In Silico Biology 2 (2002) 179–193
IOS Press

Review

Ontologies for Molecular Biology and Bioinformatics

Steffen Schulze-Kremer

RZPD Deutsches Ressourcenzentrum für Genomforschung GmbH, Heubnerweg 6, D-14059 Berlin, Germany
E-mail: steffen@rzpd.de

Edited by E. Wingender; received and accepted 18 January 2002; published 18 March 2002

ABSTRACT : About five years ago, ontology was almost unknown in bioinformatics, even more so in molecular biology. Nowadays, many bioinformatics articles mention it in connection with text mining, data integration or as a metaphysical cure for problems in standardisation of nomenclature and other applications. This article attempts to give an account of what concept ontologies in the domain of biology and bioinformatics are; what they are not; how they can be constructed; how they can be used; and some fallacies and pitfalls creators and users should be aware of.

KEYWORDS: domain ontology, biology, bioinformatics, bio-ontologies, design, guidelines, semantics, philosophy

WHY ONTOLOGIES FOR MOLECULAR BIOLOGY?

There are a multitude of heterogeneous and autonomous data resources accessible over the Internet that cover genomic [1], cellular [2], structure [3], phenotype [4] and other types of biologically relevant information [5]. Even for one type of information, e.g. DNA sequence data, there exist several databases of different scope and organisation [1,6,7].

There exist terminological differences (synonyms, aliases, formulae), syntactic differences (file structure, separators, spelling) and semantic differences (intra- and interdisciplinary homonyms). Data integration is impeded by different meaning of identically named categories, overlapping meaning of different categories and conflicting meaning of different categories. Naming conventions of data objects, object identifier codes and record labels differ between databases and do not follow a unified scheme. Even the meaning of important high level concepts that are fundamental to molecular biology is ambiguous.

One prominent example is the concept *gene*. For GDB [1], a gene is a "DNA fragment that can be transcribed and translated into a protein". For Genbank [7] and GSDB [6], however, a gene is a "DNA

Electronic publication can be found in *In Silico Biol.* **2**, 020318 <http://www.bioinfo.de/isb/2002/02/0017> 18 March 2002.

region of biological interest with a name and that carries a genetic trait or phenotype" which includes nonstructural coding DNA regions like intron, promoter and enhancer. There is a clear semantic distinction between those two 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 with some ambiguous term until now she has full responsibility to verify the semantic congruence between what she asked for and what the database returned. Even if a semantic incompatibility is known it still must be sorted out for each search result. Ontologies could help here to localise the right type of concept to be searched for as opposed to identify a mere label naming a search table.

The advent of microarray technology for mRNA expression analysis requires additional standardisation in terminology, especially for characterising experimental setup, mathematical post-processing of raw measurements, genes, tissues and samples. A comparison between different experiments is only feasible if consistent terminology and standardised input forms are used. The development of suitable ontologies is currently persued in the MGED consortium [8].

Another reason demanding for standardised nomenclatures in biology is the merging of different subfields that historically started rather independently but now with a more integrated approach to biology must be closely integrated. This concerns e.g. genetics, protein chemistry, pharmacology. Since these areas have grown quite distinguished terminology especially large pharmaceutical companies feel an urgent need to harmonise the technical language to store their corporate knowledge in a central, unified database.

The fast growth of sequence, structure, expression, metabolic and regulatory data of 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. Where currently mostly statistical and proximity approaches are applied to text analysis ontologies can support parsing and disambiguating sentences by constraining grammatically compatible concepts.

To eliminate semantic confusion in molecular biology, it will be therefore necessary to have a list of the most important and frequently used concepts coherently defined so that e.g. database managers, curators and annotators could use such set of definitions either to create new software and database schemata, to provide an exact, semantic specification of the concepts used in an existing schema and to curate and annotate existing database entries consistently.

It is important to understand that semantic ambiguities also arise between human experts. However, in the course of a conversation usually enough background knowledge and context is available so that semantic ambiguities are most often faster resolved than even consciously recognised. This is possible because of our intelligent capabilities which computers, programs and databases, at least for the near future, fall yet short of.

OVERVIEW ON ONTOLOGIES

First, one should be aware of the distinction between ontology, the study of *being* as a branch of philosophy and individual (domain) ontologies, which are the result of the analysis of a particular domain of interest (possibly as broad as the universe) and the instantiation of a concrete ontological model of that domain. Such an individual ontology represents a system of categories accounting for a particular vision of the world (or parts of it).

Ontologies are to a large extent in principle language independent, e.g. there can be a German equivalent to an English domain ontology, even if the actual translation process would not be trivial since

subtle connotations of terms and definitions must be precisely understood and appropriately retold in terms of the other language.

Domain ontologies can be of varying scope and content. One can distinguish between:

- upper-level ontologies which are primarily concerned with general high level concepts that are the basis in our understanding of a particular domain;
- application ontologies, which are centred around an application domain; or
- task ontologies that are conceived for a specific problem solving task.

Since the world around us in general and molecular biology and bioinformatics in particular are in many aspects of enormous complexity it is important to well understand beforehand the intended use for a newly to be developed ontology. Otherwise there is a great risk of loosing focus and being overwhelmed by the multitude of facets leading to a failure of finishing a sufficiently complete, useful ontology. This aspect is acknowledged by the term "situated ontologies" [9] which emphasises the fact that a domain ontology should always be evaluated with respect to its intended use.

Certainly, ontologies cannot remain constant but will need to be updated in light of new experimental evidence, new focus of knowledge and shifting semantics in our language. The good news, however, is that an ontology is much more stable than e.g. a database schema, which depends on a database representation formalism, a database management system, requirements from the applications which access the data. Since an ontology can easily be translated from one knowledge representation formalism into another (given equivalent expressive capability) it can be also converted into a database schema. Since a domain ontology addresses primarily basic, fundamental underlying relations of an application domain there is less need to modify an ontology as compared to actual knowledge bases.

The main semantic stages in information retrieval in the past were:

- (approx. 1970-1980) mainly syntax based, e.g. string search in Medline
- (approx. 1980-2000) mainly structure based, e.g. html structure of a web document.

Nowadays concept based search on a curated set of concepts is becoming more common, e.g. Ontoligua or GeneOntology. The interplay between ontologies, biology, computer science and philosophy is depicted in Figure 1.

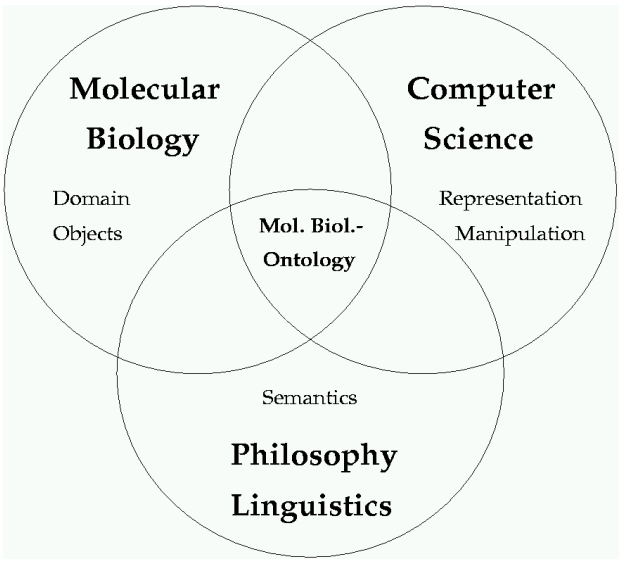

Fig. 1. Molecular biologists discover facts that need to be organised and stored in databases. Computer scientists provide techniques for data representation and manipulation. Philosophers and linguists help organise the meaning behind database labels.

Upper-level ontologies

Probably the first notable ontologist was Aristotle (384-322 BC) who among many other things pursued the question of what can be known about something - or even anything. His solution is presented in his "Categories" and can be seen as the first upper-level ontology.

- Substance
- Quantity
- Quality
- Relation
- Place
- Time
- Situation
- Condition
- Action
- Affection

In Aristotle's point of view these ten categories suffice to say anything that can be known about something. They present the essential qualities that matter. Everything else can be subsumed into one of those. Of course, for annotation of molecular biological entities this list of concepts seems too short and the concepts too general. However, if one subscribes to this set of categories as the essential fundamental ones one could continue and further subclassify these categories in more specific ones until they reach the realm of molecular biology.

Another design feature of Aristotle's ontology is the missing interconnection between his ten categories. If each of these is assumed to be an "atomic" category, i.e. it cannot be meaningfully decomposed into smaller concepts, then there cannot be much structure on top of them. However, if one might want to know more about how these ten categories basically relate to each other this information should also go into the ontology. Other ontologies try to be more explicit about the relations between their concepts (see below for examples).

```
Entity
    Particular                  (e.g. "large molecule", "green spot")
        Concrete particular
            Location
            Object
        Abstract particular
            Set
            Structure
    Universal                   (e.g. "largeness", "color")
        Property
            Property Kinds...
        Relation
```

N. Guarino offered this hierarchically composed version of an upper-level ontology. The hierarchical link between indented concepts means "is subclass of" [10]. This upper-level ontology is also rather small and stops well before biologically relevant concepts are reached.

Cyc

One of the first computational ontologies was Cyc [11]. Cyc is an ontology originally developed to cover everyday common-sense knowledge. A subset of about 6000 concepts is publicly available as HTML hypertext with ample documentation. Cyc was not built to support a specific application but with the intention to cover even subtle semantic distinctions that a person has to consider when communicating in daily life. The complete version of Cyc is commercially available. Cyc contains a large and detailed collection of well documented concepts but is of limited use for molecular biology for several reasons. Cyc does not include a significant portion of concepts relevant to molecular biology since it was designed to be a universal ontology and only very basic knowledge about chemistry and biology has been added.

Although the authors of Cyc state that they "generally only list a nonredundant series of supersets" or "the incommensurably most specific (i.e., smallest) supersets of each collection" this rule is violated on several occasions. For example, *Collection* has listed the supersets *Intangible*, *Thing* and *Set* of which *Thing* is a superset of *Intangible* which in turn is a superset of *Set*. There are also several cases where two concepts are listed to be the superset of each other, e.g. *Stuff* and *IndividualObject*. *Thing*, the "universal set of everything", has as its immediate subclasses *IndividualObject*, *Intangible* and *Role* of which all three are overlapping because there exist intangible *IndividualObject(s)* and a *Role* is something both individual and intangible (Figure 2). The definition of *Thing* as the set of everything also faces Russel's set dilemma.

Though most definitions in Cyc seem philosophically well established, what is visible to the public is counterintuitive in some places. For example, *Situation* is defined to be "a state of affairs" with superclass *IndividualObject* which is a "discrete, not abstract entity that can have parts but not elements or subsets", suggesting that not only objects involved in a *Situation* but also *Situation* itself is a tangible entity since no link to *Intangible* exists.

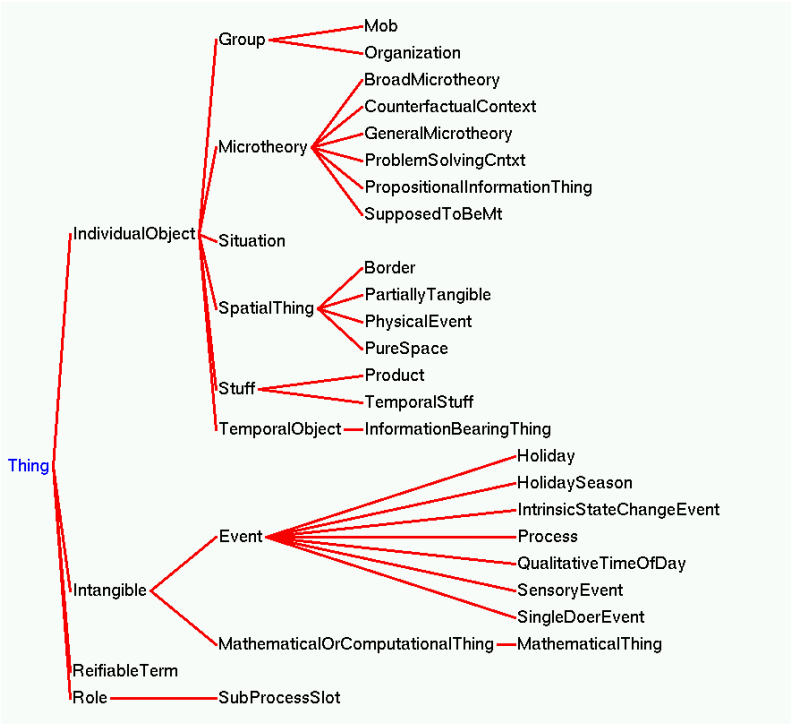

Fig. 2. Upper Level of Cyc Ontology. Straight lines indicate "is a subclass of" relation, arrows and italics denote "is a member of" relation (instances).

The concept *Stuff*, defined as a discrete object that "when divided into pieces remains of the same type" (e.g. water) includes "physical entities like wood", "temporal entities like the event of a person running" and abstract things like "a piece of English text". One problem with the definition of *Stuff* is its granularity: on a molecular scale wood can well be divided into components that no longer are wood. Similarly, English text can be divided into letters which are neither distinctively English nor text anymore.

The criterion used to subclassify a concept in Cyc is not always stated explicitly. In many cases, subclasses in one class overlap semantically or are created using different subclassifying criteria. No homonyms are found in Cyc. Naming of concepts is sometimes confusing, e.g. *Thing* vs. *SomethingExisting*; *PartiallyTangible* vs. *PartiallyIntangible*; *IntangibleObject* vs. *IntangibleStuff*. Cyc contains a hierarchy of classes containing only classes that in some cases mirrors a similar hierarchy of classes containing instances but which does not convey any new information. This adds to the confusion when searching for a concept. All these properties of the Cyc ontology make it difficult to locate the appropriate position for an existing concept or for a new one to be added.

MBO

Another philosophically motivated upper-level ontology is the author's [12]. Like Guarino's upper-level ontology it starts from a single node but also extends into physical and abstract concepts that are relevant for biology and bioinformatics.

The upper level of a prospective Ontology for Molecular Biology is shown in Figure 3. Starting from the root node *Being* which includes anything that *is*, the classes *Object* and *Event* are disjoint and discriminated based on their temporal extent. An *Object* remains an *Object* even in a single moment in time whereas an *Event* when dissected into single moments looses its identity. This holds also for all subclasses of *Object* and *Event*. The class *Object* is further subclassified into *Individual Object* and *Property*. Both can be thought of as instantaneous, i.e. they keep their identity even if looked at only for one moment. The two are discriminated based on self-contentment. An *Individual Object* can stand alone whereas a *Property* always needs another *Object* or *Event* to refer to. A *Property* is further subclassified based on arity into *Attribute*, a property with only one argument and *Relation*, a property relating two or more *Beings*.

Hereby, the logical grammar of words, not their surface structure must be considered. For example, in the statement "Paris is beautiful", beautiful is not a logical attribute to Paris because this statement necessarily involves a second entity, the speaker and thus becomes one binary and one unary relation: "She thinks, Paris is beautiful".

Attribute can be subclassified into *Identifier* and *Descriptor* based on whether it just labels an entity or whether it carries additional information about it. *Relation* can be subclassified analogous to Locke [13] into Secondary Property relations that involve personal judgement and Primary Property factuals describing intersubjective measurable relations.

Individual Object is subclassified based on physicality into Abstract Object, which has no physical equivalent per se (except capable of being represented neurologically or in writing, etc.) and Physical Object, which must have a defined spatial extension and/or energy content and is similar to Popper's "World 1" [14].

Abstract Object is further subclassified based on mentality, i.e. whether it refers to an object within the mind or to an object in the outside world, into *Mental Object* (similar to Popper's "World 2") and *Worldly Object* (similar to Popper's "World 3"). Although energy and matter are equivalent in nuclear physics a given object can be only of one type at a time. Hence, *Physical Object* has been subclassified based on mass content into *Energy* and *Matter*.

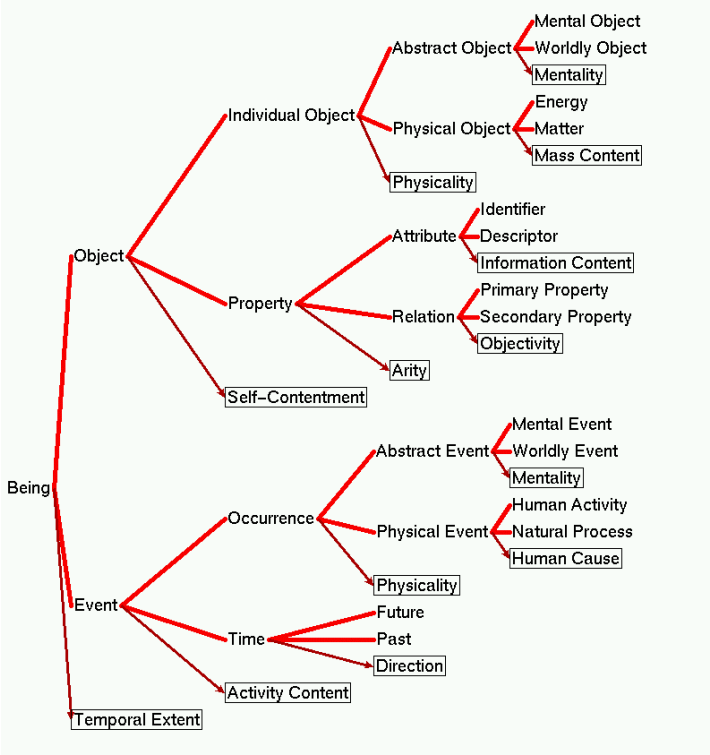

Fig. 3. Upper Level of a prospective Molecular Biology Ontology. Links represent the "is a subclass of" relation. No instances are present; discriminating criteria have arrows and boxes; thick lines denote disjunct subclasses.

On the other branch of the ontology *Event* is subclassified based on activity into *Occurrence*, where at least one object participates and (pure) *Time*, where nothing happens. This is the notion of absolute time which is no longer valid in relativistic physics and astronomy. The reason for nevertheless holding on to the belief of absolute time here is justified by the intended scope of the ontology for molecular biology: physical processes in living organisms have so far never been known to reach the realm of relativistic physics.

Time is further subclassified according direction into *Past* and *Future*. Because presence strictly lasts one moment only, it does not appear in this branch. Analogous to abstract and physical objects, *Occurrence* is subclassified based on physicality into *Abstract Event* and *Physical Event* and further *Abstract Event* based on mentality into *Mental Event* (similar to Popper's "World 2") and *Worldly Event* (similar to Popper's "World 3"). *Physical Event* is similar to Popper's "World 1" and subclassified based on whether it is done or initiated by human intention into *Human Activity* and *Natural Process*.

WHAT IS AN ONTOLOGY?

This section addresses domain or concept ontologies only. No statements should be applied to ontology as the branch of philosophy except where explicitly noted. Since there is no *a priori* definition of a domain ontology this section necessarily contains personal opinion but tries to give rational explanation whereever possible.

Here are three definitions of domain ontologies:

i "System of categories accounting for a particular vision of the world." [10]

ii "Specification of a conceptualization." [15]

iii "Concise and unambiguous description of principle relevant entities with their potential, valid relations to each other." [16]

Definition (i) is in the sprit of Aristotle's ontology and characterises well many ontological systems from philosophy but fails to impose any structure or form on them. Definition (ii) says, analyse your domain of interest, find out the basic concepts that are instantiated and specify them (but not the actual instantiations). Although this describes well in broad terms several main stages in ontology development, the definition itself is not self-explanatory.

For that reason, definition (iii) was conceived. It attempts to summarise definitions (i) and (ii) and to explain at least in some detail the scope of an ontology and a few constraints to be observed. Which are the principle relevant entities is determined by experts of the domain.

Requirements

Any domain ontology should meet the following requirements.

Each concept in the ontology should be defined as precisely as possible. Definitions are the basis for the relations between concepts, for semantic disambiguation and as such the foundation of an ontology and therefore indispensable. Although this sounds trivial and too obvious in everyday life documentation is often perceived as tedious, negligible extra work after the prototype is running. Not only that, writing good definitions is often very hard: how detailed has one to specify 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* rewriting and updating definitions is a frequent task and therefore should be adequately supported by any ontology editing software. In support of this requirement here is a brief recapitulation on definitions. There are nominal and ostensive definitions. Nominal definitions attempt to describe new concepts through a set of attributes. Of nominal definitions, there are analytic, explanatory definitions which decompose the concept to be defined into necessary and sufficient conditions, e.g. "bachelor is an unmarried adult". Further, there are synthetic, stipulative nominal definitions, which introduce a new concept, e.g. "α-helix is a polypeptide molecule of the following geometry ...". Ostensive definitions define new concepts by pointing to or enumerating a set of positive examples.

Common fallacies when defining concepts are the following:

- Definition made of only negations, e.g. "protein is not made of DNA". This leaves the definition still wide open.
- Definition too broad, e.g. "proteins are chemicals". Again, the definition is not precise enough.
- Definition too narrow, e.g. "proteins are covalent strings of amino acids". This definition misses post-translational modifications and quaterny structure.
- Using the term to be defined itself in the definition, e.g. "protein is made of protein chain".
- Verifying scope rather than defining content, e.g. "proteins can be enzymes".

The set of concepts covered by a domain ontology must include the great majority of relevant concepts of the application domain. Otherwise, a mixture of ontologically classified concepts and only vaguely defined concepts occurs with the result that computational inference stops at the undefined concepts.

There should be a formal notation of the structure of an ontology. This can include an ontology representation language or a specification of how concepts may be related.

Some documentation about extra-ontological commitments, i.e. design criteria that determine expressive capability, complexity or scope of an ontology should be given.

Standardised procedures how to add, modify or move concepts in the ontology should be defined.
Rules of how the ontology and their concepts can be used, e.g. what kind of inference is supported.

Components

The building blocks of an ontology are the following:

1. Concepts. Together with their definitions and predicates, concepts are the semantic "atoms" in an ontology.
   There are the following concept types:
   - Instances. They represent individual entities, e.g. the Eiffel tower in Paris, and are connected by the TYPE-OF relation to at least one class (see below).
   - Classes. These are generalisations of instances, e.g. *gene*, *protein*, and connected by the SUPERCLASS-OF and SUBCLASS-OF relations with each other. It is advisable to record them consistently in the singular form in an ontology.
   - Predicates. They should appear as classes themselves in the ontology so that they are properly defined. Predicates can be subdivided into attributes which are unary predicates and relations with arity of two or greater.
2. Propositions. These are definite statemants about (parts of) the world. They are used in the definitions and can be encoded in an ontology representation language.
3. Axioms. These statements are assumed to be true and cannot be proven from other first principles. The set of axioms must be logically consistent.
4. Knowledge representation formalism, e.g. first-order predicate logic with propositional logic, variables, functions and quantifiers. This is used to establish valid inferences among concepts and their predicates.

What is not an ontology?

An ontology is *not a collection of facts arising from a specific situation* but it provides all semantic entities (e.g. classes) to describe that situation. A concrete description of a situation uses those concepts to create instances and annotates them with their predicates.

An ontology is *not a model of an application domain* but a compendium of all building blocks with their valid modes of combination required to express a theory. An entire model of an application domain (e.g. enzyme chemistry) would be a set of (possibly verified) hypotheses or a theory.

An ontology is *not a database schema* which describes the categories and their data types and organisation in a database but not necessarily the relations between the actual entities in the real world they stand for. A database schema can be derived from an ontology by adding data type information and translating the knowledge representation formalism into a database management paradigm (e.g. relational). *Vice versa*, a database schema can be used as a starting point to create an ontology. The categories and their attributes can be taken as an initial set of concepts to populate an ontology.

An ontology is *not a knowledge base* which gathers knowledge about actual individual objects, events, situations, experiments etc but it holds a collection of the types of objects, events etc used to specify those objects in an actual situation. Alternatively, one could say an ontology is a particular knowledge base filled with knowledge about concepts and their ontological relations.

An ontology is *not a taxonomy* which knows only about superclass and subclass relations whereas an ontology is open to many types of relations between concepts (e.g. mereological, topological, compositional).

An ontology is *not a vocabulary or dictionary* since the words in a dictionary do not necessarily describe the hierarchy and relation between each and every concept and are not organised in a way that

supports computational inference. In an ontology one can follow a path from any one concept to another along the edges of some IS-A hierarchy or other relations.

An ontology is *not a semantic net* which is a more general representation formalism that can be used to implement an ontology but is not the only choice for that.

As an example for ontological distinctions consider the following. When we say "DNA" we can actually mean several quite different entities. First, there is the actual substance, which is physical and can drop on your foot. Second, DNA can refer to a particular class of chemical substance, which includes general features common to all DNA molecules and is used e.g. in in molecular modelling. Third, DNA can mean a certain type of sequence or string which is an abstract mathematical concept, can be subject of certain mathematical operations but cannot drop on your foot. Forth, 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. There are probably other connotations to DNA in everyday life than listed here.

HOW TO BUILD AN ONTOLOGY?

Due to various notions and uses of ontologies there are several ways of how to build an ontology (e.g. stage-based [17], iterative evolving prototypes [18]). In the following, the one of [16] is described.

Given the components described above (set of concepts, propositions about concepts, axioms, knowledge representation formalism, "is a subset" relation, "is a member" relation), apply the following steps to each concept:

- Find and write out a unique and explicit definition for each concept. This definition must be precise enough to discriminate that concept from all other concepts in the ontology and should be detailed enough to provide a clear understanding of its meaning. Although experts usually have a good understanding of their technical terms finding a proper definition can become very difficult due to overlapping meanings hidden in one word (e.g. *gene*, see above) which must be disambiguated. The ontology management software should therefore be also capable of handling homonyms.
- When adding subclasses to a class concept decide and use consistently and explicitly one (and no more) discriminating criterion for each superclass. When this design principle is followed the ontology also includes a hierarchical tree of subclasses that can be used as a decision tree when adding or searching for concepts. This will require various inheritance modes to choose from (e.g. multiple distinct inheritance or combined cumulative inheritance).
- Be explicit about the disjointness of subclasses, i.e. state where subclasses of a single class concept can overlap or not. This greatly helps later on when searching through the subclass hierarchy by focusing the search.
- Obtain complete connectivity via "is a subclass" or "is a member" relations (or their inverses) from any one concept to the rest of the ontology, i.e. at leat one "is a subclass" or "is a member" relation (or inverses) must exist for each concept. This way all concepts are defined consistently with reference to the same ontology whereas separate ontological islands could give rise to conflicting or overlapping conceptualisations which later on might require ontology integration.
- Use one root node concept only. This concept can be chosen general enough as to embrace the variety of domain relevant concepts. Otherwise different conflicting lineages could emerge.
- Add background knowledge for each concept to express domain-relevant properties. The attributes and relations should themselves be reified first in the ontology for maximal inference capability. Annotate concepts with aforementioned attributes.

- Add links from concepts in the ontology to natural language dictionaries, database keywords etc thereby interfacing the ontology with an application.

Concept naming guidelines

The following rules make an ontology more readable:

1) use singular form in a concept name
2) use lower case letters for classes
   - instances and names should begin with capital letter
   - acronyms should be all upper case
3) observe syntax requirements of selected representation formalism
   - quotes, hyphens etc may be required or forbidden
   - unique names may be required by representation formalism
4) if there is a good English word, use it
   - otherwise concatenate not more than four words to describe the concept
5) when naming a subclass specialise superclass concept name
   - specialising text should be appended, not prepended
   - makes concept easier to recognize
6) add subclassifying criterion immediately when obvious
7) always provide aliases where known

The benefits of this methodology are significant. When adding a new concept 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 the new concept should belong to. Either one finds the concepts is already there (possibly under another name). Then the insertion process is merely adding another alias to the existing concept. Or, one ends at some point in the hierarchy where no alternative seems appropriate anymore. This is then the place where the new concept should be added, either directly or using some intermediary concepts to separate the exiting concepts from the new branch. This also guarantees consistency of the existing ontology and generalisation/specialisation hierarchy after inserting a new concept. Searching for a known or even unknown concept can be done in the same way, i.e. by traversing the decision tree of discriminating criteria.

Ontological commitment

Ontological commitment refers to the choice of axioms for an ontology, i.e. the background belief which is not explicit in an ontology; the choice of granularity in the selection of concepts and definitions (coarse abstractions vs. finer details); and the choice of subclassifying criterion (content; priority). All these decisions influence the final appearance of an ontology and should at least be stated explicitly.

Difficulties

There are several difficulties to be overcome when building an ontology. Some difficulties are inherent to the ontology building process, others arise mainly from the application area at hand.

Since there is no definite rule to determine the "best" (e.g. most informative) subclassifying criterion for a given class one is left with a necessarily arbitrary decision on how to subclassify that class. This implies there will not be an optimal nor best ontology for a given set of concepts but only (in)consistent and (un)useful ontologies. Also, since the information content of the concepts that still need to be added

to an ontology is not precisely known in advance the choice of subclassifying criteria can lead to more complex inheritance structure than necessary.

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

1) Missing ontological elements
   - missing classes
   - missing attributes/relations
2) Confusing arity of relations
   - 1:1 vs. 1:Many
   - 1:Many vs. Many:Many
3) Over-elaborating
   - superfluous ontological elements
   - are all details relevant?
4) Storing important data as free text or comment fields rather than in reified predicates.

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

In toto, this leads to the conclusion that one main degree of freedom when building an ontology, i.e. the degree of abstraction, granularity and detail of the domain to be modelled determines the practical quality of an ontology in a range from useless (too abstract, does not give sufficiently detailed information) to impossible (ultimate granularity and coverage).

ONTOLOGY INTEGRATION

The feasibility and desirability of one comprehensive ontology for molecular biology versus several smaller task oriented ontologies has been extensively debated in the community. On the one hand one comprehensive domain ontology would certainly be very helpful if it could be achieved and maintained. On the other hand, it seemed much more efficient and effective to have several smaller task or subdomain ontologies which take less time and expertise to grow and maintain and therefore are in the position to be put to use much sooner.

In principle, the approach of smaller subdomain ontologies is the more practical one with the exception of a situation where eventually the goal is to combine all subdomain ontologies. In that case, much work will have to be redone since the integration of ontologies as described above can hardly be automated. Each concept must be located and identified in the various subdomain ontologies which involves manual search, reading and comparing concept definitions. A decision must be made whether the concepts are similar enough to be merged into one or if several similar concepts need to be saved. Then, these concept(s) must be added to a new ontology that will incorporate all subdomain ontology concepts.

In the special case where the root or some top-level concepts of one ontology exactly match concepts in another ontology these branches could be merged. However, in this case the data format (syntax, representation formalism) and the relations between concepts of the two ontologies still need to be checked and verified.

Since this process of ontology integration is quite laboursome 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 [12].

APPLICATIONS OF BIO-ONTOLOGIES

Ontologies can provide to computer programs much of the common sense and background knowledge that human experts use. Therefore, their range of applicability is rather broad as was indicated already in the introductory paragraphs of this document. Two examples, database integration and data annotation will be discussed here briefly.

Data integration faces the problems of syntactic and semantic heterogeneity. While regular syntactic incompatibilities can be easily aligned with pattern matching software, semantic heterogeneity needs a unified semantic repository to be resolved. For the case of *n* databases in the traditional way each table, object etc of one database has to be manually aligned with the structure and contents of every other database. Since because of different meanings in one word the mapping between database tables and attributes can be non-symmetrical this actually amounts to *n*n* integration attempts. However, if one ontology exists that can be placed "in the middle" of those *n* databases the integration effort is reduced from *n*n* to *n* only, since each database has only to be mapped to the ontology where general inference algorithms can figure out identical or similar concept in any other database [19].

For data annotation, in principle not a full fledged ontology as described above is required but only a controlled vocabulary since the main purpose is to provide constant and unique reference points. Such a controlled vocabulary is developed in the GeneOntology (GO) project [20]. GeneOntology attempts to provide continuity in 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.

However, the design principles of GO did not prevent the following shortcomings.

- The main relations (ISA and PARTOF) are used not always consistently. For example, ISA can mean "subclass of" or "instance of", i.e. there is no distinction between generic and individual entities in GO which clearly restricts its expressive capability. Similarly, PARTOF is found in places with the following meanings: "made of", "belongs to", "physical part of", "conceptual part of", "subprocess of", "controls", "causes", "activates", "inhibits", "enclosed by" and "binds to".
- GO is not very helpful as long as many concepts are lacking explicit definitions, currently about 80% of them.
- Further, GO has about 700 concepts that do not have a parent concept. Strictly speaking this says that there are about 700 independent subdomain ontologies. This is certainly not desirable since it gives away the advantage of knowing the relation of one concept in the same ontology to any other concept (but not in another ontology). Thus, the current concept collection is highly gapped.
- There are no clear design principles given for GO. The way of how a concept finds its way into GO is not well defined. Therefore it is near impossible to understand the reason why GO is as it is today, why a certain concept was placed into a particular class etc. If somebody new to GO wants to get an understanding of it there is no other way than (re-) examining each and every branch and concept individually and trying to figure out whether the assertion is acceptable or not.
- There are no integrity constraints that would guarantee the consistency and correctness of GO after adding another concept.
- The question of where to put a new concept is not answered easily by GO. It seems that this is currently done mainly by intuition. Since no subclassifiying criteria are given there is little guidance from within GO.
- No grammar or rules exist with GO that explain how to relate or use concepts in combination.
- GO is actually intended to be not one but three ontologies. Leaving the large set of parentless concepts aside (see above) there are three main root nodes. This has the disadvantage that concepts within those three hierarchies are not linked with each other and appear unrelated within GO.

All in all, GO seems currently to be more a nomenclature or controlled vocabulary for molecular biology rather than a full fledged gene ontology.

RESOURCES ON (BIO-) ONTOLOGIES

Finally, here some resources are listed that could be 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 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.
- OntoWeb is a European funded network on ontology-based information exchange for knowledge management and electronic commerce at http://www.ontoweb.org.
- 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 Workshop's webpage is at http://img.cs.man.ac.uk*/* stevens/ workshop01.
- Cycorp has ist 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.
- Ontologies for eCommerce can be found at http://www.ontology.org.

REFERENCES

[1] 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**, 57-63.

[2] Jacobson, D. and Anagnostopoulos, A. (1996). Internet resources for transgenic or targeted mutation research. Trends Genet. **12**, 117-118.

[3] 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**, 535-542.

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

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

[6] 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. and Fields, C. (1996). The Genome Sequence DataBase (GSDB): meeting the challenge of genomic sequencing. Nucleic Acids Res. **24**, 13-16.

[7] Benson, D. A., Boguski, M. S., Lipman, D. J. and Ostell, J. (1997). GenBank. Nucleic Acids Res. **25**, 1-6.

[8] 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**, 365-371.

[9] 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.

[10] 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.

[11] Lenat, D. B. (1995). Cyc: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM **38**, 33-48.

[12] Schulze-Kremer, S. (1997). Adding Semantics to Genome Databases: Towards an Ontology for Molecular Biology. Proc. Int. Conf. Intell. Syst. Mol. Biol. **5**, 272-275.

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

[14] Popper, K. R. and Eccles, J. C. (1985). The Self and Its Brain. 3. ed., Springer, Berlin.

[15] Gruber, T. R. (1993). Knowledge Acquisition **5**, 199-220.

[16] Schulze-Kremer, S. (1998). Ontologies for Molecular Biology. Pac. Symp. Biocomput. **3**, 693-704.

[17] Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. Knowledge Engineering Review, **11**(2).

[18] 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, pages 33-40, March 1997.

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

[20] 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.