NOTE TO USERS

Design and Development of an Integrated Formal Ontology for Fungal Genomics

Arash Shaban-Nejad

A Thesis
In
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

March 2005

Your file Votre référence
ISBN: 0-494-04449-7
Our file Notre référence
ISBN: 0-494-04449-7

ABSTRACT

Design and Development of an Integrated Formal Ontology for Fungal Genomics

Arash Shaban-Nejad

With the substantial increase in stored scientific data of various types a major challenge of the post-genomic era is to access the knowledge stored in a myriad of complex databases and other resources across the web.

Ontologies can play an important role in bioinformatics, as they do in other disciplines, where they will provide a shared source of precisely defined terms that can be communicated across people and applications. The Ontology Web Language (OWL) is an ontology language that has an easy to use frame feel, yet at the same time allows users to exploit the full power of an expressive Description Logic (DL).

The thesis presents a formal integrated bio-ontology design and implementation case study in the area of fungal genomics to provide simplified access to units of intersecting information from different biological databases and existing bio-ontologies. We demonstrate the capacity of the ontological conceptualization through a series of industry related queries.

Using OWL-DL highlights the features of the combination of a frame representation of OWL framework and expressive Description Logics. We also used Racer as DL reasoner to build and maintain sharable ontologies by revealing inconsistencies, hidden dependencies, redundancies and misclassifications.

Acknowledgements

I would like to express my heartfelt appreciation to my supervisor, Dr. Volker Haarslev for his supports, insights, reviews and professional advice. Without his help this work would not be possible.

A Special thanks goes out to Dr. Christopher J.O Baker, for his invaluable advice on the different parts of this thesis. He really helped me to cope with the challenges in the area of biology and bioinformatics.

I would like to express my gratitude for all my colleagues in the FungalWeb Project for their comments and discussions.

Also, thanks for the generosity of Genome Quebec in providing funding for the FungalWeb project: "Ontology, the Semantic Web and Intelligent Systems for Genomics".

Finally, I would like to express my deepest gratitude for the constant support, understanding and love that I received from my wife Shideh and my parents also my sisters and my brother during the past years.

TABLE OF CONTENTS

List of Figures

List of Tables

1. Introduction

Fungi lead a special way of life that includes spore formation and the efficient secretion of extra cellular enzymes and gene regulation. Being biochemically versatile, fungi produce a wide array of acids and degradative enzymes to support their absorptive life style, as well as an astonishing array of low molecular weight primary and secondary metabolites. Many of these metabolites have industrial and pharmaceutical applications.

Our first objective is to provide approaches and theory coupled with a flexible software platform like Protege [77] to build a formal ontology for fungal genomics to support fungi species and enzyme interactions.

Achieving simplified semantic access to units of intersecting information from different databases is the motivation of this study. To this end, ontologies written in the Ontology Web Language (OWL), representing fungal taxonomy (NCBI [12] / NEWT [13]) and enzyme attributes from BRENDA [14] are mapped to establish a knowledgebase of use to enzyme application scientists working in the field of fungal genomics.

Querying of the knowledgebase to identify instances of bio-scientific literature reporting industrially relevant enzymes produced by specific fungal taxonomic groups is described. Physio-chemical and catalytic properties of different fungal enzymes in the context of the fungal host are investigated. Enzyme substrates are described in the context of the chemical dictionary of small molecular entities (ChEBI). The new Racer Query Language (nRQL) is used for defining instance retrieval queries using DLs.

1.1 Research question

The major research question in this thesis is:

"Which mechanisms and methods can be used to build an integrated ontology in the domain of fungal genomics to provide access to information distributed in different databases, knowledgebases and other existing ontologies?"

This general question can be detailed into four smaller questions:

1. What are the specific characteristics of information in available knowledge sources?
2. How one conceptualizes the information within the ontology?
3. How we can integrate the knowledge?
4. What methods can be used for query-answering through and across the knowledgebase?

We try to explain how the state-of-the-art research and development in the Semantic Web and bioinformatics can help addressing these issues.

Because of the interdisciplinary nature of the subject only few people can claim a strong background on both sides of computer science and biology. Lack of familiarity with the intellectual questions that motivate each side can also lead to misunderstandings [2].

1.2 Approach

By analyzing the context of the problem and reviewing other existing bio-ontologies (ontologies in the area of bioinformatics) in the related areas, we try to create an integrated bio-ontology in the domain of fungal genomics. Based on this, we introduce a framework which can be used to explore some techniques to solve some of the problems in the different stages of ontologies development lifecycle.

However, the nature of the Semantic Web can cause difficulties in the evaluation of these techniques in the real world. For example, some techniques, tools and applications are available for ontology merging, but the mass of structured data and other problems can reduce the usability of these tools dramatically.

To provide some evidence of the usability of our framework, we try to apply some techniques manually to show the technical correctness and feasibility of the approach.

We used some of the developed tools and techniques in the research area. The practical studies consist of case studies in the FungalWeb Project [19].

1.3 Contributions and outline

The main contribution of this thesis is a better understanding of the problem of ontology development and integration from distributed databases and existing ontologies in the area of fungal genomics.

In the first step of building an ontology, which is called "knowledge acquisition" part, we introduce the extraction of terms and concepts from different services and distributed data sources like Entrez NCBI [12], NEWT [13], BRENDA [14] and SwissProt [15], and their roles to create an integrated ontology in the domain of fungal genomics.

In this stage we deal with data extraction and semantic interoperability problems.

The implementation part is divided into 3 stages: conceptualization, integration and encoding. We try to build an integrated ontology by reusing and merging some independent bio-ontologies like Gene Ontology [69] and TAMBIS [29]. We are still searching for better practical approaches for ontology mapping, merging and alignment.

For the encoding part, Protege with OWL support is being used. Also we used OWL-DL as the ontology language.

Racer [57] is being used as a Description Logic [127] reasoning system for the evaluation stage and asking queries.

WonderWeb OWL Ontology Validator [175] is being used, to validate the ontology by returning a description of the classes, properties and individuals in the ontology in terms of the OWL Abstract Syntax.

Parts of this thesis have been published elsewhere [164, 165, 166, 167].

2. Knowledge Representation and the Semantic Web

2.1. Introduction to the Semantic Web

Since deployment of the World Wide Web (WWW) and its advance to becoming a core part of the daily lives of many people around the world, the way in which information is transmitted, stored, and accessed has been revolutionized.

The primary idea behind the Semantic Web is having data on the web defined and linked in a way, that it can be used by machines not only just for display purposes, but also for using it in various applications [9]. One can think of it as being an efficient way of representing data on the Web, or as a globally linked database [10].

Currently the Semantic Web is an initiative by the W3C [24] to move the Web from being only human understandable, to being both human and machine understandable [25].

The current problem with the data on the Web is that it is difficult to use on a large scale, because there is no global system for publishing data in such a way that it can be easily processed by anyone. For instance the information about weather, protein sequences and genetic data all are presented by numerous sites, but all in HTML (HyperText Markup Language), which has some limitations.

The Semantic Web is generally built on syntaxes which use URIs (Uniform Resource Identifier) to represent data, usually in triple based structures: for example, many triples of URI data that can be held in databases, or interchanged on the web using a set of

particular syntaxes developed especially for the task. These syntaxes are called "Resource Description Framework" (RDF) syntaxes [10].

The information on the Web is understandable for people, and can be manipulated in different ways, to aid human understanding. For a machine, though, the material from web pages and articles from the web, for instance a protein sequence or a restaurant menu, is represented in the same way as a sequence of symbols, an effectively meaningless stream and more importantly difficult to manipulate automatically [26].

Computer programs can search through myriad of texts to find a given phrase, but they cannot easily "understand" input data, by attaching a meaning to a word.

The Semantic Web is a movement towards this machine understanding of semantic concepts inherent in any given human language. By using a mark-up language, such as XMLS and more recently OWL (Ontology Web Language), content of web pages can be 'tagged' with semantic markers, so a meaning can be assigned to these phrases [27]. In order to enable computer programs to perform such complex tasks, the information on web pages should be processable and interpretable by computers. Ontologies can be used to make web content and services interpretable and understandable by computers. As an ontology is domain knowledge captured in a form understandable both by humans and computers, the knowledge on the Web can be made computationally accessible [11].

2.2. Applying the Semantic Web for biology

The goal of the Semantic Web is to extend the existing web with conceptual metadata that are more useful to machines, revealing the intended meaning of web resources [176].

Bioinformatics is already known as an important research area for the Semantic Web. Bioinformatics resources are rich in data and knowledge, but most of that knowledge is in

the form of natural language and image annotation which need to be processed by computers. Due to growing biological annotated data, the need to make knowledge accessible by computers also increases. Ontologies can play a critical role to create a formal specification of biological knowledge.

Currently, there are some ontologies in the area of bioinformatics (usually called bio-ontologies). These ontologies can help one to make bioinformatics knowledge computationally accessible and semantically understandable for human and computers.

2.3. Knowledge Representation (KR)

The field of knowledge representation (KR) has long been a focal point of research in the Artificial Intelligence community [37]. Knowledge representation is a multidisciplinary subject that applies theories and techniques from three other fields: Logic, Ontology and Computation [3].

In fact KR is the application of logic and ontology to the task of constructing computable models for some domain.

Logic provides the formal structure and rules of inference. Without logic, a knowledge representation is not clear in meaning or intention, with no criteria for determining whether statements are redundant or contradictory.

An ontology defines the kinds of things that exist in an application domain. Without an ontology, the terms and symbols are undefined, confused, and confusing.

Computation supports the applications that distinguish knowledge representation from pure philosophy. Without computable models, the logic and ontology cannot be implemented in computer programs [3].

2.4. KR and Biology Interaction

Today, biology and knowledge representation have interactions in many areas. Biology defines an area of interest and requirements for KR. Biology also provides document support, auditing and checking the validity of reasoning from a biologist perspective. Knowledge representation facilitates to handle very large biological knowledgebases, control large numbers of instances, explain and annotate biological data, and prepare formal reasoning services.

One of the areas which can be used for collaboration between biologist and KR people is ontologies. Ontologies are produced by members from various genomics and life-science efforts. There are many projects for presentation of formal ontologies for describing the content of bioinformatics resources and services accessible on the Web.

2.5. Bioinformatics

Bioinformatics is a discipline that uses computational and mathematical techniques to store, manage and analyze biological data, in order to answer and explore biological questions [1].

The biological sciences, especially molecular biology, currently lack the laws and mathematical support of sciences such as physics and chemistry [16].

It has been said that biology is a knowledge-based discipline. Much of the biology knowledge is contained within the biology data resources. A typical resource is the SWISS-PROT protein database [15]. The protein sequence data is a small part of an entry and most of an entry is taken up by what is called `annotation' which describes: "physico-chemical features of the protein; comments on the whole sequence, such as function, disease, regulation, expression; species; names and so on [1]".

This knowledge is captured as textual terms describing the findings, not numeric data, making use of shared keywords and controlled vocabularies. However this format is human readable, but it is hardly machine understandable.

2.6. Knowledge Representation and Bioinformatics

As already mentioned, bioinformatics is an emerging field that seeks to integrate computer science with applications derived from molecular biology [1].

Bioinformatics talks about different problems in the biological database development such as integration and optimally querying data such as instance genomic DNA sequence, spatial and temporal patterns of mRNA expression, protein structure, immunological reactivity, clinical outcomes, publication records, and other sources [1].

There are some issues in bioinformatics which motivate us to use knowledge representation methods in this area. Some of the issues are:

- **Huge amount of data:** Genomic research can help us to work with biology on a scale not previously possible: all genes in a genome, all transcripts in a cell and all metabolic processes in a tissue. All approaches share the production of massive quantities of data [1]. Also gene expression patterns, protein structure, protein-protein interactions provide even more data. So we need some formal methods to deal with these data, annotate and process them to be used by biologists.
- **Complexity of data**: Biological data are complex in the type of data stored and in the richness and constraints working upon relationships between those data [29].
- **Distribution of data**: Bioinformatics is an inherently integrative discipline, requiring access to data from a wide range of sources and the ability to combine

these data in new and interesting ways [7]. More than 500 data resources and analysis tools are used in bioinformatics [20].

- **Volatility of data**: biological data is not static. As knowledge about biological entities changes and increases, so their annotations of data resources will be changed [1].
- **Heterogeneity of data**: Most knowledge and data in the area of biology are both syntactically and semantically heterogeneous [23]. Individual concepts, such as gene, have many different, but equally valid, interpretations.

These issues cause great difficulties for both curators of bioinformatics resources and their users. Some of the difficulties are: knowing which resources to use in a task; discovering instances of those resources; knowing how to use each of those resources, and how to link their content and transferring data between resources [1]. So, computational support is required for storing, exploring, representing and exploiting biological knowledge as well as knowledge in the minds of domain experts.

3. Ontology

3.1. What is Ontology?

One way of capturing knowledge within bioinformatics applications and databases is the use of ontologies to define a concrete form of conceptualizations of a community's knowledge of a domain [4]. Knowledge in ontologies can be captured and made available to both machines and humans.

Traditional ontology definition is `the specification of conceptualizations [30], used to help programs and humans share knowledge' [31]. The conceptualization is the capturing of knowledge about the world in terms of entities (things, the relationships they hold and the constraints between them). The specification is the representation of this conceptualization in a concrete form [4]. One step in this specification is the encoding of the conceptualization in a knowledge representation language.

Ontologies are being used in a wide range of application scenarios [32]:

- A community reference: Its benefits include reusing knowledge, improving maintainability and long term knowledge retention.
- Either defining database schema or defining a common vocabulary for database annotation. Benefits include documentation, maintenance, reliability and sharing.
- Providing common access to information.
- Ontology-based search by forming queries over databases.
- Understanding database annotation and technical literature. Some ontologies are designed to support natural language processing (NLP) that not only link domain

knowledge but also how knowledge is related to linguistic structures such as grammar and lexicons.

The main components of ontologies are concepts, relations, instances and axioms.

Concepts represent a set or class of entities within a domain. Relations describe the interactions between concepts or a concept's properties. Instances are the `things' represented by a concept. It is not necessary for an ontology to have instances, because an ontology is supposed to be a conceptualization of the domain. The combination of an ontology with associated instances is what is known as a knowledgebase. Axioms are being used to constrain values for classes or instances.

A common ideal for an ontology is that it should be reusable [1]. This ambition distinguishes an ontology from a database schema, even though both are conceptualizations. Usually a database schema is intended to satisfy only one application, but an ontology could be reused in many applications. However, an ontology is only reusable when it is to be used for the same purpose for which it was developed. Not all ontologies have the same intended purpose and may have parts that are reusable and other parts that are not. They will also vary in their coverage and level of detail.

3.2. Building Ontologies

The process of building ontologies is a high-cost process. Some people believe that the construction of ontologies is an art rather than a science. The goal for building ontologies is to create an agreed-upon vocabulary and a semantic structure for exchanging information about that domain [33].

There are no standard methodologies for building ontologies. Such a methodology would include a set of stages that occur when building ontologies, guidelines and principles to

assist in the different stages, and an ontology life-cycle which indicates the relationships among stages [34]. The most well known ontology construction guidelines were developed by Gruber [31], and recently, there has been increased effort in trying to develop various ontology methodologies [35].

Methodologies for ontology development can be divided into those that are phase based like TOVE [21] and those that rely on iteratively evolving prototypes like Methontology [28]. Scientists usually distinguish between formal and informal techniques for ontology development. They vary between informal methods, where the ontology is sketched out using either natural language descriptions or some diagram techniques like UML, and formal methods where the ontology is encoded in a formal knowledge representation language such as OWL, which is machine computable.

As one can see in Figure 3-1 there are common phases in most ontology life cycles:

- **Specification:** Identify purpose, scope and granularities. This phase is important for design, evaluation and reuse of ontologies.
- **Knowledge Acquisition:** the process of acquiring domain knowledge from specialists (in our domain biologists); database metadata; standard text books; research papers and other ontologies.
- **Conceptualization:** identifying the key concepts that exist in the domain, their properties and the relationships that hold between them; identifying natural language terms to refer to such concepts, relations and attributes; and structuring domain knowledge into explicit conceptual models.
- **Integration:** use or combine available data from existing databases and ontologies to obtain a consistent ontology.

- **Encoding:** representing the conceptualization in some formal language, for example frames, object models or logic.
- **Evaluation:** by assessing the competency of the ontology to satisfy the requirements of its application, including determining the consistency, completeness and conciseness of an ontology [36]. We evaluate ontologies for completeness, consistence and avoidance of redundancy.
- **Documentation:** an ontology that can not be understood can not be reused. Informal and formal complete definitions, assumptions and examples are essential to promote the appropriate use and reuse of an ontology.

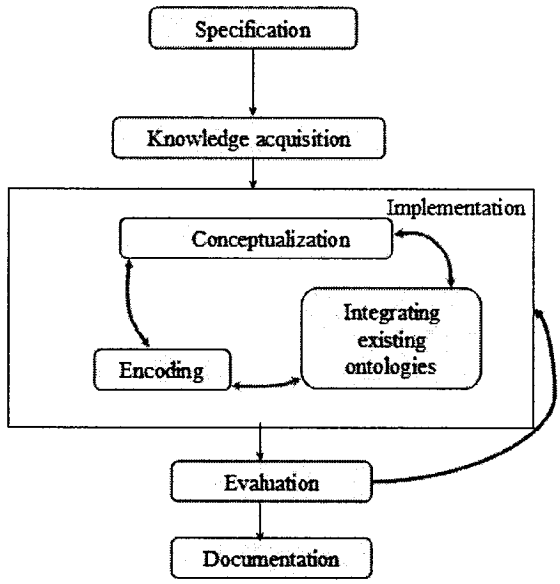


Figure 3-1 : The Ontology development life cycle [33].

3.3. Knowledge representation languages

For ontologies to be used within an application, the ontology must be specified and encoded, that is, delivered using some concrete representation. There are a variety of languages which can be used for representation of conceptual models, with varying characteristics in terms of their expressiveness, ease of use and computational

complexity. We usually choose a language based on expressivity of the encoding language, the rigour of an encoding and the semantics of a language.

The expressivity of a language is a measure of the range of constructs that can be used to formally, flexibly, explicitly and accurately describe the components of an ontology.

The rigour of an encoding is a measure of the satisfiability and consistency of the representation within the ontology.

The semantics of a language refers to the fact that it is unambiguously what the language means. i.e. when we say `A sub-concept-of B' does this mean that all the instances of A are also instances of B, or parts of B, or special kinds of B? Just because two languages use the same syntax does not mean they intend the same meaning. Especially when we want to exchange information in bioinformatics we need clearly defined and well-understood semantics for our ontology [35].

Currently there are three kinds of languages which are being used for encoding bio-ontologies: vocabularies defined using natural language; object-based knowledge representation languages such as frames and UML, and languages based on predicates expressed in logic such as Description Logics [4]. In table 1.2 one can see some of the KR languages.

	KIF/OKBC/ CG/Cycl	UML	TopicMaps /XTM	RDF(S)	DAML + OIL	OWL
Description	Legacy KR Languages	Universal Modeling Language	Topic Maps /XML Topic Maps	Resource Discription Framework	DARPA ML + Ontology Inference	Web Ontology Language
Governance	ANSI	OMG	ISO	W3C	DARPA	W3C
Years since proposed	>6	>6	>6	>4	>3	>3
Open source support	Yes	Yes	Yes	Yes	Yes	Coming

Table 3-1: Knowledge Representation Languages [168].

Vocabulary based languages: support the creation of purely hand-crafted ontologies with simple tree-like inheritance structures. The Gene Ontology has a hierarchical structure. The position of each concept and its relation with others GO are completely specified by the ontologist. However this provides more flexibility, but the lack of any structure in the representation can cause difficulties with maintenance or preserving consistency, and there are usually no formally defined semantics.

Frame-based languages: provide greater structure. Frame-based systems are based around the notion of frames or classes which represent collections of instances. Each frame has an associated collection of slots or attributes which can be filled by values or other frames. As well as frames representing concepts, a frame-based representation may also contain instance frames, which represent particular instances.

Frame-based systems have been used extensively in the KR community, particularly for applications in natural language processing. Both EcoCyc [180] and RiboWeb [181] use a frame representation. Frame-based design is similar to object-oriented design and is intuitive for many users.

In frame-based languages, it is not always clear how to interpret an assertion that a slot is filled with a particular value. All instances of the frame must have this particular attribute taking this value? Or the value represents possible fillers for the slot for each instance? For instance, we might want to say that the frame `fungi` has a slot saying `all fungi must have a name', but it is only a possibility that fungi `have an industrial usage'.

Logic based languages: for solving unclear semantic problem in frames one can use logic, notably Description Logics (DLs) [127]. DLs describe knowledge in terms of concepts and relations that are used to automatically derive classification taxonomies. A

major characteristic of a DL is that concepts are defined in terms of descriptions using other roles and concepts. For instance, in the TAMBIS ontology, the concept `Enzyme' was not simply asserted by the ontologist. Instead, a composite concept was made from `Protein' and `Reaction', joined with the relation `catalyses' - to make the concept Protein which catalyses Reaction. Thus someone viewing the ontology can see a definition for the concept Enzyme and the DL reasoner can automatically classify Enzyme as a kind of Protein. In this way, the model is built up from small pieces in a descriptive way, rather than through the assertion of hierarchies. A DL reasoner supplies a number of reasoning services which allow the construction of classification hierarchies and the checking of consistency of these descriptions [42].

The taxonomy for named concepts can be automatically established by a logic reasoning system for DLs. DLs have clear semantics, it is possible to use all of the knowledge encapsulated in the ontology to reason whether it is consistent and complete. This is not possible with simple representations such as the Gene Ontology.

3.4. The OWL Web Ontology Language

OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans [44].

Recently the W3C has released a recommendation for OWL, which is in part a mechanism for representing DL ontologies in existing W3C technologies, which are Extensible Markup Language (XML) and Resource Description Framework (RDF). XML provides a syntax for structured documents but places no semantic constraints on their meaning. XML Schema restricts the structure of XML documents and extends XML with data types. RDF is a data model for resources and the relations between them. It provides

a simple semantics for this model, and these semantics can be represented in XML syntax. RDF Schema is a vocabulary for describing properties and classes of RDF resources, with a semantics for generalization hierarchies of such properties and classes. OWL adds more vocabulary [44] for describing properties and classes, such as but not limited to relations between classes, cardinality, equality, richer typing of properties, characteristics of properties, and enumerated classes.

There are three OWL sublanguages, OWL-Lite, OWL-DL, and OWL-Full.

OWL-Lite: supports those users primarily needing a classification hierarchy and simple constraints, and it is therefore easier to provide tool support for this sublanguage.

OWL-DL: supports maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems [44]. OWL-DL is named as it corresponds to particular description logic $\mathcal{SHOIN}(\mathcal{D})$ [169]

OWL-DL includes all OWL constructs with restrictions such as type separation (a class name can not also refer to an individual or property; a property name can not also refer to an individual or class).

OWL-Full: supports maximum expressiveness and the syntactic freedom of RDF with no computational guarantees (i.e. in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right). OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning tool will be able to support every feature of OWL Full. [44]

Every OWL-Lite ontology is a legal OWL-DL ontology, but not the inverse, and so on for OWL-DL and OWL-Full.

3.5. OWL and Description Logics

The ability of using description logics makes OWL a good ontology language. The rich expressivity of OWL can be used to model the complexities of biology and bioinformatics [53]. Also automated reasoning using Racer can support the process of building and evaluating ontologies.

We argue here that Description Logics (DLs) have several key advantages in representing ontologies. These are:

Expressivity: DLs are highly expressive enabling rich and complex descriptions of domain concepts. This enables a precise interpretation of the concepts in an ontology which in turn allows machine interpretations of these concepts.

Automated Reasoning: DLs are based on formal logics. This has enabled the development of reasoners which are capable of checking ontologies for consistency. The use of automated reasoning technologies also enables the ontological engineer to adopt a different style of modeling, one which is highly compositional.

Compositionality: The first two properties enable the building of ontologies in a highly compositional fashion which makes building large ontologies more manageable.

3.6. Description Logics for building Bio-Ontologies

Knowledge representation needs theories and systems for expressing structured knowledge, accessing and reasoning with it. Description Logics are considered as one of the most important knowledge representation formalism unifying and giving a logical basis to the well known traditions of frame-based systems, semantic, object-oriented representations, semantic data models, and Type systems [40].

Currently, many Bio-ontologies are using Description Logics (DLs) for knowledge representing. GALEN [45] and SNOMED [46] were both developed in a native DL formalism [64]. Several other groups have worked at converting existing terminologies into terminologies based on a DL formalism (UMLS Metathesaurus [47] [48], UMLS semantic network [49], Gene Ontology, National Cancer Institute Thesaurus [50]), GONG (Gene Ontology Next Generation) and TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources).

Also as mentioned before Protege's OWL plug-in now also allows developers of frame-based resources to export their ontologies into DL formalism.

Description logics is also being used for ontology validation. The validation of an ontology by a DL-based classifier allows compliance with certain rules of classification and it brings also other benefits in terms of coherence checking and query optimization [51]. However, neither DLs formalisms nor the use of a classifier can ensure compliance with all principles of an ontology [63].

Description logics provide a formalism suitable for representing many features of a variety of different domains including the biomedical domain in a way that can support automatic reasoning and information retrieval.

3.7. RACER and RQL

RACER (Renamed ABox and Concept Expression Reasoner) [57] is the first full-fledged ABox description logic system for a very expressive logic and is based on optimized sound and complete algorithms. It provides a Semantic Web inference engine to implement a TBox and ABox reasoner in description logics [66]. We use Racer for developing ontologies, query answering over RDF documents and OWL ontologies and

registering permanent queries for building a document management system with notification of new results if available [65].

Racer is currently used by many people in the KR community to support a wide range of inference services about ontologies specified in OWL. Many ontology editors, ontology development and visualization tools are using Racer.

RQL - Recently nRQL (new Racer Query Language) - is an extended query language for Racer. The RQL can be seen as an extension and combination of the ABox (assertions about individuals, either that an individual is a member of some class, or related by a property to some other individual), querying mechanisms. The RQL allows the use of variables within queries, as well as much more complex queries. The variables in the queries are to be bound against those ABox individuals that satisfy the specified query. Queries will make use of concept and role terms; also the current TBox (axioms about class definitions, e.g., that A is a SubClassOf B or an EquivalentClass of C) is taken into account. However, it is possible to use ABox individuals in query expressions as well. ABox individuals and variables will be commonly referenced as objects [68].

Chapter 6 contains more information about querying, using nRQL.

3.8. Tools for Ontology Development

Tools are essential to aid the ontologist in constructing an ontology, and merging multiple ontologies. Such conceptual models are often complex, multi-dimensional graphs that are difficult to manage. These tools also usually contain mechanisms for visualising and checking the resulting model over and above the logical means for checking the satisfiability of the specified model. A survey of ontology development tools can be

found in [43, 170]. We briefly review OilEd and Protege as two tools which support DL reasoners.

3.8.1. OilEd

OilEd [171] is a graphical tool for creating and editing OIL ontologies developed at the University of Manchester. The tool is installed locally. OilEd can use DAML+OIL and OWL languages. OilEd uses the FaCT system [172], a description logic system, for checking the consequences of the statements in the ontology. The knowledge model for OilEd is based on description logics. In contrast to frame systems, OilEd allows for arbitrary boolean combinations of classes. OilEd also allows several types of constraints such as value-type, has value and cardinality restrictions [170].

3.8.2. Protege

Protege was developed by Stanford Medical Informatics at the Stanford University School of Medicine. It is considered as an integrated tool for domain experts as well as ontology developers to develop knowledge-based systems. A knowledge-based system includes information about a given domain and programs that include rules for processing the knowledge and for solving problems relating to the domain. This editor consists of a graphical user interface (GUI) whose top-level consists of overlapping tabs for classes, instances, slots, forms, and queries to allow for presentation of these parts and convenient editing and interaction between them. Protege supports users in:

- Design an ontology.
- Create a knowledge-acquisition tool for collecting knowledge.
- Enter instances of data and create a knowledgebase (KB).

• Execute applications.

The developed knowledge-based system using Protege can then be used with a problem-solving method (PSM) which is a software agent used with a KB to answer questions or solve problems regarding the domain. Protege was designed as an aid to both developers and domain experts as a user-friendly interface for creating these KB systems. It is designed to guide them through the process of system development. It also allows users to reuse existing domain ontologies and methods. It was developed originally for use in the field of clinical medicine and the biomedical sciences, but now it is being used in many areas where the concepts can be modeled as a class hierarchy.

The frame-based Protege ontology development tool has been adapted to represent ontologies in OWL-DL, so that one can build frame-based ontologies whilst gaining from the reasoning services offered by a DL. This is especially important for large, collaboratively developed ontologies that are intended to be reused and shared.

Protege is designed to support iterative development, where there are cycles of revision to the ontologies and other components of the knowledge-based system. Therefore developers should not expect to "complete" ontology development without considering other aspects of the process.

4. Overview Existing Bio-Ontologies

4.1. An Overview of the Gene Ontology

Biological knowledge is inherently complex and so cannot readily be integrated into existing databases of molecular (for example, sequence) data. An Ontology is a formal way of representing knowledge in which concepts are described both by their meaning and their relationship to each other. The use of ontologies within bioinformatics is relatively recent and consequently there are not a large number in existence [54]. This chapter reviews some most frequently used bio-ontologies in the bioinformatics community with focus on the Gene Ontology which is considered as a standard for many bioinformatics applications and the current issues in their design and development including the ability to query across databases and the problems of constructing ontologies that describe complex knowledge. All these ontologies are very different and specific to their intended use.

Bio-ontologies provide a means of formalizing biological knowledge, for example, about genes, anatomy and phenotypes in complex hierarchies that are composed of terms and rules. Most bio-ontologies are stored at http://obo.sourceforge.net and are accepted by the community as authoritative.

The primary goal for developing all bio-ontologies is to enable users aggregating several kinds of objects together such as gene sequences, literature references, web pages into custom, user defined collections. Such collections represent valuable interpretations of relevance and could then be shared, sent between colleagues, and searched [56].

The Gene Ontology project is a collaborative effort to address the need for consistent descriptions of gene products in different databases [69]. The GO consortium [69] was initiated in 1998 and is currently a collaboration among many database projects such as FlyBase, MGI, SGD, TAIR, and WormDB and covers over 5000 concepts. The goal of the GO Consortium is to produce a controlled vocabulary that can be applied to all organisms even as knowledge of gene and protein roles in cells is accumulating and changing [5]. The Gene Ontology is accepted as a one of the most important tools for representation and processing gene's products and functions.

GO is developed based on three ontologies that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner. The controlled vocabularies are structured in GO so, that use of GO terms by several collaborating databases facilitates uniform queries across them and users can query at different levels.

In GO, first of all, ontologies are written and then associations between the ontologies and the genes and gene products in the collaborating databases are made, and at the end tools for facilitating the creation, maintenance and use of ontologies are developed. This "annotations" are established electronically and then later validated by a process of manual control, so it is necessary for an annotator to be not only a GO expert but also expert in biology, the gene and its products.

4.2. The Gene Ontology Architecture

The intention of the Gene Ontology Consortium [69] is to create a shared biological resource that would enable the community to describe gene products using a common vocabulary and semantics. The GO is not intended to deal with the whole molecular

biology knowledge captured in the community databases, but captures information about the role of gene products within an organism, because the knowledge of the biological role of proteins in one organism can often be transferred to other organisms [72]. On this basis, the GO provides controlled vocabularies for the description of third-party independent ontologies:

- **Molecular function in GO:** It describes the tasks performed by individual gene products. This part is defined in the GO documentation as "the action characteristic of a gene product" [69]. Molecular function is a capability that a physical gene product or gene product group carries as a potential. It describes only what it can do without specifying where or when this usage actually occurs [73].

  Using "function" as a meaning for "action" in the molecular function hierarchy, can cause confusion, essentially where one notices some terms such as "enzyme" (defined as "a substance that catalyzes") which refer neither to functions nor to actions but rather to substances. Recently, the GO consortium for solving this problem changed all the GO molecular function term names such that the word "Activity" has been appended [74]. However, this change solves problems with some terms, but changing names is not enough at all because associated definitions are still not changed, and cause inconsistencies in this ontology.

- **Biological process in GO:** it is accomplished via one or more ordered assemblies of molecular functions. Biological process and molecular function terms are clearly interrelated. Molecular function is the initiator of biological process. GO's authors insist that part-of holds only between entities within a single hierarchy

and never between the three GO term hierarchies [69], but some terms in molecular function hierarchy have part-of relationships with some in the biological process hierarchy and it cause inconsistencies. Examples of biological process terms are "cell growth and maintenance," or "signal transduction." A biological process is not equivalent to a pathway, and the GO does not capture any of the dynamics or dependencies that would be required to describe a pathway.

- **Cellular component in GO:** It consists of sub-cellular structures, locations, and macromolecular complexes. GO includes in this vocabulary both the extracellular environment of cells and the cells themselves (cell is subsumed by cellular component) for example, terms such as virion, chromosome, nucleus, ribosome, and proteasome. Cellular components are physical and measurable entities. A gene product has one or more molecular functions and is used in one or more biological processes. It may be associated with one or more cellular components. Thus the relations between the gene product and molecular functions, biological processes and cellular components are many to many.

These three Ontologies (molecular functions, biological process and cell components) in GO are represented as directed acyclic graphs (DAGs) or networks consisting of a number of terms, represented as nodes within the graph, connected by relationships, represented as edges [75]. A DAG allows a node to have more than one parent and for the edges to distinguish between different types of relationships between nodes [76]. A DAG allows one to express multiple inheritances with the is-a relationship, where a child term may be a subclass of its parent, or with the part-of relationship, where a child is a

component of its parent term. A child term may inherit relationships of different classes from its different parents. Nevertheless, most of the relationships in GO are of type is-a and implement mainly single inheritance. Thus, the GO ontologies are built in the form of taxonomies. GO is multi-dimensional, separating the concepts of 'functional primitive', 'process' and 'localisation'. Its more complex architecture allows it to accommodate functional descriptions that are examples of more than one parent node. The scheme is being developed for classification of the gene complements of both unicellular and multi-cellular organisms.

AmiGo [78] is the new web interface for querying the "Gene Ontology" and the associated gene products. GO is represented in form of text or XML files [69]. This can facilitate the use of the GO by different external databases, which were not initially included into the GO project [83].

4.3. GO and Ontologies in Computer Science and Philosophy

In computer science an ontology is viewed as a terminology that is organized in a hierarchical structure called taxonomy, with which axioms and definitions are associated. It is often specified (for example in some description logic framework) to facilitate the support of software applications.

In philosophy ontologies can be considered as theories of the different types of entities (objects, processes, relations) existing in given domains [82]. Here both a logically rigorous formalization and a representational adequacy are important for the stability and extendibility of an ontological framework and one can not be sacrificed for another.

When we look at GO precisely, then we find that the Gene ontology in spite of its name, is not exactly an ontology as defined in computer science or philosophy. Based on the GO consortium definition, GO seems to be more a "controlled vocabulary" or nomenclature for molecular biology rather than a real ontology.

However, GO uses hierarchies of terms and taxonomies, but its focus has been directed toward providing a practically useful framework for keeping track of the biological annotations that are applied to gene products. GO has focused neither on software implementations nor on the logical expression of the theory encompassing these terms. So, when the GO consortium was faced with the trade-off between (1) formal and ontological coherence, stability and scalability and (2) the speedy population of GO with biological concepts, then preference was given to the latter. It means too little attention was paid to the significance of those ontological terms such as function, part, component, substance, action, domain and complex, which were employed in GO's construction [74].

4.4. Logical Relationships in Gene Ontology

GO uses two relations, "is-a" and "part-of" [84]:

IS-A Relation: In the GO documentation is-a is used with the meaning of "subclass of" and "instance of "[85]. The is-a relation is clearly used in such a way to indicate "is a kind of". The is-a relation is distinct also from the relation of part to whole. Confusingly, is-a is sometimes also used with the meaning part-of, as in the definition of lysosomal hydrogen-transporting ATPase V0 domain, which says the V0 and V1 domains are kinds of V-type complexes, rather than component parts thereof. Obviously, such errors derive, again, from a lack of attention to ontological principles.

Part-of Relation: This relation is totally different from subsumption, where if *X* is subsumed by *Y*, then every instance of *X* is necessarily an instance of *Y*. As GO Usage Guide says: "can be a PART-OF, (not) is always a part-of". In addition, the PART-OF relation is intended to behave transitively [85]. So, Part-of is transitive. [84]

Part-of is used in GO for representation of parts of both substances and processes and of functions/activities. Part-of has a variety of usages: [84]

- Physical- Part-of: inner membrane is part-of membrane.
- Functional-part-of: casein kinase II catalyst is part-of casein kinase II.
- Steps-in-a-process: synaptic transmission is part-of transmission of nerve impulse.
- Conceptual-part-of: the term 'molecular function' is part-of the gene ontology.
- Membrane part-of cell: means "a membrane is a Part-of every cell".
- Flagellum part-of cell: means "a flagellum is Part-of some cells".
- Replication fork Part-of cell: means "a replication fork is part-of the cell (nucleoplasm) only during certain times of the cell cycle".

As one can see above, is-a and part-of are not always used in a consistent way, i.e., in GO, is-a means "subclass of" or "instance of" [85]. Similarly, part-of is used with different meanings such as: "made of", "belongs to", "physical part of", "conceptual part of", "subprocess of", "controls", "causes", "activates", "inhibits", "enclosed by" and "binds to" [86]. For clarity and adaptability, each of these different usages for is-a and Part-of should be explicitly represented in GO as a different relation.

4.5. GO semantic integrity

The Gene ontology needs to be updated and expanded to become more useful for researchers and biologists. But as the GO expands in size and scope, its semantic integrity will at the same time become more difficult to maintain through manual inspection. In fact when GO expands, it will, as the GO consortium accepts, "be increasingly difficult to maintain the semantic consistency we desire without software tools that perform consistency checks and controlled updates" [83]

As mentioned before, one of the parameters that we have to consider for evaluating an ontology is conciseness. It means that no redundancy is allowed in ontologies and it is also checked for appropriateness. So, the addition of each new term will require the curator to understand the entire structure of the Gene Ontology in order to avoid redundancy and to ensure that all appropriate links are made with other terms.

4.6. TAMBIS, RiboWeb and EcoCyc

There are many bio-ontologies [69, 70] with different structure, scope, area of application, characteristics, representation language etc. What is important for a biologist who works with these systems are the type of knowledge represented by these ontologies and how it is represented and how it is delivered [71]. Some of the most popular bio-ontologies are:

- The TAMBIS Ontology (TaO) [29]
- The EcoCyc ontology [180]
- The RiboWeb ontology [181]

Each of them has its own benefits and weaknesses. Basically in ontology design there is no definite rule to determine the "best" [89].

The Gene Ontology as already mentioned is a controlled vocabulary for annotating gene products for molecular functions, the biological processes in which they are involved and the cellular locations in which they are found. It is also a hand-crafted DAG that facilitates multiple inheritance [80]. EcoCyc has used an ontology to specify a database schema for the E. *coli*. metabolism, signal transduction, etc. RiboWeb also uses an ontology to describe its data, but also guides its users through the analysis of their data. Finally, TAMBIS uses an ontology to allow users to query bioinformatics databases. Each of these uses a different knowledge representation system. GO is a control vocabulary (phrase-based) So, It is easy to build and access but difficult to maintain (especially for multiple hierarchies), and can cause consistency problems. Phrase-based ontologies are expressive, as they use natural language forms, but lack formality and rigour, therefore, sometimes they are difficult to re-use [90]. EcoCyc and RiboWeb are frame based systems that have advantages of an easily accessible and intuitive modeling style, reminisient of an object view of the world (considering a frame as a class and its slots as attributes). A frame encapsulates the properties of the instances [1]. These systems share a common disadvantage with phrase based vocabularies (used in GO), both of them are hand-crafted and can suffer from inconsistencies and logical mistakes. Finally, TAMBIS uses Description Logic for knowledge representation. Description Logics have a well defined semantic and powerful reasoning support that improves maintenance of logically consistent ontologies [29]. Concepts can be defined in terms of their properties and the reasoning used to classify the concepts is based on those descriptions. When a concept expression is unsatisfiable in terms of the rest of the model, the reasoning support can inform the modeller of his or her mistake. Description logic

based ontologies avoid many problems of the hand-crafted ontologies, but suffer from the complexity of the modeling style [1].

Existing bio-ontologies differ in their intention, structure, their coverage, and detail level. Although all considered ontologies are intended for molecular biology, they cover different parts of this domain and have different detail levels. In Table 4-1, a comparison is shown between "Gene Ontology" and some other bio-ontologies (extracted from a survey [72]).

Ontology	Application Scenario	Domain Oriented component	Generic Component	Detail Level	Conceptual representation	Storage and access
Gene Ontology (GO)	Controlled Vocabulary for database annotation	Drosophila, mouse and yeast gene and gene product function, process and cellular location	X	High	Taxonomies (phrases)	Text, XML Files. Java browser
EcoCyc	Database Schema	E.coli genes, metabolism, regulation, signal transduction and metabolic pathways	✓	High	Frames	storage - *DB* query processor (LISP)
RiboWeb	Database Schema	Ribosome components, Covalently bonded molecules, biological macro molecules , region of molecules	✓	High	Frames	Storage – DB Java Browser
TAMBIS Ontology (TAO)	Common Access ontology based search	Proteins, enzymes, motifs, secondary and tertiary structure, functions and processes, sub-cellular structure and chemicals.	✓	High	Description logics	query processor (Java)

Table 4-1: The summary of the content, structure and representation of some bio-ontologies [72]

4.7. Gene Ontology Next Generation (GONG) [92]

The Gene Ontology seems like an inert hierarchy of terms, that is focused not on reasoning power or on supporting software implementations but rather on providing a

robust framework for the annotations that are applied by biologists to organism gene products [91]. The complexity of an ontology required to support automated reasoning is difficult if not impossible to attain by manual curation. So, because GO is important and widely used in biological community, therefore scientists are looking to use the tools and techniques based on description logics to make the ontology amenable to automated reasoning and to deal with the added complexity of maintaining such a resource [92]. GONG is attempting to improve the Gene Ontology by rendering GO in description logics [93].

4.8. Major Strengths and Weaknesses of Gene Ontology

The GO approach has several strengths and weaknesses.

Strengths of GO:

- GO is easily accessible and has a flexible structure [80].
- Populating GO does not require the completion of complex protocols of formally determined steps but can be done intuitively by the expert biologist.
- There are a few formal constraints standing in the way of easy incorporation of existing controlled vocabularies from the biological domain.
- The principle of unique identifiers allows GO terms to be used for database annotation without consideration of their place in the GO hierarchy.

Weaknesses of GO:

- Hard to maintain when its vocabularies grow. Semantic integrity becomes more difficult to maintain when GO expands.
- Lack of semantics and rigour begin to become problematic [80].

- Enormous gap between GO-annotated docs (27,000) and full MEDLINE database (12 million entries) [69].
- When knowledge changes, then definitions will be altered [69].
- The relations is-a and part-of are not always used consistently. There is no distinction between generic and individual entities in GO which clearly restricts its expressive capability.
- GO has about 700 concepts that do not have a parent concept. It means that about 700 independent subdomain ontologies exist. This is certainly not desirable.
- No procedures are offered by which GO can be validated [81].
- There are no clear design principles given for GO. The way of how a concept finds its way into GO is not well defined. So, it is so difficult to understand the structures. For example why a certain concept was placed into a particular class. It would be difficult for reuse and maintenace.
- 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 sub-classifying criteria are given there is little guidance from within GO. [86]
- GO is actually intended to be not one but three ontologies. Leaving the large set of parentless concepts aside, 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 [86].

- Using GO "as is" takes too long and delivers too little.
- Maintenance and consistency preservation is difficult and arduous.
- It is unclear what kinds of reasoning are permissible on GO's hierarchies.
- The rationale of GO's sub-classifications 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. [74]

The Gene Ontology provided a valuable starting place for genomics, which will make the design of other genomics ontologies easier, but it is not a substitute for them.

Today people are beginning to find emerging methodologies that compare the structure and role of various ontologies [94], but judging a particular ontology is still not easy. The 'Gene Ontology' addresses many of the problems and issues that scientists have discussed in the biological domain. [76]. When one compares the Gene Ontology with other existing ontologies one has to notice that even the ontologies that cover the same parts of the same domain can differ in their detail level, which determines how deep and wide they capture the lower level concepts (e.g. different types of proteins, enzymatic reactions and cellular processes).

Ontology development is necessarily an iterative process and it is necessary in the life cycle of ontologies to change and refine frequently. One can accept GO as it is and then consider it as a base for other biological applications. But a lot of problems will migrate from GO to these systems. If scientists want GO as a basis and standard for a wide range of biological systems, then, first of all they have to solve its known problems.

5. Fungal Genomics

5.1. Fungi in Economy [59]

Fungi play an important role in recycling nutrients in the biosphere. They are the major decomposer species in terrestrial habitats. Many fungi are used directly as foods and food flavoring. Mushroom cultivation, asian food fermentations such akoji, soy sauce, oncham, and tempeh are a significant component of the diet of millions of people. Fungi can ramify through substrates; literally digesting their way along by secreting extra-cellular enzymes while their filamentous growth form facilitates mechanical penetration of potential food sources.

Fungi possess the most efficient battery of depolymerizing enzymes of all living things [59]. The FungalWeb ontology deals with fungi organisms and fungal enzymes. Fungal enzymes are already being used widely in industry. Enzymes have critical roles in different processes, such as breaking down wood by removing liquid-based materials in wood, breaking down cellulose fibres and resins in trees, or removing fat and protein in clothes. Some people call such enzymes "degradative" because they can be used to break down organic materials, including dirt in laundry [59]. These fungal enzymes can convert wood, plastic, paints and jet fuel, among other materials, into nutrients. Canada is a major producer of pulp and paper, manufacturing about 30 million metric tonnes and exporting over one third of its capacity [61].

This "degradability" has both good and bad economic effects. On the positive side, fungal enzymes drive the earth's carbon cycle, recycling ligninocellulosic remains. Moreover, many of the oldest biotechnological processes are based on fungal catalytic power:

baking, brewing and wine fermentation are examples of the way people have used fungal enzymes since ancient time. Currently, productions of laccases, xylanases, pectinases, proteases, lipases and cellulases have been turned into major industries such as:

- **Pulp and paper manufacturing:** laccases, cellulases, pectinases and xylanases;
- **Waste treatment and decontamination:** laccases and manganese lipases;
- **Food processing and functionalities:** proteases, invertases, cellulases, xylanases, , pectinases, glucomylases, lactases and glucose isomerases;
- **Baking:** xylanases, glucose oxidasess, lipases, lipoxygenases and proteases;
- **Brewing**: decarboxylases, beta-glucanases, cellulases, xylanases and proteases;
- **Wine making:** pectinases, glucosidases and cellulases;
- **Leather processing:** proteases and lipases.

The pulp and paper industry employs about 70,000 people in Canada, over half of them in Quebec. Export of pulp and paper in 1999 contributed $17.6 billion to the Canadian economy [59].

On the negative side, fungal enzymes damage standing timber, finished wood products, cotton fibers, and many other human artefacts such as fuels, paints, glues, drugs, and electrical equipment. They compete with insects and rodents as the major destroyers of foods and feeds. Estimates are that 10% of the world's food supply is lost each year due to fungal contamination. Fungi attack standing crops as plant pathogens (rusts, blights, smuts) and cause equal damage to harvested foods and feeds as storage contaminants (rots, molds, mildews). They cause billions of dollars of damage to agricultural crops each year. Although an important part of the economy of Canada, the pulp and paper industry is the third most polluting in North America [59].

Understanding fungal metabolism and activity might allow one to control unwanted fungal growth and have a better chance of defending against their bad effects by changing or removing some harmful pathways [60].

5.2. Clinically relevant fungi

Some fungal products are exploited as antibiotics (i.e. penicillin), immune response medication (e.g., cyclosporin A), blood pressure lowering agents (e.g. mevalonin), hemorrhage and migraine control drugs (i.e. ergot alkaloid) and other pharmaceuticals [59].

Fungi also cause diseases. In addition to being the major plant pathogens, hundreds of species are of medical and veterinary importance. Patients after cancer chemotherapy and organ transplantation, suffering from AIDS, or with other forms of immunocompromised status, are particularly susceptible to life threatening fungal infections.

There are two main groups of pathogenic fungi which are different from one another. Firstly, dermatophytes are a group of obligate parasites which attack human skin, nails, and hair. Secondly, dimorphic saprobes are a group of normally soil-borne fungi which have developed a different morphology in order to adapt to the hostile environment of the human body. However, antifungal drugs now constitute a billion dollar industry; most of them have side effects. Pharmaceutical manufacture using fungi constitutes a $23 billion per year industry worldwide. Worldwide annual sales of leading antifungal drugs (Diflucan, Sporanox, Nizoral,and Lamisil) were approximately $2 billion [60].

The FungalWeb ontology (FWOnt) can potentially be reused in both industrial and medical domains. An enormous number of species of fungi are already known and some groups of fungi, because of their economical or pathological importance, have been

studied more extensively. FWOnt is mostly focused on Yeast as the most well known fungi. The yeast community has taken the lead in fungal genomics because it is the only fungi which have complete genomic DNA sequences [60]. The yeast genome (A genome is the complete set of genetic material of an organism) consists of 16 chromosomes encompassing 12,067,266 base pairs. Yeast chromosomes are gene-rich.

5.3. Fungal taxonomy

Fungi, plants, and animals represent the three phylogenetic kingdoms (Phylogeny is defined as the evolutionary history of a kind of organism) within the eukaryotes. Fungi are genuine eukaryotes having cell and genome structures and metabolic organization similar to that of other eukaryotes like plants and animals. Fungi includes over 1.5 million different species which are universally used as model organisms for understanding all aspects of basic cellular regulation including cell cycle progression, gene regulation, circadian timing, recombination, protein secretion and development [60]. Fungal taxonomy is a dynamic, progressive discipline that consequently requires changes in nomenclature [95]; these changes often caused difficulties for ontologists, biologist and clinical microbiologists. Also fungi are mostly classified on the basis of their appearance rather than on their nutritional, molecular and biochemical differences. This implies that different concepts have to be applied in fungal taxonomy.

The use of classification as a technique for collecting, representing and using biological knowledge has a long history in the field. Today, increasing interest in fungal taxonomy has led to the development of new methods and approaches.

5.3.1. The concept of species and the relationships in fungi

Different concepts have been used by mycologists to define the fungal species. The biological concept, which was developed before the advent of modern phylogenetic analysis, emphasizes gene exchange (i.e., sexual and parasexual reproduction) within species and the presence of barriers that prevent the cross-breeding of species [96].

Little is known about evolutionary relationships among fungi. Only recently have some data become available, although they are still sparse [97]. The proposed phylogenetic relationships among the animal, plant, and fungi kingdoms depend on the molecular regions and methods used by different investigators. Phylogenetic analysis has shown that the fungal kingdom is part of the eukaryotic groups [98, 99]. The three main fungal, are considered as Ascomycota, Basidiomycota and Zygomycota which are thought to have diverged from the Chytridiomycota, Glomeromycota and Microsporidia 550 million years ago [100].

5.3.2. Fungal nomenclature

To be formally recognized by taxonomists, an organism must be described in accordance with internationally accepted rules. The rules that control the bio-nomenclature are very diverse and depend on the type of organism. The taxonomy of some fungi is particularly unstable and controversial at present. Changes to the names of taxa and their consequent diseases are potentially confusing [101].

For example, the name of the fungus Allescheria boydii (so called in the early 1970s) was changed to Petriellidium boydii and then to Pseudallescheria boydii within a very short

period [102]. The FungalWeb Ontology is able to clarify the confusion that changes in fungal names can cause, by specifying synonyms for terms.

Another controversy resulted from the replacement of the terms "anamorph" and "teleomorph" with "mitosporic fungus" and "meiosporic fungus", respectively, in the new scientific texts [104].

The application of nomenclatural rules to the complicated life cycle of fungi can create some confusion among the users of fungal resources [105].

The correct specification of fungi in the FungalWeb ontology could play a critical role in clinical and industrial applications.

Fungal taxonomies are still based mainly on morphological criteria. Numerous alternative approaches have been developed, including nutritional and physiological studies, serologic tests, secondary metabolites, ubiquinone systems, and fatty acids. Although some of these are very useful for identifying poorly differentiated fungi such as yeasts and black yeasts, they are only complementary tools of morphological data in most cases.

5.3.3. Morphology

Classification systems of organisms are historically based on observable characteristics. The classification and identification of fungi relies mainly on morphological criteria (unlike bacteria) [95]. In the rare instances that opportunistic fungi develop the teleomorph in vitro (this happens in numerous species of Ascomycota and in a few species of Zygomycota and Basidiomycota), there are many morphological details associated with sexual sporulation which can be extremely useful in their classification. The type of fruiting body (basidioma in basidiomycetes, ascoma in ascomycetes) is vital

for classification. The shape, color, and the presence of an apical opening (ostiole) in the ascomata are important features in the recognition of higher taxa [95].

Another problem of classification based on morphological criteria is the dual system of classification, with no consistent correlation between different taxonomies [106]. This is an important difficulty in establishing the taxonomic concept of the fungus as a whole.

5.3.4. Molecular techniques and the fungal taxonomy

Since the distinguishing morphological characteristics of a fungus are too limited to allow its identification, molecular and biochemical techniques are applied for fungi. Molecular methods are applicable and comprehensive in fungal systems [103].

We have several techniques in this area such as biochemical techniques, secondary metabolites, fatty acid composition, cell wall composition and protein composition.

The organisms usually considered to be fungi are very complex and diverse; they include multi-cellular and unicellular forms, and can reproduce by different types of propagules or even by fission. The kingdom fungi is organized into phylum (major taxonomic group in biological classification) and then into classes and orders. The major phyla presently accepted in fungi are Chytridiomycota, Zygomycota, Ascomycota, Basidiomycota, Glomeromycota and Microsporidia. Also approximately 70,000 fungal species were accepted [95]. The FungalWeb ontology taxonomy is built based on these accepted phyla. Because lack of standardized and stable terminologies, there were many fungi which were remained unclassified in the taxonomy, therefore another distinct class which is called "Unclassified fungi" is added to the ontology to cover those fungi.

Ascomycota, Basidiomycota and Zygomycota contain most of known fungal organisms.

Ascomycota: is the largest phylum of Fungi. It contains almost 50% of all known fungal species and approximately 80% of the pathogenic species.

The basic characteristic which differentiates ascomycetes from other fungi is the presence of asci inside the ascomata. However, even in the absence of these important diagnostic characteristics, the ascomycetes can be recognized by their bilayered hyphal walls with a thin electron-dense outer layer and a relatively electron-transparent inner layer [107].

Molecular data are adding a new dimension to the understanding of the relationships among the different groups of ascomycetes [107].

Pezizomycotina, Saccharomycotina, Schizosaccharomycetes, Neolectomycetes and Pneumocystidomycetes are different types of Ascomycota.

Two categories of Ascomycota are Penicillium and Aspergillus. Because of the AIDS pandemic, species of Penicillium has acquired considerable importance [108].

The taxonomy of Penicillium has always been complex due to its great number of species (approximately 250), which have very few differences. The species of Aspergillus is also being used in several industries and clinical application.

Yeasts: The term "yeast" is often taken to be a synonym for *Saccharomyces cerevisiae*, but the phylogenetic diversity of yeasts has been assigned to diverse fungal taxa. Yeasts are neither a natural nor a formal taxonomic group. In some cases, they are merely a phase of growth in the life cycle of filamentous fungi which takes place only under particular environmental conditions. Different numbers of taxa have been estimated by the different authorities. In the modern classification scheme, there are approximately

100 genera representing near 700 species [109], although the genetic relationship for some of the currently accepted taxonomies is still unknown. Yeasts are formally assigned to the Ascomycota, in the orders Saccharomycetales and Pneumocystidales, or to the Basidiomycota in the orders Tremellales and Ustilaginales [95].

Basidiomycota: (or basidiomycetes) The most characteristic feature of basidiomycetes is the formation of basidia, although they are rarely produced in vitro. Basidia are usually aseptate structures, with four tiny projections, called sterigmata. [110]. Also Basidiomycetes usually are known form their fruiting bodies or "mushrooms" and they are consumed as food all over the world [60]. Basidiomycetes of medical interest are usually placed in the orders Agaricales, Stereales, Tremellales, and Ustilaginales.

Zygomycota : The Zygomycota are a group of lower fungi whose thalli are generally nonseptate (coenocytic) after the fusion of isogamic sex organs (gametangia) they produce a single, dark, thick-walled, ornamented sexual spore, called zygospore [107].

6. The FungalWeb Ontology Design and Development

6.1. The FungalWeb Ontology Development Life Cycle

To design and development of the FungalWeb Ontology first of all we specified our domain and the level of granularity. Based on this information vocabularies were extracted from different resources such as databases, raw texts in libraries, papers and literature, available dictionaries, thesaurus and other existing bio-ontologies. After structural and morphological analysis and by integrating other ontologies by consulting with domain experts the extracted terms were organized into a hierarchical structure called taxonomy, using Protege as an ontology editor. Also RACER as verification tool and query engine is being used.

The following diagram shows the FungalWeb ontology development life cycle.

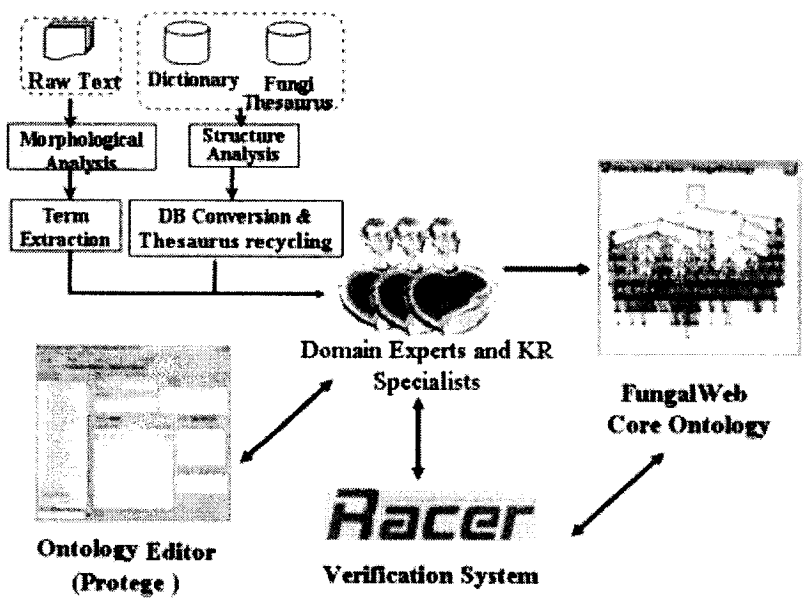


Figure 6-1: The FungalWeb ontology development life cycle

The starting stages in the ontology development are very similar to requirement gathering in the software development life cycle.

6.2. Specification

6.2.1. Purpose, Goals and Scope

The FungalWeb ontology aims to be a formal ontology in the domain of fungal genomics. By creating a shared thesaurus of concepts from different fungal biology resources and defining the relationships between the concepts, these logically defined terms can be both human and machine understandable. FWOnt is formally designed based on information from domain experts (biologists) and it is intended to be used by biologists.

6.2.2. Granularities

Every knowledge representation makes certain assumptions about the level of granularity of the resources to be represented [55, 56].

Currently the FungalWeb ontology focuses on fungi-enzyme interactions and for this purpose fungi species and enzyme instances are defined precisely with a high level of detail. However for flexibility and reusability of the ontology some functional concepts such as gene regulation and protein secretion are defined in order to be used in future work.

The ontology is modeled using a high level of granularity. Different identifiers are assigned to the general concepts, i.e., a specific name, physical attributes like color, body, functional parameters and molecular properties. After assigning specific names for the

concepts and other ontological constructors the relationships between the concepts are precisely described.

A consequence of the granular representations is the ability to annotate the elements and resource in the domain. Annotations in FWOnt are simply pieces of information, usually human-made observations that are attached to a resource. These observations are not restricted to plain text representations and can take advantage of the full expressive power of RDF. Several schemes for RDF-based annotations have been explored in the past [58].

6.3. Knowledge Acquisition

Today, as large genome projects are developed, it becomes more obvious that the ability to access knowledge and data for gene products and proteins in particular for enzymes is limited. Collecting, interpreting and standardizing those data is a very difficult job because they are highly distributed among literature, texts and papers from different fields and are often subject to experimental conditions.

Naturally it is hard to cover all the numerous literature references for each fungus and enzyme.

While advances in database and parallel processing technology over the past decade have improved the celerity with which bioinformatics computations can be performed, many obstacles persist between bioinformaticians and their data, which is often scattered over dozens of machines in incompatible data stores in a myriad of formats. One major challenge is to find a way to unifying diverse bioinformatics data sources and literature databases in a consistent format using semantic web techniques. Life scientists face many

challenges in their task of managing biological information. Looking at the bioinformatics workflow at a low level, one can observe that data from experiments is entered into databases, where scripts written in languages such as Perl are employed to filter and analyze the data and to compare them with other known samples [22].

At a more conceptual level, the information that is used and generated by life scientists, including chemical pathways, annotated gene sequences, and protein structures, is highly connected in nature. For example, an enzyme that catalyzes some specific pathway has a specific, definite structure and genetic sequence that encodes how to construct it. Connections also exist between any given protein and other proteins that are similar to it, either in terms of functionality or composition [55].

These different forms of information (i.e. annotations, pathways, structures, sequences) have been stored in a series of incompatible databases using distinct data formats. So, because these databases must be bridged, manipulated and normalized to accomplish all but the simplest of tasks, life scientists have been prevented from working with their information at the desired high level. An example is the EcoCyc ontology knowledge gathered from the research literature on E. coli. metabolism. In the former case this was a huge volume of material, which took many years to process. The TAMBIS ontology, being built to query databases, extracted a large part of its knowledge from database documentation. The information available to life scientists today is huge, and locating relevant information can be tricky if the scientists can not express their information need in terms of a query phrased with respect to standard vocabularies.

Today, the need for automatic term retrieval using Natural Language Processing (NLP) techniques (including searching and browsing) arises. And it is considered as the future work on FWOnt.

Expert's interviews, text analyses, faculty study groups and seminars, summaries of scientifically-based strategies, face-to-face workshops and courses, e-Learning opportunities, using professional libraries and access to online content and resources are considered as different ways for data extraction and knowledge acquisition [52]. What distinguishes a KR specialist from others is the knowledge of what information among these several resources is relevant and what information is not. In fact, the degree of our granularities can specify the level of information that we need.

The information for FWOnt is replicated at different data resources such as scientific literature, biological databases, and other existing bio-ontologies. First of all one should choose which sources to retrieve the information from. Selecting the appropriate source highly depends on the content and capabilities of the source, as well as the correctness of the source. Some sites publish information as soon as it is available and some are more careful on the quality and accuracy of the data.

Biological research needs tight controls on the quality and accuracy of data (especially when we reuse results of others) because in some cases invalid results can harm people's health or can have bad effects on the environment. So, with the increasing complexity and uncertainty of available data sources, we should more care about consistency and reliability of the data sources and filter out the unreliable ones.

Currently several researchers are investigating to increase the quality, reliability, accuracy, understandability and productivity of research efforts [143, 144]

6.3.1. Knowledge Resources

The following resources along with additional distributed literature are the major data sources for FWOnt:

- NCBI taxonomy database [12]: contains the names of all organisms including fungi that are represented in the genetic databases with at least one nucleotide or protein sequence.
- NEWT [13]: the taxonomy database maintained by the SwissProt. It is used in conjunction with the NCBI for extracting data for fungi concepts and instances.
- BRENDA [14]: a database of fungal enzymes and enzyme features. It gives a representative overview of enzyme characteristics, attributes, and properties.
- SwissProt [15]: a protein sequence database which provides highly curated annotations, a minimal level of redundancy and a high level of integration with other databases.
- ChEBI [8]: a dictionary of 'small molecular entities'. It encompasses an ontological classification, whereby the relationships between molecular entities or classes of entities and their parents and/or children are specified.
- CAZY (Carbohydrate Active Enzymes Database) [112, 113]: The CAZY database describes the families of structurally-related catalytic and carbohydrate-binding modules (or functional domains) of enzymes that degrade, modify, or create glycosidic bonds.
- Commercial Enzyme Vendors: Companies that retail enzymes provide detailed descriptions of the properties and benefits of their products.

Also the following fungal databases are used to find additional information about the fungal concepts and instances:

- Aspergillus nidulans Database [114]
- CandidaDB [115]
- Cryptoccocus neoformans Database at TIGR [116]
- Fungal Genome Stock Center [117]
- International Rice Blast Genome Consortium [118]
- Magnaporthea grisea Database [119]
- Neurospora crassa Database [120]
- Saccharomyces Genome Database [121]
- Schizosaccharomyces pombe GeneDB [122]

6.3.1.1. BRENDA enzyme database

BRENDA [14] is a database for fungal enzymes and enzymes in general, which is developed to work as a main collection of enzyme functional data available to the community. BRENDA intends to give a representative overview on the characteristics and variability of each enzyme. Also it has references to the primary literature for detailed information that is useful for the annotation. The data collection is being developed into a metabolic network information system with links to enzyme expression and regulation information.

This database contains a compilation of data on enzyme function, which were manually extracted directly from the primary literature, and then formal and consistency checks were done by computer programs. Each data set on a classified enzyme is checked

manually by at least one biologist and one chemist. BRENDA has useful taxonomic information as well as enzyme characteristics, with various functionalities.

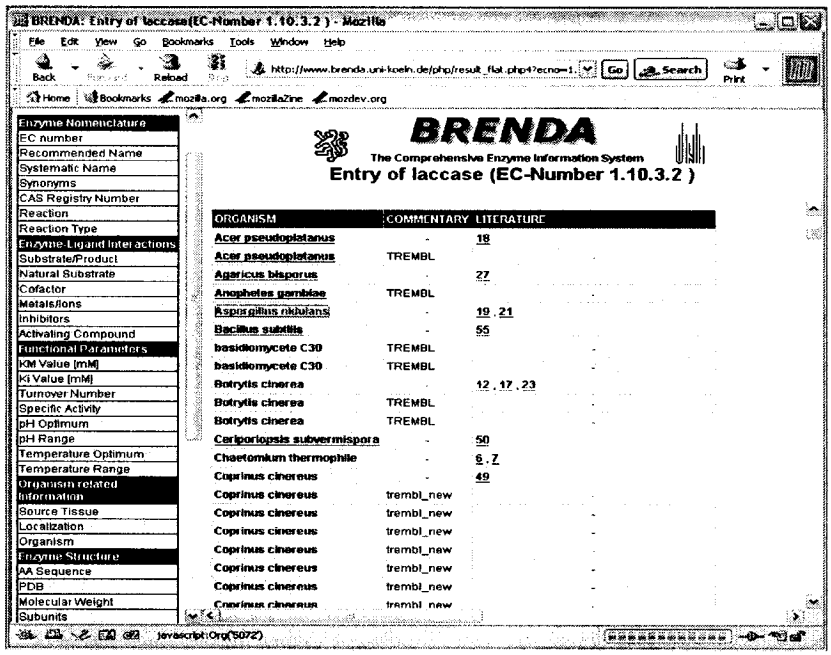


Figure 6-2: Laccase and related organisms from BRENDA.

Different enzymes are considered in FWOnt which can be known as fungal enzyme (the enzyme has been reported to be found in a particular fungi species). By browsing and searching in BRENDA database information about following fungal enzymes are reviewed and extracted:

- Endo-1,4-b-xylanase EC-Number 3.2.1.8
- Laccase EC-Number 1.10.3.2
- Manganese peroxidase EC-Number 1.11.1.13
- Cellulase EC-Number 3.2.1.4
- Protease EC-Number 3.1.2.12
- Pectinase EC-Number 3.2.1.15

- Lipase EC-Number 3.1.1.3
- Arabinase EC-Number 3.2.1.99
- Chitinase EC-Number 3.2.1.14
- Chitosanase EC-Number 3.2.1.132
- Chitin deacetylase EC-Number 3.5.1.41
- Feruloyl esterase EC-Number 3.1.1.73

For each enzyme class one can find corresponding fungi organisms and then look for the lineage for related fungi organisms to compare different fungi and find species with common lineages. This can be done by extracting data from available databases with focus on the NEWT and NCBI taxonomy databases.

6.3.1.2. NEWT

NEWT [13] is a taxonomy database maintained by the SwissProt group. It integrates taxonomy data compiled in the NCBI database and data specific to the SwissProt protein knowledgebase. NEWT is updated daily.

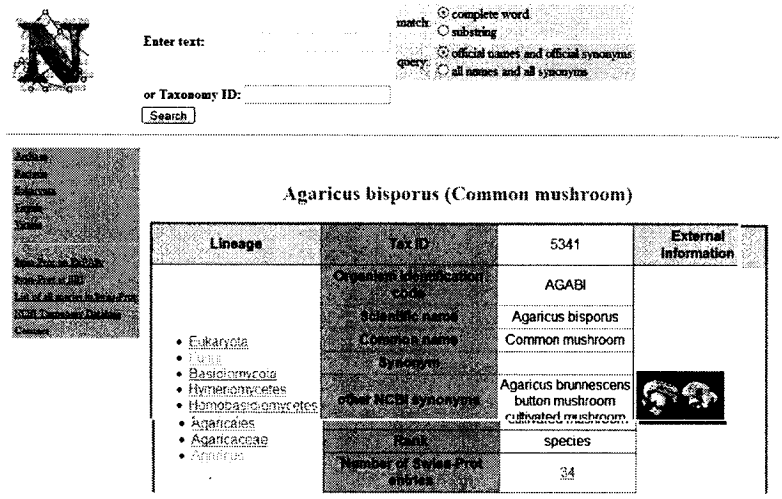


Figure 6-3: Agaricus bisporus in NEWT

6.3.1.3. NCBI taxonomy database:

It Contains the names of all organisms that are represented in the genetic databases with at least one nucleotide or protein sequence. One can browse the taxonomic structure or retrieve sequence data for a particular group of organisms.

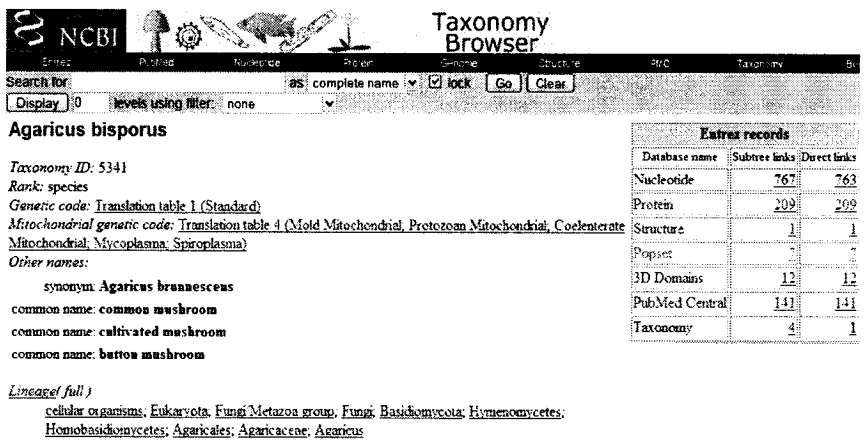


Figure 6-4: Agaricus bisporus in NCBI taxonomy browser

For each fungal enzyme there are some attributes that are divided in three categories: functional parameters, molecular properties and enzyme-ligand interactions. These properties are used in FWOnt under "Enzyme Specification" concept.

Functional Parameters	Molecular Properties	Enzyme-Ligand Interactions
KM Value [mM]	pH Stability	Substrate/Product
Ki Value [mM]	Temperature Stability	Natural Substrate
Turnover Number	General Stability	Cofactor
Specific Activity	Organic Solvent Stability	Metals/Ions
pH Optimum	Oxidation Stability	Inhibitors
pH Range	Storage Stability	Activating Compound
Temperature Optimum	Purification	
Temperature Range	Cloned	
	Engineering	
	Renatured	
	Application	

Table 6-1: Fungal Enzyme specification

The common lineage links for all fungi that contain a particular enzyme (highest phylogeny unit that links them all) can be found.

For introducing primary biological concepts such as cells, molecules, genes, and relationships between them, The European Bioinformatics Institute (EBI) presents a brief introduction to molecular biology with emphasis on genomics and bioinformatics [123]. It is prepared for scientists, engineers, computer programmers, or anybody with a background in science, but without a background in biology.

6.4. Implementation

This stage includes conceptualization, integration and encoding. Usually, because ontology development has an iterative nature, one can frequently switch from one stage to another.

6.4.1. Conceptualization

Ontologies are often classified based on their usage [4]:

1. Domain-oriented, which are either domain specific (i.e. E. coli) or domain generalizations (i.e. gene function or ribosome).
2. Task-oriented, which are either task specific (i.e. annotation analysis) or task generalizations. This kind of ontologies also known as "application ontologies";

3. Generic, which capture common high level concepts. It can be useful when trying to re-use an ontology, as it allows concepts to be correctly or more reliably placed. It can also be important when generating or analyzing natural language expressions using an ontology. Generic ontologies are also known as `upper ontologies', `core ontologies' or `reference ontologies'.

The FungalWeb ontology structure is a mixture of all three of these types. It is built in a modular way using a mixture of generic domain, generic task and application ontologies. The modular structure of FWOnt provides facilities for reusing the concepts in other ontologies.

It is important to keep the results of the first step, that of requirements gathering, in mind.

At the conceptualization stage, the key concepts that exist in the domain of FWOnt, their properties and the relationships that hold between them, are identified.

6.4.1.1. Concepts

The FungalWeb Ontology has two types of concepts:

1. **Primitive concepts:** are those which only have necessary conditions (in terms of their properties) for membership in a class. For example, a basidiomycota is fungi with a fruity body, so all basidiomycota fungi must have a fruity body, but there could be other things that have a fruity body that are not fungi.

2. **Defined concepts:** are those whose description is both necessary and sufficient for a thing to be a member of the class. For example, eukaryotic cells are kinds of cells that have a nucleus. Not only does every eukaryotic cell have a nucleus, every nucleus containing cell is eukaryotic.

The current version of the FungalWeb Ontology does not have concepts with only sufficient conditions. This is partly because limitation in tools (Protege does not allow defining concept with only sufficient condition).

6.4.1.2. Relationships

Another step in conceptualization is specifying different relationships between different concepts and instances based on the ontology goal. There are two common types of relationships in ontologies:

1. Taxonomic relationships: organizes concepts into sub- super-concept tree structures. The most common forms of these are:

- 'Is-a' relationship, i.e., an "Enzyme" is a "Protein", which in turn is a "macromolecule". The is-a relationship forms a subsumption taxonomy.
- 'Part-of' relationships describe individuals of concepts that are part of individuals of other concepts. In FWOnt, we did not define Part-of relationship explicitly. Instead to identify for example parts of Macro_molecule we defined a concept named Macro_molecule_part then we put all parts of concept Macro_molecule such as DNA_part, Protein_part and RNA_part as subclasses of this class. Therefore each subclass again has "is-a" relationship with its supper class.

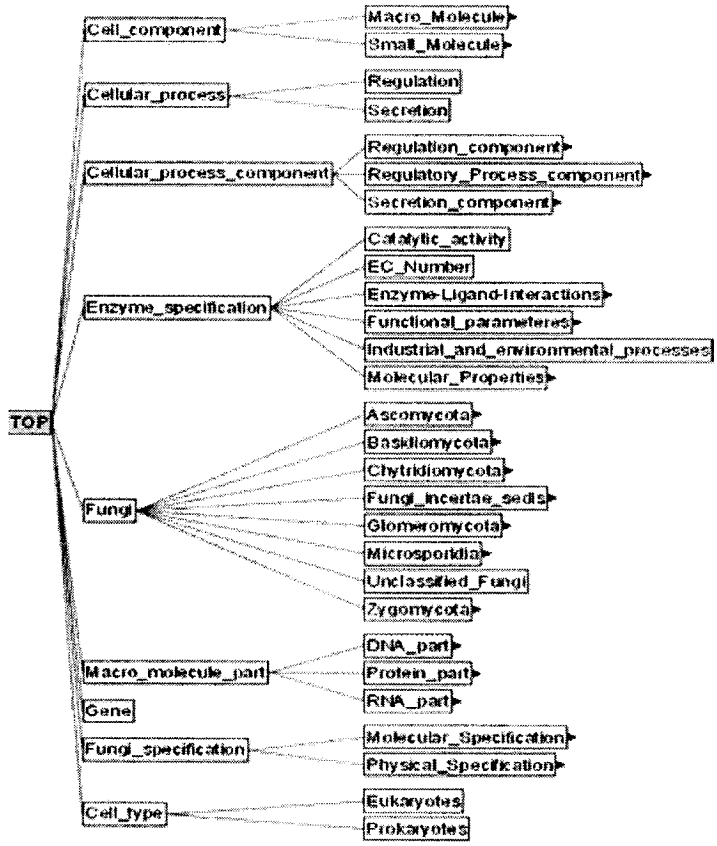


Figure 6-5: Taxonomic relationships for FWOnt top concepts

2. Associative relationships: relate individuals of concepts. Commonly found examples include the following:

- Nominative relationships describe the names of concepts, i.e., Enzyme Has_EC_number EC_number and Fungi Has_name Fungi_name.
- Locative relationships describe the location of individuals of concepts, i.e., Gene Has_Location Cell.
- Associative relationships that represent, for example, functions and processes. i.e., Enzyme Acts_On_Substrate Substrate, Gene Regulated_By Promoter.

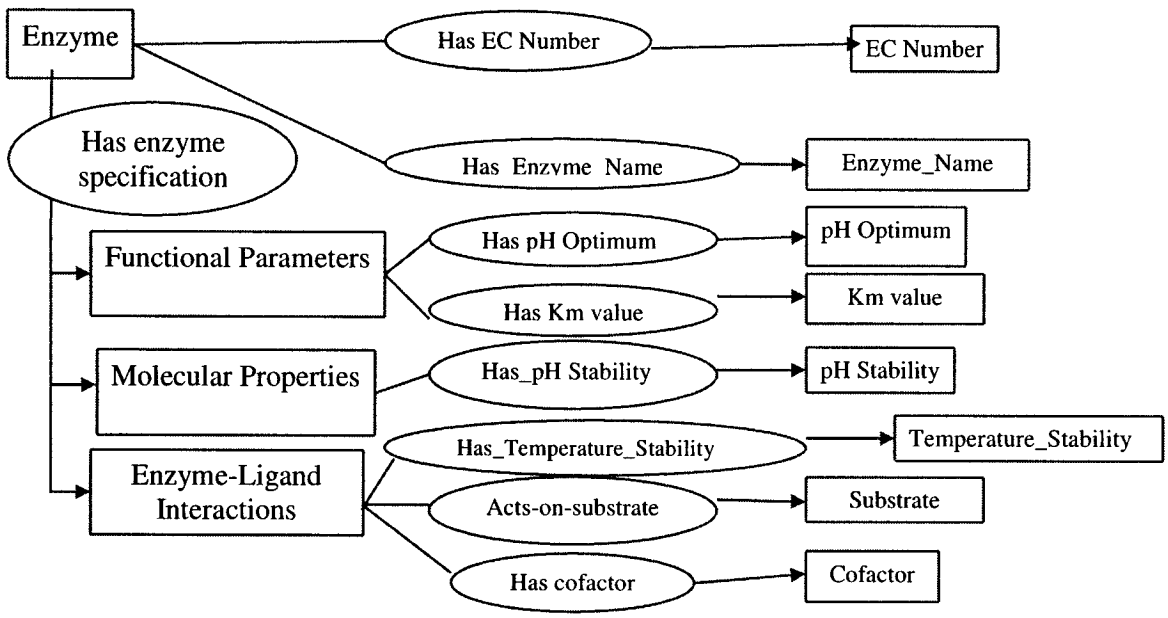


Figure 6-6: Associative relationships for Enzyme specification concepts.

The relations, like concepts, can have sub-relations, i.e., Has_Functional_Parameters is divided into Has_Temperature_Optimum, Has_Temperature_Range, Has_Ki_Value and Has_pH_Range. Relations also have properties that capture further knowledge about the relationships between individuals of concepts. These include, but are not restricted to:

- Whether it is universally necessary that a relationship must hold on all individuals of a concept. For example, when describing concept Enzyme, we might want to say that "Enzyme Has_EC_Number EC_Number" holds universally for all Individuals of concept Enzyme.
- Whether a relationship can optionally hold on individuals of a concept, for example, we might want to describe that "Enzyme Has_Cofactor Cofactor" only describes the possibility that enzymes have a cofactor, as not all enzymes do have a cofactor [90].
- The cardinality of the relationship, i.e., particular EC_Number is the EC number of only one Enzyme, but one particular Enzyme may has been reported to be found in many fungi.

Defining more complex relationships can help one to define more complex queries which help for inferences, i.e. as shown in Figure 6-8 when enzyme is related to fungi by x.y.z and to substrate by x.y'.z' then fungi and substrate can be related in some way.

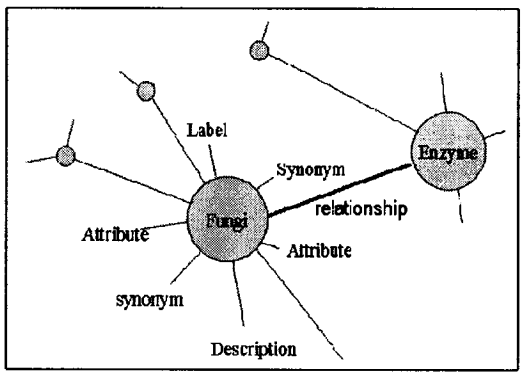


Figure 6-7: Schematic conceptual model for concept fungi in FWOnt

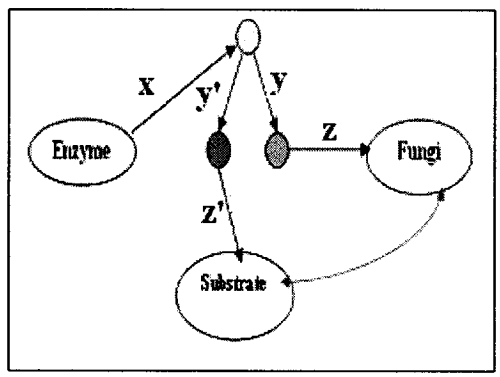


Figure 6-8: Enzyme –Fungi-substrate relation

Axioms are used to constrain values for individuals of classes. In this sense the properties of relations are kinds of axioms. Sometimes a constraint is necessary for an application

and sometimes it is not needed for another, this simply changes what knowledge is captured or how it is captured.

6.4.1.3. Instances

An instance or sometimes called 'individual' or 'particular' is a particular realization ("object") of a class, representing a real entity in the real world. Each instance of a class can have its own values for the properties. For example:

"Neurospora_crassa" is an instance of concept "Neurospora"

In the FWOnt most of the instances are fungal and enzyme fungal species. For fungal species exactly what different mycologists consider being a species can vary widely, and there are different approaches for delineating them. In mycology the distinction between a class and an instance is not always easy, and this can create confusion in genetic studies [126]. Deciding whether something is a concept of an instance is difficult, and often depends on the application [9]. For example, 'Oxidoreductases' is a concept and 'Laccase' is an instance of this concept. It could be argued that Laccase is a concept by itself representing the different instances of Laccase. This is a well known and open problem in knowledge representation research.

By instantiating the ontology we obtain a knowledgebase which can be used in a Semantic Web system for query answering and problem solving.

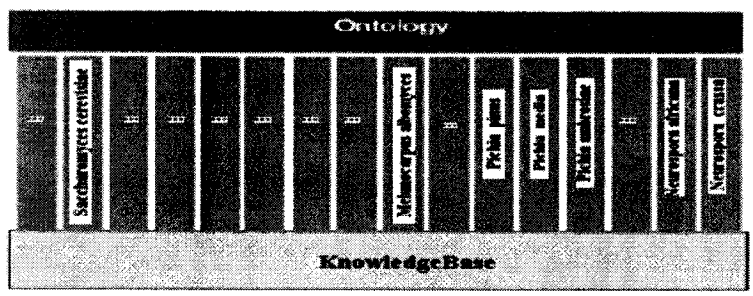


Figure 6-9: Instantiate the ontology to obtain the knowledgebase.

6.4.2. Integration

To have a Semantic Web system which allows computers to combine and infer implicit knowledge from different ontologies in a particular domain of interest, these ontologies should be linked and related to each other. The primary goal of ontologies is knowledge sharing, so ontologies are often reused and distributed in a large scale. By merging and reusing the ontologies, the system would be more effective for information retrieval, query answering and problem solving.

FWOnt is an integrated ontology which is using and re-using different accessible domain specific ontologies. By reusing concepts from other generic ontologies, a well defined concept will obtained which is easier to share.

The reuse of existing ontologies and adapting them for a particular purpose is often not possible without considerable effort [132].

The vocabularies in FWOnt come from several online databases and a few existing bio-ontologies. To reuse these ontologies together, they have to be combined in some way. This can be done by *integrating* the ontologies, which means that they are *merged* into a new ontology, or the ontologies can be kept separate. In both cases, the ontologies have to be *aligned*, which means that they have to be brought into mutual agreement. Also ontologies can be *mapped* by relating similar concepts or relations from different sources to each other by an equivalence relation [133].

The integration in the FWOnt design is done at two levels: Data and Semantic Integration.

6.4.2.1. Data Integration: at this stage data is extracted from different sources, with different data format and then the extracted data must be normalized into a consistent syntactic representation and semantic frame of reference.

6.4.2.2. Semantic Integration: for identifying the relevant data elements of individual sources for particular applications, and the semantic relationships among these data elements, in order to map heterogeneous data fragments into a common frame of reference and enabling the correct mix of data from different sources.

Most bioinformatics data sources require significant amounts of effort on the semantic aspects of data integration. Unfortunately, there is little consensus on many terms. Integrating these individual ontologies is known to be hard [140, 141], but their existence allows some meaningful information located in different parts of ontologies to be shared, with requiring a human expert to interpret every piece of information. Of course it would be desirable to do the integration automatically without human interpretation but in the meantime, partial ontology merging with human control can be very useful.

In FWOnt the data and semantic integration consists of the iteration of the following steps [134].

1. Find the places in the ontologies where they overlap.
2. Relate concepts that are semantically close via equivalence and subsumption relations (aligning).
3. Check the consistency, coherency and non-redundancy of the result.

Aligning two ontologies implies changes to at least one of them. Especially alignment of concepts is difficult because we need to understand the meaning of them.

If the ontologies are not represented in the same language, a *translation* is often required.

6.4.2.3. Issues in ontology integration

One of the most common issues in ontology integration is mismatching between ontologies in language and model levels [133].

- Language level mismatches (mismatches in mechanisms to define classes and relations): Ontologies written in different ontology languages, different syntaxes, logical notation, language expressivity and semantics of primitives (same name, different interpretations)
- Ontology or model level mismatches (difference in the ways that the domains are modeled) : Two types of mismatches at the model level are exist:

  - Conceptualization mismatch: difference in the way a domain is interpreted. (Different ontological concepts, different relations between those concepts)
  - Explication mismatch: difference in the way the conceptualization is specified.

  Figure 6-10 shows other problems in the ontology integration task.

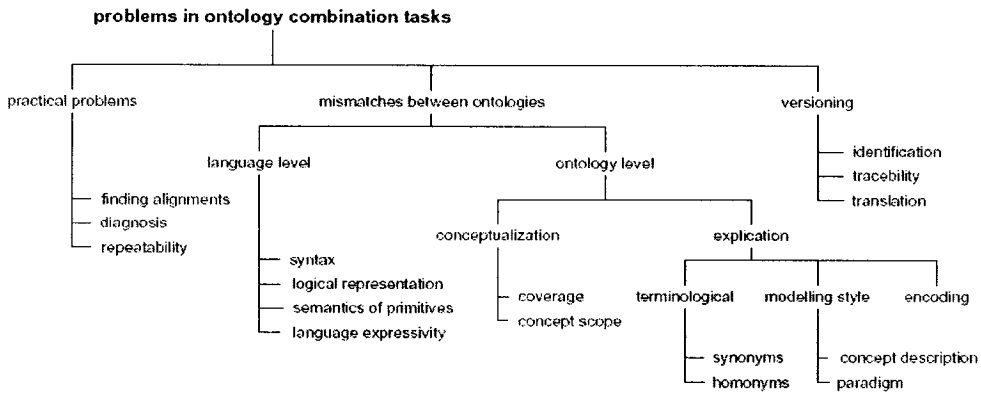


Figure 6-10: The issues that are involved in ontology integration [133].

6.4.2.4. Ontology merging tools

There are some available tools for ontology merging and combination such as COBra [136], Prompt [137], OntoMorph [138] and Chimaera [139]. However, they have some capability for finding redundant concepts and to compare concepts, but considerable manual works still needs to be done to create a consistent ontology by merging different ontologies.

Prompt is used for the FungalWeb ontology integration. For more information one can refer to a survey about existing ontology merging tools [133].

PROMPT [137]

PROMPT is implemented as a plug-in for Protege and aimed for automated ontology merging and alignment. In the merging process, PROMPT helps one by setting the preferred ontology, determining conflicts, identifying and proposing conflict-resolution strategies and providing feedback via three sub-tabs (Suggestions Tab, Conflicts Tab and New Operations Tab).

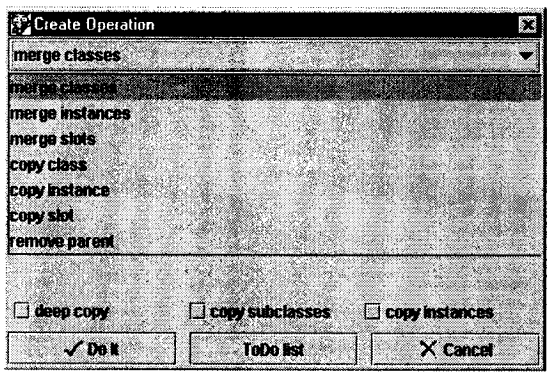


Figure 6-11: PROMPT merge mode operations

The merge mode allows one to take two existing individual Protege ontologies with different concepts, names or structures and create a new ontology which combines them, while merging similar or identical concepts.

By considering some overlaps between the FungalWeb core ontology, GO and TAMBIS, it seems one can merge these independent ontologies to obtain an integrated ontology.

PROMPT allows partial merging by helping to determine which concepts need to be merged by considering completeness and consistency.

Suggestions Tab: presents a list of suggested operations for:

Merge two similar classes from the source ontologies and copy a class from (usually a class that has no counterpart in the other ontology) one of the source ontologies. The merge suggestions are based on the linguistic and structural similarity of the class names. Some suggestions can be created, edited and removed based on the granularity and the knowledge from the source ontologies.

Conflicts Tab: shows a list of conflicts in the merged ontology that the operations have caused and proposes possible solutions to the conflict. The conflicts are such as name conflicts, dangling references, redundancy in the class hierarchy and inconsistencies.

New Operations Tab: is a working area that shows the source ontologies and allows one to create new operations.

One way for ontology integration and sharing is using common vocabularies and terms that come from standard models like UMLS (Unified Medical Language System) [62]. This approach can be a fast way, but as the model grows it is going to be hard to maintain.

Logic can be used for integrating or merging ontologies. By using intersection, union and difference, respectively, a subset ontology, joint ontology and distinct ontology would be obtained.

FWOnt reused some concepts from the Gene Ontology and TAMBIS to support knowledge sharing. Also, a future need to import some concepts from EcoCyc for defining gene structure and TRANSFAC [124] for gene regulation is predictable. As mentioned before, one of issues in ontology merging is the potential mismatch between ontologies at the language level. So, the need for translation and conversion of ontologies from one language to another language might arise. The FWOnt core is implemented in OWL-DL, fortunately TAMBIS is now available in OWL-DL format and we have access to DAML+OIL version of GO (from the GONG project (Gene ontology new generation)). So, we used a freely available DAML+OIL to OWL converter [142] to convert the Gene ontology DAML+OIL version to OWL. Because we want to have a OWL-DL file, we can not use some OWL special properties like owl:oneOf with rdf:Class. The domain of owl:oneOf is owl:Class. Though in OWL Full, rdf:Class and owl:Class are equivalent. So we did some manually adjustment to obtain an OWL-DL merged ontology.

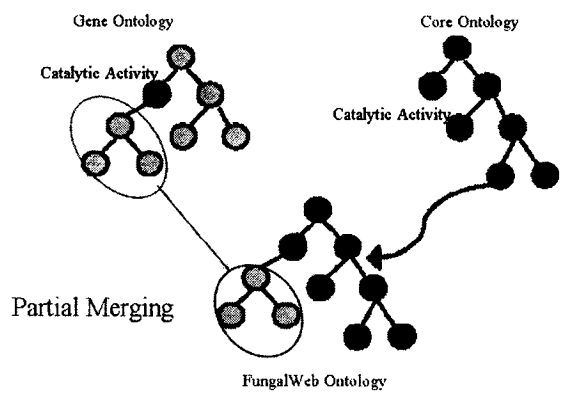


Figure 6-12: Partial merging of GO and the FungalWb core ontology.

The concept catalytic activity is defined in the Gene Ontology as subclass of molecular_function.

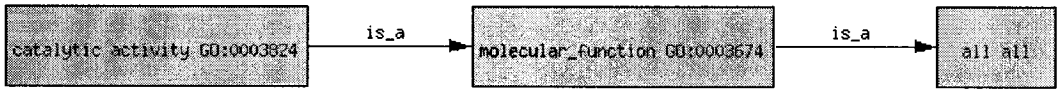


Figure 6-13: Catalytic activity in GO.

Also, in the FungalWeb core ontology all enzymes are classified based on their catalytic activity.

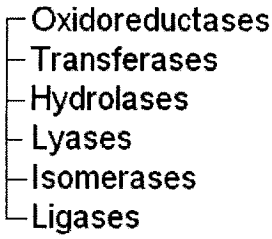


Figure 6-14: Enzyme classification basedon catalytic activity.

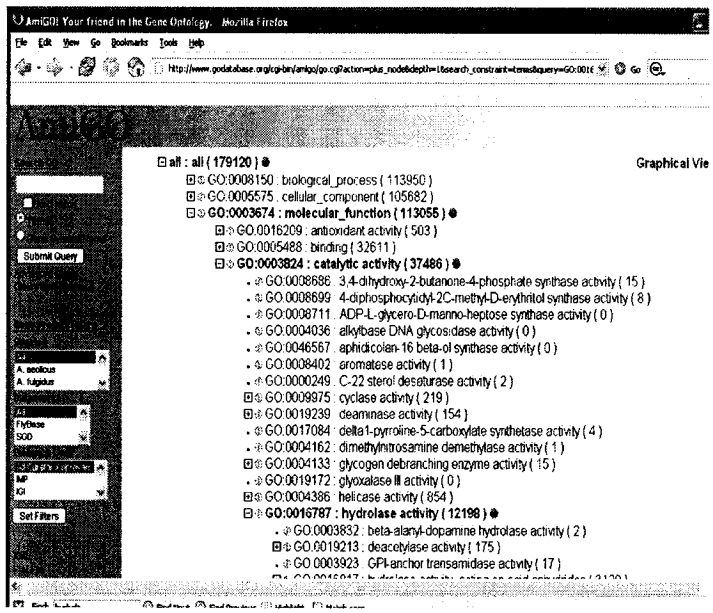


Figure 6-15: Browsing the concept catalytic activity using AmiGO.

So as shown in figure 6-15, the Gene Ontology is partially merged with the FWOnt core ontology.

When one tries to use PROMPT for merging the core ontology with, for example, GO or TAMBIS, it first generates an initial list of suggestions which is based on the similarities

in class names. For example, it proposes to merge the two classes from the FWOnt and the Gene Ontology (or TAMBIS).

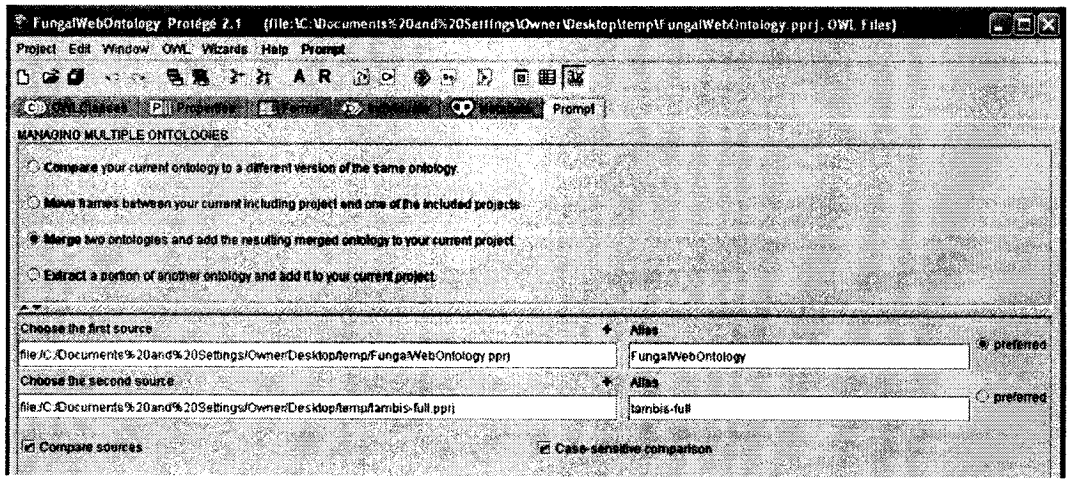


Figure 6-16: Choosing ontologies for merging operation in PROMPT.

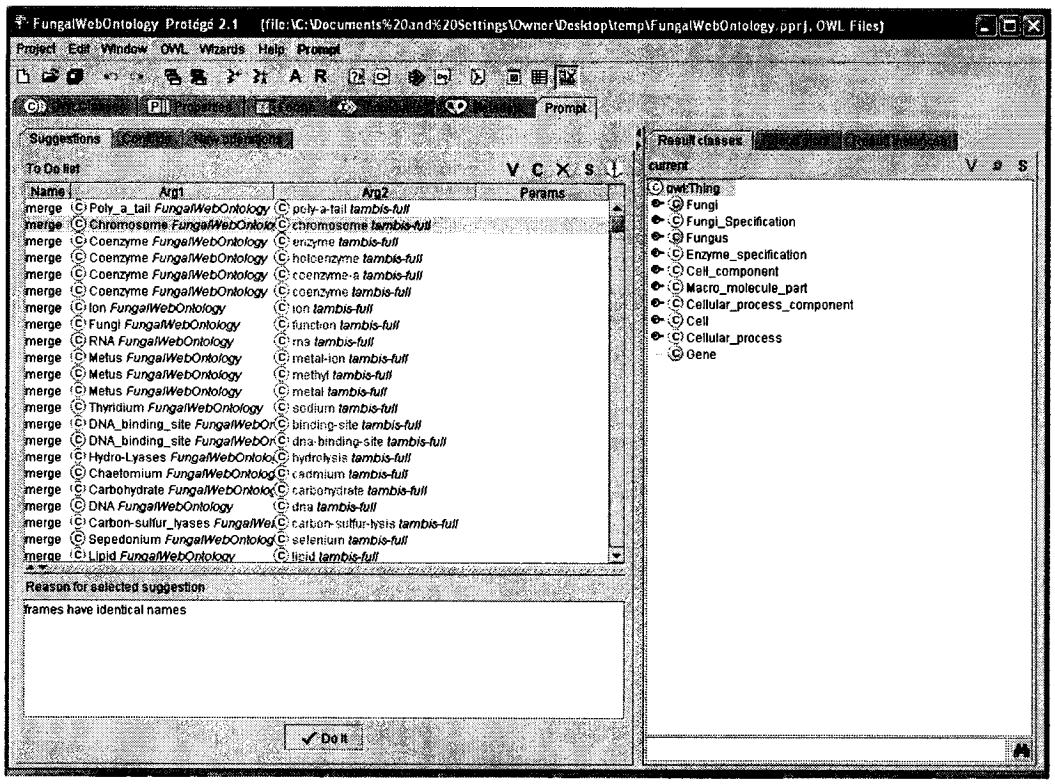


Figure 6-17: Comparing Ontologies based on similarities using PROMPT.

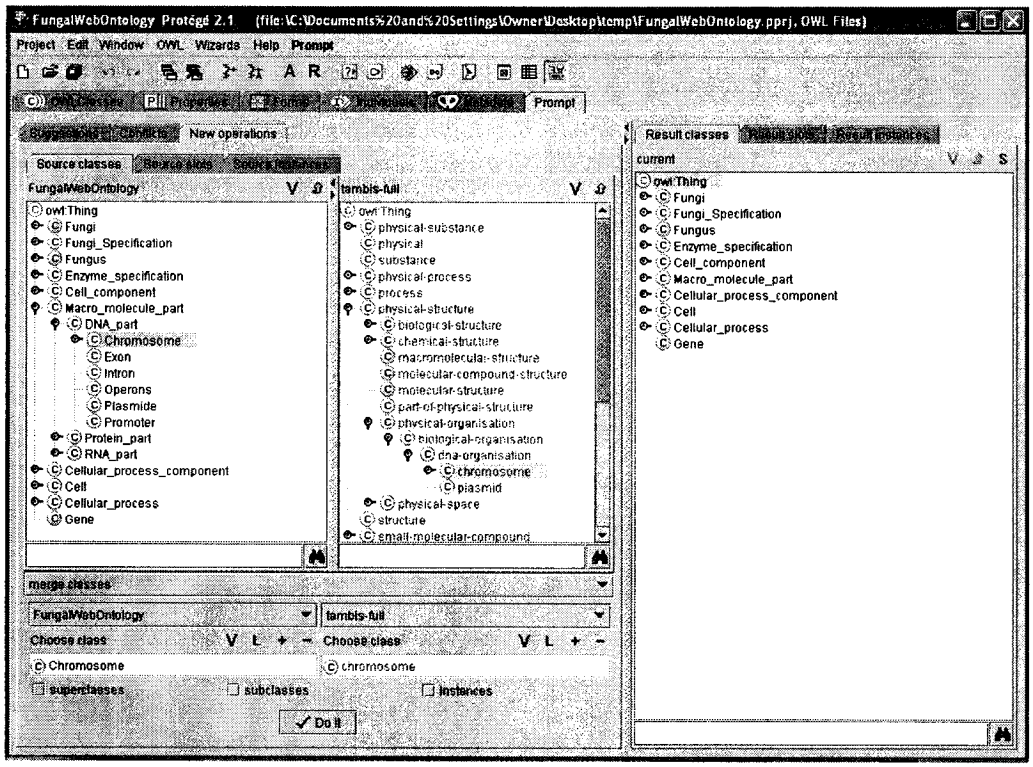


Figure 6-18: Finding common concepts in two ontologies using PROMPT.

No matter what method is picked, one needs to evaluate the integrated ontology after merging. As mentioned, the mismatches [135] between different individual ontologies are the main obstacle for ontology integration. The different ontologies are using different languages and syntax, different level of expressiveness, different platform and different tools, and also different ontologies may have a different scope, granularity, paradigm, or modeling style [133].

Some issues in ontology integration are similar with database integration. However, the data models of ontologies are more complex and they incorporate much more semantics [135]. Also databases usually have only one data model, but ontologies can be built based on several models and applications.

The FungalWeb Ontology also provides the facility to connect with chemical databases such as ChEBI. Figure 6-19, shows an enzyme substrate described in the context of the chemical dictionary of small molecular entities (ChEBI).

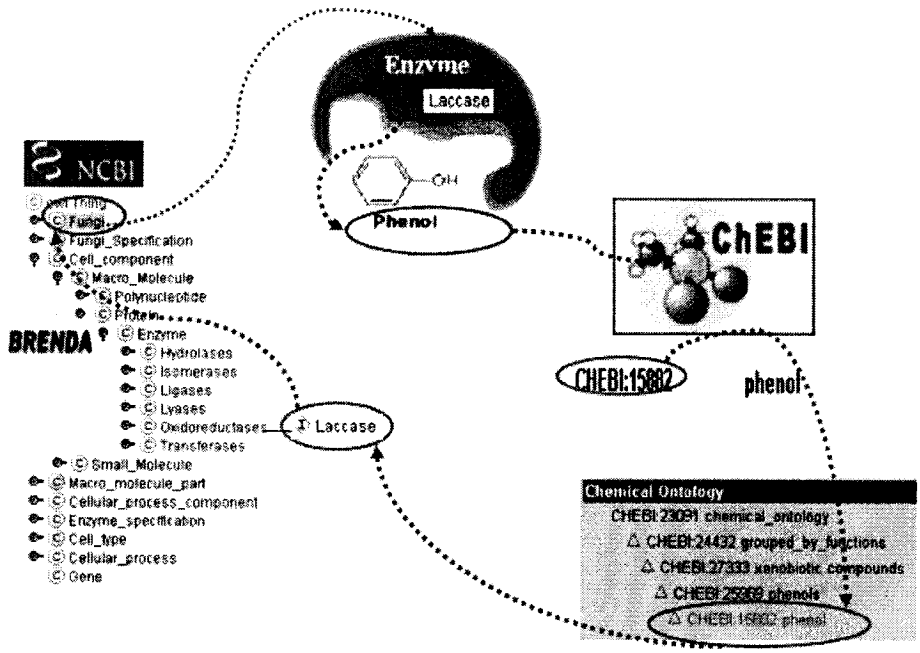


Figure 6-19: Communicating with ChEBI through enzyme substrate.

ChEBI (Chemical Entities of Biological Interest) is a dictionary of 'small molecular entities'. It contains any distinct atom, molecule, ion, ion pair, radical, radical ion, complex, conformer, etc., identifiable as a separately distinguishable entity.

ChEBI also encompasses an ontological classification, whereby the relationships between molecular entities or classes of entities and their parents and/or children are specified.

As shown in figure 6-19 'Laccase' as an enzyme acts on a substrate called 'phenol' which has an entry in ChEBI with a particular ChEBI-ID. By using this ID, FWOnt can address a place in the chemical ontology (handled by ChEBI) for the substrate.

6.4.3. Encoding

The aim of encoding is to represent the conceptualization in a formal language such as frames, logic or object models. FWOnt uses OWL-DL as the ontology implementation language and also Protege as the knowledge representation editor (with OWL plug-in) [131].

6.4.3.1. Using Protege with the OWL plugin

Protege with the OWL plug-in can be used to build OWL ontologies (which are a collection of classes, properties, and individuals) and maintain ontology consistency with a description logic classifier like Racer. It can be used to link existing Web resources such as biomedical articles and images into a semantic web.

After publishing the ontology on the Web, other OWL ontologies, resources, agents, and services can link to this file and use our ontology's concepts.

Classes (Concepts): The taxonomy (tree of concepts) is defined as the base of the FungalWeb Ontology. One can assign values from various resources to a concept which can be used for querying and annotation. FWOnt has its own defined annotation properties and sometimes we tried to reuse existing annotations. These annotation properties do not have any formal meaning for the OWL reasoners such as RACER, but they are important for the ontology maintenance and might be used by other tools (for example nRQL supports queries on annotation properties).

Properties: Defined associative relationships between the FWOnt classes are shown in figure 6-21. Protege provides access to those properties that could be used by the instances of the current class.

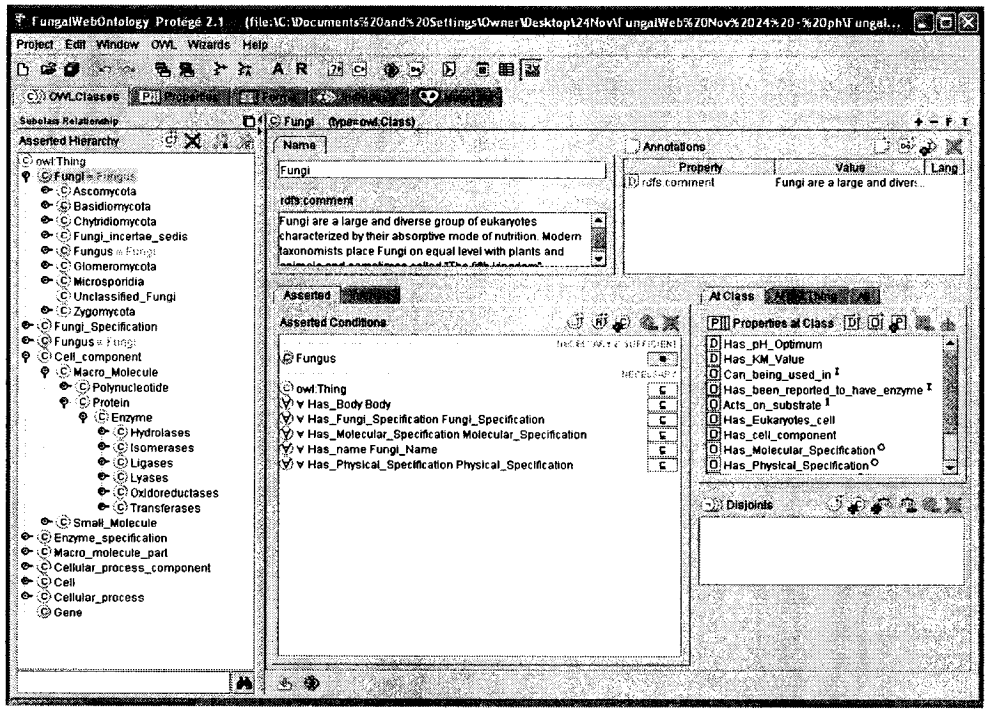


Figure 6-20: Concept definition using Protege.

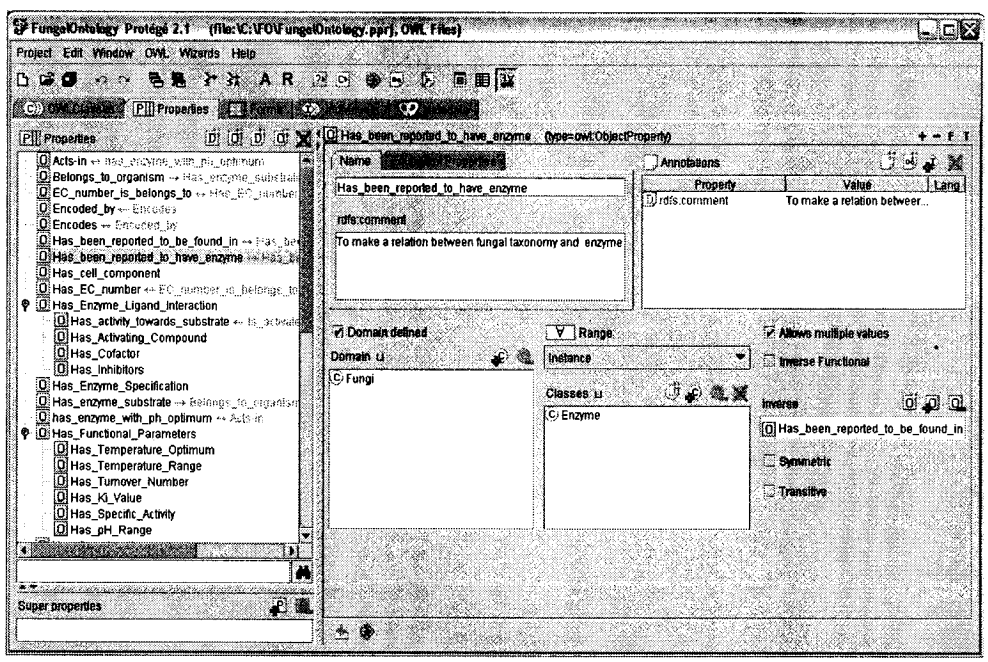


Figure 6-21: The OWL property form in Protege.

The property form provides a metadata area in the upper part, displaying the property's name, annotations, and so on, similar to the presentation in the class form.

The property can be a datatype property with primitive values, or an object property with references to other classes. Object properties can store references to individuals or classes from the ontology. For example, the object property Has_Enzyme_Specification can only take instances of Enzyme_Specification as values.

In FWOnt property characteristics are defined for both datatype properties and object properties such as whether the property is symmetric or transitive. Symmetric properties describe bidirectional relationships (if A is related to B then B is also related to A). The property R is transitive if one can conclude from that if A is related to B by R and B is related to C by R, then A is also related to C by R [131]. Part-whole relationships such as Has_DNA_Part and Has_Protein_Part are usually considered to be transitive.

Also "domain" and "range" are defined to restrict a property's domain and range in FWOnt. For instance, the domain for the property "Has_been_reported_to_have_enzyme" is limited to "fungi" and the range to "Enzyme" (Figure 6-22). Domain restriction may slow down the reasoning processes and also reduce the flexibility, so many ontology developers prefer to leave this part blank (then the property can be used for instances of any class).

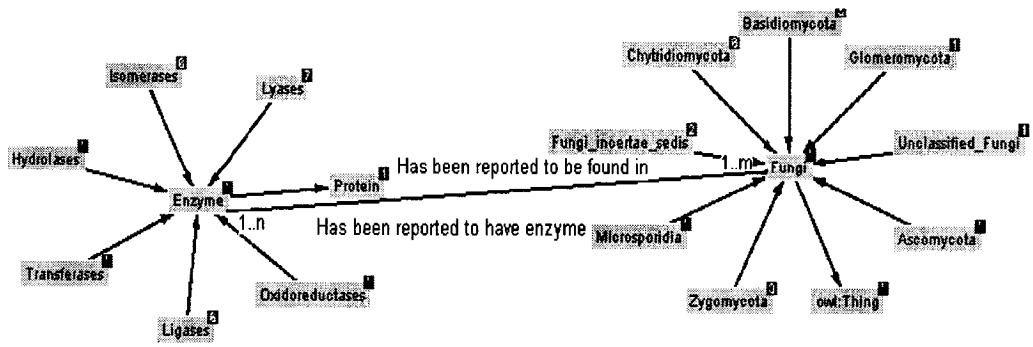


Figure 6-22: Fungi-Enzyme Relationships.

6.4.3.2 Using Description Logics

OWL has its theoretical foundation in description logics [127]. In description logics, a class describes a set of individuals. The concept corresponding to the set of all individuals in a domain is usually called Top or Thing. Whenever the set of all individuals of a class B must be a subset of the set of all individuals of a class A, B is said to be a subclass of A (Subsumption relationship). B is also said to be a kind of A. All classes are sub-concepts of Top. The "is-a" relation forms the basis of the taxonomy.

Superclasses define "necessary conditions" for class membership and conversely, subclasses define sufficient conditions for class membership. For example, being "Fungi" is a necessary condition for being a "Basidiomycota". It means, in order to be an instance of Basidiomycota, an individual has to be an instance of Fungi. Conversely, being a "Basidiomycota" is a sufficient condition for being "Fungi": every instance of "Basidiomycota" is also an instance of Fungi (but there may be other instances of "Fungi" that are not instances of "Basidiomycota").

In FWOnt one can see the inheritance between properties, i.e., if every "Basidiomycota" "Has_Fruit_Body" and "Hymenomycetes" is a subclass of Basidiomycota, then every "Hymenomycetes" also has property "Has_Fruit_Body".

Using OWL-DL allows expressing conditions on the classes based on property restrictions and other expressions. The following table shows the syntax used for OWL expressions.

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer
complementOf	$\neg C$	¬Male
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}
allValuesFrom	$\forall P.C$	∀hasChild.Doctor
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer
maxCardinality	$\leqslant nP$	⩽1hasChild
minCardinality	$\geqslant nP$	⩾2hasChild

Table 6-2: The syntaxes for OWL expressions in comparison with DL syntax [111].

The key point here is to understand that an expression involving a property and its range such as "∃ property.Concept" or "∀ property.Concept" represents a set of individuals, and therefore can be interpreted as a concept.

The logical symbols used by the OWL Plug-in are widely used in the description logics community [1]. They allow displaying even complex class expressions.

The formal definitions of the OWL primitives can be exploited by reasoners such as Racer. They compute the inheritance relationships between the classes based on their logical definitions. This reasoning support has shown to be a very valuable feature during ontology design, particularly in biomedical domains [128], [129]. Ontology designers can periodically invoke a reasoner to see whether the logical class definitions meet the expectations, and to make sure that no inconsistencies arise.

6.4.3.3. Statistics

Table 6-3 shows some statistics about current state of FungalWeb Ontology. These numbers are still growing.

Statistics	
Number of Concepts	3616
Concepts with necessary condition	3603
Concepts with necessary and sufficient condition	8
Concepts without any condition	5
Number of instances	11163
Object properties (Roles)	142
Datatype properties (Roles)	7

Table 6-3: Statistics for FungalWeb Ontology.

6.5. Ontology navigation

The user can browse the ontology in a hierarchical way from the top-level terms and down to more detailed ones.

We used the following tools for ontology navigation.

6.5.1. OntoXPL[145] : An ontology information exploration and navigation tool based on the web server tomcat. Standard HTML browsers can be used to interact with ONTOXPL. It is intended to complement existing ontology editors and does not offer any editing support. ONTOXPL uses the OWL-DL reasoner Racer via its extensive query interface in order to support the intelligent exploration of OWL ontologies.

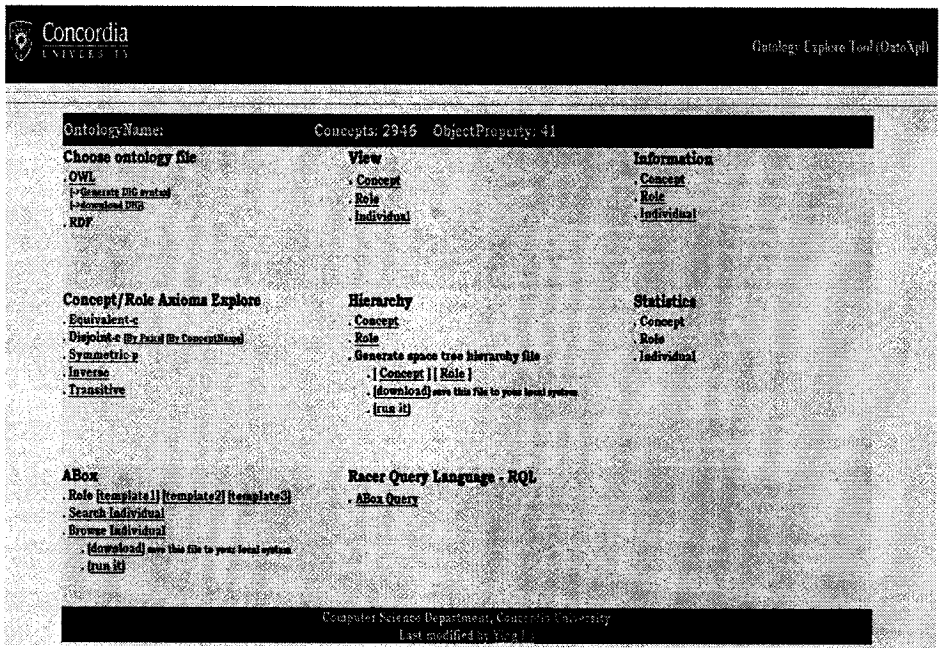


Figure 6-23: FWOnt in OntoXPL.

6.5.2. GrOWL [146]: A visualization and editing tool for OWL-DL ontologies which aims to reconcile OWL with the old semantic network philosophy by providing

a set of graphic idioms that cover almost every OWL construct. It is implemented as a java applet, stand alone editor and uses the WonderWeb OWL API [177].

Growl has capabilities to navigate and visually edit of large OWL ontologies and provides a graphic representation for OWL constructs and common DL expressions. It offers facilities to separately view TBox, ABox, RBox (axioms about properties, e.g, that property P is a subProperty of R), and the class hierarchy.

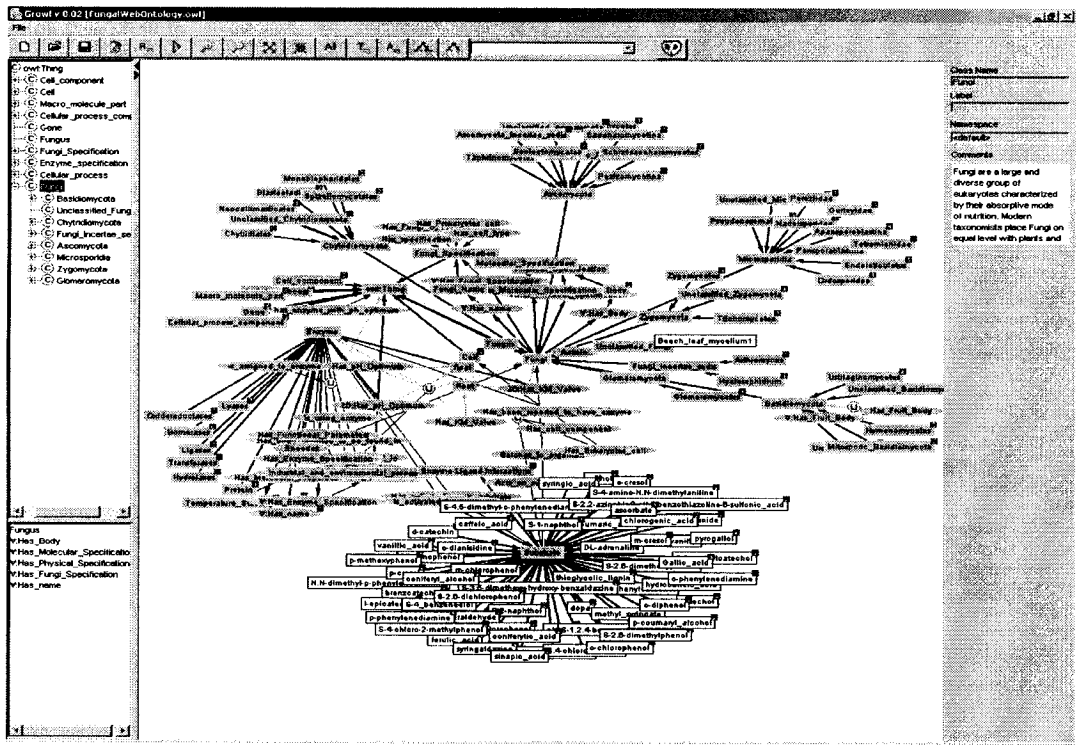


Figure 6-24: Fungi-Enzyme-Substrate in GrOWL.

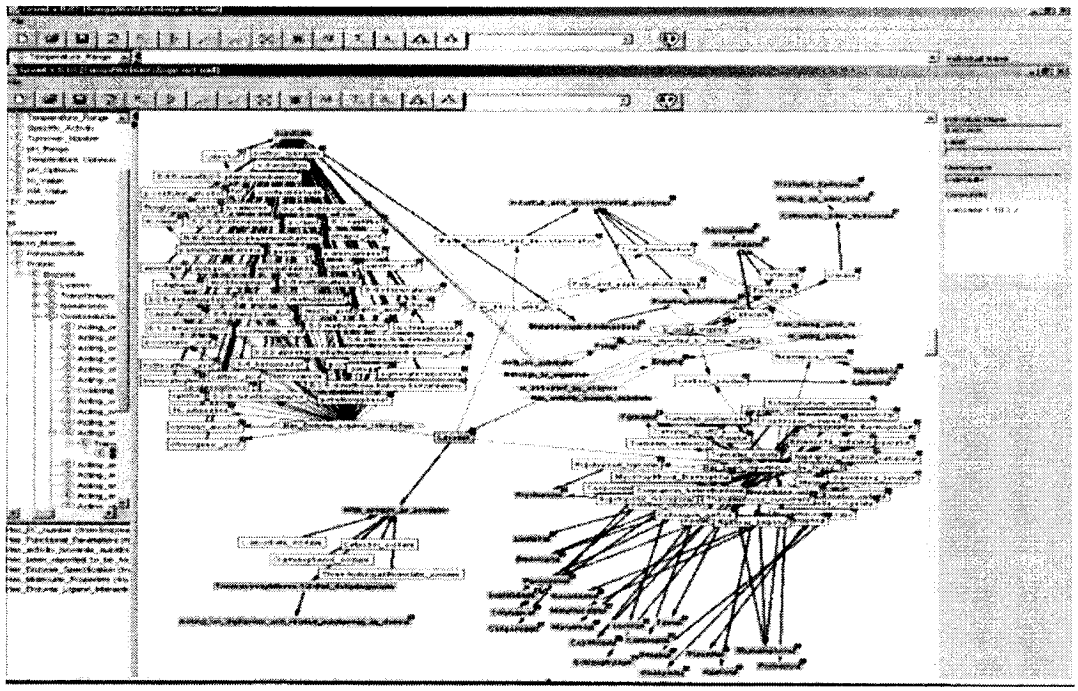


Figure 6-25: Class hierarchy and instance Laccase in GrOWL.

6.6. Evaluation

We do not have a uniform pattern for judging all ontologies. Ontoligists usually check the appropriateness of an ontology for its intended application.

The evaluation is done pragmatically, by assessing the ontology to satisfy the requirements of our application, including determining the consistency (logically and semantically), completeness and conciseness in appropriate granularity [125].

Logical consistency is checked automatically by KR editors and semantic consistency is assisted by the DL reasoner like Racer by classification or misclassification.

6.6.1. Classification and Consistency Checking

Description logics make the open world assumption, which is what is not said denotes a lack of knowledge whereas in other contexts such as databases, what is not said is assumed to be false. A direct consequence is that if we do not say explicitly that two classes such as "Macro_molecule" and "Small_molecule" are disjoint, and then it is perfectly valid for them to have individuals in common.

One of the major strengths of description logics languages like OWL is their support for reasoning. The OWL Plug-in can interact with any reasoner that supports the standard DIG interface, such as Racer [152]. During ontology design, the most interesting reasoning capabilities from these tools are classification and consistency checking.

Classification is used to infer relationships between classes from their formal definitions. A classifier takes a class hierarchy and returns a new class hierarchy, which is logically

inferred from the input hierarchy. For example, Concept "Gene" is defined in the FungalWeb ontology as:

$$DNA\_part \sqcap ((\exists\ Part\_of\ (Chromosome \sqcup Plasmide)) \sqcap (\forall\ Part\_of\ (Chromosome \sqcup Plasmide)))$$

After classifying the ontology by the reasoner it appears as a direct child of the DNA_part. This reasoning capability associated with description logic is of particular importance because it allows the user to provide intensional definitions for the classes. The relationships become consequences of these definitions, and allow constraints inheritance.

By using OWL, a DL classifier can automatically place the concepts in their correct position. This feature is very important in the domain of bioinformatics, with its deeply nested hierarchies and multi-relationships between every part of the taxonomy [127, 130].

Also by using Racer we can detect logical inconsistencies within the ontology. For example we can say that class "Pezizomycotina" is inconsistent, if it is both an "Ascomycota" and a "Basidiomycota". Since the last two concepts are defined to be disjoint, the reasoner reports that no individual can be an instance of this class.

An important issue with reasoning in OWL is that many reasoners are not able to handle the full expressivity of OWL. The OWL specification distinguishes between OWL Full and OWL-DL to indicate which language elements are typically tractable for reasoners. Ontologies that use OWL Full elements such as metaclasses cannot be classified. Since OWL Full ontologies can state anything about anything, available reasoners can not support full reasoning for the complete OWL Full syntax.

To be sure that FWOnt remains in OWL-DL format the WonderWeb ontology validator [175] is frequently used to check any constructs found in the ontology which relate to particular species of OWL.

6.7. Querying

One way to test the appropriateness of an ontology is to use querying across the knowledgebase which is built based on the ontology. Biologists and web agents need to use and query ontologies and the resources committed to them, thus the need for ontology querying arises. The Semantic Web provides a uniform view of resources, to automatically select appropriate sources for each query, based on the researcher's requirements. Some servers have only partial information about a subject and some have limitations in quality of data. So, the answers to a particular query may require an unpredictable amount of time to compute and may be unpredictable in size and quality [148].

In some formal ontology query languages the answering agent may use automated reasoning methods to derive answers to queries, in which the knowledge to be used in answering a query may be in multiple knowledgebases. In the current state of FWOnt the knowledge source for querying is determined manually.

The major challenges in querying over a large bioinformatics data sources are [147]:

- Queries are often complex associative queries over multiple Web documents and most of them involve complex data extraction.

- Complex genomics-specific queries are often reused many times by other genomics researchers, either directly or through some refinements.

The Semantic Web needs to have different query services with access to different information with different formats. Figure 6-26 shows the schematic querying across multiple biological data sources; NCBI, NEWT, ChEBI, SwissProt and BRENDA are different data sources that are used in the FungalWeb ontology. To provide access to these integrated data, a mapping of database entities to ontological concepts is necessary. By mapping database entities to the concepts in the ontology a variety of complex querying would be possible. In the FWOnt domain, these entities are mostly enzymes, organisms (fungal species) and enzyme properties.

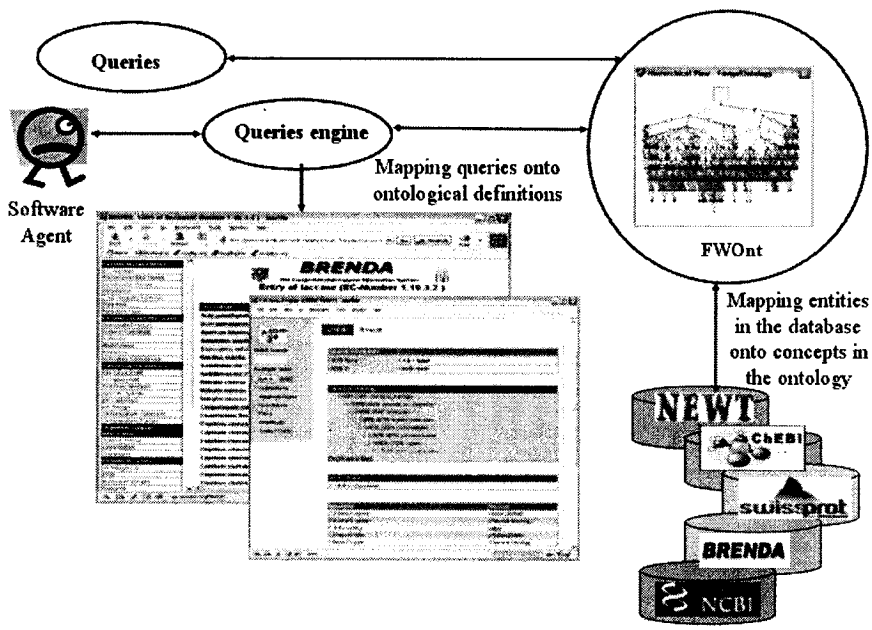


Figure 6-26: Querying across multiple resources through ontologies.

6.7.1. Ontology Query Languages

Traditional database query languages such as SQL (Structured Query Language) and other Web languages like XML query are not suitable for supporting such heterogeneity. Query languages for description logics and other logic-based knowledge representation formalisms often include "structural queries" asking about the subsumers, subclasses, and instances of classes [149] [150].

In languages like nRQL the queries can be formulated by taking advantage of the expressiveness of OWL [148].

Table 6-4 presents a list of query languages and their description for querying knowledgebases aiming to support large-scale applications in the Semantic Web.

Language	Description	Query Language
XML Extensible Markup Language	Structured Documents	XQuery , XPath
RDF Resource Description Framework	Data Model for Objects	RDQL, RQL, Versa, Squish
OWL Ontology Web Language	Data model + Relations	OWL-QL, Jena, nRQL (New Racer Query Language)
SWRL Semantic Web Rule Language	Data model + relations + rules (owl + RuleML)	N/A

Table 6-4: Different query languages.

These query languages are supposed to be able to extract the presence of ontology knowledge, for example, if the ancestor/descendant traversal of class/property hierarchies can be performed and what filtering conditions can be posed on class/property hierarchies. They also can provide constructs for calculating the extent of a property/class (the ability to find all the resources defined as instances of a particular property/class).

Also, they support Boolean filters (negation, conjunction, disjunction), set operations (union, intersection, difference), arithmetic operations on data values and container value constructors. When a query language is part of real, large scale applications, it usually supports other features such as aggregate (min, max, average, sum and count), grouping (e.g., an SQL-equivalent to group by clause), sorting (for ordering querying results) and built-in data functions (e.g., math and string/date converting functions) [154].

nRQL and OWL-QL both are two description logics based query languages, which are widely used in OWL ontologies. In the FungalWeb Ontology we used nRQL with Racer. For more information about other ontology query languages one may refer to available surveys on this topic [154].

6.7.1.1 nRQL (The New Racer Query Language)

nRQL is implemented in Racer with its applicability to OWL Semantic Web repositories to retrieve A-box individuals under specific conditions. It is more expressive than traditional concept-based retrieval languages offered by previous DLs reasoning systems. OWL/RDF is designed based on the advantages of description logics [147]. Racer is a true A-box reasoner; it not only provides means for representing metadata schema described in OWL ontologies, but also extensionally specified information, e.g., representing the actual web resources with concrete individuals and their interrelationships.

More information about nRQL and its syntax can be found in the Racer documentation [178] and nRQL handbook [68]. Some sample nRQL queries for FWOnt are available in the appendix 1.

6.7.1.2. OWL Query Language (OWL-QL)

OWL-QL is a formal language for a querying agent and an answering agent to use in conducting a query-answering dialogue using knowledge represented in OWL [151]. OWL-QL queries are full OWL KBs, so not only extensional queries like in nRQL must be answered, but also "structural queries" are possible. Racer's API presents similar functions. nRQL is not a subset of OWL-QL. In OWL-QL, neither negation as failure nor disjunctive A-boxes can be expressed. Moreover, binary constraint query atoms of nRQL as well as negated has-known-successor query atoms "are missing" in OWL-QL [153].

6.8 Performance Analysis

We used stand-alone windows Racer server version 1.7.23 for the analysis with JRacer2 [179] as a TCP-based client for Racer. The Racer server was started locally and TCP/IP communication did not involve any network latencies.

The tests were carried out on an IBM, P4, 2.4 GHz, with 512 MB main memory under the Windows XP professional. However, nowadays CPU and memory are not too expensive to upgrade and these factors are not the serious bottleneck for the performance.

We considered "response time" (in second) as our primary metric.

Checking the ABox and TBox consistency is done when initial loading of a knowledgebase was in progress.

During the development phase we called the check-tbox-coherence and check-abox-coherence, functions frequently, to check which atomic concepts or instances in the TBox or ABox are inconsistent. Response times (RT) for those functions are as following:

check-tbox-coherence:	RT = 12 Sec
check-abox-coherence:	RT = 42 Sec

Racer is strong in class subsumption problems. On average, Racer solves the posed subsumption problems within fraction of a second.

Also, because the FWOnt contains small number of concepts with necessary and sufficient condition (see table 6-4), Racer can run the reasoning process very fast.

By doing the tests in different stages of ontology development, it was discovered that with a growing size of the knowledgebase the response time for answering the queries

was increased. So, the performance of Racer was highly dependent on the number of individuals and grows with the number of individuals.

Also, we discovered that the number of properties does not have an important affect on the response time, but, the strongest influence on the performance of Racer with respect to instance retrieval appears to be the number of property fillers.

We tried to find the response time for some of our sample queries (see appendix 1). Some results (without considering the time for initial loading of the knowledgebase in the Racer) are:

Query 3 (Retrieving Fungi has contain Pectin lyase) RT = 24 Sec

Query 4 (Retrieving the taxonomic lineage for one individual RT = 155 Sec

However the response time is still acceptable for us, but as the number of instances and property fillers in ontology grows, and also if one wants to use the ontology in some ontology-based application such as an ontology-based NLP tool, then the response time will not be in a satisfactory range.

7. Application scenarios for the FungalWeb Ontology [164]

Fungi are microorganisms well known for the range of novel enzymes they produce and enzymes of fungal origin are now used in industrial processes which amount to billions of dollars of revenue annually. The path to product development, namely gene discovery, enzyme characterization, mutational improvement and industrial application is long and fraught with numerous hurdles, both with respect to the domain knowledge and technical challenges. Both within an academic setting and in industrial RnD many decisions are made on incomplete knowledge. This is partly the result of the information overload scientists are currently experiencing and partly since the required knowledge is distributed between numerous disciplines [155]. The current need in the enzyme RnD environment is to have an integrated framework for discovery and decision support. This must integrate data from laboratory research, data accessed from distributed databases and textual resources as well as the results of bioinformatics computations. Existing technologies can be assigned to this need for data integration.

Using ontologies and advanced query tools with simple semantic query access will enable the research manager to address the range of questions involved. To provide a reliable semantic resource in a contemporary RnD environment the scientific and technical span of the ontology can encapsulate a more inter disciplinary range of concepts. The full range of conceptualizations required includes gene discovery, protein family classification, enzyme characterization, enzyme improvement, enzyme production, enzyme substrates, enzyme performance benchmarking, enzyme assays, and market niche. Inclusion of such concepts and instance data will provide for knowledge

repositories that support the scientific manager in a scientific discovery process at its inception, through the data production phase and resulting interpretation phase and is one of the goals of the Fungal Web data integration initiative.

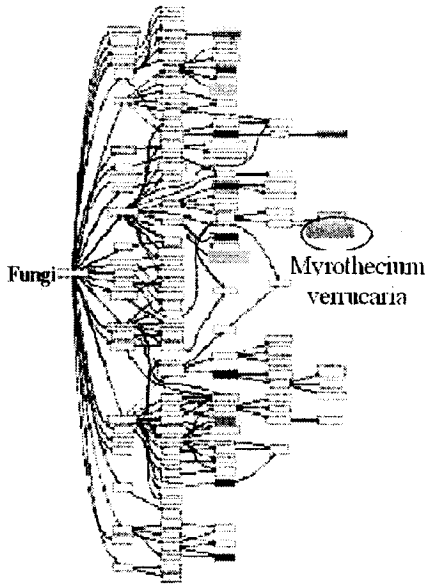
The goal of this part is to demonstrate the scope of our ontological conceptualization and demonstrate the range of cross disciplinary queries that can be posed. In the following section we describe *junction scenarios* where a biotechnologist would ask support from the ontology; thereby illustrating how the diverse needs of the fungal biotechnology manager can be accommodated. An explanation of the scientific context of these semantic queries and the conceptual frames designed to support the needs are outlined in each case.

7.1. Scenario 1: Finding alternatives for pulp chlorine delignification

New environmental legislation forces pulp manufacturers to consider reducing the chlorine used for pulp delignification, implementing new bleaching technology or finding alternatives to chlorine delignification include enzyme treatments.

- Laccase is known to bleach pulp when used in conjunction with mediator compounds.
- Pulp liquor is highly alkaline (pH 9-11) at the treatment step before chlorine bleaching normally occurs. The enzyme treatment with laccase is only possible if the laccase can tolerate and operate in alkaline conditions.
- Since the majority of laccases have pH profile of 3-6 it would seem unlikely that laccase could help.
- How can one find out if there is any report of a laccase with a pH optimum of 9.0?

- If there is such enzyme what species of organisms does this enzyme come from.
- How can one produce it in large quantities required for industrial scale bleaching



Query: retrieving all Fungi which have been reported to have enzyme **Laccase** with **pH optimum of 9.0** and act on the Phenol

The answer is:

```
(((?X !http://a.com/ontology#Myrothecium_verrucaria!)))
```

Figure 7-1: Schematic diagram to visualize the query result

7.2. Scenario 2: Identifying enzymes acting on Substrates

Enzyme technologies pose significant opportunities to treat waste products that are a nuisance within an industrial process or natural environment, i.e., bioremediation. Enzyme technologies can additionally provide access to otherwise wasted natural resources such as cellulose for ethanol production. A research manager can frequently pose the question, "Could an enzyme be used to degrade this novel chemical substrate?" To determine if an enzyme is suitable to act on such a substrate a domain expert knowledgeable of biochemistry would typically evaluate a chemical analysis of the substrate and consider what enzymes carry out modifications of the bonds in the substrate. To replace the need for a domain expert we consider this in four steps, namely, chemical analysis, semantic translation, ontology query, and knowledge retrieval.

Presented here is an approach where a key term in a written report from an analytical chemist describing the chemical nature of the substrate is used as a query to the ontology. The analytical chemist has analyzed the naturally occurring polymer 'pectin' known for its gelling properties and use in food conservation. It is composed of multiple units of the monomer glucuronic acid. The chemist's description 'A *polymer* containing repeated units of *galacturonic* acid' (Figure 7-2) contains the key semantic terms polymer and galacturonic.

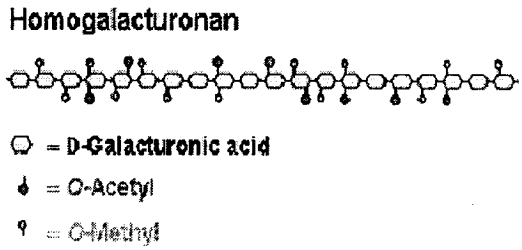


Figure 7-2: Structure of poly galacturonic acid, pectin.

In order to identify enzymes able to modify the substrate, the semantic word stems of the substrate, *poly* and *galacturon*, are used to query concepts within the FWOnt. The key concept in the ontology that facilitates the link of chemical substrates with enzyme reaction is the concept, "semantic word stem". This concept can be instantiated by an NLP application summarizing the word stem of the most common terms found in the semantically rich enzyme descriptions provided by the systematic enzyme reaction classification scheme introduced by the International Union of Biochemistry (IUB) Enzyme Commission. By integrating the semantically rich descriptions of the IUB (Figure 7-3) into the ontology we are able to query the different enzymes known to degrade / modify poly-galacturonan, better known as pectin, from the "semantic word stem" concept. Such enzymes are generally referred to as 'pectinases'.

Enzyme Class:	EC 3.2.1.67
Enzyme Name:	Exo**polygalacturonase**; acts on non methoxylated **polygalacturonic** acid
Common name:	**polygalacturonase**
Reaction Type:	hydrolysis of O-glycosyl bond, Random hydrolysis of 1,4-?-D-galactosiduronic linkages in pectate and other **galacturonans**
Other name(s):	pectin depolymerase; pectinase; endo**polygalacturonase**; pectolase; pectin hydrolase; pectin **polygalacturonase**; endo-**polygalacturonase**; poly-?-1,4-**galacturonide** glycanohydrolase; endo**galacturonase**; endo-D-**galacturonase**
Systematic name:	**poly** (1, 4-?-D-**galacturonide**) glycanohydrolase

Figure 7-3: Semantically rich enzyme descriptions provided by the systematic classification system introduced by IUB Enzyme Commission.

Pectinases; pectinesterase, pectin acetylesterase, endopectinase, exopolygalacturonase, pectate lyase, pectate disaccharide-lyase (exo-polygalacturonate lyase), pectin lyase have the word stem concept instantiated with 'galacturon' and are found when querying the 'word stem' concept using nRQL/DL with Racer (Figure 7-6). These are graphically displayed in figure 7-4 in the context of the concepts supporting the identification of the reaction classes of pectinases.

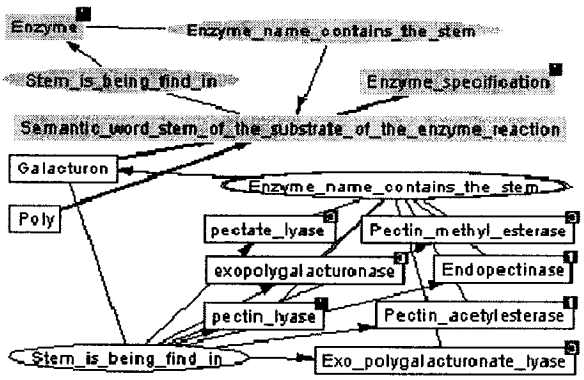


Figure 7-4: Conceptual frame supporting the identification of pectinase enzymes using substrate word stems.

Having identified what enzymes could be used to modify the substrate of interest the manager would further pose the question 'where are such enzymes found'. In the next section we describe how the FungalWeb ontology can be used to answer this question.

7.3. Scenario 3: Ontology for determining enzyme taxonomic provenance

Taxonomic provenance (which enzymes are found in which fungal species) of enzymes with industrial potential has long been an interest of microbial biotechnologists. In spite of the recent trend, for finding new genes in environmental DNA samples from diverse environments and cloning genes directly from these sources [159], scientists continue to use bioinformatics techniques to infer a taxonomic origin for these sequences. Taxonomic groups that have provided the most useful enzymes or natural products to date continue to be investigated as providers of new enzymes / genetic material or small molecules [160, 161]. Furthermore, laboratory isolation to the taxa of interest can be enhanced on the basis of the specific nutritional and growth requirements of the specific taxonomic group in question. Identifying taxonomic provenance is an important step within the gene discovery process.

Having identified enzymes able to modify pectin, the same research manager wishes to know which fungi are known to produce pectinases and the common lineage that these species all share. The common lineage query requires identifying the highest taxonomic group that unites all species known to produce the enzyme of interest, akin to finding a common ancestor. For a small number of species this is a relatively simple task that can be accomplished using online web site resources but it becomes significantly more challenging for a large number of species producing the same enzyme. Within the FungalWeb Ontology a fungal taxonomy is represented as a deep hierarchy of individual units / concepts. The key relationship between fungi and enzyme permits the query of species found to have pectinases and their common lineage. Figure 7-6, Query 3 retrieves all fungal instances in the knowledgebase that are known to produce pectin lyase and

Query 4 shows the common lineage for samples species known to produce this enzyme. The common lineage is the sub phylum of the *Ascomycota* called *Pezzimycotina* known for anatomically producing mycelia that make ascocarps (ascus-bearing structures also called ascomata) with hymenia.

7.4. Scenario 4: Ontology for enzyme benchmark testing

As gene discovery initiatives result in the production of an enzyme suitable for an industrial application, biotechnologists carry out benchmark performance testing with commercially available competitor enzymes under the same assay conditions to determine the likely potency of the product and to consider whether further enzyme improvement is necessary. Identification of vendors supplying enzymes suitable for their given application is a necessary step in this process.

To benchmark a polygalacturonic acid degrading enzyme it is necessary to retrieve all commercial enzyme products sold to the fruit processing industry. The product name, the vendor and product parameters are instantiated to the ontology. It can be done by using software agents [162] able to locate and retrieve information from commercial product literature in heterogeneous formats posted on distributed web sites. This information is instantiated to the concepts of the ontology described in figure 7-5, where Depol_40L-040L and Cellulase_13P-013P are commercial enzyme products that contain pectinases and are used in the fruit processing industry.

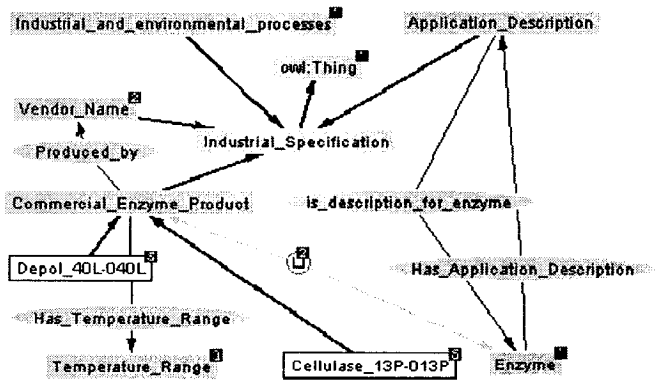


Figure 7-5: Conceptual frame supporting the identification of pectinase vendors, the characteristics and application of the products.

Using up-to-date information the scientist can coordinate a comprehensive benchmarking study of competitor enzymes, flag promising enzymes for further mutational improvement and identify a suitable market niche for new enzymes. Further vendor related queries to the ontology are shown in Figure 7-6, queries 6, 7 and 8 where enzyme vendors and products of use in the dairy and pulp and industries are returned.

```
C:\WINDOWS\System32\cmd.exe - java Q13
C:\JRacer2>javac Q13.java
C:\JRacer2>java Q13

1 - Is 'Galacturon'an instance of
Semantic_word_stem_of_the_substrate_of_the_enzyme_reaction ?

True

2 - Find all Enzyme names which contain semantic word stem
 of the substrate of the enzyme reaction that matches with
Galacturon

<<<?X :http://a.com/ontology#exopolygalacturonase:>>
<<?X :http://a.com/ontology#pectin_lyase:>>
<<?X: http://a.com/ontology#Pectin_methyl_esterase.>>
<<?X:http://a.com/ontology#Exo_polygalacturonate_lyase:>>
<<?X :http://a.com/ontology#Endopectinase:>>
<<?X :http://a.com/ontology#pectate_lyase:>>
<<?X :http://a.com/ontology#Pectin_acetylesterase:>>>

3- All Fungi have been reported to have enzyme Pectin lyase.

<<<?X :http://a.com/ontology#Aspergillus_niger:>>
<<?X :http://a.com/ontology#Aspergillus_oryzae:>>
<<?X :http://a.com/ontology#Aspergillus sojae:>>
<<?X :http://a.com/ontology#Aspergillus_japonicus:>>
<<?X :http://a.com/ontology#Glomerella_lindemuthiana:>>
<<?X :http://a.com/ontology#Penicillium_expansum:>>
<<?X :http://a.com/ontology#Fusarium_oxysporum:>>
<<?X :http://a.com/ontology#Penicillium_italicum:>>
<<?X :http://a.com/ontology#Penicillium_paxilli:>>>

4- For all Fungi that contain pectin lyase what is common lineage.

<:http://a.com/ontology#Pezizomycotina.>
<:http://a.com/ontology#Ascomycota:>
<:http://a.com/ontology#Fungi.>

5- I am looking for commercial enzyme product(s) which
can being used in fruit processing industry in TEMP RANGE 50-70
celsius

<<<?X :http://a.com/ontology#Cellulase_13P-o13P:>>>

6- Give me the name of Enzyme production companies
selling products to the dairy industry.

<<<?Xhttp://a.com/ontology#Specialty_Enzymes_&_Biochemicals_Co.:>>
<<?X :http://a.com/ontology#Danisco_Enzymes.>>>

7- Which enzymesare used in pulp and paper manufacturing?

<<<?X :http://a.com/ontology#Pectinase.>>
<<?X :http://a.com/ontology#Laccase:>>
<<?X :http://a.com/ontology#Lipase:>>
<<?X :http://a.com/ontology#Cellulase:>>>

8-Which companies are producing enzymes for pulp and paper
manufacturing?

Vendor name is.<<<?X:http://a.com/ontology#DYADIC.>>>

C:\JRacer2>
```

Figure 7-6: Query results generated by nRQL and Racer

We have already provided a rich and unique set of data for semantic queries. In summary: More than 1965 enzyme products, 116 enzyme products containing two or more enzymes, 38 Enzyme Vendors, 34 applications and/or industries for 83 enzyme types.

Detailed information is provided for the following vendors:

Amano Enzyme Inc., Bio-Cat Inc., Biocatalysts Enzymes, Biochem Europe, Biozyme Laboratories, Danisco Enzymes, Deerland Enzymes Inc., Diversa, Dyadic International Enzyme Development Corporation, Enzyme Technical Association, Genencor International, National Enzyme Company and Specialty Enzymes & Biochemicals Co.

Enzyme Parameters listed include:

Application/Purpose, Benefits, CAS No., Density, Description, EC No., EINCS, Enzyme/Category, IUB, Origin/Source, pH (Optimum and Stable), Product, Specification /Activity, Systematic Names and Temperature (Optimum and Stable)

Ontology for enzyme improvement (A scenario for future work)

Where existing enzymes are known to have a suitable catalytic capability for a given application, and where further benchmarking studies show sufficient enzymatic potency for a given application, there may still be further considerations.

Such an enzyme may not have the required functional properties such as pH range or temperature optimum. The enzyme however can still be a good candidate for further mutational enhancement. When considering improving the properties of an existing enzyme there are a range of options. To determine which would be the most successful strategy the scientist would review literature describing the methods employed with close attention to the success of the strategy in improving enzyme properties. Of particular

importance is the extent to which a particular approach has improved a property and how much additional improvement is possible. Such questions are inherently difficult to answer and require considerable literature review across mutational studies in many different protein families. Ontological and text mining technologies can render and provide access to knowledge concerning the mutational approaches and improvements along with wild type properties of the individual enzymes investigated. The current access route to this information requires manual browsing of distributed database resources such as BRENDA [14, 157] and the scientific literature [156].

To provide access to such knowledge multiple instance data can be generated using a custom designed tool for natural language processing of scientific full text articles [158]. The instance data shown in the XML format (Figure 7-7), describes the protein name, the wild type organism, the PMID: PubMed identification number of the paper citing the mutation, the GI: Genbank accession number of the protein, the mutation (Wild Type Residue-Location-Mutant Residue) and the impact of the mutation. Currently this data is instantiated manually to the concepts in FWOnt but it can be done automatically in the future. The instance data in Figure 7-7 enables our industry manager to query the ontology for xylanase enzymes that have been modified by site directed mutagenesis. To obtain a deeper semantic access to the impact of a mutation from the <Conext> field, additional concepts are needed within the ontology. Currently concepts are being introduced to permit the ranking of such sentences based on a descriptions of changes in enzyme properties (shift, increase, more active, fold, destabilize, decrease, remain same), the direction of the change (positive / negative), units of measurement (half life (s), Kcat, hydrolysis efficiency, pH) and the biological property of the enzyme that has been altered

(denaturation, catalysis, stability, folding). Appropriate synonym lists for these concepts are additionally being developed. These additions to the FungalWeb Ontology will facilitate semantic access in a manner far superior to manual browsing of texts and database content. Answers to questions such as 'Find the locations of all mutations in all xylanases that have been reported to have delivered enhanced temperature or pH profile' will soon be enabled. Such a query is currently not possible from any bioinformatics database.

```
<Protein>
    <Name>xylanase</Name>
    <Organisms>
      <Name>Bacillus circulans</Name>
<label>PMID: 9930661; GI: 17942986</label>
<Mark>D37N</Mark>
<Context>The upward shift of the optimum pH of the
D37N mutant was predictable from the results of structural
and amino acid sequence comparison.
</Context>
```

Figure 7-7: Instance data produced by the Mutation Miner [158].

8. Summary, Conclusion and Future work

8.1. Summary

In this work we have introduced and followed the large-scale ontology design and development case study in the domain of fungal genomics using state-of-the-art semantic technologies. In particular, we looked at the ontologies as a core for a semantic web system which can be used by human or some intelligent systems for ontology-based information retrieval and providing extended interpretations and annotations that can better serve the purpose of communication over the Semantic Web.

After, reviewing the basic concepts, other similar systems and current available tools and technologies and uncovering their problems, we turned to a description of the context of our ontology design and development by focusing on the following key areas:

- Knowledge gathering: at this stage we studied which resources to use in our tasks and how to use each of them, how to discover instances from those resources how to link their content and understand the content of the resources and interpret the results.
- Ontology implementation: includes conceptualization, integration and encoding. By using some tools and techniques with combination of manual and automated methods, we formally implemented the ontology.
- Ontology evaluation and querying: We pragmatically assessed the ontology to satisfy the requirements of our application, including determining the consistency (logically and semantically), completeness and conciseness in appropriate

granularity. For this purpose we used Racer as a DL reasoner. For testing the appropriateness of the ontology for our purpose, we used querying across the ontology.

- Application scenarios: Lastly, by defining some application scenarios, we tried to show, what a bioinformatics application can gain from using ontology-based technologies. Ultimately, how could one argue for the commercial usage and business feasibility of such an ontology?

The major issues in our work are the following:

- Deal with highly heterogeneous and volatile data.
- Integration of ontologies implemented in different languages, semantic, tools and platform, and lack of trustable tools for this purpose.
- Lack of uniform pattern to judge and verify the quality of the ontology.
- Managing the ontology for updating and versioning.

8.2. Conclusion and Future work

In the "postgenomic era" it is predictable that bioinformatics would have a great impact on improving our understanding of such diverse areas as the regulation of gene expression, protein structure determination, comparative evolution, and drug discovery.

Bio-ontologies are currently being used for communication of knowledge, as well as database schema definition, query formulation and annotation. We introduced the need and use of ontologies in bioinformatics. The need for ontologies arises from the need to be able to cope with the size and complexity of biological knowledge and data.

We have used Semantic Web technology to create an ontology and a large knowledgebase in the domain of fungal biotechnology and genomics from trusted biological sources to provide unified semantic access to these heterogeneous sources.

By browsing and querying the FWOnt for concepts of interest instead of digesting full text articles we have reduced the time it takes to gather corresponding information.

We have demonstrated the capacity of the ontological conceptualization through a series of industry related queries. Our conceptualizations are designed to integrate well with existing ontologies through fundamental concepts such as protein, enzyme or fungus but extend to practical and commercial concepts of the biotechnology industry.

Our ongoing research involves improvement of querying capabilities and using Natural Language Processing (NLP) techniques for ontology update and change management.

We need further additions to the ontology to cover microarray data that will draw on the MicroArray and Gene Expression (MAGE) [173] object model MGED (Microarray Gene

Expression Data Society) [174]. For coverage of metabolic pathways and regulatory networks we will use the model of BioCyc [163] as a starting point.

References

1. Robert Stevens, Chris Wroe, Phillip Lord, and Carole Goble. Ontologies in bioinformatics. In Stefan Staab and Rudi Studer, editors, Handbook on Ontologies in Information Systems, pages 635-657. Springer, 2003.
2. David S. Roos . COMPUTATIONAL BIOLOGY: Bioinformatics - Trying to Swim in a Sea of Data, 16 Feb 2001.
3. John F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, 2000.
4. Robert Stevens, Carole A. Goble and Sean Bechhofer. Ontology-based Knowledge Representation for Bioinformatics , Department of Computer Science and School of Biological Sciences, University of Manchester September 26, 2000.
5. S. Schulze-Kremer. Ontologies for Molecular Biology. In Proceedings of the Third Pacific Symposium on Biocomputing, pages 693-704. AAAI Press, 1998.
6. A. Rector. Description logics in medical informatics. Chapter 13 from the Description Logic, Handbook Theory, Implementation and Applications Edited by Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, Peter Patel-Schneider. January 2003.
7. S. F. Altschul et al., J. Mol. Biol. 215, 403 (1990).
8. Chemical Entities of Biological Interest (ChEBI) http://www.ebi.ac.uk/chebi/
9. The Semantic Web Community Portal, http://semanticweb.org/
10. The Semantic Web: An Introduction, http://infomesh.net/2001/swintro/
11. Robert Stevens . A Semantic Web of Bioinformatics Resources . Fifth Annual Bio-

Ontologies Meeting, 2002 http://www.cs.man.ac.uk/~stevensr/meeting02/

12. National Centre for Biotechnology Information (NCBI) (http://www.ncbi.nlm.nih.gov/Entrez/)
13. NEWT, UniProt taxonomy browser http://www.ebi.ac.uk/newt/display
14. BRENDA is one of the main collection of enzyme functional data available to the scientific community. (http://www.brenda.uni-koeln.de/)
15. SwissProt is a curated protein sequence database which strives to provide a high level of annotation. (http://ca.expasy.org/sprot/)
16. T.K. Attwood, D.J. Parry-Smith. Introduction to bioinformatics. Addison Wesley Longman, 1999.
17. P.G. Baker, C.A. Goble, S. Bechhofer, N.W. Paton, R. Stevens, and A Brass. An Ontology for Bioinformatics Applications. Bioinformatics, 1999.
18. A. Bairoch and R. Apweiler. The SWISS-PROT Protein Sequence Data Bank and its Supplement TrEMBL. Nucleic Acids Research , 1999.
19. FungalWeb : " Ontology, the Semantic Web, Intelligent Systems for Fungal Genomics " : http://www.cs.concordia.ca/FungalWeb/
20. C. Discala, X. Benigni, E. Barillot, and G. Vaysseix. DBcat: A Catalog of 500 Biological Databases. Nucleic Acids Research, 2000.
21. Fox, M.S., "The TOVE Project: A Common-sense Model of the Enterprise", the Proceedings of the International Conference on Object Oriented Manufacturing Systems, Calgary Manitoba, pp.176-181, 1992.
22. D. Buttler, M. Coleman1, T. Critchlow1, R. Fileto, W. Han, L. Liu, C. Pu, D. Rocco, and L.Xiong. Querying multiple bioinformatics data sources: Can Semantic Web research help? SIGMOD Record, 2002.

23. International Union of Biochemistry . Enzyme Nomenclature : Recommendations of the Nomenclature Committee of the International Union of Biochemistry on the Nomenclature and Classification of Enzyme-Catalyzed Reactions. Academic Press (For the International Union of Biochemistry by), Orlando, FL, 1984.

24. World Wide Web Consortium (http://www.w3.org/)

25. Web Content Accessibility Guidelines 2.0 March 2004. http://www.w3.org/TR/2004/WD-WCAG20-20040311/

26. Gangemi, A., Mika, P., Understanding the Semantic Web through Descriptions and Situations. Lecture Notes in Computer Science, Springer-Verlag GmbH , pp. 689 - 706 2003.

27. Ciaran Mandal. Statutes on the Semantic Web, Dublin University, 2004.

28. Corcho O, Fernández-López M, Gómez-Pérez A, López-Cima A. Building legal ontologies with METHONTOLOGY and WebODE. Law and the Semantic Web. Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications. Springer-Verlag, LNAI 3369. March 2005.

29. P.G. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens. TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. An Overview. In Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology (ISMB'98), pages 25-34, California, June 1998.

30. T. R. Gruber. A translation approach to portable ontologies, Knowledge Acquisition, Vol. 5, No. 2, 1993.

31. T.R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. International Workshop on Formal Ontology, Padova, Italy, 1993.

32. R. Jasper and M. Uschold. A Framework for Understanding and Classifying Ontology Applications. KAW'99, 1999. http://sern.ucalgary.ca/KSI/KAW/KAW99/

33. Robert Stevens, Building an Ontology. July 2001. http://www.cs.man.ac.uk/~stevensr/onto/node12.html#building

34. M. Uschold and M. Gruninger. Ontologies: Principles, Methods and Applications. University of Toronto, Knowledge Engineering Review Volume 11 Number 2, June 1996.

35. D.M. Jones, T.J.M. Bench-Capon, and P.R.S. Visser. Methodologies for Ontology Development. In Proceedings of 15th IFIP World Computer Congress, London, 1998.

36. A. Gomez-Perez. Some Ideas and Examples to Evaluate Ontologies. Technical Report KSL-94-65, Knowledge Systems Laboratory , Stanford, 1994.

37. D. Duce G.A. Ringland. Approaches to Knowledge Representation: An introduction. Knowledge-Based and Expert Systems Series. John Wiley, Chichester, 1988.

38. I. Horrocks, D. Fensel, J. Broekstra, M. Crubezy, S. Decker, M. Erdmann,W. Grosso, C. Goble, F. Van Harmelen, M. Klein, M. Musen, S. Staab, and Studer. The ontology interchange language oil: The grease between ontologies. http://www.cs.vu.nl/~dieter/oil

39. A. Farquhar, R. Fikes, and J.P. Rice. The ontolingua server: A tool for collaborative ontology construction. Journal of Human-Computer Studies, 46:707-728, 1997.

40. Description logics reference page (http://dl.kr.org/)

41. A. Borgida. Description Logics in Data Management. IEEE Trans Knowledge and Data Engineering, 1995.

42. S Bechhofer and C.A. Goble. Delivering Terminological Services. AI*IA Notizie, Periodico dell'Associazione Italiana per l'intelligenza Artificiale.Vol.12, No.1, March 1999.

43. A. J. Duineveld, R. Stoter, M.Weiden, B. Kenepa, and V. Benjamins. Wondertools A Comparative Study of Ontological Engineering Tools. KAW1999.

44. W3C OWL Web Ontology Language Overview http://www.w3.org/TR/owl-features/

45. GALEN is a technology that is designed to represent clinical information http://www.opengalen.org/

46. SNOMED Clinical terms: a universal health care terminology that makes health care knowledge usable and accessible http://www.snomed.org/

47. Pisanelli, Gangemi , Steve G. An ontological analysis of the UMLS Methathesaurus. Proc AMIA Symp 1998.

48. Hahn U, Schulz S. Towards a broad-coverage biomedical ontology based on description logics. PSB 2003.

49. Kashyap V, Borgida A. Representing the UMLS Semantic Network using OWL. International Semantic Web Conferences ISWC 2003.

50. Golbeck J, Fragoso G, Hartel F, Hendler J, Oberthaler J, Parsia B. The National Cancer Institute's Thesaurus and Ontology. Journal of Web Semantics 2003.

51. Horrocks I, Rector A, Goble C. A Description Logic based schema for the classification of medical data. Proceedings of KRDB'96; 1996.

52. Stevens R, Baker P, Bechhofer S, Ng G, Jacoby A, Paton NW, et al. TAMBIS: transparent access to multiple bioinformatics information sources. Bioinformatics 2000 pp. 184-186. volume 16.

53. Phillip Lord. Description Logics: OWL and DAML+OIL. March 15, 2004.

54. Jonathan B. L. Bard & Seung Y. Rhee, Ontologies in biology: Design, Applications and future challenges. Nature Reviews Genetic, March 2004, Vol. 5, pp. 213-222.

55. Dennis Quan, Sean Martin, David Grossman. Applying Semantic Web Techniques to Bioinformatics, IBM Internet Technology, ISWC 2003.

56. Davis, R., Shrobe, H., and Szolovits, P. What is a Knowledge Representation? AI Magazine, 1993.

57. Volker Haarslev, Ralf Möller. RACER System Description. Proceedings of International Joint Conference on Automated Reasoning, IJCAR'2001, R. Goré, A. Leitsch, T. Nipkow (Eds.),June 18-23, 2001, Siena, Italy, Springer-Verlag, Berlin,pp. 701-705.

58. Kahan, J. and Koivunen, M. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In Proceedings of the 10 International World Wide Web Conference 2001.

59. Bruce Birren, Gerry Fink, and Eric Lander, Whitehead Institute, Center for Genome Research, Fungal Genome Initiative , February 8, 2002.

60. An initiative to sequence the genome of the filamentous fungus Aspergillus Nidulans. http://gene.genetics.uga.edu/white_papers/anidulans.html

61. Concordia Fungal Genomics Project https://fungalgenomics.concordia.ca/home/index.php

62. Unified Medical Language System (UMLS), by National Library of Medicine http://www.nlm.nih.gov/research/umls/about_umls.html

63. Ceusters W, Smith B, Flanagan J. Ontology and medical terminology: Why

Description Logics are not enough. Proceedings of TEPR 2003 San Antonio, 2003.

64. Olivier Bodenreider, Barry Smith , Anand Kumar , Anita Burgun. Investigating subsumption in DL-Based Terminologies: A Case Study in SNOMED CT. KR-MED 2004 Proceedings.

65. Ralf Möller and Volker Haarslev, RACER system description page http://www.cs.concordia.ca/~haarslev/racer/

66. Volker Haarslev , Ralf Möller, Description of the RACER System and its Applications. Proceedings of the International Workshop on Description Logics (DL-2001), Stanford, USA, 1.-3. August 2001, pp. 132-141.

67. Volker Haarslev , Ralf Möller. Racer: An OWL Reasoning Agent for the Semantic Web. Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems, in conjunction with the 2003 IEEE/WIC International Conference on Web Intelligence, Halifax, Canada, October 13, 2003, pp. 91-95.

68. Volker Haarslev, Ralf Möller, Michael Wessel. The New Racer Query Language – nRQL documentation. http://www.cs.concordia.ca/~haarslev/racer/racer-queries.pdf

69. Gene Ontology documentation, http://www.geneontology.org/doc/GO.doc.html

70. Biological Ontology Committee in Japan http://ontology.ims.u-tokyo.ac.jp/OntologyCommittee/Collection.html

71. The Gene Ontology Consortium. 2000, Gene Ontology: tool for the unification of biology. http://www.geneontology.org/GO_nature_genetics_2000.pdf

72. Nataliya Sklyar, A survey of existing Bio-Ontologies . Technical Report September 2001. http://dol.uni-leipzig.de/pub/2001-30/en

73. Carole Goble. The Story of GO: How a community built an ontology , July 2003.

74. Barry Smith, Jennifer Williams, Steffen Schulze-Kremer, The Ontology of the Gene Ontology, Published in Proceedings of AMIA Symposium 2003.

75. P.W.Lord , R.D. Stevens, A. Brass and C.A.Goble , Semantic similarity measures as tools for exploring the Gene Ontology. Pacific Symposium on Biocomputing 8:601-612, 2003.

76. Rison S., Hodgman T.C., Thornton J.M., Comparison of functional annotation schemes for genomes. Functionmal & Integrated Genomics 1:56-69. PMID: 11793222, 2000. http://nucleus.cshl.edu/agsa/Papers/Analysis/function.pdf

77. Protege : an ontology editor and a knowledgebase editor developed by Stanford University. http://protege.stanford.edu/

78. AmiGo: Gene Ontology browser http://www.godatabase.org/cgi-bin/go.cgi

79. Carole Goble. University of Manchester, Building Ontologies. Presentation www.cs.man.ac.uk/~carole/old/GGF/part4-building.ppt

80. Peter Karp , Robert Stevens , Carole Goble, 2001 , Bio-Ontologies: Their Creation and Design, July 2001.

81. Jennifer Williams. Bringing Ontology to the Gene Ontology. Comparative and Functional Genomics, January 2003, vol. 4, no. 1, pp. 90-93(4).

82. Barry Smith , Ontology: Philosophical and Computational http://ontology.buffalo.edu/smith//articles/ontologies.htm

83. Gene Ontology Consortium 2001, Creating the Gene Ontology resource: Design and Implementation.

84. Jennifer Williams, November 2002, Formal Ontology and the Gene Ontology. Comparative and Functional Genomics, January 2003, vol. 4, no. 1, pp. 68-70(3).

85. Gene Ontology Usage Guide http://www.geneontology.org/doc/GO.usage.html

86. Steffen Schulze-Kremer, 2002, Ontologies for molecular biology and bioinformatics. In Silico Biol. 2002; 2(3):179-93.

87. Barry Smith, Realism, Concepts and Categories or: how realism can be pragmatically useful for information system http://www.humaniora.sdu.dk/ifki/ontoquery/presentations/Smith3.ppt

88. Anand Kumar, Barry Smith, A Framework for Protein Classification. Proceeding Of German Conference on Bioinformatics, Munich, Oct 12-14. 2003. 2; 55-57

89. Natalya F. Noy and Deborah L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

90. Carole A. Goble and Robert D. Stevens, Managing Biological Information Using Biological Knowledge. In NETTAB – Network Tools and Applications in Biology CORBA and XML: Towards a Bioinformatics Integrated Network Environment, Advanced Biotechnology Center, Genoa, Italy, 2001.

91. Anand Kumar, Barry Smith, The Universal Medical Language System and the Gene Ontology: Some Critical Reflections. Lecture Notes in Computer Science. 2003 Sep; 2821/2003: 135 – 148. http://ontology.buffalo.edu/medo/UMLS_GO.pdf

92. GONG (Gene Ontology Next Generation) project: http://gong.man.ac.uk/

93. C.J. Wroe, R. Stevense, C. A. Goble, A Methodology to migrate the Gene Ontology to a Description logic environment using DAML+OIL. Pacific Symposium on Biocomputing 8:624-635, 2003.

94. Nicola Guarino and C.Welty. Identity, Unity, and Individuality: Towards a Formal Toolkit for Ontological Analysis. In Proceedings of ECAI-2000 Amsterdam.

95. Josep Guarro, Josepa Gene´ and Alberto M. Stchigel. Developments in fungal taxonomy. Clinical Microbiology Reviews, July 1999, p. 454–500 Vol. 12, No. 3

96. Davis, J. I. Species concepts and phylogenetic analysis. Introduction. Syst. Bot. 20:555-5591995.

97. Berbee, M. L., and J. W. Taylor. Detecting the morphological convergence in true fungi using 18S RNA sequence data. BioSystems 1992; 28(1-3):117-25.

98. Sogin, M. L. Evolution of eukaryotic microorganisms and their small subunit ribosomal RNAs. Amer. Zool. 29:487-499 (1989).

99. Taylor, J. W., F. C. Swann, and M. L. Berbee. Molecular evolution of ascomycete fungi: phylogeny and conflict, p. 201-212. In D. L. Hawksworth (ed.), Ascomycete systematics: problems and perspectives in the nineties. Plenum Press, New York, N.Y. 1994.

100. Berbee, M. L., and J. W. Taylor. Ascomycete relationships: dating the origin of asexual lineages with 18S ribosomal RNA gene sequence data. p. 67-78. In D. R. Reynolds, and J. W. Taylor (ed.), The fungal holomorph: mitotic, meiotic and Pleomorphic speciation in fungal systematics. CAB International, Wallingford, United Kingdom. 1993.

101. McGinnis, M. R., and I. F. Salkin. A clinical user perspective of anamorphs and teleomorphs, p. 87-92. InD. R. Reynolds, and J. W. Taylor (ed.), The fungal holomorph: mitotic, meiotic and pleomorphic speciation in fungal systematics. CAB International, Wallingford, United Kingdom., 1993.

102. Odds, F. C., T. Arai, A. F. Di Salvo, E. G. V. Evans, R. J. Hay, H. S. Randhawa, M. G. Rinaldi, and T. J. Walsh. Nomenclature of fungal diseases, A report and recommendations from a Sub-Committee of the International Society for Human and Animal Mycology (ISHAM). 1992.

103. Bruns, T. D., T. J. White, and J. W. Taylor. Fungal molecular systematics. Ann. Rev. Ecol. Syst. 22: 525-564 (1991).

104. Hawksworth, D. L., P. M. Kirk, B. C. Sutton, and D. N. Pegler. Ainsworth and Bisby's dictionary of the fungi. Mycologist's Handbook, 6th edition. 1995.

105. Hazen, K. C. Methods for fungal identification in the clinical mycology laboratory. Clin. Microbiol. Newsl. 18:137-144, 1996.

106. Hennebert, G. L., and B. C. Sutton. Unitary parameters in conidiogenesis, p. 65-76. In D. L. Hawksworth (ed.), Ascomycete systematics: problems and perspectives in the nineties. Plenum Press, New York, N.Y 1994.

107. Hawksworth, D. L., P. M. Kirk, B. C. Sutton, and D. N. Pegler. Ainsworth and Bisby's dictionary of the fungi. 8th ed. International Mycological Institute, Egham, United Kingdom. 1995.

108. Drouhet, E. Penicilliosis due to Penicillium marneffei: a new emerging systemic mycosis in AIDS patients travelling or living in Southeast Asia. 1993.

109. Kurtzman, C. P., and J. W. Fell. T he yeasts, a taxonomic study, 4th ed. Elsevier Science B.V., Amsterdam, the Netherlands. 1998.

110. Kendrick, B. The fifth kingdom, 2nd ed. Mycologue Publications, Waterloo, Canada. 1992.

111. Ian Horrocks, University of Manchester, OWL: An Ontology language for the

Semantic Web, 2003.

http://www.openmath.org/cocoon/openmath/meetings/eindhoven2003/proceedings/horrocks.pdf

112. Carbohydrate Active Enzymes Database (CAZY) http://afmb.cnrs-mrs.fr/CAZY/

113. Coutinho, P.M. & Henrissat, B. (1999) The modular structure of cellulases and other carbohydrate-active enzymes: an integrated database approach. In "Genetics, Biochemistry and Ecology of Cellulose Degradation". K. Ohmiya, K. Hayashi, K. Sakka, Y. Kobayashi, S. Karita and T. Kimura eds., Uni Publishers Co., Tokyo, pp. 15-23.

114. Aspergillus nidulans Database (http://www.broad.mit.edu/annotation/fungi/aspergillus/)

115. CandidaDB (http://genolist.pasteur.fr/CandidaDB/)

116. Cryptoccocus neoformans Database at TIGR (http://www.tigr.org/tdb/e2k1/cna1/)

117. Fungal Genome Stock Center (http://www.fgsc.net/)

118. International Rice Blast Genome Consortium (http://www.riceblast.org/)

119. Magnaporthea grisea Database http://www.broad.mit.edu/annotation/fungi/magnaporthe/

120. Neurospora crassa Database (http://www.broad.mit.edu/annotation/fungi/neurospora/)

121. Saccharomyces Genome Database (http://www.yeastgenome.org/)

122. Schizosaccharomyces pombe GeneDB http://www.broad.mit.edu/annotation/fungi/neurospora/

123. A. Brazma, H. Parkinson, T. Schlitt, Mohammadreza Shojatalab. A quick introduction to elements of biology: cells, molecules, genes, functional genomics, microarrays. EMBL-EBI, October 2001.

124. TRANSFAC (collection of databases which deal with information about gene expression) http://www.gene-regulation.com/pub/databases.html#transfac

125. A. Gomez-Perez. Some Ideas and Examples to Evaluate Ontologies. Technical

Report KSL-94-65, Knowledge Systems Laboratory , Stanford, 1994.

126. Carlile, M. J., and S. C. Watkinson. The fungi. Academic Press, 1994., London, United Kingdom.

127. F. Baader, D. Calvanese, D. McGuineness, D. Nardi, and P. Patel-Schneider, editors. The Description Logic Handbook. Cambridge University Press, 2003.

128. J. Golbeck, G. Fragoso, F. Hartel, J. Hendler, B. Parsia, and J. Oberthaler. The national cancer institute's thesaurus and ontology. 2003.

129. Alan Rector. Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In proceedings of K-CAP 2003.

130. Alan L. Rector. Clinical terminology: Why is it so hard? Methods Inf Med. Dec;38(4-5):239-52, 1999.

131. Holger Knublauch Olivier Dameron Mark A. Musen, Medical Informatics, Stanford University. Weaving the Biomedical Semantic Web with the Protege OWL Plugin. KR-MED 2004 Proceedings.

132. M. Uschold, M. Healy, K. Williamson, P. Clark, and S. Woods. Ontology reuse and application. FOIS'98, Treno, Italy, 1998.

133. M. Klein, Vrije Universiteit Amsterdam . Combining and relating ontologies: an analysis of problems and solutions, IJCAI01, 2001.

134. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies, KR2000.

135. Klein, M. Supporting evolving ontologies on the internet. In Lindner, W. and stuller, J., editors, Proceedings of the EDBT 2002 PhD Workshop, number 2490 in LNCS, pages 597-606, Prague, Czech Republic.

136. An Ontology mapping tool http://www.xspan.org/applications/cobra/index.html

137. PROMPT: tool for managing multiple ontologies in Protege http://protege.stanford.edu/plugins/prompt/prompt.html

138. OntoMorph: A Translation System for Symbolic Knowledge http://www.isi.edu/~hans/ontomorph/presentation/ontomorph.html

139: Chimaera: creating and maintaining distributed ontologies on the web http://www.ksl.stanford.edu/software/chimaera/

140. A. Gupta et al., Registering scientific information sources for semantic mediation, 21st International Conference on Conceptual Modeling, (2002).

141. B. Ludäscher et al., Model-Based Mediation with Domain Maps, 17th Intl. Conference on Data Engineering (2001).

142. Owl Converter: Maryland Information and Network Dynamics Lab Semantic Web Agents Project (Mindswap) http://www.mindswap.org/2002/owl.shtml

143. K. J. Kochut et al., IntelliGEN: a distributed workflow system for discovering protein-protein interactions, International Journal on Distributed and Parallel Databases, Special Issue on Bioinformatics (2002).

144. G. Wiederhold et al., Composing Diverse Ontologies, Technical Report, Stanford University (1998).

145. Volker Haarslev, Ying Lu, Nematollah Shiri. OntoXpl - Intelligent Exploration of OWL Ontologies. Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004), Beijing, China, Sept. 20-24, 2004, pp. 624-627.

146. Growl Web site , Ecoinformatics Collaboratory Group, The university of Vermont

http://ecoinformatics.uvm.edu/dmaps/growl/

147. David Buttler, Matthew Coleman, Terence Critchlow, Renato Fileto, Wei Han, Ling Liu, Calton Pu, Daniel Rocco, Li Xiong. Querying Multiple Bioinformatics Information Sources: Can Semantic Web Research Help?

148. Richard Fikes, Patrick Hayes, and Ian Horrocks. OWL-QL – A Language for Deductive Query Answering on the Semantic Web. J. Web Sem. 2(1): 19-29 , 2004.

149. Alex Borgida and Deborah McGuinness; " Asking Queries about Frames" ; Proceedings of Principles of Knowledge Representation and Reasoning (KR '96); Cambridge, Massachusetts; Morgan Kaufmann; November 5-8, 1996.

150. Sean Bechhofer, Ian Horrocks, Peter F. Patel-Schneider and Sergio Tessaris; "A Proposal for a Description Logic Interface"; Proceedings of Description Logic Conference (DL'99); pp. 33-36; August 11-14, 1999.

151. Deborah McGuinness and Frank van Harmelen (editors); "OWL Web Ontology Language Overview"; August 18, 2003; http://www.w3.org/TR/owl-features/.

152. Holger Knublauch Olivier Dameron Mark A. Musen, Medical Informatics, Stanford University. Weaving the Biomedical Semantic Web with the Protege OWL Plugin. KR-MED 2004 Proceedings.

153. Volker Haarslev, Ralf Moller, and Michael Wessel Querying the Semantic Web with Racer + nRQL. Proceedings of the KI-2004 International Workshop on Applications of Description Logics (ADL'04), Ulm, Germany, September 24, 2004.

154. A. Magkanaraki, G. Karvounarakis, Ta Tuan Anh, V. Christophides, D. Plexousakis. Ontology storage and querying. Technical report No. 308, April 2002.

155. W3C Workshop on Semantic Web for Life Sciences 27-28 October 2004,

Cambridge, Massachusetts USA http://www.w3.org/2004/07/swls-cfp.html

156. Wheeler DL, Chappey C, Lash AE, Leipe DD, Madden TL, Schuler GD, Tatusova TA, Rapp BA (2000). Database resources of the National Center for Biotechnology Information. Nucleic Acids Res 2000 Jan 1;28(1):10-4.

157. Schomburg I, Chang A, Ebeling C, Gremse M, Heldt C, Huhn G, Schomburg D. BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. 2004 Jan 1;32(Database issue):D431-3.

158. Baker C.J.O. and Witte R. 2004 Enriching Protein Structure Visualizations with Mutation Annotations Obtained by Text Mining Protein Engineering, Literature The 3rd Canadian Working Conference on Computational Biology Co-located with IBM CASCON, Markham, Ontario.

159. Solbak AI, Richardson TH, McCann RT, Kline KA, Bartnek F, Tomlinson G, Tan X, Parra-Gessert L, Frey GJ, Podar M, Luginbuhl P, Gray KA, Mathur EJ, Robertson DE, Burk MJ, Hazlewood GP, Short JM, Kerovuo J.J. Biol Chem. Discovery of pectin-degrading enzymes and directed evolution of a novel pectat lyase for processing cotton fabric. 2004 Dec 23.

160. Bull AT, Ward AC, Goodfellow M. Search and discovery strategies for biotechnology: the paradigm shift. Microbiol Mol. Biol Rev. 2000 Sep; 64(3): 573-606.

161. Kroken S, Glass NL, Taylor JW, Yoder OC, Turgeon BG. Phylogenomic analysis of type I polyketide synthase genes in pathogenic and saprobic ascomycetes. Proc Natl Acad Sci U S A. 23;100(26):15670-5. Epub 2003 Dec 15.

162. Graham J, Windley V, McHugh D, McGeary F, Cleaver D, and Decker K, Tools for

Developing and Monitoring Agents in Distributed Multi Agent Systems, Workshop On Agents in Industry at the Fourth International Conference on Autonomous Agents, Barcelona, Spain, June, 2000.

163. Karp P.D., Arnaud M., Collado-Vides J., Ingraham J., Paulsen I.T., Saier M.H. Jr. (2004). "The E. coli EcoCyc Database: No Longer Just a Metabolic Pathway Database." ASM News 70(1): 25-30.

164. Christopher J. O. Baker, René Witte, Arash Shaban-Nejad, Greg Butler and Volker Haarslev Department of Computer Science and Software Engineering, Centre for Structural and Functional Genomics. The FungalWeb Ontology: Application Scenarios, IDEAS 2005.

165. Shaban-Nejad A. "FungalWeb Ontology: Design a Formal Ontology of Fungal Genomics to analyzing the large-scale enzyme-fungi interaction in OWL-DL Environment". Robert Cedergren Bioinformatics Colloquium, September 23-24, 2004, Université de Montréal, Montreal, Quebec.

166. Baker C.J.O, Shaban-Nejad A., Haarslev V., Semantic query of a fungal enzyme knowledgebase in the Ontology Web Language – Description Logics Environment. Standards and Ontologies for Functional Genomics 2 (SOFG2), October 23-26 2004 , The University of Pennsylvania School of Medicine, Philadelphia, Pennsylvania.

167. Shaban-Nejad A., Baker C.J.O., Butler G. Haarslev V., The FungalWeb Ontology: The Core of a Semantic Web Application for Fungal Genomics. The 1ST Canadian Semantic Web Interest Group Meeting (SWIG'04), November 19 2004, Université du Quebec a Montreal, Montreal, Quebec.

168. Topquadrant inc., Ontology languages,

http://www.coolheads.com/egov/opensource/topicmap/s167/img21.html

169. Ian Horrocks and Peter Patel-Schneider. Reducing OWL entailment to description logic satisfiability. J. of Web Semantics, 1(4):345-357, 2004.

170. Patrick Lambrix , Manal Habbouche and Marta P´erez, Evaluation of ontology development tools for bioinformatics. BIOINFORMATICS Vol. 19 no. 12 2003, pages 1564–1571.

171. OilEd: Bechhofer et al., 2001; Stevens et al., 2001; http://oiled.man.ac.uk

172. Fast Classification of Terminologies (FaCT) is a DL reasoner, Horrocks, 1999; http://www.cs.man.ac.uk/~horrocks/FaCT/

173. MicroArray and Gene Expression (MAGE), aims to provide a standard for the representation of microarray expression data to facilitate the exchange of microarray data between different data systems. http://www.mged.org/Workgroups/MAGE/mage.html

174. Microarray Gene Expression Data Society –MGED aims to facilitate the sharing of microarray data generated by functional genomics and proteomics experiments. http://www.mged.org/

175. WonderWeb Ontology validator is developed by Sean Bechhofer (University of Manchester) and Raphael Volz (University of Karlsruhe) as part of the EU IST Project WonderWeb. http://phoebus.cs.man.ac.uk:9999/OWL/Validator

176. Tim Berners-Lee, James Hendler and Ora Lassila. "The Semantic Web": A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American magazine, may 2001 issue.

177. Sean Bechhofer, Phillip Lord, Raphael Volz. Cooking the Semantic Web with the

OWL API. 2nd International Semantic Web Conference, ISWC, Sanibel Island, Florida, October 2003.

178. Volker Haarslev, Ralf Möller, Michael Wessel. The RACER User's Guide and Reference Manual describes all features (updated for 1.7.19). http://www.sts.tu-harburg.de/~r.f.moeller/racer/racer-manual-1-7-19.pdf

179. JRACER2 is a TCP-based client for Racer and developed by Daniel Hartwig, Rene Weissmann, Univ. of Applied Sc., Wedel. It is accessible from the Racer system download page: http://www.sts.tu-harburg.de/~r.f.moeller/racer/download.html

180. I.M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I.T. Paulsen, M. Peralta-Gil and P.D. Karp. EcoCyc: A comprehensive database resource for Escherichia coli, Nucleic Acids Research 33:D334-7 2005.

181. R.O. Chen, R. Felciano, and R.B. Altman. RiboWeb: Linking Structural Computations to a KnowledgeBase of Published Experimental Data. In Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology, pages 84-87. AAAI Press, 1997.

Appendix 1: Sample Queries using Racer + nRQL

1-Is Galacturon an instance for Semantic_word_stem_of_the_substrate_of_the_enzyme_reaction?

```
(Retrieve () (|http://a.com/ontology#Galacturon|
        |http://a.com/ontology#Semantic_word_stem_of_the_substrate_of_the_enzyme_reaction|)
)
```

2-Find all Enzyme names which contain semantic word stem of the substrate of the enzyme reaction that matches with Galacturon ");

```
(Retrieve (?x) (AND (?x |http://a.com/ontology#Enzyme|)
                    (?x |http://a.com/ontology#Galacturon|
                        |http://a.com/ontology#Enzyme_name_contains_the_stem|)
                )
)
```

3-All Fungi that have been reported to have enzyme Pectin lyase

```
(Retrieve (?x) (AND (?x |http://a.com/ontology#Fungi|)
                    (?x |http://a.com/ontology#pectin_lyase|
                        |http://a.com/ontology#Has_been_reported_to_have_enzyme|)
                )
)
```

4-What is the taxonomical lineage for "Aspergillus japonicus"?

```
( individual-types |http://a.com/ontology#Aspergillus_japonicus|)
```

The Racer function "individual-types" returns all atomic concepts of which the individual is an instance. For retrieving the most-specific atomic concepts of which an individual is an instance, one can use function individual-direct-types instead. (See Racer documentation).

5- All commercial enzyme product(s) which can being used in fruit processing industry and produced by Biocatalysts Co. in TEMP RANGE 50-70 Celsius.

```
(Retrieve (?x) (AND
                    (AND
                        (AND
                            (AND (?x |http://a.com/ontology#Commercial_Enzyme_Product|)
                                    (?x |http://a.com/ontology#Fruite_and_Vegetable_Processing|
                                        |http://a.com/ontology#Can_being_used_in|)
                             )
                            (?x |http://a.com/ontology#Biocatalysts|
                                |http://a.com/ontology#Produced_by|))(?x |http://a.com/ontology#C50-70|
                                |http://a.com/ontology#Has_Temperature_Range|)
                         )
                    )
                )
)
```

6- All Enzyme production companies which have products that can be used in dairy industries.

```
(Retrieve (?x) (AND (?x |http://a.com/ontology#Vendor_Name|)
                    (?x |http://a.com/ontology#Dairy_products|
                        |http://a.com/ontology#Producing_enzyme_for|)
               )
)
```

7-Which type of enzymes are being used in pulp and paper manufacturing?

```
(Retrieve (?x) (AND (?x |http://a.com/ontology#Enzyme|)
                    (?x |http://a.com/ontology#Pulp_and_paper_manufacturing|
                        |http://a.com/ontology#Can_being_used_in|)
               )
)
```

8- Which companies are producing enzymes for pulp and paper manufacturing?

```
(Retrieve (?x) (AND (?x |http://a.com/ontology#Vendor_Name|)
                    (?x |http://a.com/ontology#Pulp_and_paper_manufacturing|
                        |http://a.com/ontology#Producing_enzyme_for|)
               )
)
```

9- Give me the name of commercial enzyme products available in the market which have benefits such as increasing bread volume and improving rumb softness and structure of bread.

```
(Retrieve (?x) (AND
                  (AND
                     (AND
                        (AND (?x |http://a.com/ontology#Commercial_Enzyme_Product|)
                             (?x |http://a.com/ontology#Baking|
                                 |http://a.com/ontology#Can_being_used_in|)
                        )
                             (?x |http://a.com/ontology#Improving_rumb_softness_and_structure|
                                 |http://a.com/ontology#Has_benefit|)
                     )
                           (?x |http://a.com/ontology#Increasing_bread_volume|
                               |http://a.com/ontology#Has_benefit|)
                  )
               )
)
```

10- All Fungi which has been reported to have both enzyme, Laccase and Cellulase.

```
(Retrieve (?x) (AND
                  (AND (?x |http://a.com/ontology#Fungi|)
                       (?x |http://a.com/ontology#Laccase|
                           |http://a.com/ontology#Has_been_reported_to_have_enzyme|)
                   )
                       (?x |http://a.com/ontology#Cellulase|
                           |http://a.com/ontology#Has_been_reported_to_have_enzyme|)
                )
)
```