in a monotonic logic such as OWL where appropriate, for which [Rector, 2004] provides a solution. The solution in [Rector, 2004] to the example of arteries carrying oxygenated blood, except the pulmonary artery, has the problem that it must be explicitly known that some artery is *not* the pulmonary artery, in order to conclude that this artery carries oxygenated blood. There may be cases where this is not wanted, especially if the exception occurs very rarely. In particular, if there is only one rare exception to a rule and some statement influencing the property which changes with this exception is asserted, then the knowledge engineer may want to make this exception explicit, and ignore it otherwise. Then, a question whether an artery carries oxygenated blood evaluates to true, except when it is *proven* that this artery is the pulmonary artery. On the other hand, the solution proposed by [Rector, 2004] is guaranteed to provide the correct inference in every case. Depending on the users and uses of a knowledge base or ontology, different representations for this case may be selected, and in many cases the treatment in [Rector, 2004] is adequate.

Case two is solved by explicitly introducing a context argument, in the form of additional properties, e.g., by introducing some relation **has-anatomical-status** which maps to *normal*. Then, a *Mouse* that has an anatomical status *normal* could have, e.g., a tail and a head as part. If a mouse had no tail, it can be concluded that it is an anatomically abnormal mouse. However, then it would be impossible to conclude that it still has a head. An extension to the solution in [Rector, 2004] would be to make the context more fine-grained, by specifying mouse with anatomically normal tails, heads, and so on. This comes down to specifying an enormous number of exceptions in a monotonic logic, and in order to obtain a correct answer to a query for all the parts of some individual mouse, all these exceptions must be explicitly excluded. It would not be possible to simply state that some entity is a mouse in order to obtain its parts. Instead it is required to specify explicitly which parts are normal and abnormal, which means in essence to add the answers to the query asked.

The third case in [Rector, 2004] is closest in spirit to our work, as one of the proposals is to use a hybrid reasoning system in order to deal with it. We have extended this idea by giving a formal account of our treatment of exceptions, which is based on a well-studied nonmonotonic logic, and is implemented in a computationally tractable framework. It can also be used in conjunction with appropriate upper ontologies. Further, we have shown how to use this formalism to achieve interoperability between canonical and phenotype ontologies in biology. And finally, we give an implementation of our ontology and support for reasoning over exceptions. This could be achieved because recent years have seen an increasing effort in developing reasoners for the Semantic Web and extending them in various ways, among them the implementation we are using, DLVHEX.

We believe that our solution to the problem of exceptions and deviations from a canonical ontology is more general than the proposal in [Rector, 2004]. In our opinion, the knowledge contained in a canonical ontology is inherently default knowledge. There is no adequate solution for representing this type of knowledge in a monotonic knowledge representation formalism. Representation in monotonic logic requires exceptions to be encoded in the ontology either as a list of exceptions to an axiom, or using a general *abnormality* predicate. For example, the fact that mice usually have some tail as part can be represented as "*Mouse* **has-part** *Tail* except when ..." followed by a complete list of exceptions. Alternatively, *Mouse* can be replaced by *Normal mouse* in the rule, and a mouse without a tail is not normal. The first solution requires complete knowledge of all known exceptions. These must additionally be explicitly excluded in every query for parts of the mouse. The second way does not require this knowledge of exceptions, but allows for no further inferences once a mouse is known to be not normal. Defaults and exceptions cannot be dealt with in a monotonic logic without substantially modifying the canonical ontology, and limiting the ability to query the ontology.

Limitations of the method

A major drawback of the software implementation we are using, DLVHEX, is its use of RACER [Haarslev and Möller, 2003] as a description logic reasoner and of DLV [Leone et al., 2006] as a datalog system. RACER and DLV are proprietary software. In order to be of general use and high quality, and to allow for general adoption, an implementation entirely based on free software is required [Raymond, 1999, Stallman et al., 2002].

A number of formalisms have been proposed as a solution to handling defaults in Semantic Web representation languages or other knowledge representation formalisms. Many require modifying the language, and therefore changing tools that are used to develop ontologies. Many biomedical ontologies are developed using tools such as OBO-Edit [Richter et al., 2007] by biology experts, but not necessarily experts in logic or formal ontology. The solution we propose requires no changes to existing tools, since we are using a hybrid reasoning mechanism. Tools that are currently in use can therefore be used further by the ontology developers. The additional semantic features that allow for the treatment of canonical relations as defaults are maintained separately from the ontologies in which they are used.

6 Ontology-based knowledge acquisition

Never pay more for an acquisition
than you have to.

Third Rule of Acquisition

Even when all the problems pertaining to the *representation* of knowledge are solved, problems regarding the acquisition of knowledge remain. Knowledge acquisition is often an expensive and error-prone process, requiring highly skilled professionals. The difficulty is to create a bridge between the knowledge engineer and the expert who has the knowledge. It is rare to have a domain expert who is also sufficiently experienced in knowledge representation to create a representation of her knowledge independently. Even if this were the case, creating this representation of knowledge would be time-consuming – especially in the case of biology where knowledge accumulates and evolves at a rapid pace.

It would benefit ontology-based knowledge representation in biology if the knowledge acquisition process could be performed on a large scale by the trained experts in a domain. Achieving this goal requires the development of software tools that allow many trained experts to collaborate on a knowledge base. It must also allow a means to identify and correct errors and settle conflicts between the domain experts.

I describe the development of several such tools; two require the active use by domain experts. While using them, they are aware that they are developing a knowledge base. This eases the task of interpreting and formalizing the knowledge, but the difficulty is motivating the experts to use the software. Therefore, I describe a third approach that is based on text- and data-mining. This creates more knowledge in less time but is less reliable. All these knowledge-acquisition software applications are ontology-based: they use formal ontologies to verify knowledge, access their own conceptual model and perform queries on the information stored in them. The knowledge that is acquired using these tools is integrated directly within a formal ontology.

6.1 The role of reasoning in collaborative knowledge aqcuisition

In knowledge acquisition, several sources of inconsistency can arise. Some of these can be excluded or detected using formal knowledge and automated reasoners. While it is possible to handle many difficulties manually when only a small group of people is involved in the knowledge acquisition process, large-scale collaborative knowledge acquisition necessitates the development of automated methods to solve potential conficts. Disagreements can arise on several levels and at several stages in the knowledge acquisition process.

One kind of disagreement is with regard to a scientific fact which is under dispute. This can come in several forms. An observation from an experiment can be contested. This may be due to variations in the experiment, because the equipment used to make an observation is known or suspected to produce unreliable results, or biased observations. In this case, the vocabulary used to describe the observations is undisputed. Under dispute are facts from reality or interpretations of these facts. With a sufficient formalization of biological theories, it may become possible to employ automated reasoners to detect inconsistencies between a biological fact (e.g., an individual observation) and biological theories. At the moment, no such theories exist with a sufficient degree of completeness to allow such checks on a regular basis.

Another problem, likely to occur in large-scale collaborative knowledge acquisition, is the use of different conceptualizations of a domain. Given an (observed) individual, it may be contested whether it is an instance of the category A or B. Ontologies are intended to solve this problem by making the *meaning* of A and B explicit. Formal ontologies can then be used to automatically detect inconsistencies that arise through the use of incompatible conceptualizations of a domain, if these incompatibilities are explicitly (i.e., through the use of formal deduction) derivable from the ontologies. In this case, it is advantageous to have *complete* theories that define the meaning of all the categories used in the ontologies.

A complete theory is a theory T over a language $L(\Sigma)$ such that for every $\phi \in L(\Sigma)$, either $T \models \phi$ or $T \models \neg\phi$. Incomplete theories do not fix the intension of the whole vocabulary but only provide restrictions on it. It is then possible to have a statement ϕ such that both $T \cup \{\phi\}$ and $T \cup \{\neg\phi\}$ are consistent. In this case, when two users disagree about ϕ, it is more difficult to detect this disagreement automatically[1].

Ontologies fix the intended meaning of a vocabulary. Formalized ontologies can then be used to detect automatically inconsistencies that arise due to dif-

[1] The addition of $\phi \wedge \neg\phi$ to the theory T will result in an inconsistency that is automatically detectable. However, the disagreement is not always so obvious. For example, it may be contested whether an individual is an instance of a specific category or not. This disagreement is not necessarily obvious and can be hidden in complex assertions about an individual.

ferent conceptualizations. However, most biomedical ontologies do not permit this derivation because they do not use negation. Without the use of negation, deriving inconsistencies is impossible. As such, they are not suitable for automatically enforcing the use of a common conceptualization.

Top-level ontologies, on the other hand, use rich axiomatizations that permit the detection of inconsistencies. However, they usually do not contain domain knowledge and are therefore only of limited use for enforcing a single conceptualization in domain-specific terminologies and knowledge bases.

An upper domain ontology for biology, however, is a bridge between a top-level and the biological domain ontologies, and provides definitions for general domain-specific terms and some constraints that are specific to the biological domain.

6.2 BOWiki

6.2.1 Motivation

The use of ontologies for the description of biological knowledge has increased rapidly as the community has recognized the value of this approach. Annotating biological data with ontological terms provides an explicit description of some of the data's features.

Developing and maintaining the ontologies in biomedicine requires manual creation, deletion and correction of concepts and their definitions within the ontology, as well as annotating biological data to concepts of the ontology. While the development and maintenance of the ontologies themselves is almost exclusively performed manually, the annotation of data to ontologies can be automated [Fleischmann et al., 1999]. However, the quality of automatic annotations remains inferior to manual annotation. As increasing quantities of data are generated and published, large-scale *manual* annotation becomes increasingly time-consuming and costly.

Several authors suggest using a community-based tool such as a wiki for the description, discussion and annotation of the functions of genes and gene products [Wang, 2006, Hoehndorf et al., 2006, Giles, 2007]. A wiki is a collaboratively maintained website, that can be modified by all its users [Leuf and Cunningham, 2001].

Using a wiki for annotating biological data with an concepts of ontologies could shift the work from few curators to a large number of scientists. Many are experts in specific sub-domains of biology. However, these specialists are not necessarily experts in ontology curation and annotation. Therefore, the use of a freely accessible wiki introduces additional difficulties for maintaining the correctness and consistency of added data, and for accurately representing biological information.

The information represented in the wiki should adhere to criteria of quality, such as internal consistency (the content of the wiki does not contain contradictory information) and consistency with biological background knowledge (the content of the wiki should be semantically correct). In order to ensure internal consistency, logic-based tools can be employed to detect contradictory information. To support consistency with biological background knowledge, parts of this background knowledge must be formalized as a logical theory. Once this is achieved, it becomes possible to use automated reasoners for verifying consistency between the knowledge compiled in a collaboratively developed knowledge base and the formal theory of the biological background knowledge.

A starting point for formalizing biological background knowledge can be found in core ontologies [Valente and Breuker, 1996]. Core ontologies are formal theories about basic types of entities and their interrelations within a domain. We expect that core ontologies are more robust and stable, and achieve a higher degree of support and agreement among the participants of a community, than more specialized ontologies.

We have developed the BOWiki, a wiki system based on the application of a core ontology together with an automated reasoner that can maintain a consistent and correct knowledge base. It is specifically targeted at small- to medium-sized communities for the collaborative annotation of data with concepts of imported ontologies and their integration.

6.2.2 Functionality and Implementation

The BOWiki is a semantic wiki based on the MediaWiki [Wikimedia Foundation, 2008] software. Wikis began as web-based software that permit the collaborative development of text-centered resources [Leuf and Cunningham, 2001]. Semantic wikis extend this idea through functions that maintain parts of the wiki content in the form of structured data [Völkel et al., 2006]. This allows for improved information processing, e.g., by querying the data collection. For instance, *inline queries* [Völkel et al., 2006] can be added to the source code of a wikipage, which produce an integrated form of displaying query results on a wikipage.

While a standard wiki allows for the creation of wikipages and links to other wikipages, it remains unclear what *type of entity* a wikipage describes and what *relation* a hyperlink represents. The explicit specification of types and relations can be exploited for diverse problems, e.g., to connect the domain knowledge created within a wiki with other knowledge- and databases, or to perform complex queries and ensure internal consistency.

Within our MediaWiki extension, users can specify the type of the entity described by wikipage (see table 6.1). One of the central ideas of the BOWiki is to be equipped with a pre-defined set of types and relations (and corresponding restrictions among them). The types and relations should be the basic categories within the domain of application. They should form a core ontology [Valente and Breuker, 1996]. We deliver the BOWiki with the biological core ontology GFO-Bio [Hoehndorf et al., 2007], but any OWL file can be used as the type system. The types in the OWL file should form a core ontology for the application domain.

For the purpose of automated reasoning over such a core ontology, the BOWiki requires the core ontology in the form of an OWL-DL [Mcguinness and van Harmelen, 2004] ontology. Types are modelled as OWL classes and binary relations as OWL properties (using OWL datatype properties is possible, and accordingly, XML Schema datatypes may be used as type restrictions for relation definitions in wiki syntax). Relations of higher arity are modelled according to the third use case in [Noy and Rector, 2006], i.e., as classes whose individuals model relation instances. Interconnections among types and relations are formulated by means of OWL expressions. Wikipages as (descriptions of) instances of types give rise to OWL individuals, which are members of the OWL classes that correspond to their types.

A core ontology in OWL provides background knowledge about the domain in the form of axioms that restrict the basic types and relations within it. To be usable for the BOWiki, the core ontology must satisfy certain adequacy conditions related to the domain's conceptualization. This allows for automatic verification of the content created in the BOWiki: users may introduce a new

BOWiki syntax	**OWL abstract syntax**
Generic	
1 `[[OType:C]]`	Individual(**page** type(C))
2 `[[R::page2]]`	Individual(**page** value(R page2))

3 `[[R::role1=page1;...;roleN=pageN]]`	Individual(R-id type(R))
	Individual(R-id value(subject **page**))
	Individual(R-id value(R-role1 page1))
	. . .
	Individual(R-id value(R-roleN pageN))

4 `[[has-argument::`
`name=roleName;type=OType:C]]`
	SubClassOf(**page** gfo:Relator)
	ObjectProperty(R-roleName domain(**page**) range(C))

Examples	
1 on page Apoptosis: `[[OType:Category]]`	Individual(Apoptosis, type(Category))
2 on page Apoptosis: `[[CC-isa::Biological_process]]`	Individual(Apoptosis value(CC-isa Biological_process))

3 on page HvSUT2:
`[[Realizes::`
`function=Sugar_transporter_activity;`
`process=Glucose_transport]]`
	Individual(Realizes-0 type(Realizes))
	Individual(Realizes-0 value(Realizes-subject HvSUT2))
	Individual(Realizes-0 value(Realizes-function Sugar_transporter_activity))
	Individual(Realizes-0 value(Realizes-process Glucose_transport))

4 on page Realizes:
`[[has-argument::`
`name=function;`
`type=OType:Function_category]]`
	SubClassOf(Realizes gfo:Relator))
	ObjectProperty(Realizes-function domain(Function_category))

Table 6.1: Syntax and semantics of the BOWiki extensions. The table shows the syntax constructs used in the BOWiki for semantic markup. The second column provides a translation to OWL. (**page** refers to the wikipage in which the statement appears; “R-id” is a name for an individual whose “id” part is unique and generated automatically for the occurrence of the statement). Because OWL has a model-theoretic semantics, this translation yields a semantics for the BOWiki syntax. In the lower half of the table we illustrate each construct with an example and present its particular translation to OWL.

page in the wiki and describe some entity; they may then add type information about the described entity; and this added type information is then automatically verified. The verification checks the logical consistency of the BOWiki's content – as OWL individuals and relations among them – with the restrictions of core ontology types and relations, like those in GFO-Bio.

The BOWiki uses a description logic [Baader, 2003] reasoner to perform these consistency checks. A layer of abstraction is needed between the BOWiki application and the description logic reasoner in order to support more than one reasoner. While the DIG protocol [Bechhofer, 2003] provides such an abstraction layer and is implemented by many description logic reasoners, it does not support operations required by the BOWiki. Among the missing features are the removal of instances, rollbacks of the knowledgebase or explanations of detected inconsistencies. In order to address these problems, we implemented the BOWikiServer, a stand-alone server that provides access to a description logic reasoner using the Jena 2 Semantic Web Framework [Carroll et al., 2003] and a custom-developed protocol [Hoehndorf, 2007]. A schema of the BOWiki's architecture is illustrated in figure 6.2.

Whenever a user edits a wikipage in the BOWiki, the consistency of the changes with respect to the core ontology is verified using the BOWikiServer. Only consistent changes are permitted. In the event of an inconsistency, an explanation for the inconsistency is given, and no change is made until the problem is resolved by the user.

In addition to verifying the consistency of captured knowledge with respect to a core ontology, the BOWikiServer can be employed to perform complex queries over the data captured within the wiki.

Inline queries are performed as retrieval operations for description logic concepts [Baader, 2003], i.e., as queries for all individuals that satisfy a description

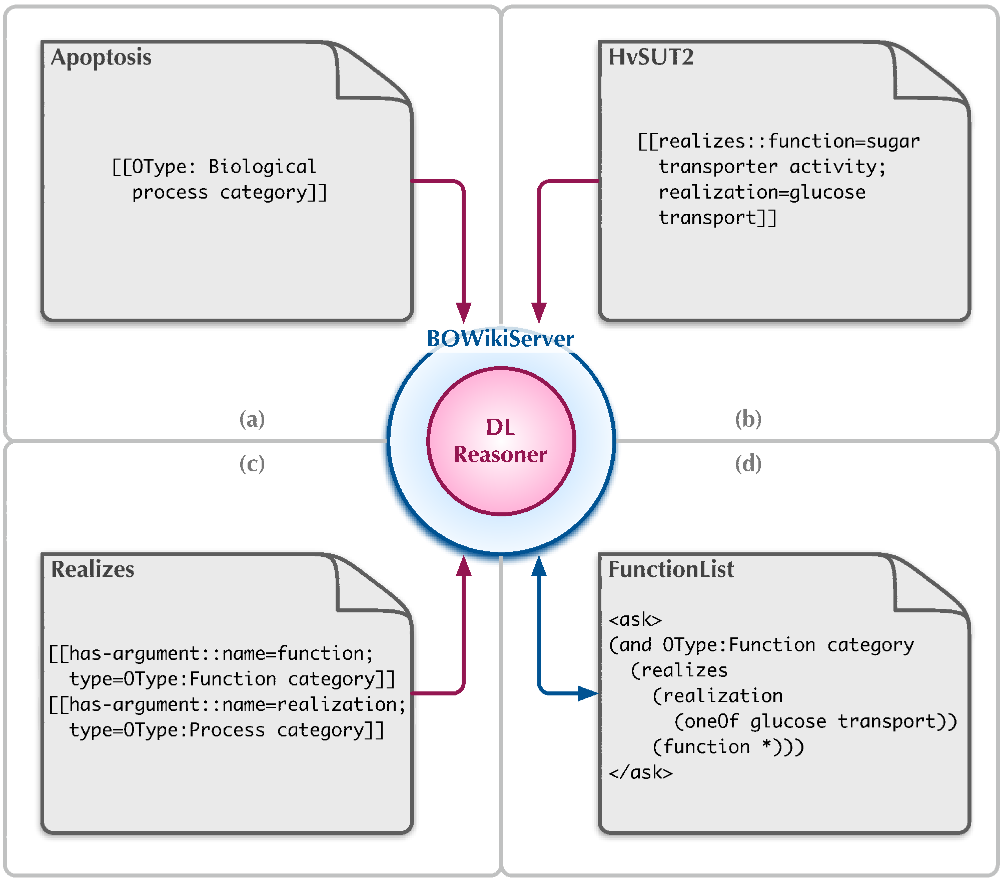


Figure 6.1: Overview of basic BOWiki functionality. (a) The `OType` statement is used to declare the entity described by a wikipage to be an instance of a certain type. The syntax for using a defined relation is shown in (b). Note that the *subject* role is implicitly filled in by HvSUT2, since the relation is used on this page. The relation **realizes** is therefore a ternary relation. (c) A relation's arguments are defined using the `has argument` statement. The example shows the definition for two roles and their restriction to specific OWL categories. An inline-query for all functions realized by *Glucose transport* appears in part (d).

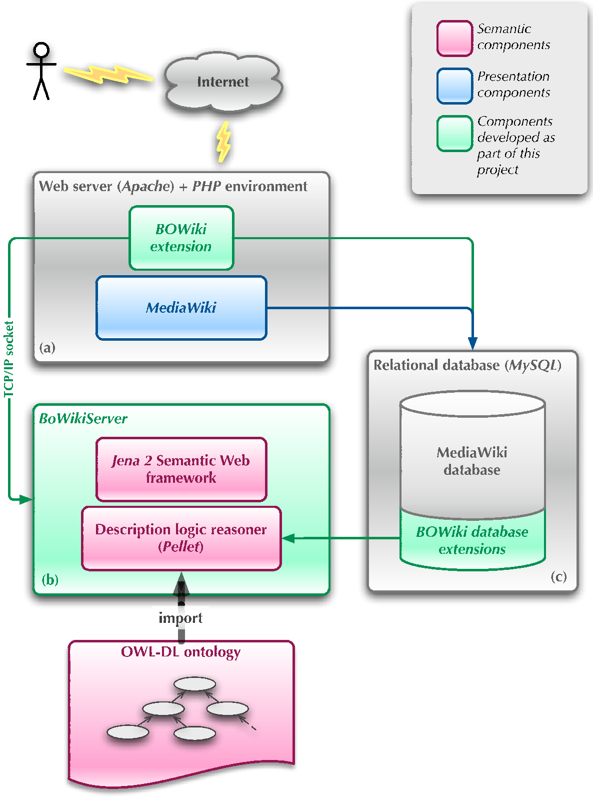

Semantic components
Presentation components
Components developed as part of this project
Internet
Web server (Apache) + PHP environment
BOWiki extension
MediaWiki
(a)
TCP/IP socket
Relational database (MySQL)
MediaWiki database
BOWiki database extensions
(c)
BoWikiServer
Jena 2 Semantic Web framework
Description logic reasoner (Pellet)
(b)
import
OWL-DL ontology

Figure 6.2: BOWiki Architecture. (a) The BOWiki extension to the MediaWiki software processes the semantic data added to wiki pages. The semantic data is subsequently transferred to the BOWikiServer using a TCP/IP connection. (b) To evaluate newly entered data or semantic queries, the BOWikiServer requires an ontology in OWL-DL format (provided during installation of the BOWiki). Consistent semantic data will be stored. If an inconsistency is detected, the edited page is rejected with an explanation of the inconsistency. The BOWikiServer currently uses the Jena 2 Semantic Web framework together with the Pellet reasoner. (c) After successful verification the semantic data is stored in a separate part of the SQL database.

logic concept description. An example of such an *inline query* is shown in figure 6.1 (d).

In a performance evaluation of our implementation, we obtained good results for knowledge bases with several thousand individuals (a wiki with several thousand wikipages). The time required to add individuals (wikipages) to the knowledge base of the BOWiki increases linearly with the number of individuals, and was between one second (for an empty knowledge base) and 4 seconds (for 3000 individuals) in our tests. Complex queries require about the same time as adding individuals, and the time increases linearly with the number of individuals. The limiting factor in the number of individuals and relations within the BOWiki appears to be system memory: for 3000 individuals with few relations among them, 3GB of system memory were required. Possibilities for improving performance are discussed in a later section.

6.2.3 Application of the BOWiki

The BOWiki is a semantic wiki that can be specialized for a domain. While semantic wikis allow for the structured representation of information, they provide little or no quality control, and no assistance to users in verifying the consistency of captured knowledge. An upper domain ontology provides background knowledge about the domain, which the BOWiki can use to verify the correctness of its content with respect to the provided domain knowledge. The upper domain ontology, together with a reasoner, therefore provides a form of quality control for the BOWiki content.

We envision two main applications for the BOWiki in the biomedical domain, the annotation of data and the integration of knowledge bases.

Annotating data

In conjunction with a biological core ontology like GFO-Bio [Hoehndorf et al., 2008a], BioTop [Schulz et al., 2006b] or the Simple Bio Upper Ontology [Rector et al., 2006b], the BOWiki can be used to annotate data. For this purpose, we developed a module that allows the import of OBO ontologies [Smith et al., 2007] in the OBO Flatfile Format [Golbreich and Horrocks, 2007] into the BOWiki. By default, these ontologies are only accessible, but not considered in the reasoning of the BOWiki. Users can create wikipages containing information about biological entities, and describe the entities both in text and in a structured form on these wikipages using relations available within the BOWiki.

In contrast to annotating data with ontological categories, i.e., the assertion of an arbitrary association relation between some biological data and an ontological category, it is possible in the BOWiki to make the relation between a biological entity (e.g. a protein) and a category precise: a protein may not only be **annotated** to *Transcription factor activity*, *Nucleus*, *Sugar transport* and *Glucose*: it stands in the **has function** relation to transcription factor activity; it can be **located at** a nucleus; it can **participate in** a *Sugar transport* process; it can **bind** glucose. The ability to make these relations explicit renders annotations using a semantic wiki both exceptionally powerful and precise.

Integrating knowledge bases

The BOWiki can also be used to connect different ontology-based knowledge bases using explicit relations. It is possible to create explicit (partial) definitions for terms from ontologies using terms and relationships from other ontologies. This may be useful for creating so-called cross-products [Smith et al., 2007] between different ontologies. Cross-products define categories in one ontology using relations and categories from other ontologies. Currently, such cross-products are created by a few ontologists using text-extraction and manual curation [Bada and Hunter, 2007]. A community effort to create the appropriate relations and definitions may contribute to the more rapid creation of these cross-products, and to a richer selection of such cross-products. In addition, the BOWiki provides quality control for the creation of these definitions by verifying both internal consistency and consistency with a background ontology.

For example, the category *Germ cell migration* can be defined as an instance of *Cell migration category* which **is-a** *Cell migration* and **results in movement of** *Germ cell* [Mungall, 2007]. Here, *Cell migration category* is a category that has as instances only (and all) categories that are subcategories of *Cell migration*. This kind of meta-instantiation is available in the General Formal Ontology (GFO) and the biological core ontology GFO-Bio, and has been applied to the integration of anatomy and phenotype ontologies [Hoehndorf et al., 2007].

6.2.4 Discussion

Comparison with other approaches in biology

WikiProteins [Giles, 2007] is a software project based on the MediaWiki software [Wikimedia Foundation, 2008] aimed at using a wiki for the annotation of Swissprot [Boeckmann et al., 2003]. Similar to the BOWiki, it utilizes ontologies like the Gene Ontology [Ashburner et al., 2000] as a foundation for the annotation. However, WikiProteins does not include a description logic reasoner to retrieve or verify information, and therefore lacks the features of quality control and retrieval that are central to the BOWiki.

The Semantic Mediawiki [Völkel et al., 2006] is a semantic wiki, also based on the Mediawiki software [Wikimedia Foundation, 2008]. It is designed to be applicable within the online encyclopedia Wikipedia[2]. Because of the high number of Wikipedia users, performance and scalability requirements are of much greater importance for the Semantic Mediawiki than they are for the BOWiki. Therefore, they also do not provide a description logic reasoner or ontologies for content verification. Furthermore, their underlying datamodel is RDF [Beckett, 2004], which only permits binary relations. The BOWiki is based on GFO's theory of roles and relations [Loebe, 2007, Herre et al., 2006], and supports relations with any number of arguments.

Other semantic wikis such as pOWL [Auer, 2005] or OntoWiki [Auer et al., 2006] allow for the modification and creation of description logic knowledge bases or editing their instances. These wikis almost exclusively use RDF or OWL as a knowledge model. The BOWiki uses an ontologically founded data model, that can be converted to either OWL or RDF, but also to other languages. While the idea of using upper domain and core ontologies for type-checking in a wiki is not new [Vrandecic and Krötzsch, 2006, Hoehndorf et al., 2006], there is currently, to the best of our knowledge, no other semantic wiki that incorporates a description logic reasoner for verification or retrieval of content.

Quality control

A wiki allows for the quick correction of errors in its content, and the BOWiki implements an additional form of quality control using a description logic reasoner. The reasoner identifies logical inconsistencies in the semantic information added to the BOWiki by referring to the knowledge provided in the form of a core ontology or upper domain ontology. There are several upper domain

[2] `http://www.wikipedia.org`

ontologies in biology that can be used for this purpose [Rector et al., 2006b, Schulz et al., 2006b, Hoehndorf et al., 2008a].

While these ontologies contain only a few (between 100 and 500) categories that can be used as types in the BOWiki, we believe that their use may help to provide better-integrated and coherent knowledge resources. The use of types in the BOWiki forces users to use relations in a similar way. This will facilitate both the retrieval and sharing of the information contained in the BOWiki.

Using description logic reasoners

The BOWikiServer provides a layer of abstraction between the description logic reasoner and the BOWiki. Depending on the description logic reasoner used, different features can be supported. Different reasoners support different expressivity, and therefore more or less stringent checking can be employed, permitting the increase of data that is processable by the BOWiki. Currently, the BOWikiServer uses the Pellet reasoner [Sirin and Parsia, 2004]. Pellet supports the explanation of inconsistencies, which can be shown to users to help them correct any inconsistent statements submitted to the BOWiki. It also supports the nonmonotonic description logic ALCK with the auto-epistemic **K** operator [Donini et al., 1997]. This permits the combination of both open- and closed-world reasoning [Reiter, 1980].

On the other hand, reasoning in the description logic fragment of OWL [Mcguinness and van Harmelen, 2004] is NExpTime-complete [Schaerf, 1994]. As our performance tests have shown, only small to medium-sized knowledge bases are currently supported by the BOWiki. Our tests were performed using the Pellet reasoner. While it supports several features that are beneficial for the BOWiki, it may be replaced by different, perhaps more efficient description logic reasoners like Fact++ [Tsarkov and Horrocks, 2006b], which does not provide as many features as does Pellet. It is also possible to use a reasoner for a weaker logic like OWL-Lite [Mcguinness and van Harmelen, 2004] or RDFS [Brickley and Guha, 2004], but expressivity in the type system would be sacrified for higher performance. In addition, several projects attempt to implement description logic reasoners that are capable of handling large ABoxes [Bechhofer et al., 2005, Chen et al., 2005]. Since the BOWiki mainly supports the modification of an ABox, using these systems may help to further improve the performance.

6.3 Social Tagging

In the BOWiki, structured content is added in the form of relations in which all arguments are fixed. A weakened form of acquiring structured knowledge is the used of only partially specified relations. In these, the arguments are not fixed, but identified by free-text keywords chosen by a user. Similarily, the kind of relation may not be fixed but only partially specified or unknown. A form of personal knowledge management that incorporates some of these features is *tagging*.

Tagging refers to the association of a set of keywords with some object. Collaborative social tagging enables multiple users to individually tag objects and share the tagged objects or the tags for these objects. There are an enormous number of available systems for the collaborative tagging of entities. Users can tag movies (YouTube[3]), pictures (Flickr[4]), Websites (del.icio.us[5]) or scientific

[3] http://www.youtube.com

[4] http://www.flickr.com

[5] http://del.icio.us

documents (CiteULike[6]). Depending on the *type* of tagged object, different tagging platforms are implemented.

Tagging is primarily intended as a form of personal information management or to easily annotate entities for information retrieval and information sharing [Marlow et al., 2006]. It is not intended as a form of knowledge acquisition, and users of a tagging system are rarely consciuously contributing to the creation of a knowledge base through their tagging. Nevertheless, tagging creates informal vocabularies (folksonomies) that can be further analyzed [Mathes, 2004, Wu et al., 2006]. Here I will defend the claim that an ontological analysis of the entities that participate in a tagging event helps to formalize the meaning of a tag, and provides additional benefits to the tagger.

6.3.1 Problem statement

The type of the tagged object determines the attributes that are stored with it. For documents, these may be the author, date of publication, journal, etc. For a webpage, it may be its URL, for photographs the type of camera used to take it or for movies the actors and director.

Depending on the type of tagged object, the tags may describe different aspects or *facets* of it. Some tagging systems support the use of facets: tags can be associated to different aspects of the tagged object. Often, several default facets like “theme” or “topic” are used. While these facets are common to many tagging systems, several possible facets depend on the type of tagged object: videos may not only have a topic, but also temporal duration or temporal parts. Photographs have color-schemes. Molecules have functions, structural parts, shape and weight.

[6] `http://www.citeulike.org`

I describe a collaborative tagging system[7] that allows for tagging objects of different types. Depending on the type of tagged object, different information about the object is stored. In addition, tags can be associated to *facets* of objects. Some facets depend on the type of tagged object, while others are applicable to any tagging action.

6.3.2 Basic design: Tagging core ontology and foundation in GFO

I outline the tagging ontology of [Uciteli, 2008], which forms the foundation of the tagging software discussed here[8]. It is based on [Newman, 2005] and the GFO [Herre et al., 2006].

A basic entity in the tagging domain is *Tag*, which is the role played by the string a tagger enters during a tagging action. The tag is a concrete individual. It instantiates a special kind of category, a *Symbol structure*. The tag is a **token-of** the symbol structure.

The tag is associated with an object. This object can be any entity. While the object referred to by a URI is often identified with its URI [Newman, 2005, Pepper and Schwab, 2003], in the ontological analysis of tagging, it becomes important to distinguish between the object **described by** a URI and the URI itself. A tag can relate to either of these, and the nature of this relationship differs. Therefore, [Uciteli, 2008] distinguishes two kinds of entity, an information object containing information about some other entity, and the entity that is described by the information object.

[7] `http://bioonto.de/pmwiki.php/Main/CollaborativeTaggingSystem`

[8] The Tagging Ontology can be found in OWL format at `http://bioonto.de/uploads/Main/gfo_tag_ont.owl`.

It is assumed in [Uciteli, 2008] that tags are always associated to objects, not information resources describing them. In order to specify the way that some entity relates to whatever is denoted by a tag, *facets* are introduced. Facets are relationships that an entity can have to other entities. For example, physical objects can have a **part-of** facet, a **participates-in** facet, but also facets relating it to categories, like an **instance-of** facet. According to the tagging ontology, every entity can be denoted by some other entity (an information resource). When the denotation itself is used as a facet and combined with relations available for information resources, entities can be tagged so that the meaning of the tag relates to the information resource describing the entity. Figure 6.3 shows an example RDF graph of a tagging event.

Tagging is an intensional act, i.e., it involves a conceptualization of the tagger. In particular, not every instance of the same symbol structure is used to denote the same thing. Therefore, different taggers associate different concepts with tokens of a symbol structure. For example, a German tagger using the tag “tag” probably refers to a “day” and not a “tag”. The domain and range restrictions of the relations used to construct facets for tagged entities can also be used to clarify the different meanings of tags depending on the tagger.

While [Uciteli, 2008] provides a comprehensive core ontology for the tagging domain, most tagging systems do not use all of the features described in the tagging ontology. For example, faceted tagging systems are rarely employed, concepts are almost never used and the distinction between an entity and the information resource that describes it is seldom explicated. However, this tagging ontology provides a means for analyzing collaborative tagging systems. Even if only a fragment of the ontology is used as the conceptual schema of a concrete implementation, the ontology can be used to share information with other tagging systems, if they employ a similar schema.

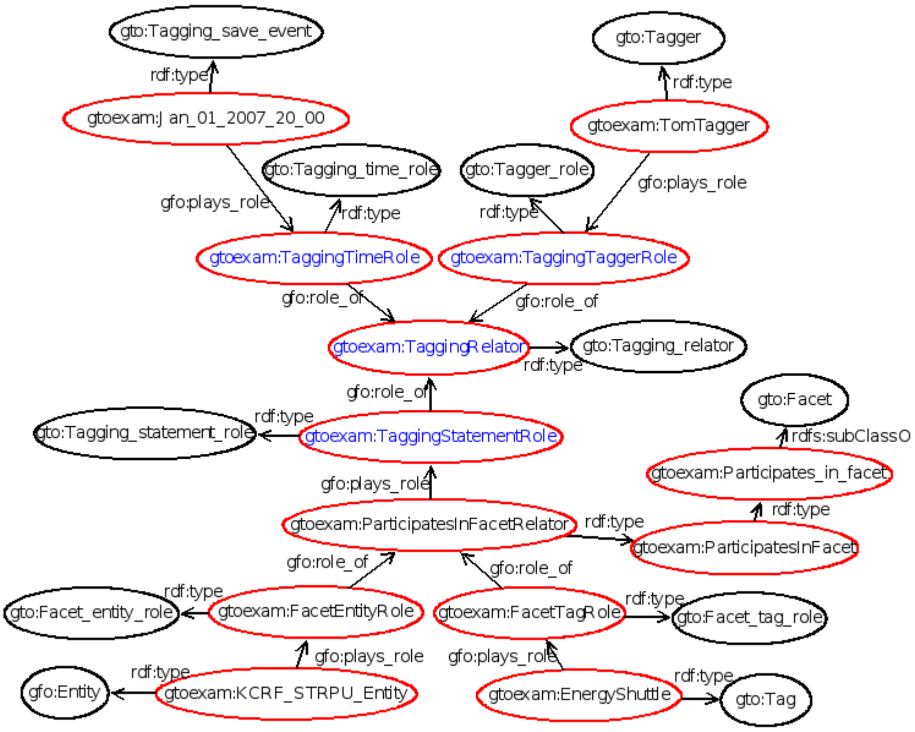


Figure 6.3: The figure (from [Uciteli, 2008]) shows a tagging of the protein KCRF by Tom using the **participates-in** facet.

6.3.3 Application: GFO-Bio

A domain ontology like GFO-Bio can be used to specialize the tagging software to a domain. In particular, it can be used to generate the facets applicable to the tagged object, and the properties of the tagged object. The domain ontology provides the knowledge about the entities that play the role of the tagged object in the tagging relation.

An example application in the biological domain, described in [Uciteli, 2008], is the organization of information about proteins and related kinds of entities. Researchers investigate proteins from different perspectives and associate them with different features depending on the kind of interest they take in a protein. A shared organization of proteins according to freely chosen keywords permits the agile organization of information pertaining to these proteins.

To describe different facets of the tagged proteins, relations in GFO-Bio plus the relations in the tagging ontology are used. These provide basic aspects of descriptions of a protein. GFO-Bio provides ontological relations that permit relation a protein to the processes in which it participates or the functions a protein can have. The tagging ontology that comes with the tagging software provides facets that associate a protein with meta-information like a webpage or publication describing the protein.

The specification of a facet between a protein or other kind of molecule and the tag is voluntary. In large-scale applications it suffices to have a minority of users use the facet features of the tagging software. Since the keywords associated with a tagged entity will often be shared by different users, the facets can be infered by the facets used by other users to relate a protein (or a similar protein) with the same tag.

To completely formalize the knowledge acquired through the use of the ontology-based tagging system described here, the tags used for each tagged entity must be further analyzed and their referents identified. The facets used to tag an entity can provide a starting point. However, this analysis will most likely have to be performed manually, assisted by methods from natural language processing to automatically identify or suggest an entity refered to by a set of keywords.

6.3.4 Implementation

A prototype of the tagging system described in [Uciteli, 2008] was implemented and the prototypical implementation uses GFO-Bio in conjunction with the tagging ontology. In addition to common components of tagging systems such as user management and means for sharing information, the implementation contains features that permit the extraction of facets from ontologies in the OWL format. The implementation is ontology-based and represents its content in the GFO tagging ontology [Uciteli, 2008].

The architecture of the tagging software is divided into two major components: a configuration component in which the tagging system is initialized with an OWL ontology, and modules that are used during the runtime of the tagging software.

During initialization of the tagging software, the types of objects that can be tagged are determined and the facets that are applicable to these types generated. For this purpose, an OWL file must be provided that contains classes and relations. A list of all OWL classes is generated. From this list, one or more classes must be selected to provide the basic types for the tagging software. These types of objects can be tagged. Using the Pellet [Sirin and Parsia, 2004] reasoner, all facets that are applicable to these kinds of entities are generated.

For this purpose, the relations in the provided OWL ontology and the GFO Tagging Ontology are used. The facets that are applicable to the types selected for tagging are the relations that permit instances of these types as arguments. Based on the selection of types and facets, a database schema is generated.

To improve performance, the tagging software uses a database in addition to the OWL file that was supplied during the initialization phase. Tags are classified based on the GFO Tagging Ontology and the supplied OWL file. Additionally, the same content is stored in a database that partially mirrors the OWL files used[9]. A library of screen scrapers[10] can be employed to automatically obtain information about the tagged object from websites. Screen scrapers must be indexed with the OWL class or classes for which they are able to obtain data.

The prototype implementation of the tagging software[11] uses GFO-Bio and employs screen scrapers for data about proteins and protein domains. The screen scrapers obtain their data from UniProt [Consortium, 2007]. Based on GFO-Bio, protein can have parts, participate in processes, be located in structures or have functions. These are generated as facets that are applicable to proteins. The GFO Tagging Ontology defines additional relations, i.e., a time of the tagging event or a document that describes the tagged object. Based on these, additional facets are generated.

A tagger can tag a protein or a protein domain. For this purpose, the tagger can use one of the screen scrapers that are supplied for the *Protein* or the *Molecule* OWL classes. The tag can be classified using a facet, i.e., the **participates-in** facet. Selection of a facet is optional. When the **participates-in** facet is

[9] Only a part of the database schema is generated from the OWL ontologies. Other aspects store information about users and passwords, or other aspects of the tagging software's model.

[10] A screen scraper is a program that extracts structured data from the output of another program, usually a web browser.

[11] `http://bioonto.de/pmwiki.php/Main/CollaborativeTaggingSystem`

used, it is assumed that the tag refers to a process (based on **participates-in**'s domain and range restrictions in GFO-Bio). A more elaborate description of the tagging software and more examples can be found in [Uciteli, 2008].

6.4 Information extraction and text-mining

Both the BOWiki and the collaborative tagging system we developed require the manual assertion of knowledge by human agents. Most of the information is already present in publication in scientific journals. If this information could be extracted and formalized, it would provide a rich source of knowledge without the manual effort of humans.

A large body of biological knowledge has been accumulated in scientific literature. These articles provide a large resource for the acquisition of knowledge and the extraction of formal biological theories. The knowledge contained in literature contains more facts than biomedical databases, and covers a larger period of time, therefore permitting not only the construction of single scientific theories, but also the analysis of their evolution over time.

Literature is intended for humans. Understanding and interpreting natural language is complicated by unclear semantics of natural language statements, context dependency, different and unspecified terminology or the evolution of term meanings. Therefore, methods to automatically obtain facts from literature often focus on narrow tasks and specialized methods.

Methods from computer linguistics, natural language processing and text-mining are used to extract biological facts or full-fledged biological theories from texts in natural language. In the context of biomedical ontologies, three tasks are of particular relevance.

i) The first is the automatic creation of formalized ontologies from text. Several steps would be involved in such an endeavour: identifying important concepts within the domain, identifying relationships, generating definitions for both these concepts and relationships, and finally generating an axiom system or multiple axiom systems for the remaining, undefined relations and concepts. Although significant progress has been made in this area [Brewster et al., 2008], the problem of generating ontologies from text is far from being solved, and no established method currently exists in the domain of biology.

ii) Another problem for which linguistic methods are being proposed is the automated extraction of annotations from text. This task has primarily been investigated for annotations of gene products with GO categories. The BioCreative evaluation challenge [Hirschman et al., 2005b] evaluated how well various methods solved this task. However, the evaluated methods did not produce results that came close to human annotators.

   Identifying annotations from text can be broken down into several subproblems: identifying the occurrence of a GO category in the text, identifying the occurrence of a gene product name in text, and identifying the kind of relationship between both from the text in order to find out whether or not an annotation should be generated. Each of these problems is being still researched. For example, the state of the art for identifying gene names in text is an F-measure[12] of 0.92 for yeast genes and 0.79 for mouse genes [Hirschman et al., 2005a]. Identifying category names from the GO reaches an F-measure of 0.34 for untrained methods [Gaudan et al., 2008] and up to 0.9 for methods that employ machine learning [Leaman and Gonzalez, 2008]. Extracting annotations from text is more

[12] The F-measure is the harmonic mean between precision and recall. Precision is the fraction of the retrieved results that are correct with respect to the task. Recall is the fraction of relevant results that is correctly retrieved.

difficult and requires both the identification of occurrences of gene names and GO category names as well as identifying the relation between them [Blaschke et al., 2005].

iii) Finally, literature can be used as the basis for identifying mappings or alignments between ontologies. These alignments are called *cross-products* within the OBO Foundry project. In the context of text mining this task is called *relationship extraction*. While many of these cross-products are created through a manual curation effort, some methods use the textual definitions of the ontology's categories, their names or scientific literature. Again, the occurrence of a category in text must first be recognized for text-based methods, and the kind of relationship identified. This relationship must then be further investigated in order to create axioms for it (which is not necessary when only annotations are extracted). Finally, the resulting facts must be verified for their consistency with established ontologies in the field.

The general task of extracting relationships between categories from text can be divided into several steps:

1. Identify occurrences of entities in text[13].
2. Identify occurrences of relationships in text[14].
3. Identify the relationship that holds between two identified categories.
4. Test the relevance of the detected information.
5. Verify the extracted information and integrate in an ontological model.

[13] An entity occurs in a text if the interpretation (semantics) of the text must refer to the entity. In particular, this goes beyond the recognition of a *name* as in the task of *named entity recognition*. Additionally, most entities (in particular categories) are considered *intensional* entities, and reference to entities must be understood as reference to the entities' *intension*.

[14] The same conditions apply for the occurrence of relations as do for the occurrence of other entities.

For the task of ontology alignment, the entities identified in the first step will be ontological categories and the relations identified in the second step ontological relations. In the third step, the instances of the relations are extracted. In RDF, these instances would be represented as triples, but the instances extracted will be in general instances of relations with higher arity[15]. In step four, the information that is extracted is analyzed for its relevance. Depending on the method chosen for performing the first three steps, this step could be omitted. In the last step, the extracted information is verified and embedded in an ontologically founded knowledge base.

I participated in the development of software to detect categories and relationships in text as well as a method and software to measure the relevance of an extracted relation or association. Furthermore, in line with the aim and underlying theme of this thesis, I have participated in the development of a method for integrating results of a text mining analysis in a formal ontological framework. This integration serves as the foundation for verifying the extracted information.

6.4.1 Named-Entity Recognition for ontological categories

Names of categories in biomedical ontologies are complex, dense and descriptive. They are not usually *used* within a text. It is therefore a particularly difficult problem to identify the occurrence of a category in a text. While most methods that address this problem are based on a combination of pattern matching and measures of information content [Ruch, 2005, Gaudan et al., 2008, Doms and Schroeder, 2005, Couto et al., 2005], we have developed an alternative method to identify category names in text. The method is based

[15] Furthermore, an instance of a relation in the GFO is described using relational roles that specify the mode of the relation's argument's participation in the relation.

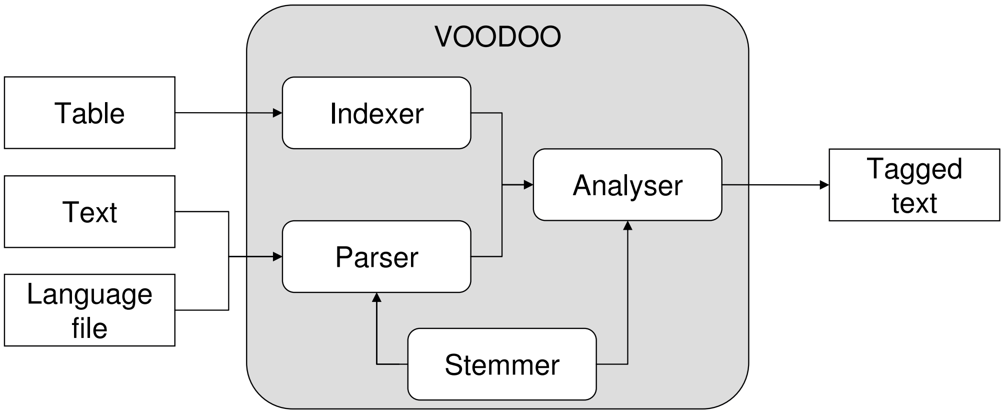


Figure 6.4: Flow diagram of the steps performed by VOODOO. The input consists of a text in which VOODOO detects category names, and a file containing the names and their synonyms. The text is parsed using the Stanford statistical parser and the parsed sentences are subsequently stemmed using the Porter stemming algorithm. The vocabulary is first stemmed and then used to generate a set of bidirectional index maps: a map between each stemmed word and its numerical index, a map between a numerical word index and a multiword term, and a map between a multiword term and a category identifier. These maps are then used to index the stemmed forms of the parsed sentences. The sentences, their dependency parse trees and the index maps generated from the vocabulary are subsequently used by the analyzer to generate the tagged text output.

on dependency parsing, stemming and pattern matching and can generally be applied to identify multi-word names or phrases in a text.

The method used in the VOODOO software combines stemming, pattern matching and dependency parsing. It takes as input a text and a list of category names together with their synonyms and identifiers. VOODOO processes sentences within the text separately. Therefore, VOODOO's output contains the identified categories for each sentence.

The processing is carried out in four steps, as illustrated in the flow diagram in figure 6.4. First, all names in the input file containing the category names and identifiers for each category are tokenized, stemmed and indexed. Tokenization splits multi-word phrases into single words. These single words are stemmed using the Porter stemming algorithm with stop-words [Porter, 1997]. No word containing 3 or fewer letters is stemmed[16]. Finally, the resulting tokens are assigned an index. For example, the GO category `GO:0000354`, *cis assembly of pre catalytic spliceosome*, is first tokenized in the tokens *cis*, *assembly*, *of*, *pre*, *catalytic* and *spliceosome*. Then, these tokens are stemmed. *cis*, *of* and *pre* contain only three letters and are not stemmed. Then, the resulting tokens are assigned indices.

Second, the text in which the category names are to be identified is parsed using the Stanford Parser [Klein and Manning, 2002]. For each sentence, a dependency parse tree is generated. Figure 6.5 shows a dependency parse tree for an examplary sentence taken from the BioCreative corpus.

After the generation of the dependency parse tree, each word in the parsed sentence is stemmed using the stemming procedure used for stemming category names, i.e., using the Porter stemming algorithm with the same list of stop words.

As the final step, VOODOO identifies category names which satisfy two conditions. Given a category and a sentence, the first condition is met when all words that constitute the category's name or synonym occur in their stemmed form within the stemmed form of the sentence. For example, consider the category *Osteoclast differentiation* and here its exact synonym *Osteoclast cell differentiation* (`GO:0030316`), the category *Multinuclear osteoclast* (`CL:0000779`) and the sentence

[16] All words with three or fewer letters are considered to be on the list of stop words.

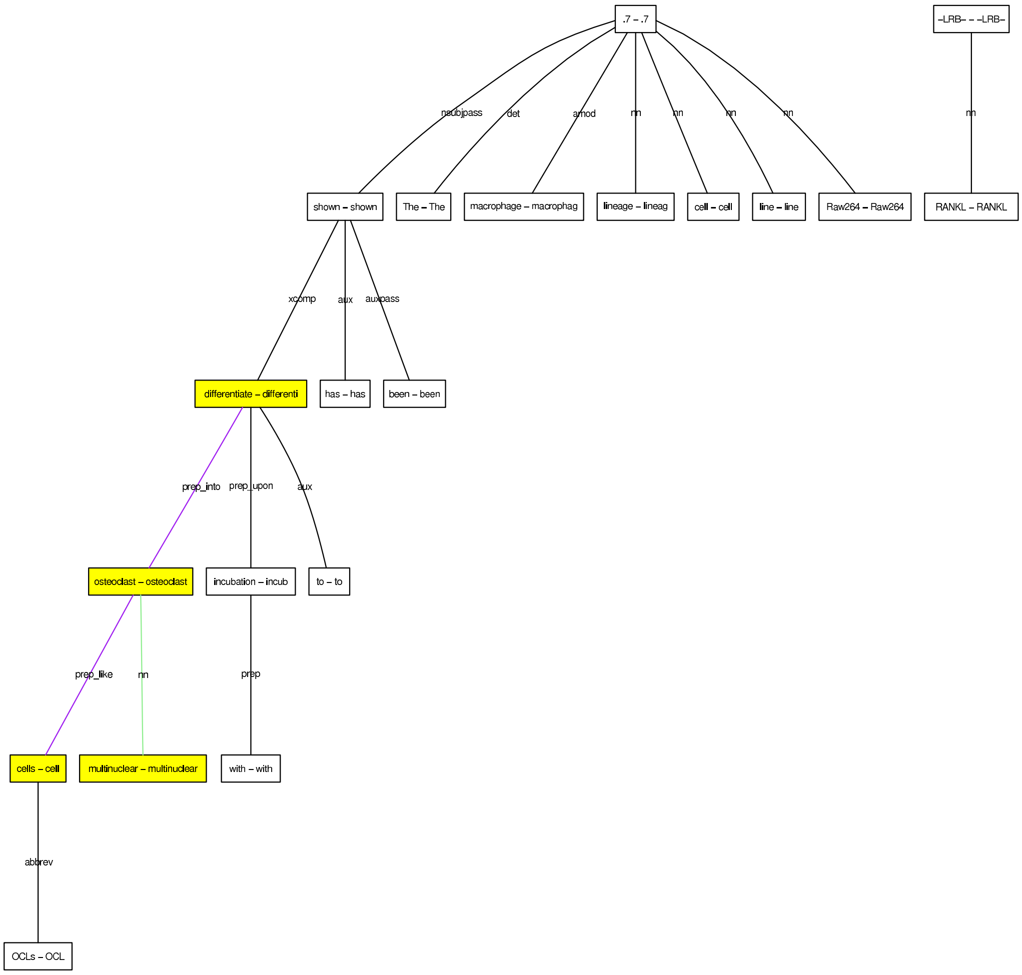


Figure 6.5: Graph structure of the sentence “The macrophage lineage cell line Raw264 .7 has been shown to differentiate into multinuclear osteoclast like cells (OCLs) upon incubation with RANKL ().”. The GO category `GO:0030316` (connected with purple edges) and CL category `CL:0000779` (connected with green edge).

> The macrophage lineage cell line Raw264 .7 has been shown to differentiate into multinuclear osteoclast like cells -LRB- OCLs -RRB- upon incubation with RANKL -LRB- -RRB- .

The stemmed form of the categories are *osteoclast cell differenti*, *multinuclear osteoclast* and the stemmed form of the sentence

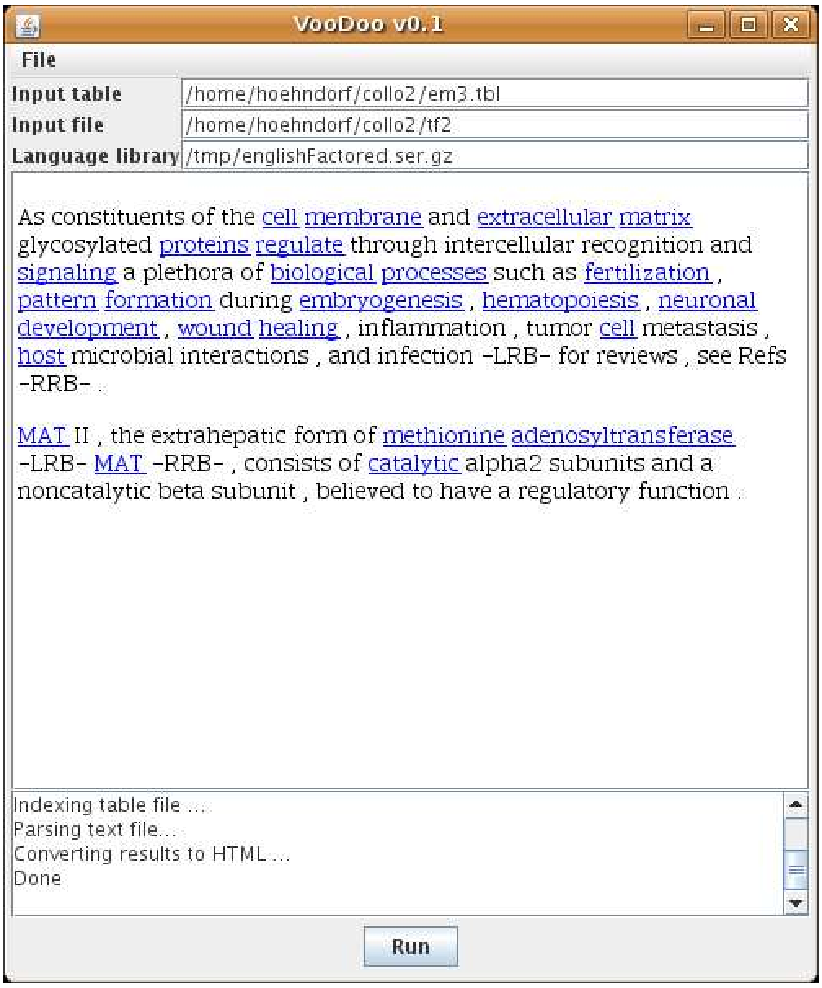
> The macrophag lineag cell line Raw264 .7 ha been shown to differenti into multinuclear osteoclast like cell -LRB- OCL -RRB- upon incub with RANKL -LRB- -RRB- .

Each token that constitutes the stemmed category name also occurs in the stemmed sentence; this satisfies the first condition.

The second condition that must be met is that all matching tokens must form a connected subgraph in the dependency parse tree of the sentence. The implications of this condition are illustrated in figure 6.5 for the category *Osteoclast cell differention* and *Multinuclear osteoclast*. We implemented the second condition to recover parts of the information that is lost when the Porter stemming algorithm is applied; connectivity also reflects the condition that multi-word terms stand for a single, biologically meaningful category.

Implementation: VOODOO

The VOODOO software is implemented in Java and Groovy. It makes use of the Stanford Parser [Marneffe et al., 2006], an entropy-based statistical parser. Each step illustrated in the flow diagram in figure 6.4 can be run individually by using either a Java package or a Groovy script. To automate the process conveniently, we provide a graphical user interface based on Java to perform all necessary steps and display the results of the analysis. A screenshot of the user interface is shown in figure 6.6. Additionally, we provide a web-interface



Input table /home/hoehndorf/collo2/em3.tbl

Input file /home/hoehndorf/collo2/tf2

Language library /tmp/englishFactored.ser.gz

As constituents of the cell membrane and extracellular matrix glycosylated proteins regulate through intercellular recognition and signaling a plethora of biological processes such as fertilization , pattern formation during embryogenesis , hematopoiesis , neuronal development , wound healing , inflammation , tumor cell metastasis , host microbial interactions , and infection -LRB- for reviews , see Refs -RRB- .

MAT II , the extrahepatic form of methionine adenosyltransferase -LRB- MAT -RRB- , consists of catalytic alpha2 subunits and a noncatalytic beta subunit , believed to have a regulatory function .

```
Indexing table file ...
Parsing text file...
Converting results to HTML ...
Done
```

Figure 6.6: A screenshot of VOODOO's graphical user interface.

at `http://onto.eva.mpg.de/VOODOO` to demonstrate VOODOO's functionality.

6.4.2 Testing the significance of extracted relations

We have developed a set of novel statistical tests that can be used to identify whether an extracted relationship is significant or detected by accident [Hoehndorf et al., 2008c]. To test the method we have applied it to co-occurrences of ontological categories in text, i.e., leaving the relation unspecified. However, with a sufficient amount of extracted instances of a particular relation, the tests can be applied to the task of relationship extraction as well.

We assume that ontologies form at least a taxonomy, i.e., in their graph representation, edges are labeled with *isA* denoting the ontological **is-a** relation. We call the set of all *isA*-successors of a category A the sub-categories $subcat(A) = \{B|isA(B,A)\}$ and its predecessors the super-categories $supcat(A) = \{B|isA(A, B)\}$. The direct successors and predecessors of A in the taxonomy are called children ($child(A) = \{B|isA(B,A) \wedge B \neq A \wedge \forall X(isA(B,X) \wedge isA(X,A) \rightarrow X = B)\}$) and parents, respectively. The test on two ontologies is based on a number of further assumptions.

1. The ontologies are represented as directed acyclic graphs G_1 and G_2 that have no nodes in common.
2. Each pair of nodes from G_1 and G_2 is connected by an inter-graph edge.
3. There is a graph decoration for each graph plus the inter-graph edges.
4. For each pair of nodes from G_1 and G_2, a scoring function generates a single real value using the graph decorations.

The first assumption is often satisfied in the case of biomedical ontologies that are represented in the OBO Flatfile Format, which specifies a directed acyclic graph. Secondly, the edges between the nodes of the two graphs represent the connections between two ontological categories. The graph decoration may contain multiple values. An example of a graph decoration is the number of occurrences and co-occurrences of each category name in a text corpus. Finally, a scoring function is used to calculate a single value for each inter-graph edge. The tests are designed to rank the values of the scoring function depending on their statistical significance.

The score between two categories C and D may be influenced by the topology of the ontology: categories that are more general may occur and co-occur more often. Therefore, it is insufficient to test for a high or low score between categories in order to determine significant edges. Furthermore, since our application is text mining, the score may also depend on the original graph decorations, and therefore the text corpus and the method for identifying occurrences and co-occurrences.

We simulate the random distribution of the scores of each category pair through multiple random permutations of the original graph decorations. We then calculated and recorded co-occurrence scores for all pairs of categories. In addition, for each category D, such that $isA(D,C_1)$, the score difference $score(C_1,C_2) - score(D,C_2)$ was recorded. Further, for each category E with $isA(C_1,E)$, the score difference $score(E,C_2) - score(C_1,C_2)$ was recorded.

Hence, the results of this step are threefold. First, we approximate the random score distribution for each pair of categories. Second, each triple of categories C, D and $E \in child(C)$ gives rise to a random distribution of score differences between (C,D) and (E,D). Third, each triple C, D and $E \in parent(C)$ yields a random distribution of score differences between (E,D) and (C,D).

Based on these distributions, we developed a set of novel statistical tests that test the significance of a score value for a pair of categories [Hoehndorf et al., 2008c]. The tests combine measures of relevance and specificity. In [Hoehndorf et al., 2008c], we applied the method to the extraction of associations between categories, and assumed that significant co-occurrences represent significant associations. However, if more specific relations between categories are used instead of co-occurrences, the tests can be applied to identify significant relations as well.

6.4.3 Verification and Ontological Interpretation

After information is extracted using methods from named entity recognition, relationship extraction and the corresponding significance tests, they can be embedded in an ontology to connect it with further knowledge and verify the ontological adequacy of the extracted information, i.e., the use of compatible conceptualizations in its representation. I describe and extend the work presented in [Hoehndorf et al., 2008b] here.

For embedding the extracted information in an ontology, the basic conceptualization of the text mining domain must first be analyzed. For our purpose, text mining identifies references to four kinds of ontological entities in text: categories C, individuals I, relations R and instances of relations T. Without loss of generality, I restrict my discussion to binary relations and $R \subseteq (C \cup I) \times (C \cup I)$. I call the structure $\mathcal{TM} =< C, I, R, T >$ resulting from a text mining analysis a *text mining structure* (TMS).

The aim is to provide an ontological interpretation of such a TMS. We can then apply this ontological interpretation for the refinement of the TMS using the axioms of an ontology. In order to deal with inconsistent and incomplete knowledge, we use a non-monotonic form of logical deduction as a method to consistently generate explanations for facts resulting from this ontological interpretation [Hobbs et al., 1988].

For the purpose of this analysis, an ontology is a structure $O =< C', R', ::, isa, Ax >$ of categories C' and relations R' together with a set of axioms Ax. Ontologies contain as relations at least the instantiation relation **(::)** and the **is-a** relation.

Definition 3. *An* ontological interpretation I *of a TMS* $\mathcal{TM} =< C, I, R, T >$ *with respect to the ontology* $O =< C', R', ::, isa, Ax >$ *satisfies the conditions:*

- *for each* $c \in C$, $c^I = c'$ *such that* $c' \in C'$ *and either* $c :: c'$ *or* $isa(c, c')$,
- *for each* $i \in I$, $i^I = i'$ *such that there exists a* $c' \in C'$ *and* $i :: c'$,
- *for each* $r \in R$, $r^I = r'$ *such that* $r' \in R'$ *and* $isa(r, r')$,
- *for each* $t \in T$, $t^I = t'$ *such that there exists a* $r' \in R'$ *and* $t' :: r'$.

According to this definition, an ontological interpretation performs the following functions: for each category identified in the text, it identifies at least one category in the ontology O of which the category found in the text is either a sub-category or an instance; for each individual in the text, it identifies at least one category of which this individual is an instance; and similarly for relations and their instances. This definition assumes an ontology which supports higher-order categories. If the ontology O does not support these, the first part of the definition must be restricted to the case where the identified category is a sub-category of one of the ontology's categories.

Two major difficulties can arise when trying to find an ontological interpretation of a TMS. First, it may occur that no ontological interpretation exists due to an inconsistency. In this case, we call the TMS $\mathcal{TM}$ classically inconsistent with the ontology O. Second, there may be many possible ontological interpretations for a TMS, and some measure of preference should be established to select the most appropriate ontological interpretation.

In order to deal with inconsistencies, we can establish classical consistency by extending the ontological interpretation such that identified categories (or instances) are subclasses (or instances) of more general categories. For example, consider a TMS containing the following three relation instances:

$$IsA(Arsenic, Poison) \tag{6.1}$$

$$PlaysRole(Arsenic, Poison) \tag{6.2}$$

$$HasFunction(Arsenic, Poison) \tag{6.3}$$

Here, poison is used in three mutually exclusive meanings: as a substance, a role and a function; any ontological interpretation interpreting *Poison*, *IsA*, *PlaysRole* and *HasFunction* in their usual understanding will be classically inconsistent. The cause of the inconsistency is a too specific interpretation of *Poison*. Interpreting *Poison* as a subclass of *Entity* avoids the inconsistency, but does not permit inferences based on axioms pertaining to more specific categories.

The general problem is finding the most specific consistent ontological interpretation for a TMS. We propose the use of abductive reasoning over ontologies [Elsenbroich et al., 2006] to fill this gap: abduction is a non-classical form of inference that generates an explanation for an observation. The general form of abductive inference is inference is $B, A \rightarrow B \vdash A$.

Several additional conditions can be employed in abductive inference systems. Let Γ be a knowledge base. These conditions are popular restrictions on abductive inferences:

- Consistency: $\Gamma \cup A \nvdash \bot$
- Minimality: A is a *minimal* explanation for B
- Relevance: $A \nvdash B$
- Explanatoriness: $\Gamma \nvdash B$

In order to identify the most specific consistent ontological interpretation, we use minimal consistent abduction. We add the following formula as additional assumption, where C_i ranges over all categories from O:

$$\begin{aligned} isa(Poison, C_1) \vee \dots \\ \vee isa(Poison, C_n) \rightarrow isa(Poison, Entity) \end{aligned} \tag{6.4}$$

Then, minimal consistent abduction can generate the desired explanation for (6.4):

$$\begin{aligned} isa(Poison, Substance) \vee isa(Poison, Role) \vee \\ isa(Poison, Function) \end{aligned} \tag{6.5}$$

We have not yet implemented the presented method here, because tools to support abductive reasoning over ontologies are not easily available. Therefore, a large-scale evaluation of the method's results and performance is still pending. Nevertheless, the method can currently be used manually to provide ontological interpretations of results of text-mining analyses.

6.5 Discussion: need for ontology-*driven* software

A common theme underlies the software tools that I discussed in this section. The BOWiki is an ontology-based wiki which is based on an ontology-based, conceptual model that is integrated with a domain model to facilitate the acquisition of ontology-based information. The tagging system described in section 6.3 uses a tagging core ontology based on the GFO that is extended with GFO-Bio to provide domain-specific information on the types of objects that can be tagged. The text-mining method described in 6.4.3 analyzes the results of text mining analyses using a text mining core ontology (called a text mining structure) and a domain core ontology such as GFO-Bio, and produces an interpretation of the text mining results within these ontologies.

Each of the described methods or software applications utilize the ontologies as a component of their basic operation during run-time. The BOWiki's quality-control features, query capabilities and description of an entities properties and relation depend on the used ontologies, and are generated from these ontologies during run-time. The facets that are applicable to a tagged objects in the tagging software are generated using the provided ontologies. Ontological interpretations situate data that was generated using text mining within the combination of the text mining core ontology and a domain ontology.

The combination of domain ontologies with the conceptual model of the software application has several advantages: domain specific knowledge is separated from the conceptual model; information can flow in both directions between the domain model and the conceptual model of the software; and the data collected or processed by the software application can be verified during the run-time of the software.

The separation of the conceptual model and the domain ontology permits the development of flexible software applications that can be reused within multiple domains. For example, the BOWiki software application uses GFO-Bio as domain ontology within the biological domain, but can be configured to operate with any OWL file. However, the OWL ontology that is used by the BOWiki must be integrated with the top-level ontology used by the BOWiki's conceptual model to permit the successful interpretation of the domain categories, relations and axioms within the BOWiki.

Information flow between the conceptual model and the domain model is important to develop modular and reusable software. For example, an OWL class may represent a kind of relation, and the BOWiki's conceptual model contains a *Relation* category. Without the information that a category and its sub-categories represent relations, the BOWiki would treat these as any other OWL class, i.e., not as relations. This example also illustrates the need of an ontological foundation of both the conceptual model and the domain ontology in a common upper-level ontology, which realizes this information flow.

Finally, employing an automated reasoner to reason over both the conceptual model and the domain model during runtime facilitates the detection of errors and inconcistencies as well as possible explanations for these. This property of ontology-driven information systems contributes to their robustness. Additionally, the knowledge that is acquired using the tools described here is from the beginning consistently integrated within an ontology.

7 Summary and Conclusions

> 無
>
> Chao-chou Ts'ung-shen

Interoperability between ontology-based information systems in biology is a goal which the biological community still attempts to achieve. For this purpose, organizations such as the OBO Foundry were developed which enforce several criteria pertaining to ontology development and maintenance. These criteria are primarily technical and social criteria. They are intended to facilitate interoperability between the information systems based on these ontologies.

In chapter 3, I have analyzed the problem of interoperability between ontology-based information systems. Interoperating systems must allow for a *flow of information* between them. Information flows when the classifications made by one information systems have consequences on the classifications made by another information system such that the relations between both form an *infomorphism*.

I identified several issues pertaining to the interoperability between ontology-based information systems in biology and biomedicine. I have grouped these issues in three classes: logic and knowledge representation, ontology and knowledge acquisition. The first addresses how ontologies are represented in a formal language and which semantics is employed in this representation. Furthermore, the kinds of inferences that are permitted fall into this group. The second refers to the ontological commitment of the biological domain ontologies. I claim that only an explicit statement of this commitment permits establishing information flow between multiple ontologies. Finally, the information flow must be constructed, which necessitates the acquisition of domain knowledge, i.e., the specific relation that exists between two domain categories.

In chapters 4, 5 and 6, I have addressed the issues identified in chapter 3. Chapter 4 introduced an extension of the semantics of the OBO Flatfile Format and discussed the semantics for frame-based ontologies like the Foundational Model of Anatomy. The OBO Flatfile Format's semantics was given by a translation to the OWL-DL language, for which a model-theoretic semantics exists. I extended this translation to permit more flexible translations to OWL. The primary motivation for extending the translation was the need for expressing negation which is needed to adequately define some relations that are used in biomedical ontologies.

In chapter 5, I outlined the categories of the biological core ontology GFO-Bio. GFO-Bio contains categories and meta-categories for the biological domain. I introduced a set of axioms to illustrate the specification of several categories in GFO-Bio, and explained how GFO-Bio can be used for the integration of biological domain ontologies. Two ontological issues within the biological domain were discussed in greater detail. First, I described an ontology of functions and its application within biological ontologies. Second, I analyzed the integration of ontologies that serve as reference models and phenotype ontologies. The integration of these kinds of ontologies required an extended logical framework that allows for non-monotonic inferences. I used default logic in the form of answer set programming to formulate axioms for relationships between categories in in reference models.

In chapter 6, I discussed a number of approaches for acquiring biological knowledge. These were divided in three groups. In the first, knowledge is directly acquired, i.e., the relation and its arguments are known and directly asserted. For this purpose, I introduce the BOWiki in section 6.2, an ontology-based semantic wiki for the acquisition of biological knowledge.

In the second group, discussed in section 6.4, knowledge is extracted from natural language texts. When used for the purpose of knowledge acquisition, an additional step must be performed: the identification of the entity that is mentioned in text. When a relationship is extracted from text, the kind of relationship as well as the arguments must be identified.

A hybrid approach was proposed in section 6.3, where I discussed a form of collaborative tagging. In the software developed for this purpose, both the tagged object and the relationship to the tag can be specified explicitly. One argument in the relation (the tag) remained specified in natural language, and must be analyzed using techniques from natural language processes.

Every application and software developed in chapter 6 utilized ontologies for their operation. The ontologies are used to enforce the use of a common conceptualization. For this purpose, the acquired data is classified with respect to an ontology and its consistency verified. GFO-Bio is currently used in all these applications. However, the applications are general enough to allow the use of other ontologies as well.

The biological core ontology GFO-Bio is central to the suggestions I make here. It ties together the software applications that I described, utilizes a nonmonotonic logic for integrating domain ontologies and provides ontological analyses for multiple biological domain categories, therefore providing the foundation for a flow of information between domain ontologies.

GFO-Bio is represented in the Web Ontology Language (OWL) [Mcguinness and van Harmelen, 2004]. This makes GFO-Bio interoperable with other ontologies developed in the Semantic Web [Berners-Lee et al., 2001]. But the application for which biological domain ontologies were developed transcends traditional forms of knowledge representation used on the Semantic Web. Forms of non-classical, common-sense forms of reasoning must complement an ontological analysis of the biological domain ontologies. The domain ontologies were developed pragmatically for use in specific applications, and not all these applications satisfy the constraints encountered in a classical logic framework. In particular the *monotonicity* of classical logics hinders the development of knowledge based applications that utilize multiple ontologies. GFO-Bio is accompanied with axioms that permit the use of *non-monotonic* logics. This leads to the development of versatile and flexible ontologies that are applicable in multiple application scenarios.

Within the Semantic Web, software applications that utilize ontologies are being developed. The combination of ontologies with automated reasoners permits the development of information systems that cannot only perform queries on stored *data*, but that have access to the knowledge and constraints that underlie the *schema* or *conceptual model* of the information system itself. I have described several novel applications for the use in biomedical knowledge acquisition. Each of these applications utilizes ontologies to classify data, explain the data's *meaning* to users, verify the ontological adequacy of statements and enforce a common basic conceptualization of a domain when multiple users are involved. With increasing development and formalization of biological domain ontologies, these applications become more powerful. Furthermore, the integration of Semantic Web technology like automated reasoners [Sirin and

Parsia, 2004] and OWL libraries [Carroll et al., 2003] into these applications permits the flexible modification and customization of the software. Novel knowledge can be integrated by providing updated ontologies that serve as the schemata for these applications.

Interoperability between ontology-based information systems is not a state, but a continuing process that never ends. Novel knowledge will continuously lead to changes and modifications in the ontologies. Scientific breakthroughs will shake the foundations of a domain, and ontologies in these domains will have to be developed anew to keep the pace and continue to play the role they have been assigned. Sound and flexible principles for developing domain ontologies and for establishing and maintaining interoperability between information systems based on them will continue to remain an important area of research. The division into the three categories logic, ontology and knowledge acquisition that underlies my investigations may provide a continuing insight into the facets of interoperability, and serve as a foundation for improving both the representation of knowledge in biology and the development of ontology-based applications in this domain.

A Theory of Sequences: Implementation

What follows is the implementation of the ontology of sequences described in section 5.1.3. The implementation is based on the SPASS theorem prover [Weidenbach et al., 2002]. The input file of the SPASS theorem prover can be download from the project webpage [Hoehndorf, 2009].

```
begin_problem(Sequences).
list_of_descriptions.
name({* SequenceAxioms *}).
author({* Robert Hoehndorf *}).
status(unsatisfiable).
description({* s *}).
end_of_list.

list_of_symbols.
predicates[(Seq,1),(Mol,1),(sPO,2),(PO,2),(binds,2),(inst,2),
(sPPO,2),(PBS,1),(soverlap,2),(sdisjoint,2),
(between,4),(end,3),(in,2),(Jun,1),(conn,2),(conn2,2),
(PPO,2),(At,1),(overlap,2),(disjoint,2),
(CSeq,1),(LSeq,1)].
end_of_list.
```

```
list_of_formulae(axioms).

%instead of basic GFO import, disjointness axioms
formula(forall([X],implies(Seq(X),not(Mol(X))))).
formula(forall([X],implies(Seq(X),not(Jun(X))))).
formula(forall([X],implies(Mol(X),not(Jun(X))))).
formula(forall([X],implies(Mol(X),not(Seq(X))))).
formula(forall([X],implies(Jun(X),not(Mol(X))))).
formula(forall([X],implies(Jun(X),not(Seq(X))))).

%existence axioms, to exclude trivial models
formula(exists([X],Seq(X))).
formula(exists([X],Mol(X))).
formula(exists([X],Jun(X))).

%argument restrictions
formula(forall([X,Y],implies(sPO(X,Y),and(Seq(X),Seq(Y))))).
formula(forall([X,Y],implies(PO(X,Y),and(Mol(X),Mol(Y))))).
formula(forall([X,Y],implies(binds(X,Y),and(Mol(X),Mol(Y))))).
formula(forall([X],implies(Seq(X),forall([Y],implies(inst(Y,X),
                               Mol(Y)))))).

%ground mereology for sequences
formula(forall([X],implies(Seq(X),sPO(X,X)))).
formula(forall([X,Y],implies(and(sPO(X,Y),sPO(Y,X)),
      equal(X,Y)))).
formula(forall([X,Y,Z],implies(and(sPO(X,Y),sPO(Y,Z)),
      sPO(X,Z)))).
```

```
%definitions of sequence-atoms (PBS),
%proper sequence part, overlap, disjoint
formula(forall([X,Y],equiv(sPPO(X,Y),and(sPO(X,Y),
      not(sPO(Y,X)))))).
formula(forall([X],equiv(PBS(X),and(Seq(X),not(exists([Y],
      sPPO(X,Y))))))).
formula(forall([X,Y],equiv(soverlap(X,Y),exists([Z],
      and(sPO(Z,X),sPO(Z,Y)))))).
formula(forall([X,Y],equiv(sdisjoint(X,Y),not(soverlap(X,Y))))).

%atomar mereology for sequences
formula(forall([X],implies(Seq(X),exists([Y],and(PBS(Y),
      sPO(Y,X)))))).
formula(forall([X],implies(Seq(X),not(exists([Y],and(sPPO(Y,X),
      forall([U],implies(and(sPPO(U,X),PBS(U)),sPO(U,Y))))))))).

%weak supplementation principle
formula(forall([X,Y],implies(sPPO(X,Y),exists([Z],
      and(sPO(Z,Y),sdisjoint(Z,X)))))).

%strong supplementation principle
formula(forall([X,Y],implies(sPPO(X,Y),exists([Z],
      and(sPO(Z,Y),sdisjoint(Z,X)))))).

%argument restrictions for between, end; definition of in
formula(forall([J,P1,P2,S],implies(between(J,P1,P2,S),
      and(Jun(J),PBS(P1),PBS(P2),Seq(S))))).
formula(forall([J,P,S],implies(end(J,P,S),and(Jun(J),
      PBS(P),Seq(S))))).
```

```
formula(forall([J,S],equiv(in(J,S),or(exists([P1,P2],
      between(J,P1,P2,S)),exists([P],end(J,P,S)))))).

%axioms for conn
formula(forall([J1,J2],implies(conn(J1,J2),conn(J2,J1)))).
formula(forall([J1,J2],implies(conn(J1,J2),not(equal(J1,J2))))).
formula(forall([J1,J2],implies(conn(J1,J2),conn2(J1,J2)))).
formula(forall([J1,J2,J3],implies(and(conn2(J1,J2),
      conn2(J2,J3)),conn2(J1,J3)))).
formula(forall([J1,J2,S],implies(and(in(J1,S),in(J2,S)),
      conn2(J1,J2)))).
formula(forall([J1,J2,S1,S2],implies(and(in(J1,S1),in(J2,S2),
      not(soverlap(S1,S2))),not(conn2(J1,J2))))).
formula(forall([J1,J2,S],implies(and(conn(J1,J2),in(J1,S)),
      in(J2,S)))).

%junctions belong to exactly one sequence
formula(forall([J,P1,P2,S],implies(between(J,P1,P2,S),
      between(J,P2,P1,S)))).
formula(forall([J,P1,P12,P2,P22,S1,S2],implies(and(
      between(J,P1,P2,S1),between(J,P12,P22,S2)),
and(or(and(equal(P1,P12),equal(P2,P22)),
      and(equal(P1,P22),equal(P2,P12))),soverlap(S1,S2))))).
formula(forall([J,P1,P2,S1,S2],implies(and(end(J,P1,S1),
      end(J,P2,S2)),and(equal(P1,P2),soverlap(S1,S2))))).
formula(forall([J1,J2,J3,P],implies(and(end(J1,P,P),
      end(J2,P,P),end(J3,P,P),not(equal(J1,J2))),
or(equal(J3,J1),equal(J3,J2))))).
```

```
%=============================================================
%axioms for tokens
%=============================================================

%ground mereology for token
formula(forall([X],implies(Mol(X),PO(X,X)))).
formula(forall([X,Y],implies(and(PO(X,Y),PO(Y,X)),equal(X,Y)))).
formula(forall([X,Y,Z],implies(and(PO(X,Y),PO(Y,Z)),PO(X,Z)))).

%definitions of token-atoms (At), proper part, overlap, disjoint
formula(forall([X,Y],equiv(PPO(X,Y),and(PO(X,Y),not(PO(Y,X)))))).
formula(forall([X],equiv(At(X),and(Mol(X),
        not(exists([Y],PPO(X,Y))))))).
formula(forall([X,Y],equiv(overlap(X,Y),exists([Z],
        and(PO(Z,X),PO(Z,Y)))))).
formula(forall([X,Y],equiv(disjoint(X,Y),not(overlap(X,Y))))).

%atomar mereology on token level
formula(forall([X],implies(Mol(X),exists([Y],
        and(At(Y),PO(Y,X)))))).
formula(forall([X],implies(Mol(X),not(exists([Y],
        and(PPO(Y,X),forall([U],
implies(and(PO(U,X),At(U)),PO(U,Y))))))))).

%weak supplementation principle
formula(forall([X,Y],implies(PPO(X,Y),exists([Z],
        and(PO(Z,Y),disjoint(Z,X)))))).

%strong supplementation principle
```

```
formula(forall([X,Y],implies(PPO(X,Y),exists([Z],
        and(PO(Z,Y),disjoint(Z,X)))))).

formula(forall([A,X,Y],implies(and(Seq(A),inst(X,A),
        At(Y),PO(Y,X)),
exists([B],and(sPO(B,A),PBS(B),inst(Y,B),
        forall([C],implies(inst(Y,C),equal(B,C))))))))).
formula(forall([A,X],implies(and(PBS(X),inst(A,X)),At(A)))).

%axioms for binds
formula(forall([X,Y],implies(binds(X,Y),and(At(X),At(Y))))).
formula(forall([X,Y],implies(binds(X,Y),exists([U,V],and(PBS(U),
        PBS(V),inst(X,U),inst(Y,V)))))).
formula(forall([X],not(binds(X,X)))).
formula(forall([X,Y],implies(binds(X,Y),binds(Y,X)))).

%axioms for linearity of sequences
formula(forall([X,Y],implies(and(Seq(X),PBS(Y),sPO(Y,X)),
forall([A,B],implies(and(inst(A,X),inst(B,Y)),
        not(exists([U,V,W],
and(binds(U,B),binds(V,B),binds(W,B),not(equal(U,V)),
        not(equal(V,W)),not(equal(U,W)))))))))).

formula(forall([A,X],implies(and(Seq(X),not(PBS(X)),inst(A,X)),
forall([B],implies(and(PO(B,A),At(B)),
        exists([C],binds(B,C))))))).

formula(forall([A,X],implies(and(Seq(X),inst(A,X)),
        not(exists([U,V,W],
and(not(equal(U,V)),not(equal(V,W)),not(equal(U,W)),
exists([P],and(binds(U,P),forall([Q],
            implies(binds(U,Q),equal(P,Q))))),
exists([P],and(binds(V,P),forall([Q],
            implies(binds(V,Q),equal(P,Q))))),
exists([P],and(binds(W,P),forall([Q],
            implies(binds(W,Q),equal(P,Q)))))))))).

formula(forall([X],equiv(CSeq(X),and(Seq(X),not(PBS(X)),
        forall([A,B],
implies(and(inst(A,X),PO(B,A),At(B)),
exists([C,D],and(binds(B,C),binds(B,D),
forall([E],implies(binds(B,E),or(equal(C,E),
                    equal(D,E)))))))))))).

formula(forall([X],implies(CSeq(X),not(exists([J,P],
        and(in(J,X),end(J,P,X))))))).

formula(forall([X,J],implies(and(CSeq(X),in(J,X)),
               exists([P1,P2],between(J,P1,P2,X))))).


formula(forall([X],implies(LSeq(X),
         exists([J1,J2,P1,P2],
         and(not(equal(J1,J2)),
 end(J1,P1,X),
 end(J2,P2,X),
 forall([J3],implies(exists([P],end(J3,P,X)),
 or(equal(J3,J1),
 equal(J3,J2))))))))).

formula(forall([X,J],implies(and(LSeq(X),in(J,X),
            not(exists([P],end(J,P,X)))),
    exists([P1,P2],between(J,P1,P2,X))))).

formula(forall([X],implies(Seq(X),or(LSeq(X),CSeq(X))))).
formula(exists([X],and(Mol(X),not(exists([Y],binds(X,Y)))))).

formula(forall([J,P,S],
 implies(end(J,P,S),
   exists([J1],
and(conn(J,J1),
 forall([J2],
  implies(conn(J,J2),
   equal(J2,J1)))))))).

formula(forall([J,P1,P2,S],
 implies(between(J,P1,P2,S),
  exists([J1,J2],
   and(not(equal(J1,J2)),
conn(J,J1),
conn(J,J2),
forall([J3],
 implies(conn(J,J3),
or(equal(J3,J1),
  equal(J3,J2))))))))).

formula(forall([J,P1,P2,S,M],
       implies(and(between(J,P1,P2,S),
  inst(M,S)),
  exists([A1,A2],
 and(At(A1),
       At(A2),
       PPO(A1,M),
       PPO(A2,M),
       binds(A1,A2)))))).



end_of_list.

list_of_formulae(conjectures).
formula(and(forall([X],sPO(X,X)),not(forall([X],
        sPO(X,X))))).


end_of_list.
end_problem.
```

Bibliography

RL Albin, AB Young, and JB Penney. The functional anatomy of basal ganglia disorders. *Trends Neurosci*, 1989:366–75, 1910.

Eric E. Allen and Jillian F. Banfield. Community genomics in microbial ecology and evolution. *Nature Reviews Microbiology*, 3(6):489–498, June 2005. ISSN 1740-1526.

J.F. Allen and P.J. Hayes. Moments and points in an interval-based temporal logic. *Computational Intelligence*, 5(3):225–238, 1989.

D. M. Armstrong. *A World of States of Affairs (Cambridge Studies in Philosophy)*. Cambridge University Press, March 1997. ISBN 0521589487.

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, Issel L. Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1): 25–29, May 2000.

Sören Auer. Powl – a web based platform for collaborative semantic web development. In *Proceedings of the 1st Workshop on Scripting for the Semantic Web, SFSW'05, Heraklion, Greece, May 30*, 2005.

Sören Auer, Sebastian Dietzold, and Thomas Riechert. Ontowiki – a tool for social, semantic collaboration. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora Aroyo, editors, *Proceedings of the 5th International Semantic Web Conference, ISWC 2006, Athens, Georgia, USA, Nov 5-9*, volume 4273 of *Lecture Notes in Computer Science*, Berlin, 2006. Springer.

Franz Baader. *The Description Logic Handbook : Theory, Implementation and Applications*. Cambridge University Press, January 2003. ISBN 0521781760.

Michael Bada and Lawrence Hunter. Enrichment of OBO ontologies. *J Biomed Inform*, 40(3):300–315, 2007.

Michael Bada, Robert Stevens, Carole Goble, Yolanda Gil, Michael Ashburner, Judith A. Blake, Michael J. Cherry, Midori Harris, and Suzanna Lewis. A short study on the success of the gene ontology. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(2):235–240, February 2004.

Jonathan Bard, Seung Y. Rhee, and Michael Ashburner. An ontology for cell types. *Genome Biol*, 6(2), 2005.

J. Barwise. *Model-Theoretic Logics (Perspectives in Mathematical Logic)*. Springer, 1985. ISBN 0387909362.

J. Barwise and J. Perry. *Situations and Attitudes*. MIT Press, Cambridge, MA, 1983.

Jon Barwise. *The situation in logic*. CSLI Publications, 1988.

Jon Barwise and Jerry Seligman. *Information Flow : The Logic of Distributed Systems*. Cambridge University Press, 1997.

Sean Bechhofer. The DIG description logic interface: DIG/1.1. Technical report, University of Manchester, 2003.

Sean Bechhofer, Ian Horrocks, and Daniele Turi. The OWL Instance Store: System description. In Robert Nieuwenhuis, editor, *Proceedings of the 20th*

International Conference on Automated Deduction, CADE-20, Tallinn, Estonia, Jul 22-27, volume 3632 of *Lecture Notes in Computer Science*, Berlin, 2005. Springer.

Dave Beckett. RDF/XML syntax specification (revised). W3C recommendation, World Wide Web Consortium (W3C), February, October 2004.

D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank. *Nucleic Acids Res*, 33 Database Issue, January 2005. ISSN 1362-4962.

T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):28–37, 2001.

C. Blaschke, E. A. Leon, M. Krallinger, and A. Valencia. Evaluation of biocreative assessment of task 2. *BMC Bioinformatics*, 6 Suppl 1, 2005. ISSN 1471-2105.

Olivier Bodenreider. Ontologies and data integration in biomedicine: Success stories and challenging issues. *Data Integration in the Life Sciences*, pages 1–4, 2008.

B. Boeckmann, A. Bairoch, R. Apweiler, M. C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic Acids Res*, 31(1):365–370, January 2003. ISSN 1362-4962.

Christopher Boorse. Wright on functions. *The Philosophical Review*, 85(1): 70–86, 1976.

A. Borgida, R.J. Brachman, D.L. McGuinness, and L.A. Resnick. CLASSIC: a structural data model for objects. *Proceedings of the 1989 ACM SIGMOD international conference on Management of data*, pages 58–67, 1989.

R.J. Brachman and J.G. Schmolze. An overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, 9(2):171–216, 1985.

Franz Brentano. *Philosophische Untersuchungen zu Raum, Zeit und Kontinuum*. Meiner, Hamburg, 1976.

Christopher Brewster, Simon Jupp, Joanne Luciano, David Shotton, Robert Stevens, and Ziqi Zhang. Issues in learning an ontology from text. In Phillip Lord, Nigam Shah, Susanna-Assunta Sansone, and Matthew Cockerill, editors, *Proceedings of The 11th Annual Bio-Ontologies Meeting*, 2008.

Dan Brickley and Rahmanathan V. Guha. RDF vocabulary description language 1.0: RDF Schema. W3C recommendation, World Wide Web Consortium (W3C), 2004.

C. J. Bult, J. T. Eppig, J. A. Kadin, J. E. Richardson, and J. A. and Blake. The Mouse Genome Database (MGD): mouse biology and model systems. *Nucleic acids research*, 36(Database issue), January 2008. ISSN 1362-4962.

P. Burek, R. Hoehndorf, F. Loebe, J. Visagie, H. Herre, and J. Kelso. A top-level ontology of functions and its application in the open biomedical ontologies. *Bioinformatics*, 22(14):e66–e73, July 2006.

Patryk Burek. *Ontology of Functions*. PhD thesis, University of Leipzig, Institute of Informatics (IfI), 2006.

Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: Implementing the Semantic Web recommendations. Technical Report HPL-2003-146, Hewlett Packard, Bristol, UK, 2003.

Werner Ceusters, Peter Elkin, and Barry Smith. Referent tracking: The problem of negative findings. *Stud Health Technol Inform*, 124, 2006.

Cuiming Chen, Volker Haarslev, and Jiaoyue Wang. LAS: Extending Racer by a large Abox store. In Ian Horrocks, Ulrike Sattler, and Frank Wolter, editors, *Proceedings of the 2005 International Workshop on Description Logics, DL2005, Edinburgh, Scotland, UK, Jul 26-28*, volume 147 of *CEUR Workshop Proceedings*, Aachen, Germany, 2005. CEUR-WS.org.

J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. Sgd: Saccharomyces genome database. *Nucleic Acids Res*, 26(1):73–79, January 1998. ISSN 0305-1048.

M. Chicurel. Bioinformatics: bringing it all together. *Nature*, 419(6908), October 2002. ISSN 0028-0836.

Alonzo Church. A note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1:40–41, 1936.

Uniprot Consortium. The universal protein resource (uniprot). *Nucleic Acids Res*, 35(Database issue), January 2007. ISSN 1362-4962.

Fabrice Correia. *Existential Dependence and Cognate Notions*. Analytica. Philosophia Verlag, Muenchen, 2005.

F. M. Couto, M. J. Silva, and P. M. Coutinho. Finding genomic ontology terms in text using evidence content. *BMC Bioinformatics*, 6 Suppl 1, 2005.

O. Dameron, D.L. Rubin, and M.A. Musen. Challenges in Converting Frame-Based Ontology into OWL: the Foundational Model of Anatomy Case-Study. *AMIA Annual Symposium Proceedings*, 2005:181, 2005.

K. Degtyarenko, P. Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcantara, M. Darsow, M. Guedj, and M. Ashburner. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, 2007.

Keith Devlin. *Logic and Information*. Cambridge University Press, 1991.

Theodosis Dimitrakos and Tom Maibaum. On a generalized modularization theorem. *Inf. Process. Lett.*, 74(1-2):65–71, 2000. ISSN 0020-0190.

Pavlin Dobrev, Ognian Kalaydjiev, and Galia Angelova. From conceptual structures to semantic interoperability of content. In Uta Priss, Simon Polovina, and Richard Hill, editors, *Proceedings of the 15th International Conference on Conceptual Structures (ICCS 2007)*, volume 4604 of *Lecture Notes in Artificial Intelligence*, pages 192–205, Berlin, Heidelberg, July 2007. Springer-Verlag.

A. Doms and M. Schroeder. GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Res*, 33(Web Server issue):783–786, Jul 2005.

Francesco M. Donini, Daniele Nardi, and Riccardo Rosati. Autoepistemic description logics. In Martha E. Pollack, editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 1997, Nagoya, Japan, Aug 23-29*, volume 1, San Francisco, 1997. Morgan Kaufmann.

K. Eilbeck, S. E. Lewis, C. J. Mungall, M. Yandell, L. Stein, R. Durbin, and M. Ashburner. The sequence ontology: a tool for the unification of genome annotations. *Genome Biol*, 6(5), 2005a. ISSN 1465-6914.

Karen Eilbeck, Suzanna E. Lewis, Christopher J. Mungall, Mark Yandell, Lincoln Stein, Richard Durbin, and Michael Ashburner. The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biology*, 6 (5), 2005b.

Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. A uniform integration of higher-order reasoning and external evaluations in answer set programming. In Leslie P. Kaelbling and Alessandro Saffiotti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh, Scotland, UK, Jul 30 - Aug 5*, Denver, Colorado, 2005. Professional Book Center.

Corinna Elsenbroich, Oliver Kutz, and Ulrike Sattler. A case for abductive reasoning over ontologies. In *Proc. OWL: Experiences and Directions*, pages 10–11, 2006.

Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, 1 edition, 2007. ISBN 3540496114.

R. Fikes and T. Kehler. The role of frame-based representation in reasoning. *Communications of the ACM*, 28(9):904–920, 1985.

Wolfgang Fleischmann, Steffen Möller, Alain Gateau, and Rolf Apweiler. A novel method for automatic functional annotation of proteins. *Bioinformatics*, 15(3), 1999.

Consortium Flybase. The flybase database of the drosophila genome projects and community literature. the flybase consortium. *Nucleic Acids Res*, 27(1): 85–88, January 1999. ISSN 0305-1048.

D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors. *Handbook of Logic in Artificial Intelligence and Logic Programming (Vol 3): Nonmonotonic Reasoning and Uncertain Reasoning*, volume 3. Oxford University Press, Oxford, UK, 1994.

S. Gaudan, Jimeno A. Yepes, V. Lee, and Rebholz D. Schuhmann. Combining evidence, specificity, and proximity towards the normalization of gene ontology terms in text. *EURASIP journal on bioinformatics & systems biology*, 2008.

A. Geraci. *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*. Institute of Electrical and Electronics Engineers Inc., The, 1991.

Jim Giles. Key biology databases go wiki. *Nature*, 445(7129):691, 2007.

Claudio Gnoli and Roberto Poli. Levels of reality and levels of representation. *Knowledge organization*, 31(3), 2004.

Christine Golbreich and Ian Horrocks. The OBO to OWL mapping, GO to OWL 1.1! In Christine Golbreich, Aditya Kalyanpur, and Bijan Parsia, editors, *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Innsbruck, Austria, Jun 6-7*, volume 258 of *CEUR Workshop Proceedings*, Aachen, Germany, 2007. CEUR-WS.org.

Jorge J. E. Gracia. *Metaphysics & Its Task: The Search for the Categorial Foundation of Knowledge*. State University of New York, 1999.

Richard E. Green, Johannes Krause, Susan E. Ptak, Adrian W. Briggs, Michael T. Ronan, Jan F. Simons, Lei Du, Michael Egholm, Jonathan M. Rothberg, Maja Paunovic, and Svante Pääbo. Analysis of one million base pairs of neanderthal dna. *Nature*, 444(7117):330–336, 2006. ISSN 0028-0836.

Pierre Grenon. Bfo in a nutshell: A bi-categorial axiomatization of bfo and comparison with dolce. Technical report, IFOMIS, Faculty of Medicine, University of Leipzig, June 2003a.

Pierre Grenon. Bfo in a nutshell: A bi-categorial axiomatization of BFO and comparison with DOLCE. Technical report, University of Leipzig, Leipzig, 2003b.

Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43 (5-6), 1995.

N. Guarino and C. A. Welty. An overview of ontoclean. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 151–172. Springer, 2004.

Nicola Guarino. Formal ontology and information systems. In Nicola Guarino, editor, *Formal Ontology in Information Systems: Proceedings of the First International Conference (FOIS'98), Trento, Italy, 6-8 June 1998*, volume 46 of *Frontiers in Artificial Intelligence and Applications*, Amsterdam, June 1998. IOS Press. URL `http://www.loa-cnr.it/Papers/FOIS98.pdf`.

G. Guizzardi. *Ontological foundations for structural conceptual models*. PhD thesis, University of Twente, Enschede, The Netherlands, Enschede, October 2005.

Volker Haarslev and Ralf Möller. Racer: A core inference engine for the Semantic Web. In York Sure and Oscar Corcho, editors, *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), Sanibel Island, Florida, USA, Oct 20*, volume 87 of *CEUR Workshop Proceedings*, Aachen, Germany, 2003. CEUR-WS.org.

M.A. Haendel, F. Neuhaus, DS Osumi-Sutherland, P.M. Mabee, JLV Mejino, C.J. Mungall, and B. Smith. CARO–the common anatomy reference ontology. *Anatomy Ontologies for Bioinformatics*, 2007.

Nicolai Hartmann. *Teleologisches Denken*. Walter de Gruyter, 1966.

Nicolai Hartmann. *Neue Wege der Ontologie*. Kohlhammer, Stuttgart-Berlin, 1942.

Terry F. Hayamizu, Mary Mangan, John P. Corradi, James A. Kadin, and Martin Ringwald. The Adult Mouse Anatomical Dictionary: a tool for annotating and integrating data. *Genome Biology*, 6(3), 2005.

Barbara Heller and Heinrich Herre. Ontological categories in GOL. *Axiomathes*, 14(1):57–76, 2004.

Thorsten Henrich, Mirana Ramialison, Beate Wittbrodt, Beatrice Assouline, Franck Bourrat, Anja Berger, Heinz Himmelbauer, Takashi Sasaki, Nobuyoshi Shimizu, Monte Westerfield, Hisato Kondoh, and Joachim Wittbrodt. MEPD: a resource for medaka gene expression patterns. *Bioinformatics*, 21(14):3195–3197, 2005.

H. Herre and B. Heller. Ontology of time and situoids in medical conceptual modeling. In S. Miksch, J. Hunter, and E. T. Keravnou, editors, *Proceedings of the 10th Conference on Artificial Intelligence in Medicine (AIME 05)*, volume 3581, pages 266–275, Berlin, 2005. Springer.

Heinrich Herre and Barbara Heller. Semantic foundations of medical information systems based on top-level ontologies. *Journal of Knowledge-Based Systems*, 2006.

Heinrich Herre, Barbara Heller, Patryk Burek, Robert Hoehndorf, Frank Loebe, and Hannes Michalek. General Formal Ontology (GFO) – A foundational ontology integrating objects and processes [Version 1.0]. Onto-med report, Research Group Ontologies in Medicine, Institute of Medical Informatics, Statistics and Epidemiology, University of Leipzig, Leipzig, 2006.

Philip Hieter and Mark Boguski. Functional genomics: It's all how you read it. *Science*, 278(5338):601–602, October 1997.

David Hilbert, W. Ackermann, and Robert E. Luce. *Principles of Mathematical Logic*. American Mathematical Society, 1999.

L. Hirschman, M. Colosimo, A. Morgan, and A. Yeh. Overview of biocreative task 1b: normalized gene lists. *BMC Bioinformatics*, 6 Suppl 1, 2005a. ISSN 1471-2105.

L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia. Overview of biocreative: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6 Suppl 1, 2005b. ISSN 1471-2105.

Jerry R. Hobbs, Mark Stickel, Paul Martin, and Douglas D. Edwards. Interpretation as abduction. In *26th Annual Meeting of the Association for Compu-*

tational Linguistics: Proceedings of the Conference, pages 95–103, Buffalo, New York, 1988.

Robert Hoehndorf. BOWikiServerProtocol, 2007. URL `\url{http://onto.eva.mpg.de/trac/BoWiki/wiki/BoWikiServerProtocol}`.

Robert Hoehndorf. Ontology of situations in the framework of the gol ontology. Master's thesis, Department of Computer Science, University of Leipzig, 2005.

Robert Hoehndorf. Formal ontology of sequences. `http://bioonto.de/pmwiki.php/Main/FormalOntologyOfSequences`, 2009.

Robert Hoehndorf, Kay Prüfer, Michael Backhaus, Heinrich Herre, Janet Kelso, Frank Loebe, and Johann Visagie. A proposal for a gene functions wiki. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *Proceedings of OTM 2006 Workshops, Montpellier, France, Oct 29 - Nov 3, Part I, Workshop Knowledge Systems in Bioinformatics, KSinBIT 2006*, volume 4277 of *Lecture Notes in Computer Science*, Berlin, 2006. Springer.

Robert Hoehndorf, Frank Loebe, Janet Kelso, and Heinrich Herre. Representing default knowledge in biomedical ontologies: Application to the integration of anatomy and phenotype ontologies. *BMC Bioinformatics*, 8(1), 2007.

Robert Hoehndorf, Frank Loebe, Roberto Poli, Janet Kelso, and Heinrich Herre. Gfo-bio: A biological core ontology. *Applied Ontology*, 3(4):219–227, 2008a.

Robert Hoehndorf, Axel-Cyrille Ngonga Ngomo, and Michael Dannemann. Towards ontological interpretations for improved text mining. In Tapio Salakoski, Dietrich Rebholz-Schuhmann, and Sampo Pyysalo, editors, *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland*, pages 165–166. Turku Centre for Computer Science (TUCS), 2008b.

Robert Hoehndorf, Axel-Cyrille Ngonga Ngomo, Michael Dannemann, and Janet Kelso. From terms to categories: Testing the significance of co-occurrences between ontological categories. In Tapio Salakoski, Dietrich Rebholz-Schuhmann, and Sampo Pyysalo, editors, *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland*, pages 53–60. Turku Centre for Computer Science (TUCS), 2008c.

M. Horridge, N. Drummond, J. Goodwin, A. Rector, R. Stevens, and H.H. Wang. The Manchester OWL Syntax. *Proc. of the 2006 OWL Experiences and Directions Workshop (OWL-ED2006)*, 2006.

Illumina. Dna sequencing with solexa technology. Technical report, Illumina Systems & Software, 2007.

I. Johansson and KD Althoff. Qualities, Quantities, and the Endurant-Perdurant Distinction in Top-Level Ontologies. *Büchel G, Klein B, Roth-Berghofer Th (eds) WSPI 05. Proceedings of the Second International Workshop on Philosophy and Informatics, CEUR-WS*, 130, 2005.

Jin D. Kim, Tomoko Ohta, Yuka Tateisi, and Jun I. Tsujii. GENIA corpus - a semantically annotated corpus for bio-textmining. In *ISMB (Supplement of Bioinformatics)*, pages 180–182, 2003.

Dan Klein and Christopher D. Manning. Fast exact inference with a factored model for natural language parsing. In *NIPS*, pages 3–10, 2002.

V. Kolovski, B. Parsia, and Y. Katz. Implementing owl defaults. In *Proceedings of OWL-ED*, 2006.

Kurt Konolige. On the relation between default and autoepistemic logic. *Artif. Intell.*, 35(3):343–382, 1988.

O. Lassila and D. McGuinness. The Role of Frame-Based Representation on the Semantic Web. *Linköping Electronic Articles in Computer and Information Science*, 6(5):2001, 2001.

R. Leaman and G. Gonzalez. Banner: an executable survey of advances in biomedical named entity recognition. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 652–663, 2008. ISSN 1793-5091.

Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, 7(3), 2006.

Bo Leuf and Ward Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley, Reading, Massachusetts, 2001.

David Lewis. *On the Plurality of Worlds*. Blackwell Publishing Limited, February 2001. ISBN 0631224262.

Vladimir Lifschitz. Answer set programming and plan generation. *Artificial Intelligence*, 138(1-2), 2002.

Frank Loebe. Abstract vs.
social roles – Towards a general theoretical account of roles. *Applied Ontology*, 2(2), 2007.

Frank Loebe and Heinrich Herre. Formal semantics and ontologies: Towards an ontological account of formal semantics. In *Proceedings of Formal Ontologies in Information Systems (FOIS)*, 2008.

Marcel Margulies, Michael Egholm, William E. Altman, Said Attiya, Joel S. Bader, Lisa A. Bemben, Jan Berka, Michael S. Braverman, Yi-Ju Chen, Zhoutao Chen, Scott B. Dewell, Lei Du, Joseph M. Fierro, Xavier V. Gomes, Brian C. Godwin, Wen He, Scott Helgesen, Chun H. Ho, Gerard P. Irzyk, Szilveszter C. Jando, Maria L. I. Alenquer, Thomas P. Jarvie, Kshama B.

Jirage, Jong-Bum Kim, James R. Knight, Janna R. Lanza, John H. Leamon, Steven M. Lefkowitz, Ming Lei, Jing Li, Kenton L. Lohman, Hong Lu, Vinod B. Makhijani, Keith E. Mcdade, Michael P. Mckenna, Eugene W. Myers, Elizabeth Nickerson, John R. Nobile, Ramona Plant, Bernard P. Puc, Michael T. Ronan, George T. Roth, Gary J. Sarkis, Jan F. Simons, John W. Simpson, Maithreyan Srinivasan, Karrie R. Tartaro, Alexander Tomasz, Kari A. Vogt, Greg A. Volkmer, Shally H. Wang, Yong Wang, Michael P. Weiner, Pengguang Yu, Richard F. Begley, and Jonathan M. Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, July 2005. ISSN 0028-0836.

Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, May 2006.

M. Marneffe, B. Maccartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*, pages 449–454, 2006.

Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. WonderWeb Deliverable D18: Ontology library (final). Technical report, Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy, 2003.

Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. December 2004. URL `http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html`.

H. Maturana and F. J. Varela. *Autopoiesis and Cognition: The Realization of the Living*. D. Reidel, Boston, 1980.

John Mccarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39, 1980.

John Mccarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 28(1):89–116, 1986.

Deborah L. Mcguinness and Frank van Harmelen. OWL Web Ontology Language overview. W3C recommendation, World Wide Web Consortium (W3C), February 2004.

John Mcneish. Embryonic stem cells in drug discovery. *Nat Rev Drug Discov*, 3(1):70–80, January 2004.

Hannes Michalek. *A Formal Ontological Approach to Causality Embedded in the Top-Level Ontology of GFO*. PhD thesis, University of Leipzig, 2009. forthcoming.

Ruth G. Millikan. *Language, Thought, and Other Biological Categories: New Foundations for Realism*. MIT Press, 1988.

M. Minsky. Frame-system theory. *Thinking: Readings in Cognitive Science*, pages 355–376, 1977.

N. J. Mulder, R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bradley, P. Bork, P. Bucher, L. Cerutti, R. Copley, E. Courcelle, U. Das, R. Durbin, W. Fleischmann, J. Gough, D. Haft, N. Harte, N. Hulo, D. Kahn, A. Kanapin, M. Krestyaninova, D. Lonsdale, R. Lopez, I. Letunic, M. Madera, J. Maslen, J. Mcdowall, A. Mitchell, A. N. Nikolskaya, S. Orchard, M. Pagni, C. P. Ponting, E. Quevillon, J. Selengut, C. J. Sigrist, V. Silventoinen, D. J. Studholme, R. Vaughan, and C. H. Wu. Interpro, progress and status in 2005. *Nucleic Acids Res*, 33(Database issue), January 2005. ISSN 1362-4962.

Chris Mungall. Logical definitions, 2007. URL `\url{http://wiki.`

geneontology.org/index.php?title=Logical_Definitions\&oldid=4157}.

Frank H. Netter. *Atlas of Human Anatomy*. Rittenhouse Book Distributors Inc., 2nd edition, January 1997. ISBN 0914168819.

Richard Newman. Tag ontology design. http://www.holygoat.co.uk/projects/tags/, 2005. Last accessed: Nov 20, 2007.

Ian Niles and Adam Pease. Towards a standard upper ontology. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9, New York, NY, USA, 2001. ACM Press. ISBN 1581133774.

Natalya F. Noy. Semantic integration: a survey of ontology-based approaches. *SIGMOD Rec.*, 33(4):65–70, December 2004. ISSN 0163-5808.

Natasha Noy and Alan Rector. Defining N-ary relations on the Semantic Web. W3C working group note, World Wide Web Consortium (W3C), 2006.

P. V. Ogren, K. B. Cohen, G. K. Acquaah-Mensah, J. Eberlein, and L. Hunter. The compositional structure of gene ontology terms. *Pac Symp Biocomput*, pages 214–225, 2004.

OMG. Unified Modeling Language: Infrastructure. Specification v2.0, Object Management Group (OMG), Needham (Massachusetts), Mar 2006. http://www.omg.org/docs/formal/05-07-05.pdf.

Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax Section 5. RDF-Compatible Model-Theoretic Semantics. Technical report, W3C, December 2004. URL http://www.w3.org/TR/owl-semantics/rdfs.html#built_in_vocabulary.

Helen Pearson. What is a gene? *Nature*, 441(7092):398–401, May 2006. ISSN 0028-0836.

S. Pepper and S. Schwab. Curing the web's identity crisis: Subject indicators for rdf. In *Proceedings of the XML conference 2003*, 2003.

I. Q. H. Phan, Sandrine F. Pilbout, Wolfgang Fleischmann, and Amos Bairoch. NEWT, a new taxonomy portal. *Nucleic Acids Research*, 31(13), 2003.

Roberto Poli. The basic problem of the theory of levels of reality. *Axiomathes*, 12(3-4), 2001.

Karl R. Popper. *Logik der Forschung.* Mohr, January 1994. ISBN 3161462343.

M. F. Porter. An algorithm for suffix stripping. *Readings in information retrieval*, pages 313–316, 1997.

Kay Prufer, Bjoern Muetzel, Hong-Hai Do, Gunter Weiss, Philipp Khaitovich, Erhard Rahm, Svante Paabo, Michael Lachmann, and Wolfgang Enard. Func: a package for detecting significant associations between gene sets and ontological annotations. *BMC Bioinformatics*, 8:41+, February 2007. ISSN 1471-2105.

Eric S. Raymond. The cathedral and the bazaar. *Knowledge, Technology, and Policy*, 12(3), 1999.

A. L. Rector, W. A. Nowlan, and A. Glowinski. Goals for concept representation in the GALEN project. *Proc Annu Symp Comput Appl Med Care*, pages 414–418, 1993.

Alan Rector. Barriers, approaches and research priorities for integrating biomedical ontologies. Technical report, University of Manchester, 2008.

Alan Rector, Jeremy Rogers, and Thomas Bittner. Granularity, scale and collectivity: When size does and does not matter. *Journal of Biomedical Informatics*, 39(3):333–349, June 2006a.

Alan Rector, Robert Stevens, Jeremy Rogers, and the CO-ODE and BioHealth Informatics Teams. Simple bio upper ontology, 2006b. URL `http://www.cs.man.ac.uk/\~rector/ontologies/sample-top-bio/`.

Alan L. Rector. Defaults, context, and knowledge: Alternatives for OWL-indexed knowledge bases. In Russ B. Altman, Keith A. Dunker, Lawrence Hunter, Tiffany A. Jung, and Teri E. Klein, editors, *Proceedings of the 9th Pacific Symposium on Biocomputing (PSB 2004), Hawaii, USA, Jan 6-10*, London, 2004. World Scientific.

R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

Alexandre Riazanov and Andrei Voronkov. Vampire. In *CADE-16: Proceedings of the 16th International Conference on Automated Deduction*, pages 292–296, London, UK, 1999. Springer-Verlag. ISBN 3540662227.

John D. Richter, Midori A. Harris, Melissa Haendel, The Gene Ontology OBO-Edit Working Group, and Suzanna Lewis. OBO-Edit – an ontology editor for biologists. *Bioinformatics*, June 2007.

A. Rogers, I. Antoshechkin, T. Bieri, D. Blasiar, C. Bastiani, P. Canaran, J. Chan, W. J. Chen, P. Davis, J. Fernandes, T. J. Fiedler, M. Han, T. W. Harris, R. Kishore, R. Lee, S. Mckay, H. M. Muller, C. Nakamura, P. Ozersky, A. Petcherski, G. Schindelman, E. M. Schwarz, W. Spooner, M. A. Tuli, K. V. Auken, D. Wang, X. Wang, G. Williams, K. Yook, R. Durbin, L. D. Stein, J. Spieth, and P. W. Sternberg. Wormbase 2007. *Nucleic Acids Res*, 36, 2007.

Cornelius Rosse and José L. V. Mejino. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of Biomedical Informatics*, 36(6), 2003.

Patrick Ruch. Automatic assignment of biomedical categories: toward a generic approach. *Bioinformatics*, November 2005.

Andrea Schaerf. Reasoning with individuals in concept languages. *Data & Knowledge Engineering*, 13(2), 1994.

Sebastian Schaffert, Rupert Westenthaler, and Andreas Gruber. IkeWiki: A user-friendly semantic wiki. In Holger Wache, editor, *Demos and Posters of the 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, Jun 11-14*, 2006.

Marco Schorlemmer and Yannis Kalfoglou. On semantic interoperability and the flow of information. In *Proceedings of ISWC03 Semantic Integration Workshop*, 2003.

S. Schulz, H. Stenzhorn, and M. Boeker. The ontology of biological taxa. *Bioinformatics*, 24(13):i313, 2008.

Stefan Schulz, Elena Beisswanger, Udo Hahn, Joachim Wermter, Anand Kumar, and Holger Stenzhorn. From genia to bioTop: Towards a top-level ontology for biology. In Brandon Bennett and Christiane Fellbaum, editors, *Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, pages 103–114, Amsterdam, 2006a. IOS Press.

Stefan Schulz, Elena Beisswanger, Joachim Wermter, and Udo Hahn. Towards an upper-level ontology for molecular biology. *AMIA Annu Symp Proc*, 2006, 2006b.

John R. Searle. *The Construction of Social Reality*. Free Press, January 1997.

Johanna Seibt. Forms of emergence in general process theory. *Synthese*, 166 (3):479–512, 2008.

Evren Sirin and Bijan Parsia. Pellet: An OWL DL reasoner. In Volker Haarslev and Ralf Möller, editors, *Proceedings of the 2004 International Workshop on Description Logics, DL2004, Whistler, British Columbia, Canada, Jun 6-8*, volume 104 of *CEUR Workshop Proceedings*, Aachen, Germany, 2004. CEUR-WS.org.

B. Smith, J. Williams, and S. Schulze-Kremer. The ontology of the gene ontology. *AMIA Annual Symposium proceedings*, pages 609–613, 2003. ISSN 1559-4076.

Barry Smith. Beyond concepts: Ontology as reality representation. In A. Varzi and L. Vieu, editors, *Proceedings of FOIS*, 2004.

Barry Smith, Jakob Köhler, and Anand Kumar. On the application of formal principles to life science data: A case study in the gene ontology. In *Proceedings of DILS 2004 (Data Integration in the Life Sciences)*, volume 2994 of *Lecture Notes in Bioinformatics*, pages 79–94, Berlin, 2004a. Springer.

Barry Smith, Werner Ceusters, Bert Klagges, Jacob Köhler, Anand Kumar, Jane Lomax, Chris Mungall, Fabian Neuhaus, Alan L. Rector, and Cornelius Rosse. Relations in biomedical ontologies. *Genome Biol*, 6(5), 2005a.

Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J. Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J. Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H. Scheuermann, Nigam Shah, Patricia L. Whetzel, and Suzanna Lewis. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotech*, 25 (11):1251–1255, November 2007.

C. L. Smith, C. A. Goldsmith, and J. T. Eppig. The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biol*, 6(1), 2005b. ISSN 1465-6914.

Cynthia L. Smith, Carroll, and Janan T. Eppig. The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biology*, 6(1), 2004b.

Peter G. Smith. Functions: Consensus without unity. *Pacific Philosophical Quarterly*, 74:196–208, 1993.

L.N. Soldatova, A. Clare, A. Sparkes, and R.D. King. An ontology for a Robot Scientist. *Bioinformatics*, 22(14), 2006.

J.F. Sowa and A.K. Majumdar. Analogical Reasoning. *Lecture Notes in Computer Science*, pages 16–36, 2003.

John F. Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole, Pacific Grove, 2000.

Judy Sprague, Leyla Bayraktaroglu, Yvonne Bradford, Tom Conlin, Nathan Dunn, David Fashena, Ken Frazer, Melissa Haendel, Douglas G. Howe, Jonathan Knight, Prita Mani, Sierra A. Moxon, Christian Pich, Sridhar Ramachandran, Kevin Schaper, Erik Segerdell, Xiang Shao, Amy Singer, Peiran Song, Brock Sprunger, Ceri E. Van Slyke, and Monte Westerfield. The zebrafish information network: the zebrafish model organism database provides expanded support for genotypes and phenotypes. *Nucl. Acids Res.*, pages gkm956+, November 2007.

Richard M. Stallman, Joshua Gay, and Lawrence Lessig. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Free Software Foundation, Boston, 2002.

Robert W. Taylor and Doug M. Turnbull. Mitochondrial dna mutations in human disease. *Nature Reviews Genetics*, 6(5):389–402, May 2005. ISSN 1471-0056.

The Plant Ontology Consortium. The plant ontology consortium and plant ontologies. *Comparative and Functional Genomics*, 3(2):137–142, 2002. ISSN 1531-6912.

Stephan Tobies. *Complexity results and practical algorithms for logics in Knowledge Representation*. PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.

A. Tolk and J. A. Muguira. The levels of conceptual interoperability model (lcim). In *Proceedings IEEE Fall Simulation Interoperability Workshop*. IEEE CS Press, 2003.

D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4130 LNAI:292–297, 2006a.

Dmitry Tsarkov and Ian Horrocks. FaCT++ description logic reasoner: System description. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning: Proceedings of the Third International Joint Conference, IJCAR 2006, Seattle, Washington, USA, Aug 17-20*, volume 4130 of *Lecture Notes in Computer Science*, Berlin, 2006b. Springer.

S. N. Twigger, M. Shimoyama, S. Bromberg, A. E. Kwitek, and H. J. and Jacob. The rat genome database, update 2007–easing the path from disease to data and back again. *Nucleic Acids Res*, 35(Database issue), January 2007. ISSN 1362-4962.

Alexandr Uciteli. Ontologien und kollaborative Taggingsysteme. Master's thesis, Department of Computer Science, University of Leipzig, 2008. forthcoming.

Abel Ureta-Vidal, Laurence Ettwiller, and Ewan Birney. Comparative genomics: Genome-wide analysis in metazoan eukaryotes. *Nat Rev Genet*, 4(4):251–262, April 2003.

André Valente and Joost Breuker. Towards principled core ontologies. In Brian R. Gaines and Mark A. Musen, editors, *Proceedings of the 10th Knowl-*

edge Acquisition Workshop (KAW'96), Banff, Alberta, Canada, Nov 9-14, 1996.

Francisco J. Varela, Humberto R. Maturana, and R. Uribe. Autopoiesis: The organization of living systems, its characterization and a model. *Biosystems*, 5(4):187–196, 1974.

Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *Proceedings of the 15th International Conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26*, New York, 2006. ACM.

Denny Vrandecic and Markus Krötzsch. Reusing ontological background knowledge in semantic wikis. In Max Völkel and Sebastian Schaffert, editors, *Proceedings of the First Workshop on Semantic Wikis – From Wiki to Semantics, SemWiki 2006, Budva, Montenegro, Jun 12*, volume 206 of *CEUR Workshop Proceedings*, Aachen, Germany, 2006. CEUR-WS.org.

Kai Wang. Gene-function wiki would let biologists pool worldwide resources. *Nature*, 439(7076):534, 2006.

Christoph Weidenbach, Uwe Brahm, Thomas Hillenbr, Enno Keen, and Christian Theobald. Spass version 2.0. In *In Proc. CADE-18*, pages 275–279. Springer, 2002.

D.L. Wheeler, D.M. Church, R. Edgar, S. Federhen, W. Helmberg, T.L. Madden, J.U. Pontius, G.D. Schuler, L.M. Schriml, E. Sequeira, et al. Database resources of the National Center for Biotechnology Information: update. *Nucleic Acids Research*, 32:D35, 2004.

P.L. Whetzel, R.R. Brinkman, H.C. Causton, L. Fan, D. Field, J. Fostel, G. Fragoso, T. Gray, M. Heiskanen, T. Hernandez-Boussard, et al. Development of FuGO: An Ontology for Functional Genomics Investigations. *OMICS: A Journal of Integrative Biology*, 10(2):199–204, 2006.

Wikimedia Foundation. MediaWiki. `http://www.mediawiki.org`, 2008.

World Health Organization. *ICD-9 CM 2001*. Practice Management Information, 2001. ISBN 1570661774.

Larry Wright. Functions. *Philosophical Review*, 1973.

C.J. Wroe, R. Stevens, C.A. Goble, and M. Ashburner. A methodology to migrate the gene ontology to a description logic environment using DAML+ OIL. *Pac Symp Biocomput*, 8:624–635, 2003.

Harris Wu, Mohammad Zubair, and Kurt Maly. Harvesting social knowledge from folksonomies. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 111–114, New York, NY, USA, 2006. ACM. ISBN 1-59593-417-0.

Satoko Yamamoto, Takao Asanuma, Toshihisa Takagi, and Ken I. Fukuda. The molecule role ontology: an ontology for annotation of signal transduction pathway molecules in the scientific literature. *Comparative and Functional Genomics*, 5(6-7), 2004.

Ernst Zermelo. Untersuchungen über die grundlagen der mengenlehre. *Mathematische Annalen*, 65:261–281, 1908.

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbstständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, 05.03.2009

Robert Hoehndorf

Scientific History

- Diploma in Computer Science, University of Leipzig, 2005.
- PhD in Computer Science, University of Leipzig, *expected 2009*.
  - Supervisors: Prof. Dr. Heinrich Herre and Dr. Janet Kelso.
  - Member of the graduate school *Knowledge Representation*.

List of publications and presentations

Parts of this work were previously published or presented at conferences.

Journal

- GFO-Bio: A biological core ontology. Robert Hoehndorf, Frank Loebe, Roberto Poli, Heinrich Herre, Janet Kelso. Applied Ontology, 2008.
- Representing default knowledge in biomedical ontologies: Application to the integration of Anatomy and Phenotype Ontologies. Robert Hoehndorf, Frank Loebe, Janet Kelso, Heinrich Herre. BMC Bioinformatics, 2007.

- A top-level ontology of functions and its application in the Open Biomedical Ontologies. Patryk Burek, Robert Hoehndorf, Frank Loebe, Johann Visagie, Janet Kelso, Heinrich Herre. Bioinformatics, 2006.

Conferences and Workshops

- Towards Ontological Interpretations for Improved Text Mining. Robert Hoehndorf, Axel-Cyrille Ngonga Ngomo and Michael Dannemann. Proceedings of 3rd International Symposium on Semantic Mining in Biomedicine, 2008.
- From Terms to Categories: Testing the Significance of Co-occurrences between Ontological Categories. Robert Hoehndorf, Axel-Cyrille Ngonga Ngomo, Michael Dannemann and Janet Kelso. Proceedings of 3rd International Symposium on Semantic Mining in Biomedicine, 2008.
- BOWiki: An ontology-based wiki for annotation of data and integration of knowledge in biology. Robert Hoehndorf, Joshua Bacher, Michael Backhaus, Sergio E. Gregorio, Jr., Frank Loebe, Kay Prufer, Alexandr Uciteli, Johann Visagie, Heinrich Herre and Janet Kelso. The 11th Annual Bio-Ontologies Meeting, 2008.
- BOWiki: A collaborative annotation and ontology curation framework. Michael Backhaus, Janet Kelso, Heinrich Herre, Robert Hoehndorf, Frank Loebe, Kay Pruefer, Johann Visagie. Proceedings of Workshop on Social and Collaborative Construction of Structured Knowledge, 2007.
- A proposal for a gene function wiki. Robert Hoehndorf, Kay Pruefer, Michael Backhaus, Heinrich Herre, Janet Kelso, Frank Loebe, Johann Visagie. OTM Workshop Proceedings, 2006.

- The design of a wiki-based curation system for the Ontology of Functions. Robert Hoehndorf, Kay Pruefer, Michael Backhaus, Johann Visagie, Janet Kelso. Proceedings of the Joint BioLINK and 9th Bio-Ontologies Meeting, 2006.

Other

- Ontologies in Biology. Janet Kelso, Kay Pruefer and Robert Hoehndorf. In *Theory and Application of Ontologies. Volume II: Ontology: the Information-science stance*. Editors: Michael Healy, Achilles Kameas, Roberto Poli. In Press.
- General Formal Ontology (GFO): A Foundational Ontology Integrating Objects and Processes. Part I: Basic Principles (Version 1.0).Heinrich Herre, Barbara Heller, Patryk Burek, Robert Hoehndorf, Frank Loebe and Hannes Michalek. Onto-Med Report Nr. 8. Research Group Ontologies in Medicine (Onto-Med), University of Leipzig, 2006.

Talks

- Representing defaults in biomedical ontologies. Presented at Dagstuhl Seminar on Ontologies and Text Mining for Life Sciences : Current Status and Future Perspectives, Dagstuhl, Germany, 2008.
- Ontology-integration and non-monotonic reasoning. Presented at the Dagstuhl Seminar “Towards Interoperability of Biomedical Ontologies”, Dagstuhl, Germany, 2008.
- Towards interoperability between anatomy and phenotype ontologies. Presented at 3rd Student Council Symposium at the ISMB/ECCB 2007.