Ontologies for the life sciences

Steffen Schulze-Kremer and Barry Smith

Preprint version of article in *Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics*, New York and London: John Wiley and Sons, vol. 4, 2005, http://www.wiley.com/legacy/wileychi/ggpb/toc4.html.

Abstract

Where humans can manipulate and integrate the information they receive in subtle and ever-changing ways from context to context, computers need structured and context-free background information of a sort which ontologies can help to provide. A domain ontology captures the stable, highly general and commonly accepted core knowledge for an application domain. The domain at issue here is that of the life sciences, in particular molecular biology and bioinformatics. Contemporary life science research includes components drawn from physics, chemistry, mathematics, medicine and many other areas, and all of these dimensions, as well as fundamental philosophical issues, must be taken into account in the construction of a domain ontology. Here we describe the basic features of domain ontologies in the life sciences and show how they can be used.

Keywords: domain ontology, molecular biology, bioinformatics, philosophy

Motivation

The multitude of heterogeneous and autonomous data resources available to life scientists today includes genomic (Fasman et al, 1996), cellular (Jacobson, Anagnostopoulos, 1996), structural (Bernstein et al, 1977), phenotype (McKusick, 1994) and a range of other types of biologically relevant information (Bairoch, 1993). Even for one type of information, e.g. DNA sequence data, there exist several databases of different scope and organisation (Fasman et al, 1996; Keen et al, 1996; Benson et al, 1997).

There exist terminological differences (alternative e.g. synonyms, aliases), syntactic differences (in file structure, separators, spelling) and semantic differences (the same words are used to mean different things in different sources). Conventions for naming data objects, object identifier codes and record labels differ between databases and do not follow a unified scheme. Even terms for important high-level concepts and relations that are fundamental to life science are often used in conflicting or ambiguous ways (Smith and Rosse, 2004).

One prominent example is the concept *gene*. For GDB (Fasman et al, 1996) a gene is a "DNA fragment that can be transcribed and translated into a protein". For GenBank (Benson et al, 1997) and GSDB (Keen et al, 1996), a gene is a "DNA region of biological interest with a name and that carries a genetic trait or phenotype". The latter definition is problematic, not only because it makes the answer to the question of which genes exist depend on the vagaries of human naming acts, but also because it comprehends non-structural coding DNA regions like intron, promoter and enhancer. There is a clear distinction between the two underlying notions of *gene*, but both continue to be used, thereby adding another level of complexity to

data integration. Another term with multiple meanings is *protein function* (biochemical function, e.g. enzyme catalysis; genetic function, e.g. transcription repressor; cellular function, e.g. scaffold; physiological function, e.g. signal transducer).

If a user queries a database using an ambiguous term she must herself take responsibility for verifying the congruence in meaning between her use of the term and what the database returns. Those semantic incompatibilities which are known must be resolved with each new search result, while those which are unknown propagate errors behind the scenes. Ontologies can help to resolve such incompatibilities in a global fashion, for example by flagging cases where a single term is used for entities in distinct ontological categories and enforcing manual disambiguation.

The advent of microarray technology for mRNA expression analysis requires additional standardisation in terminology for characterising not only genes, tissues and samples but also experimental setups and the factors involved in mathematical post-processing of raw measurements. A comparison between different experiments is only feasible if consistent terminology and standardised input forms are used. The development of ontologies suitable for this purpose is pursued in the MGED consortium (Brazma et al., 2001).

Standardised nomenclatures are required also where the new more integrated approach to biology has led to the merging of subfields with historically independent origins. This applies for example to genetics, protein chemistry, and pharmacology. Pharmaceutical companies have expressed an urgent need to harmonise the technical languages of these fields so that the knowledge derived from each can be stored in a unified way. The fast growth of sequence, structure, expression, metabolic and regulatory data pertaining to many organisms adds additional pressure to utilise standardised and compatible nomenclature in molecular biology.

Text mining and natural language understanding in biology can also profit from ontologies. Currently it is mostly techniques based on string-based statistical and proximity approaches that are applied to text analysis. Ontologies can however support parsing and disambiguating sentences by enforcing grammatically compatible uses of terms via rules for ontologically compatible combinations of referents (Jackson and Ceusters, 2002; Nirenburg and Raskin, 2001).

To support consistent reporting of results in molecular biology, it will be necessary to develop controlled vocabularies comprehending the most important and frequently used terms with coherent definitions in such a way as to allow database managers, curators and annotators to create new and more coherent software and database schemata, to provide exact, semantically precise specification of the concepts used in existing schemata and to curate and annotate existing database entries in a consistent way.

It is important to understand that terminological ambiguity affects not only interoperability on the level of computers but also communication between human beings. But where humans have the facility to resolve ambiguities in an efficient way, computer programs and databases have, for the moment at least, no analogues of the capabilities which allow such concurrent disambiguation. Ontologies are one important tool designed to make up for this shortfall.

Data integration must overcome the problem of syntactic and semantic heterogeneity. While some syntactic incompatibilities, e.g. prefix versus postfix operators, can be easily aligned automatically, incompatibilities which arise from differences in meaning require manual resolution within a common unifying framework. Thus for example each table, object, etc., of

one database must be manually aligned with the corresponding components of each other database that is to be integrated. If we begin with *n* databases, and if the process of integration is carried out in pairwise fashion, then this requires *n*n* attempts at resolution of differences in meaning. If, however, a single ontology exists that can serve as a central switchboard for those *n* databases, then the integration effort is reduced from *n*n* to *n*, since each database has to be mapped to the ontology only once in order to become interoperable with any other database (Köhler and Schulze-Kremer, 2002).

Overview of ontologies

Work in ontology can be classified along a number of distinct dimensions. Most important is the distinction between 1) ontology as the study of beings or entities, the study of what exists at the highest level of abstraction – ontology as a branch of philosophy – and 2) domain ontologies, which result from the analysis of particular domains of reality and correspond broadly to separate areas of scientific inquiry.

Ontologies in either of these senses are in principle language independent. Thus there can be a German equivalent to an English domain ontology, even if the actual translation process need not be trivial. A domain ontology may also be formalizable in some artificial language such as the language of first-order predicate logic, Description Logic (Rector et al., 2003), or some other form of representation formalism.

Ontology as a branch of philosophy can be important in bringing clarity to the life science field even before we enter the specific territory of biological domain ontologies. Thus consider the fact that "DNA" can be used to designate quite different entities. First, there is DNA as physical stuff which can be measured with a spectrophotometer. Second, there is the class of all of chemical substances which share the general features common to DNA molecules. Third, there is the family of specific types of sequences or strings in the sense of abstract structures that can be subject to mathematical operations but cannot be measured or detected in reality. Fourth, "DNA" is often used in the lab to refer to a particular instance of a sequence, e.g. the DNA sequence of E. coli K12 which can be stored in a database and needs carrier (memory chip, paper) to survive. These and similar distinctions – between classes and instances, between sequences and stuffs – are distinctions between philosophical categories, which have analogues in a number of different domains, and which are unfortunately often the source of confusions, not least when attempts are made at systematic representation of empirical knowledge for purposes of automatic information retrieval.

Among domain ontologies we can draw distinctions between ontologies of varying scope and content. Thus we can distinguish between

- upper-level ontologies domain which are primarily concerned with the small number of general categories (such as *cell*, or *gene*, or *molecule*) that serve as the basis of our understanding of a particular domain;
- terminology-based ontologies, which are centred around the many highly specific terms used in the formulation of the results of scientific inquiry (such as *enzyme active site formation*, *postsynaptic membrane* or *receptor signaling protein activity*).

Since the world around us in general, and molecular biology and bioinformatics in particular, are such as to manifest an enormous multi-dimensional complexity, no single ontology can suffice for every purpose. Rather we must content ourselves with ontologies representing different views of reality created in connection with practical goals – often views of reality at

different levels of granularity (Bittner and Smith, 2003). Thus, before building an ontology it is important to understand what is its intended use, since otherwise there is a risk of being overwhelmed by the multitude of facets by which we are confronted. This aspect is acknowledged by the use of the term "situated ontologies" (Mahesh and Nirenburg, 1995) to emphasise the fact that a domain ontology should be evaluated with respect to its intended use.

The larger terminology-based ontologies clearly must be constantly updated in light of new experimental evidence and developments in language usage. At the highest levels, however, an ontology is designed to be much more stable than e.g. a database schema. The latter is dependent on specific choices concerning a database representation formalism, database management system, and requirements from the applications which access the data. Since an upper-level ontology is of its nature designed to be easily translatable from one knowledge representation formalism to another (given equivalent expressive capability) it can also be converted into a database schema. But a domain ontology addressing the fundamental categories and relations of an application domain is designed to be independent of given software implementations. When new knowledge classes are discovered the ontology should be extendible in relatively straightforward ways, along lines to be described below.

The interplay between ontologies, biology, computer science and philosophy is depicted in Figure 1.

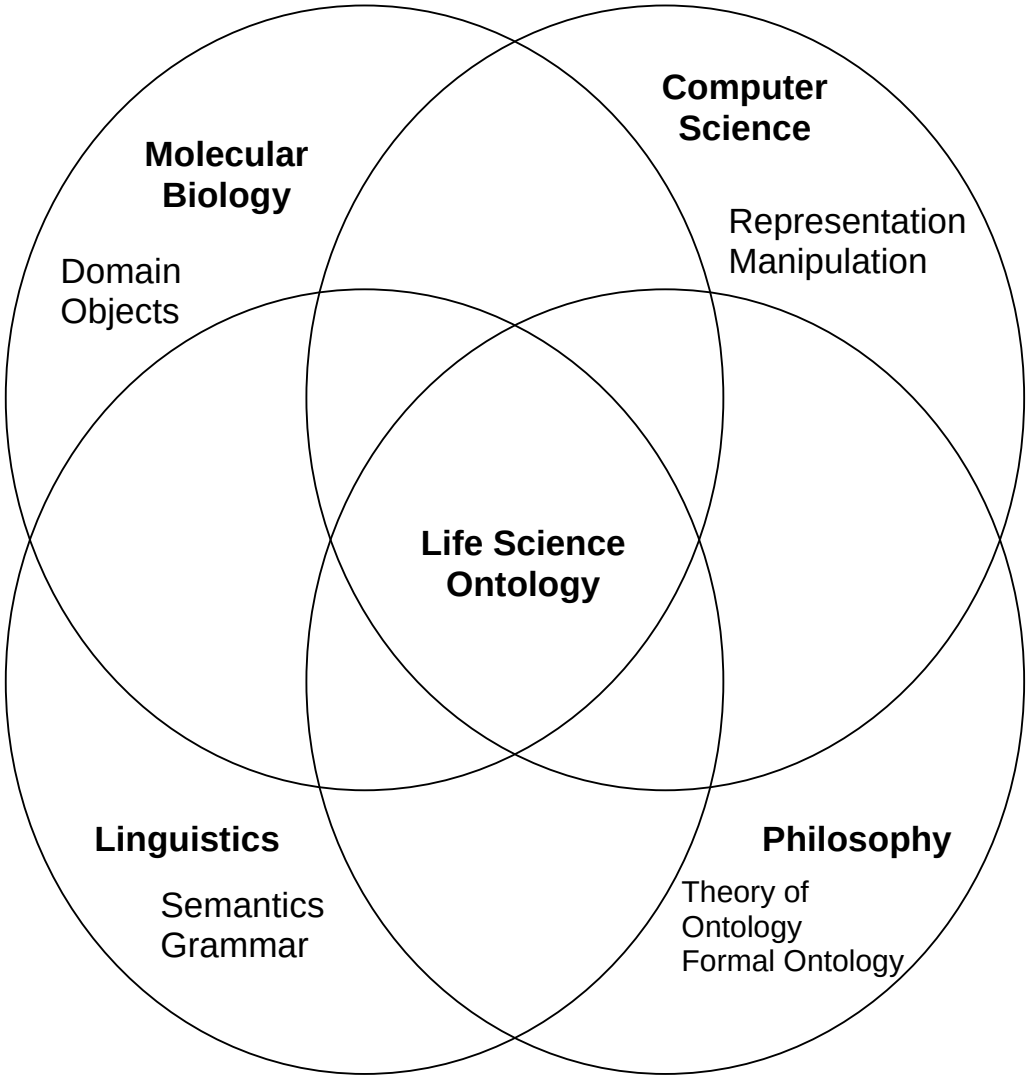


Figure 1: Molecular biologists discover facts that need to be organised and stored in databases. Computer scientists provide techniques for data representation and manipulation. Linguists help organise the meanings underlying database labels. Philosophers provide formal theories of basic ontological relations and principles governing best practice in definition and classification.

Upper-level ontologies

The first important ontologist was Aristotle (384-322 BC) who, among many other things, pursued the question of how reality is organized into classes or universals. His solution is presented in his *Categories* which can be seen as the first upper-level ontology (Barnes, 1984). From Aristotle's point of view, ten categories suffice to express anything that can be known about something:

- Substance
- Quantity
- Quality

- Relation
- Place
- Time
- Situation
- Condition
- Action
- Affection

Of course, from the point of view of the annotation of entities in molecular biology, the categories here distinguished will not suffice. However, if one subscribes to the view that Aristotle provides a still serviceable account of the most fundamental set of categories, then one could see molecular biology and other special life sciences as the results of further sub-classifications of Aristotle's categories into ever more specific kinds.

Another feature of Aristotle's ontology is the paucity of interconnections between his ten categories, each of which is assumed to be an atomic category in the sense that it cannot be meaningfully decomposed into smaller units. Aristotle does allow that substance is the primary category, so that instances of all other categories are *dependent on* instances of substance. As concerns the interrelations between the nine 'accidental' categories, however, he tells us too little. Later ontologists added further interrelations between their basic categories, including a taxonomy of different kinds of dependence relations provided by Husserl in his *Logical Investigations* (English translation, 2 vols., London: Routledge and Kegan Paul, 1970) and the related taxonomy offered in our own day by the DOLCE ontology (Guarino, 1997, Masolo et al., 2003).

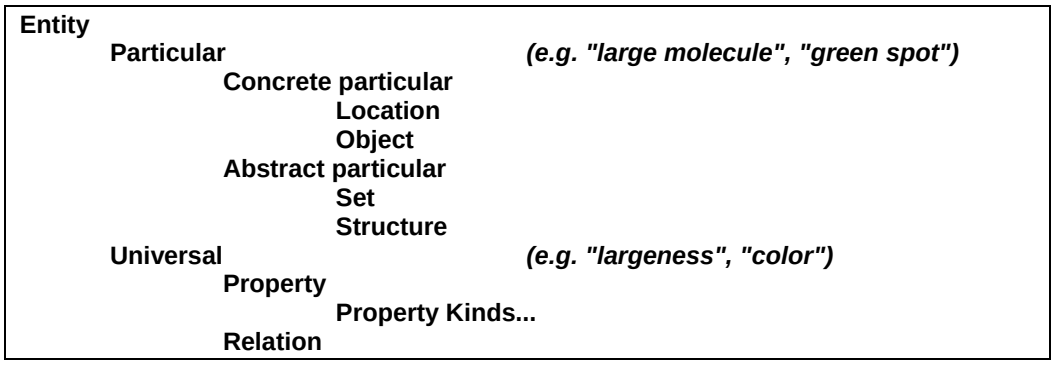


Figure 2: DOLCE top-level ontology.

A contemporary philosophically motivated upper-level ontology for Molecular Biology is advanced in (Schulze-Kremer, 1997). Like DOLCE, this starts from a single node, and it also extends to incorporate those physical and abstract entities that are relevant for biology and bioinformatics.

The upper level of the Molecular Biology Ontology is shown in Figure 3. Starting from the root node *Being*, which includes all entities of any sort, it distinguishes two disjoint classes of *Object* and *Event*, which are discriminated based on their mode of existence in time. An *Object* retains its identity from one moment to the next; an *Event*, in contrast, is divided into temporal parts or phases and unfolds itself through these phases in successive moments of time. This distinction is passed on to all subclasses of *Object* and *Event*. The class *Object* is further subclassified into *Individual Object* and *Property*. Both preserve their identity from one moment to the next. They are discriminated on the basis of their ability to exist in a self-

contained way. An *Individual Object* can stand alone, whereas a *Property* always needs another *Object* or *Event* which it is the property of. *Property* is further subclassified on the basis distinctions in arity, into *Attribute*, a property with only one argument, and *Relation*, a property relating two or more *Beings*.

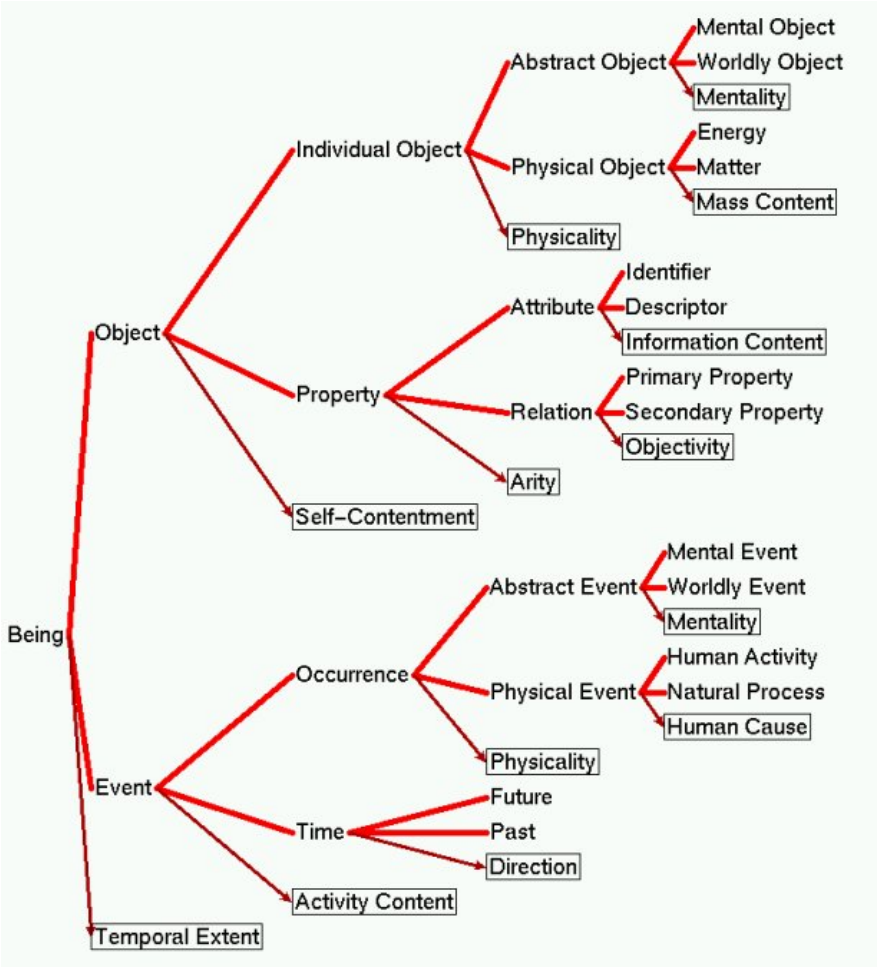


Figure 3: Upper Level of the Molecular Biology Ontology of (Schulze-Kremer, 1997). Links represent the IS-A-SUBCLASS-OF relation. Discriminating criteria are marked by arrows and boxes; thick lines denote disjoint subclasses.

Attribute is subclassified into *Identifier* (for example "ID-2394873")_and *Descriptor* (for example "E. coli R12 DNA") on the basis of whether an item simply labels an entity or carries additional information about it. *Relation* can be subclassified into *Secondary Property*, each instances of which involves some relation to a cognitive subject, and *Primary Property*, whose instances are objective and measurable entities such as mass or charge. (Locke, 1975)

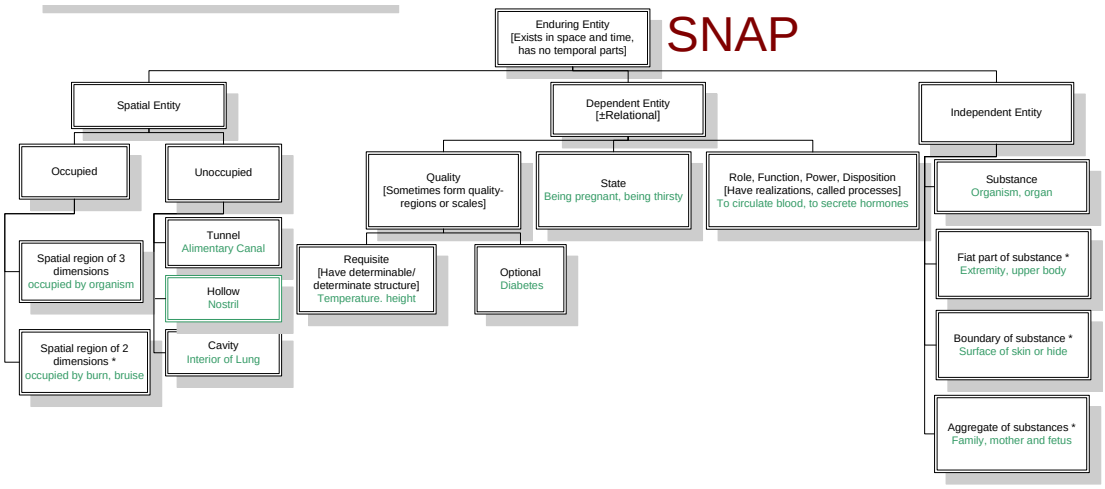
Individual Object is subclassified via the criterion of physicality into *Abstract Object*, which has no physical equivalent *per se* (except the capacity for being represented in writing, etc.), and *Physical Object*, which must have a defined spatial extent and/or energy content. *Physical Object* is further subclassified on the basis of mass content into *Energy* and *Matter*.

Abstract Object is further subclassified via the criterion of mentality – i.e. according to whether it refers to an object within the mind or to an object in the outside world – into *Mental Object* (e.g. thought, love) and *Worldly Object* (e.g. circle, sequence).

The category *Event* is subclassified via the criterion of activity into *Occurrence*, in the instances of which at least one object participates, and *Time*, where this is not the case.

Time is further subclassified according direction into *Past* and *Future*. Because the present moment is strictly speaking instantaneous, it does not appear in this branch. Analogous to abstract and physical objects, *Occurrence* is subclassified via the criterion of physicality into *Abstract Event* and *Physical Event.* The former is further classified on the basis of the criterion of mentality into *Mental Event* (e.g. thinking, feeling) and *Worldly Event* (e.g. binding, transport). *Physical Event* is subclassified on the basis of whether it is initiated by human intention into *Human Activity* and *Natural Process*. The operations of man-made devices in laboratory experimentation fall under the first of these two headings, the natural processes subject to molecular biological analysis under the second.

Another closely related more general purpose ontologies is the Basic Formal Ontology (Grenon et al, 2003; Grenon and Smith, forthcoming) which draws a still more radical distinction than that between *Objects* and *Events* by distinguishing two separate ontologies, called SNAP, for enduring entities (such as organisms, cells, enduring attributes, functions and dispositions) and SPAN, for processes or events.



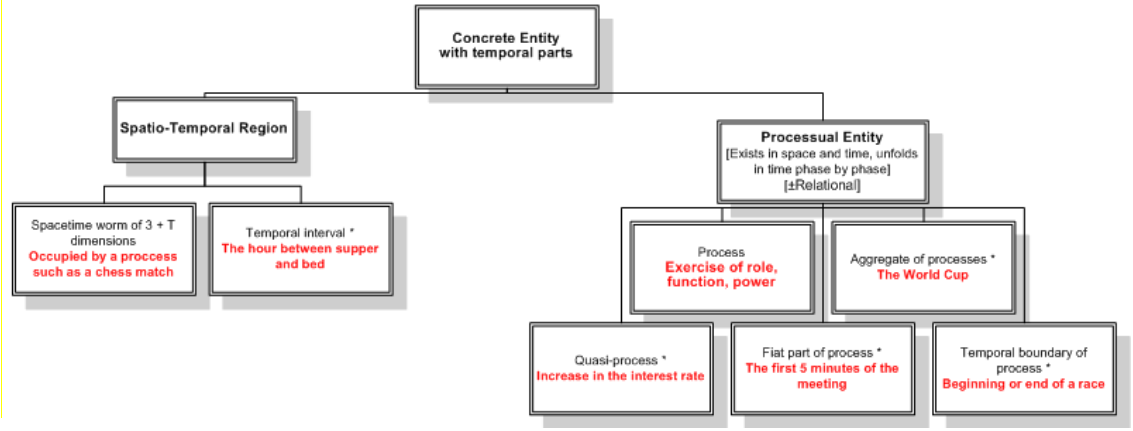


Figure 4: Top-level categories of the SNAP an SPAN ontologies of Basic Formal Ontology (BFO).

What is an ontology?

Three conceptions of domain ontologies can be distinguished:

(i) A concise and unambiguous description of principal relevant entities with their potential, valid relations to each other (Schulze-Kremer, 1998).

(ii) A system of categories accounting for a particular vision of the world (Guarino, 1998).

(iii) A specification of a conceptualization (Gruber, 1993).

(i) and (ii) represent a view of ontology broadly in the spirit of Aristotle's ontology and of traditional philosophical ontology and realized in the top-level category systems illustrated above. (iii) rests on a view of ontologies rooted in logic-based knowledge representation; it tells us that to build an ontology we must analyse our domain of interest and represent the basic concepts that are exemplified therein in some formal language. Although this describes in broad terms some of what is involved in ontology development, the definition itself does not yet go far enough, and we here specify some further requirements which a domain ontology should satisfy.

1. Each term in an ontology should be defined as precisely as possible. Definitions are the basis for establishing the relations between terms in an unambiguous way and are indispensable when laying down the foundation of an ontology. But writing good definitions is often very hard (not least in the domains of the life sciences). How detailed does one have to be in specifying the concept at hand to make it distinguishable from others already present and those that are to be added in the future? Since this question cannot be answered *a priori*, definitions frequently need to be updated and such updating should be supported by ontology editing software.

2. The set of concepts covered by an ontology must comprehend all the categories instantiated by the entities in the application domain at the pertinent level of granularity.
3. There should be a specification of the structure or organization of an ontology. This can be by means of an ontology representation language such as KIF (Genesereth and Fikes, 1992) or DAML+OIL (Joint United States/European Union ad hoc Agent Markup Language Committee, 2001), or it can be a semi-formal or informal specification, for example utilizing UML diagrams or natural language.
4. An ontology should be associated with documentation specifying its expressive capabilities, as well as its scope and the levels of granularity of the entities with which it deals.
5. Standardised procedures should be defined specifying how to add, modify or remove categories.
6. Indications should be given specifying how the ontology and its categories can be used, e.g. what kind of inference is supported, how it can be applied for example to support information retrieval.

Components

The building blocks of an ontology are as follows:

1. *Terms*; these are the atomic building blocks of an ontology conceived as a syntactic structure.

2. *Links between terms*, for instance the link between ‘mammal’ and ‘animal’ representing the fact that the former class is included as subclass within the latter.

3. *Predicates*, for instance TRANSCRIBED-BY, CATALYSED-BY.

4. *Propositions*. These are definite statements about (parts of) the world, sometimes encoded in an ontology representation language and typically representing relations between classes. They include as a special distinguished class: *axioms*, which are fundamental statements which are assumed to be true and which are given without proof. The set of axioms must be logically consistent. Some propositions can be derived from axioms via logical reasoning.

5. A *logical formalism*, including logical constants such as *and* and *or*, variables and quantifiers.

6. *Definitions*, which can be divided into real, nominal and ostensive. Real definitions capture the essence or nature of the entities referred to by the term to be defined: they specify the universal marks which all instances of a defined category share (Michael et al, 2001). Nominal definitions reflect the way a given term is used. They may be analytic, which means that they decompose the concept to be defined into its necessary and sufficient conditions (e.g. bachelor is an unmarried man). Or they may be stipulative, which means that they serve to introduce a new concept (e.g. an alpha-helix peptide is a polypeptide molecule with the following geometry ..., non-lytic viral exocytosis is the exit of the virion particle from the host cell by exocytosis, without causing cell lysis). Ostensive definitions define concepts by pointing to or by enumerating examples, as when we define “yellow” by pointing to yellow things.

Some principal rules governing the formulation of good definitions are as follows:

- Definition should not be negative in form, e.g. protein is not made of DNA.
- Definition should not be too broad. Thus: *proteins are chemicals* is to be rejected.
- Definition should not be too narrow. *Proteins are covalent strings of amino acids* is to be rejected because it does not embrace post-translational modifications and quaterny structure.
- Definitions should not be circular (*A protein is made of a protein chain*). Definitions should not convey extra or redundant information. Consider e.g.: *regulation of protein catabolism is any process that modulates the frequency, rate or extent of the breakdown into simpler components of a protein by the destruction of the native, active configuration, with or without the hydrolysis of peptide bonds*

In addition we have that in the world toward which an ontology is directed. This includes:

1. The *classes of entities* in reality to which terms refer. These are generalisations of instances, e.g. *gene*, *protein*, and connected with each other *inter alia* by the SUPERCLASS-OF and SUBCLASS-OF relations. Often SUBCLASS-OF is also called IS-A. If one class stands in an immediate SUCLASS-OF relation to a second class we say that they stand in the relation of *child* to *parent*; two classes with the same parent are called *siblings*; classes with no children are called *leaves*; a class with no parents is called a *root*.

2. These *entities* themselves, i.e. the instances of classes, which are individuals (such as this organism or this sample of protein) connected by the IS-OF-TYPE relation to at least one class. Normally instances play a role only generically – thus we define the PART-OF relation between classes as follows:

   > *A* PART-OF *B* =def. all instances of *A* are such that there is some instance of *B* of which they are a part. <*REFER TO SMITH-ROSSE PAPER AGAIN)*

   However, specific experiments or specific research findings may enjoy a biological significance and may be referred to explicitly in a corresponding domain ontology.

3. *Attributes* and *Relations* to which the predicates refer.

Distinguishing Marks of Ontologies

An ontology is to be distinguished from a *knowledge base* conceived as a *collection of statements of fact.* Rather an ontology is a specification of the principal categories instantiated by the entities in a given domain of reality and at a given level of granularity. Thus one might say that an ontology is a certain highly general type of knowledge base filled with knowledge about categories and their ontological relations.

An ontology is to be distinguished from *a model of an application domain* ; rather it is a compendium of the building blocks of such a domain together with their valid modes of combination, the whole formulated as a theory.

An ontology is *not a database schema,* i.e. it does not describe the categories, data types and organisation in a database. Rather it is a specification of the classes and relations among entities in the real world. Such datatypes may represent such classes and relations also, which means that a database schema can be derived from an ontology by adding data type information and translating the knowledge representation formalism into a database management format. A database schema can also be used as a starting point for ontology building. The corresponding entity-types and attributes can then be taken as an initial set of categories to populate an ontology.

An ontology is *not a taxonomy* which knows only about superclass and subclass relations; an ontology is open to other types of fundamental relations, including temporal, mereological (part-whole), topological, compositional and casual relations, as well as dependence relations e.g. between qualities and functions and the objects which they are the qualities and functions of.

An ontology is *not a vocabulary* or *dictionary* or *thesaurus,* since all these lack the logical organization which an ontology demands in order to support computational inference. Moreover they standardly do not describe the hierarchy and relations between the entities designated by the terms they include. In an ontology one can follow a path from any term to any other along the edges of some IS-A hierarchy or by following other relations.

An ontology is *not a semantic net.* Rather, a semantic net is one sort of formalism that can be used to represent ontologies, but this formalism can be used for other purposes also.

How to build an ontology?

There are several ways to build an ontology, some of which are surveyed in (Fernandez et al, 1997). A currently popular methodology, especially in the biological and medical domains, is text mining (Maedche and Staab, 2003; Kashyap et al, 2003). Here we describe the method used in (Schulze-Kremer, 1998).

In order to assemble the components described above (terms, propositions, axioms, formalism, subclass relations, etc.), we apply the following steps in succession:

- Collect an initial list of domain relevant terms. These can be taken e.g. from database tables or text books. This list will have to cover the main central objects and processes in that application domain and will be extended when populating the ontology.
- Provide a unique and explicit definition for each high-level term. This definition must be precise enough to discriminate the reference of that term from all other entities referred to in the ontology and it should be detailed enough to provide a clear representation of the term's meaning. Experts often have only a tacit understanding of the technical terms they use; thus it is often difficult to provide an explicit formal definition, not least given the ambiguities by which many terms are affected. Ontology management software should therefore be capable of disambiguating terms with multiple meanings, for example by imposing unique identifiers.

With the move to lower-level terms as the details of an ontology are filled in, we need to add subclasses to classes already recognized. Here it is important to fix upon and to use consistently and explicitly one and only one discriminating criterion for each superclass. When this design principle is followed the ontology automatically manifests the properly hierarchical structure of a tree of subclasses that can also be used as a decision tree when

adding or searching for terms. To use the hierarchical tree of subclasses one starts at the single top-most node and applies the discriminating criterion at each level to the new term. Depending on the characterization of the new term with respect to the discriminating criterion a decision is reached which subclass to follow. For full expressivity this will require a choice among a number of inheritance modes (e.g. multiple distinct inheritance, where the subclass can be distinctly interpreted depending on its parent types, e.g. *queen*, seen as a monarch or as a piece on the chess board; or combined cumulative inheritance, where all properties of all parents are inherited by the child term, e.g. a protein-DNA complex inherits features from both DNA and protein). Ideally one should use the same classification criterion throughout the ontology, as is done in the Foundational Model of Anatomy where the single criterion of *structure* is used (Rosse and Mejino, 2003).

- Be explicit about the disjointness of subclasses, i.e. state where subclasses of a single class can or cannot overlap. For example, the distinction of molecules into protein and DNA is disjoint, since no molecule can be both at the same time. This greatly helps to focus searches through the subclass hierarchy, since if it is known in advance that a subclassification is disjoint then only one of its subclasses need be followed when proceeding further down the hierarchy. Using only disjoint subclasses is also called the 'single inheritance' mode and implies the creation of a true hierarchy with no fusion between branches as one moved down the tree to more specialized classes.
- Obtain complete connectivity via IS-A-SUBCLASS or INSTANTIATES relations (or their inverses) from any one term in the ontology to any other term. (Thus at least one IS-A-SUBCLASS or INSTANTIATES relation (or their inverses) must exist for each term. In this way we ensure that all terms are defined consistently in such a way as to form a single ontology with no separate ontological islands which would require integration later on.
- Use one root node only. This root must be general enough to embrace the entirety of the domain-relevant categories, since otherwise different conflicting lineages could emerge.
- Add background knowledge for each term to express domain-relevant properties, but keep this strictly separate from the definitions. The attributes and relations should themselves be reified first (i.e. added as individual terms to the ontology) for maximal inference capability.
- Add links from terms in the ontology to other ontologies, natural language dictionaries, database keywords, etc., thereby interfacing the ontology with applications of various types and supporting its integration with other information sources.

Guidelines on syntax

The following have emerged as rules of good syntactic practice which serve to make an ontology more manageable for human users.

- Use singular rather than plural forms in a term name.
- Use lower case letters only for terms for classes.
    - Names of instances should begin with a capital letter, e.g. E.coli-Strain-K12-Sequence.
    - Acronyms should be upper case throughout.
- Observe syntax requirements of the selected representation formalism.
    - Quotes, hyphens, etc., may be required or forbidden.
    - Unique names may be required by the representation formalism.

- When naming a subclass start by specialising the name of the superclass.
    - The specialising text should be appended, not prepended.
    - This makes the term easier to be recognized.
- Always provide aliases where known and keep records of equivalences.

If these rules are followed this means that when adding a new term one can use the discriminating criteria of the ontology as a decision tree to travel down from the root and at each branch deterministically decide where each new term should belong. One then either finds that the term is already there (possibly under another name), and the insertion process consists merely in the addition of another alias to the existing term. Or one ends at some point in the hierarchy where no appropriate superclass can be found. This is then the place where the new term should be added, either directly or by introducing intermediary terms designed to separate already existing terms and branches from the branch to be newly created. This also guarantees that the ontology remains consistent after a new term has been inserted. Searching for a known or even unknown term can be done in the same way, i.e. by traversing the decision tree of discriminating criteria.

There are several difficulties to be overcome when building an ontology. Some difficulties are inherent to the ontology building process, others reflect specific application areas. First is the problem of determining the best (e.g. most informative) criterion of subclassification for a given class. Here one faces a to some degree arbitrary decision as to how to proceed in creating subclasses, and this implies that there will in general not be one single optimal ontology for a given domain at a given level of granularity but rather only ontologies more or less well-integrated with other information resources, have greater or lesser reasoning power, and so on. Also, since the information content of the terms that will need to be added to an ontology in the future cannot be known in advance, the choice of subclassifying criteria may lead to a more complex inheritance structure than necessary, and may thus itself have to be revised.

Other difficulties arising in the ontology building process are the following:

- For many application domains it is unrealistic to aim for exhaustiveness of an ontology. However, each domain ontology must cover all entities that are of practical relevance for its application domain in practice.
- The arity of relations may be a source of confusion.
    - Relations may be 1:1 (e.g. each person has a social security number which is unique to that person) or 1:Many (e.g. each single person has one weight under standard conditions but several people may have the same weight).
    - There are also 1:Many relations (e.g. a fountain pen writes in a single colour but one colour may be used by several pens) and Many:Many relations (e.g. a shirt may have several colours and each colour may be present in several shirts).
- There is the danger of over-elaborating an ontology by getting lost in those branches which are already well-understood and thus face few representational difficulties, but which are of little relevance to applications.
    - Therefore avoid superfluous ontological elements.
    - Check whether all details are really relevant to the intended purpose.
- Storing important data as free text or in comment fields rather than as defined terms can lead to confusion since free text fields are not well accessible for automatic reasoning. Thus wherever possible one should encode the quality with which to

annotate another term as a term itself in the ontology thereby making its scope explicit and enabling links to its inverse and other relations.
- Multiple inheritance should be carefully applied to make sure that the resulting subclasses really exist. Single inheritance is generally safer and easier to understand.

Of the domain specific difficulties in ontology building ill-defined technical terms, controversial technical terms, difficulty of analysing and separating homonyms, imprecise or lacking documentation of database categories are the most common.

The degree of abstraction and detail one chooses to adopt in building an ontology of a given domain at a given level of granularity determines the practical quality of the ontology which results, in a range from useless (too abstract, only upper-level terms defined which do not give sufficiently detailed information) to impossible to complete (ultimate granularity by going to the finest level of detail irrespective of application needs).

Ontology integration

Ontologies can be distinguished according to choice of axioms, which reflect those highly general background beliefs which are taken for granted by those working in the corresponding field. They can also be distinguished by the level of detail in the terms and definitions used and by the choice of subclassifying criterion. All these decisions should be stated explicitly. Important, too, is choice of *domain* – which can extend from single cell to whole populations of organisms – and of *granularity* (from molecule to whole organism).

Given these distinctions the goal of constructing one comprehensive ontology for the life sciences begins to seem like an unreachable goal. Many groups have thus concluded that they must rest content with several smaller task-oriented ontologies, although the question is still been extensively debated in the bio-ontologies community (Bio-Ontologies Workshop, 2004). The approach of building smaller ontologies must eventually however come to terms with the goal of combining ontologies together, for example via techniques for ontology integration of the sort outlined in (Ceusters et al, 2004). Such integration is by no means a simple matter, for given the heterogeneity of the domain ontologies contained in a system like the UMLS (Lindberg, 1990), the relevant integrating steps can hardly be carried out automatically. Each concept must be located and identified in the various sub-domain ontologies on the basis of manual search and comparison of definitions, decisions must be made whether concepts are similar enough to be merged or if several similar concepts need to be defined and cross-related. the corresponding concepts must then be added to a new ontology that will incorporate all sub-domain terms within a single consistent framework.

In the special case where the top-level terms of one ontology exactly match those of another ontology, the corresponding branches can be merged. However, in this case the data format (syntax, representation formalism) and the relations between terms of the two ontologies still need to be verified and if necessary manually cross-calibrated.

Since this process of manual ontology integration is quite cumbersome it might be more sensible to start of with an ontology that has a rather general upper level and can accommodate all of the diverse ontological types that are to be expected from the application domain. This was exactly the motivation for starting the MBO ontology described in (Schulze-Kremer 1997).

Applications of bio-ontologies

Ontologies can provide computer programs with a counterpart of much of the common-sense background knowledge that human experts bring to bear in processing information. The range of applicability of ontologies is thus rather broad, and two examples, database integration and data annotation, will be discussed here briefly.

Data annotation is the process of linking data records for example in a gene product database to other knowledge resources, for example cellular locations. (It is comparable to the process of indexing or cataloguing books or other literature items.) It is not a full-fledged ontology as described above that is required for this purpose, but rather only a controlled vocabulary, whose main purpose is to provide a fixed and unambiguous terminology for communication of research results. A controlled vocabulary of this sort is developed in the Gene Ontology (GO) project (Ashburner et al, 2000), which attempts to ensure consistency in gene product annotations by means of the so-called GO identifiers (GO ID). This means that new concepts get new GO IDs, old concepts keep their GO IDs even if they are moved to another location within the hierarchy, and GO IDs of deleted concepts are not reused. As an ontology GO has a rather simple, informal structure, which rests on the use of only two kinds of links: IS-A and PART-OF.

The GO approach has brought considerable benefits:
1) Work on populating GO could start immediately, without its authors needing to solve some of the intricate problems which face ontologies when formalized as logical theories.
2) Extending GO does not require the completion of complex protocols of formally determined steps but can be done intuitively by the expert biologist.
3) There are few formal constraints standing in the way of easy incorporation of existing biological terms into the GO vocabulary.
4) The principle of unique identifiers allows GO terms to be used for database annotation without consideration of their place in the GO hierarchy.

Focusing on the rapid population of GO has, however, a number of drawbacks (Smith et al, 2004):
1) It is unclear what kinds of reasoning are permissible on the basis of GO's hierarchies.
2) The rationale of GO's subclassifications is unclear. The reasoning that went into current choices has not been preserved and thus cannot be explained to or re-examined by a third party.
3) No procedures are offered by which GO can be validated.
4) There are insufficient rules for determining how to recognize whether a given concept is or is not present in GO. The use of a mere string search presupposes that all concepts already have a single standardized representation, which is not the case.

Open Biological Ontologies

GO is a part of the Open Biological Ontologies project (http://sourceforge.net/projects/obo), which offers a framework for the development of well-structured controlled vocabularies for shared use across different biological domains. Contributions to OBO obey the following guidelines:

1. The ontologies must be open source, which means that they may be used by all without any constraints other than that their origin is acknowledged and they are not redistributed in altered form. OBO ontologies are intended to be resources for the entire biological community.

2. The ontologies employ, or can be instantiated in, or can be easily converted into, a common shared syntax. This may be either the GO syntax, extensions of this syntax, or OWL. This criterion is not met in all of the OBO ontologies currently listed.

3. The ontologies are orthogonal to other ontologies already lodged within OBO. Thus different ontologies, for example ontologies for anatomy and process, can be combined through additional relationships, and the latter can then be used to constrain when terms from different ontologies can be jointly applied to describe one and the same biological entity from distinct perspectives.

4. The ontologies share a unique identifier space. The source of concepts from any ontology can be immediately identified by the prefix of the identifier of each concept. It is, therefore, important that this prefix be unique.

5. The ontologies include textual definitions of their terms. Many biological terms are ambiguous; thus each term should be defined in such a way that its precise meaning within the context of a particular ontology is clear to a human user.

Resources on (Bio-)Ontologies

The following include information relevant to work on bio-ontologies.

- Protégé 2000, an ontology editing software from Stanford Medical Informatics, is at http://smi.stanford.edu/projects/protege.
- GKB Editor, the Generic Knowledge Base Editor of Peter Karp and SRI <avoid acronyms> can be found at http://www.ai.sri.com/~gkb.
- OilEd, a simple ontology editor resides at http://www.ontoknowledge.org/oil/tool.shtml.
- The Semantic Web Community Portal at http://www.semanticweb.org has lot's of ontology related information and pointers.
- Ongoing KBS/Ontology Projects and Groups are listed at http://www.cs.utexas.edu/users/mfkb/related.html.
- On-To-Knowledge: Content-driven Knowledge-Management through Evolving Ontologies is a European funded research project at http://www.ontoknowledge.org.
- The previous Bio-Ontologies Workshops and other material on ontologies is compiled by Robert Stevens at http://img.cs.man.ac.uk/stevens.
- Cycorp has its own webpage at http://www.cyc.com.
- Formal Ontology in Information Systems is an international conference series on ontologies with a webpage at http://www.fois.org.
- Barry Smith has an extensive collection of works on ontology development in general and biomedical ontologies in particular at http://ontology.buffalo.edu/smith.

References

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. and Sherlock, G. (2000). Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. Nature Genet. 25, 25-29.

Bairoch, A. (1993). The ENZYME data bank. Nucleic Acids Res. 21, 3155-3156.

Barnes, J. (1984).The Complete Works of Aristotle. (Ed.) 2 vols, Princeton, 1984.

Benson, D. A., Boguski, M. S., Lipman, D. J. and Ostell, J. (1997). GenBank. Nucleic Acids Res. 25, pp1-6.

Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F., Brice, M.D., Rodgers, J.R., Shimanouchi, O.K.T. and Tasumi, M. (1977). The Protein Data Bank: a computer-based archival file for macromolecular structures. J. Mol. Biol. 112, pp535-542.

Bio-Ontologies Workshop (2004). http://bio-ontologies.man.ac.uk

Bittner, T., Smith B. (2003). A Theory of Granular Partitions. In: Duckham, Matthew, Michael F. Goodchild, Michael F. Worboys (eds.): Foundations of Geographic Information Science. Taylor & Francis, London: 2003, pp117-151.

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C., Gaasterland, T., Glenisson, P., Holstege, F.C., Kim, I. F., Markowitz, V., Matese, J.C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J. and Vingron, M. (2001). Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nat Genet. 29, pp365-371.

Ceusters, W., Smith B., Matthew Fielding J. (2004). LinkSuite: Formally Robust Software Tools for Ontology-Based Data and Information Integration. In: Proceedings of DILS 2004 Data Integration in the Life Sciences (Lecture Notes in Computer Science 2994) Springer, Berlin: 2004, pp79-94

Fasman, K. H., Letovsky, S.I., Cottingham, R.W. and Kingsbury, D.T. (1996). Improvements to the GDB Human Genome Data Base. Nucleic Acids Res. 24, pp57-63.

Fernandez, M., Gomez-Perez, A. and Juristo, N. (1997). METHONTOLOGY: From Ontological Arts Towards Ontological Engineering. In: Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering, Stanford, USA, pp33-40, March 1997.

Genesereth, M.R., Fikes, R.E. (1992). Knowledge interchange format reference manual, version 3.0, Technical Report Logic-92-1, Computer Science Department, Stanford University.

Grenon, P., Smith B. (forthcoming). SNAP and SPAN: Prolegomenon to Geodynamic Ontology. In: Spatial Cognition and Computation, http://ontology.buffalo.edu/bfo/SNAP.pdf, http://ontology.buffalo.edu/bfo/SPAN.pdf

Grenon, P., Smith B., Goldberg L. (2003). Biodynamic Ontology: Applying BFO in the Biomedical Domain. In: Pisanelli, Domenico M. (ed.): Ontologies in Medicine: Proceedings of the Workshop on Medical Ontologies, Rome, October 2003. IOS Press, Amsterdam.

Gruber, T. R. (1993). Knowledge Acquisition 5, pp199-220.

Guarino, N. (1997). Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In: Pazienza, Maria Teresa (Ed.): Information

extraction. A multidisciplinary approach to an emerging information technology. International summer school SCIE-97. Springer pp139-170.

Guarino, N. (1998). Some Ontological Principles for Designing Upper Level Lexical Resources. In: Proceedings of First International Conference on Language Resources and Evaluation, Granada, Spain.

Jackson B., Ceusters W. (2002). A novel approach to semantic indexing combining ontology-based semantic weights and in-document concept co-occurrences. In: Baud R, Ruch P. (Eds) EFMI Workshop on Natural Language Processing in Biomedical Applications, 8-9 March, 2002, Cyprus, pp75-80.

Jacobson, D. and Anagnostopoulos, A. (1996). Internet resources for transgenic or targeted mutation research. Trends Genet. 12, pp117-118.

Kashyap V., Ramakrishnan C., Rindflesch T.C. (2003). Towards (Semi-)automatic Generation of Bio-medical ontologies. In: AMIA 2003 Annual Symposium on Biomedical and Health Informatics.

Keen, G., Burton, J., Crowley, D., Dickinson, E., Espinosa-Lujan, A., Franks, E., Harger, C., Manning, M., March, S., McLeod, M., O'Neill, J., Power, A., Pumilia, M., Reinert, R., Rider, D., Rohrlich, J., Schwertfeger, J., Smyth, L., Thayer, N., Troup, C., Fields, C. (1996). The Genome Sequence DataBase (GSDB): meeting the challenge of genomic sequencing. Nucleic Acids Res. 24, pp13-16.

Köhler, J., Schulze-Kremer, S. (2002). The Semantic Metadatabase (SEMEDA): Ontology Based Integration of Federated Molecular Biological Data Sources. In Silico Biology 2, 0021.

Lindberg C. (1990). The Unified Medical Language System (UMLS) of the National Library of Medicine. J Am Med Rec Assoc. 1990 May;61(5) pp40-2.

Locke, J. (1975). An Essay Concerning Human Understanding (originally published 1690). P. H. Nidditch (Ed.), Oxford University Press, Oxford.

Maedche A., Staab S. (2003). Ontology Learning. In: S. Staab & R. Studer (Eds.) Handbook on Ontologies in Information Systems. Springer, Proc AMIA Symp. 2003, pp886.

Mahesh, K. and Nirenburg, S. (1995). A Situated Ontology for Practical NLP. In: Proc. Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95), Aug. 19-20, Montreal, Canada.

Masolo C., Borgo S., Gangemi A., Guarino N., Oltramari A. (2001). Ontology Infrastructure for the Semantic Web: WonderWeb Deliverable D18, IST Project 2001-33052 WonderWeb, Laboratory For Applied Ontology - ISTC-CNR, Via Solteri, 38, 38100 Trento, Italy http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf

McKusick, V.A. (1994). Mendelian Inheritance in Man. Catalogs of Human Genes and Genetic Disorders. Baltimore, MD: Johns Hopkins University Press, 11 edition.

Michael, J., Mejino, J.L.V., Rosse, C. (2001). The Role of Definitions in Biomedical Concept Representation. In: Proceedings, American Medical Informatics Association Fall Symposium, pp463-467.

Nirenburg S., Raskin V. (2001). Ontological semantics, formal ontology, and ambiguity. FOIS 2001 (Formal Ontology and Information Systems, N. Guarino, ed., ACM Press), pp151-161.

Rector A.L., Rogers J.E., Zanstra P.E., Van Der Haring E. (2003). OpenGALEN: Open Source Medical Terminology and Tools. Proc AMIA Symp. 2003, p982.

Rosse C., Mejino J.L.V. (2003). A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J Biomed Inform. vol 36, pp478-500.

Schulze-Kremer, S. (1997). Adding Semantics to Genome Databases: Towards an Ontology for Molecular Biology. Proc. Int. Conf. Intell. Syst. Mol. Biol. 5, pp272-275.

Schulze-Kremer, S. (1998). Ontologies for Molecular Biology. Pac. Symp. Biocomput. 3, pp693-704.

Smith, B., Rosse C. (2004). The Role of Foundational Relations in the Alignment of Biomedical Ontologies. In: Proceedings of MedInfo 2004, San Francisco.

Smith, B., Köhler J., Kumar, A. (2004). 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 (Lecture Notes in Computer Science 2994) Springer, Berlin, pp. 124-139.

The Joint United States / European Union ad hoc Agent Markup Language Committee (2001). http://www.daml.org/.