



Research Article

Using ontologies for life science text-based resource organization

Giulia Panzarella^{a,*}, Pierangelo Veltri^b, Stefano Alcaro^{a,c,d}^a Dipartimento di Scienze della Salute, Università "Magna Græcia" of Catanzaro, Campus Universitario "S. Venuta", Viale Europa, 88100 Catanzaro, Italy^b Dipartimento di Medicina Clinica e Sperimentale, University Magna Græcia of Catanzaro, Catanzaro, 88100, Italy^c Net4Science Academic Spinoff, Università "Magna Græcia" of Catanzaro, Campus Universitario "S. Venuta", Viale Europa, 88100 Catanzaro, Italy^d Associazione CRISEA-Centro di Ricerca e Servizi Avanzati per l'Innovazione Rurale, Località Condoleo, Belcastro, Catanzaro, Italy

ARTICLE INFO

Keywords:

Information overload

Ontology

Semantic web

Life science terms

ABSTRACT

Ontologies are used to support access to a multitude of databases that cover domains relevant information. Heterogeneity and different semantics can be accessed by using structured texts and descriptions in a hierarchical concept definition. We are interested in Life Sciences (LS) related ontologies including components taken from molecular biology, bioinformatics, physics, chemistry, medicine and other related areas. An Ontology comprises: (i) term connections, (ii) the identification of core concepts, (iii) data management, (iv) knowledge classification and integration to collect key information. An ontology may be very useful in navigating through LS terms. This paper explores some available biomedical ontologies and frameworks. It describes the most common ontology development environments (ODE): Protégé, Topbraid Composer, Ontostudio, Fluent Editor, VocBench, Swoop and Obo-edit, to create ontologies from textual scientific resources for LS plans. It also compares ontology methodologies in terms of Usability, Scalability, Stability, Integration, Documentation and Originality.

1. Introduction

Ontologies are increasingly being employed to enhance understanding and transferability of knowledge in Life Science [1–7]. Research groups are interested in developing ontologies [8] to arrange and connect data into searchable libraries [9, 10], to establish frameworks of essential concepts (e.g., OWS framework [11]), to share information, to use knowledge to minimize confusion [12] and to allocate the growing volume and data multiplicity in biomedical research, chemistry, medicine, genomics, biology and pharmacology [13]. Further reasons for appreciating its utility are a wider understanding of the value of standardized vocabularies and codified semantics in various domains [8] and the development or extension of construction tools. To enable widespread access [14] and give users the ability to publish, explore, create and update, a few works have been published regarding procedures [15–18] plans, criteria, activities and methodologies adapted to build them. The state of the art in ontologies, as well as the methods and challenges on semi-automatic and automatic ontology generation, have been widely examined by various articles [19–23], over the past few years. Publications on this topic have not presented, so far, a list of biomedical ontologies and open-source bioinformatics databases, also considering an overview of the ontology development tools (ODEs) needed for their investigation and comparing them. In this paper, we investigated the landscape of ontologies in the field of life sciences,

responding to the above proposition. Such ontologies are created and maintained by scientists to support the retrieval, integration and analysis of their data. The issues pertaining to ontology development, mapping and maintenance are critical key areas that must be comprehended and solved [24]. The thesis to be defended is that life science ontologies, here extensively explored, provide unequalled support for scientific study, and it is simple and intuitive for scientists to create or integrate ontologies. In this paper, we have been investigating and sharing efficient tools for achieving and building ontologies. We discuss the use of ontology developments (ODEs) from textual scientific resources, such as Protégé [25], Topbraid Composer [26], Ontostudio [27], Fluent Editor [28], VocBench [29], Swoop [30], and Obo-edit [31]. We begin by listing some available open source biomedical ontologies (Table 1) and bioinformatic databases (Table 2).

2. Biomedical ontologies and databases

Ontology is a multi-disciplinary field drawing upon the knowledge of natural language processing, information organization and extraction, artificial intelligence, knowledge acquisition and representation [19]. The commonly used ontology definition is adopted from Gruber [32] where an "ontology is a formal, explicit specification of a shared conceptualization". By serving as a common conceptualization, ontologies can lead to lower costs and better flexibility in data recognition and

* Corresponding author.

E-mail address: giulia.panzarella@unicz.it (G. Panzarella).

Table 1
Summary of specific ontologies.

Ontologies	Short Description
Uberon [48]	Open Biomedical and integrated cross-species ontology consisting of over 6,500 classes representing a variety of anatomical entities, allowing integration of model organism and human data.
Cell ontology [69]	Ontology for cell types in animals, not organism specific.
OMP [60]	Ontology of phenotypes observed in all microbes: bacteria, archaea, protists, fungi, and viruses.
ChEBI [70]	Freely available dictionary of molecular entities focused on 'small' chemical compounds. It includes an ontological classification.
EFO [71]	Experimental Factor Ontology. It is the core ontology for Open Targets [72]. The ontology pulls together classes from reference ontologies such as disease, cell line, cell type and anatomy and adds axiomatisation as necessary to connect areas such as disease to phenotype.
PATO [61]	An ontology of phenotypic qualities, mostly used in conjunction with other ontologies such as GO or anatomical ontologies to refer to phenotypes. E.g., it is used by the Human Phenotype Ontology (HPO) for logical definitions of phenotypes that facilitate cross-species integration.
GO [33]	The world's largest source of information on the functions of genes biological processes and cellular components.
FlyBase [62]	Database of Drosophila, Genes and Genomes. It uses Gene Ontology (GO) controlled vocabulary (CV) terms for cellular component, biological process and molecular function to describe properties of gene products.
WormBase [63]	Database concerning the genetics, genomics and biology of <i>C. elegans</i> and related nematodes.
CIA [73]	Ontology of the anatomy and development of <i>Ciona intestinalis</i> .
PLANA [74]	Extendable relational framework of defined <i>Schmidtea mediterranea</i> (Smed) anatomical terms, from subcellular to system levels across all life cycle stages.
NCI Thesaurus [55]	It is a reference terminology including broad coverage of the cancer domain, including cancer related diseases, findings and abnormalities.
SNOMED CT [56]	Systematized Nomenclature of Medicine—Clinical Terms. It is a comprehensive medical terminology ontology used for standardizing the storage, retrieval, and exchange of electronic health data.
ORDO [75]	Structured vocabulary for the Orphanet Rare Disease capturing relationships and genes.
Mondo Disease Ontology [76]	Semi-automatically ontology that merges multiple disease resources. It aims to harmonize disease definitions across the world.
AEO [77]	Ontology of anatomical entities that expands CARO, the Common Anatomy Reference Ontology, facilitating annotation of tissues that share common features, and enabling interoperability across anatomy ontologies.
ChemFonT [78]	Chemical Functional Ontology describing the functions and actions of more than 341.000 biologically important chemicals.

Table 2
Summary of bioinformatic databases.

Bioinformatic Databases	Short Description
OLS Ontology Search [53]	Ontology Lookup Service
OBO Library [79]	Open Biological and Biomedical Ontology
OMIM [57]	Public database of bibliographic information about human genes and genetic disorders.
BioPortal [80]	Repository of biomedical ontologies
AberOWL [81]	Ontology repository, semantic search engine
OntoBee [82]	A linked ontology data server to support ontology term dereferencing, linkage, query and integration
DiseaseCard [83] [84] [85]	Web-based tool for the collaborative integration of genetic and medical information
MalaCards [86] [87]	Integrated compendium for human diseases and their annotation
GeneCard [88]	Human Gene Database
DISEASES [89]	Text mining and data integration of dis-ease–gene associations
SIGNOR [90]	SIGNaling Network Open Resource Database of causal relationships between biological entities
KEGG [91]	Kyoto Encyclopedia of Genes and Genomes. Knowledge base for systematic analysis of gene functions, linking genomic information
MENTHA [92]	Resource for browsing integrated proteininteraction networks
PhosphositePlus [93]	Knowledge base dedicated to mammalian post-translational modifications (PTMs)
PhosphoELM [94]	Database of phosphorylation sites—update
UniProtKB [95]	Universal protein resource
HGMD [96]	Human Gene Mutation Database
CTD [97]	Comparative toxicological studies resource
PedAM [98]	Database for pediatric disease annotation and medicine

classification. In particular, when ontology provides a shared framework of common understanding of specific domains that can be communicated between people and application systems, then it can have a significant impact on areas, such as those biomedical and bioinformatics, dealing with large amounts of distributed and heterogeneous computer-based information. In this section, we present a descriptive list of existing ontologies (Table 1) and knowledge databases (Table 2) in the biomedical area. The use of ontologies as automatic reasoning mechanisms for database entities organization, management, understanding and representation, has proven to be advantageous in the biomedical field. The Gene Ontology (GO) [33] is the most extensive repository currently accessible for computable knowledge regarding the function of genes and gene products (proteins and non-coding RNAs). It offers sub ontologies for cellular components, biological processes, and molecular functions (CC). With the aid of GO-CAMs [34–36], users can link GO annotations [37], such as words for molecular functions, and enhance them with a biological context (Fig. 1).

For many species, GO annotations are provided via a variety of data sources, including GOA [38], Swiss-Prot [39], Ensembl [40], MGD [41] or AgBase [42]. GOChase [43] corrects errors in gene product annotations through the Gene Ontology. The GoPubMed application [44] also uses GO to categorize millions of articles in PubMed [45]. It enables biology domain specialists to read, search and update Open Biomedical Ontologies (OBO) [46] pertaining to anatomy, health, biochemistry or phenotype, including the cell ontology [47] and an increasing variety of species specific anatomy ontologies, like Uberon [48,49]. Uberon comprises the terminology for the anatomical domains of an expanding range of existing and emerging research organisms, allowing comparative evolutionary investigations. Online ontology tools like AmiGO [36] and QuickGO [50] can additionally query the Gene Ontology. The Ontology of Biological Reality framework (OBR), which uses the SNAP and SPAN ontologies [51], has also made suggestions for ways to enhance and integrate biomedical ontologies. Drug Ontology is one example of the approximately 277 life science ontologies that can be searched

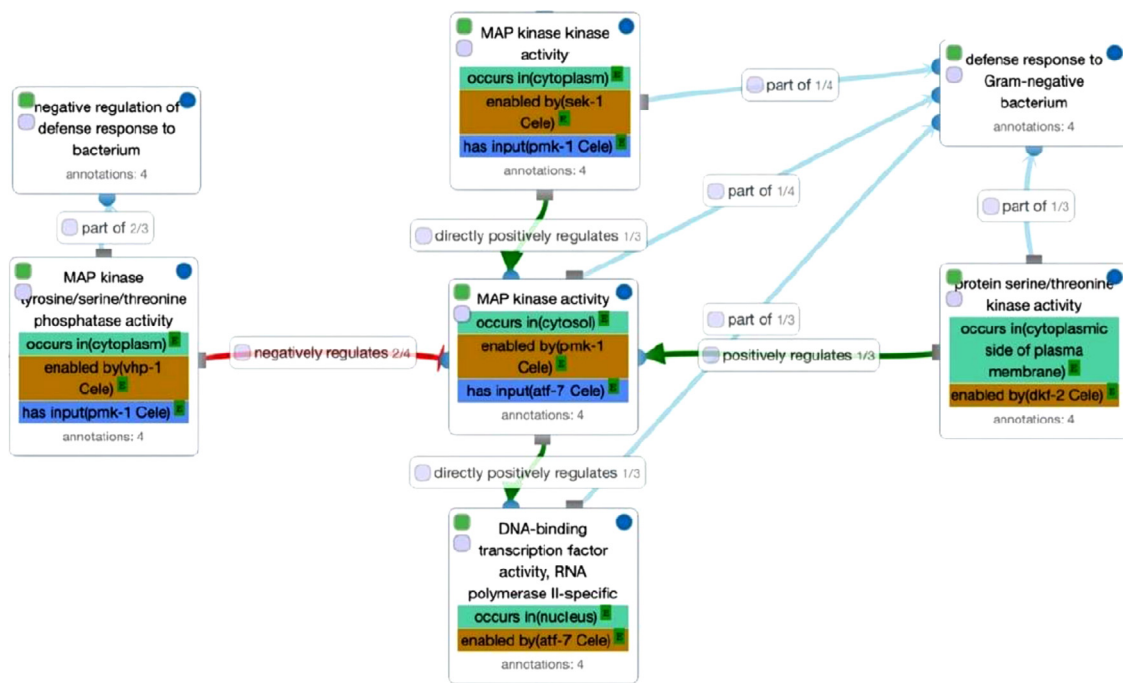


Fig. 1. Gene Ontology Causal Activity Model (GO-CAM) of 'C. elegans defense response to Gram negative bacterium in the intestine'. Extracted from [34].

and browsed using the EBI Ontology Lookup Service (OLS) [52,53] online portal. In Table 1 we report some OLS ontologies [54], species specific and biomedical ontologies such as the NCI Thesaurus [55], SNOMED CT [56] or OMIM [57] that can be considered for addressing clinical and disease related data. The NCI Thesaurus, for instance, is used by the caBIG project [58] to define metadata for exchanging items in their grid environment [59]. Among the species specific ontologies for research applications, we mention the Ontology of Microbial Phenotypes (OMP) [60], the Phenotypic Quality Ontology (PATO) [61], FlyBase [62] and WormBase [63]. In this regard, Textpresso Central [64] is an online literature search platform which covers specific research papers on model organisms such as the WormBase C. elegans bibliography [63]. Building an ontology for a research organism may seem like a big undertaking, but it is a necessary investment in the community. Data is acquired and handled according to FAIR, findable, accessible, interoperable and reproducible principles. FAIR [65] was born with the need to improve the infrastructure supporting the reuse of academic data. All researches can access data more easily adhering to FAIR practices. Anatomy ontologies are findable and accessible when available through the Ontology Lookup Service (OLS) [53]. Ontologies are interoperable when using relationships found in the Relationship Ontology [49, 66] and are Reproducible when reported in adherence to the Minimum Information for Reporting an Ontology (MIRO) practice [67]. Table 1 shows a summary of some species specific ontologies, e.g., the pathway ontology [68]. Table 2 shows a summary of bioinformatic databases.

2.1. Semantic web technologies

By the middle of 1970s, AI researchers focused on developing efficient strategies for acquiring, representing, and reasoning with large amounts of domain knowledge [99], overcoming the “knowledge acquisition (KA) bottleneck” [100], resulting in the new wave of Semantic Web of data. In this regard, the novel lines of research specified reusable models, i.e., ontologies, and reasoning components, i.e., problem-solving procedures [101, 102], which had an impact on contemporary Semantic Web technologies. The Semantic Web represents a technological infrastructure to enable large-scale data interoperability, discover ontologies

relevant to the task at hand and reuse of semantically characterized information [99]. Ontologies play a crucial role in enabling content-based access and standardized concepts reusability, describing Semantic Web entities, relationships, and categories of things. The term “Semantic Web” refers to a Web of linked data, e.g., dates, titles, part numbers, chemical properties, empowered by technologies such as RDF, GRDDL, SPARQL, OWL and SKOS. People are enabled to create data stores on the Web, build vocabularies, and write rules for handling data. Semantic Web has been gaining huge momentum in recent years, becoming widely applied to many domains such as medicine, health care, finance, geology [99, 103, 104]. Applying the appropriate ontology is considered necessary for getting the correct sense of the meaning of the entities or terms, avoiding ambiguity and any deviations that persist on the concepts. By aiding decision-making in clinical research, Semantic Web technologies can also bridge many forms of biological and medical information allowing vertical applications across institutions, industries and research communities [105].

2.2. Ontology languages and description logics

Since the 1990s, numerous ontology languages have been defined to allow NLP and support the Semantic Web [106]. Fig. 2 shows some of the defined languages divided into traditional syntactic ontology languages and markup ontology languages. The Knowledge Interchange Format (KIF) has been a revolutionary computer-oriented language for knowledge interchange among various computer programs [107]. The KIF syntax ontology language [108] is used additionally by the Ontolingua language [109]. Frame-Logic (F-Logic) stands for a subsequent frame-based and object-oriented language in order to characterize knowledge and data [110, 111]. Data representation and storage, database information exchange, data filtering and web service interaction [112] have also been skills well covered by the eXtensible Markup Language (XML) [113]. RDF (Resource Description Framework) [114] has been proposed to unify knowledge on the Web and to describe and exchange graph data. Furthermore, XOL [115], an XML-based ontology language, has been created. LOOM [116] has also been developed as a knowledge representation language, based on description logics (DL). DAML (DARPA Agent Markup Language) [117, 118] is

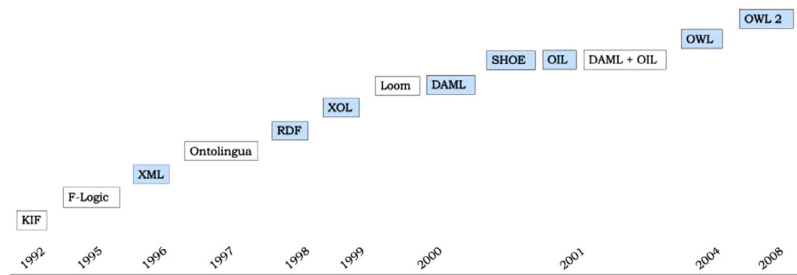


Fig. 2. Ontology languages timeline. Traditional syntax ontology languages are depicted in white; the Markup ontology languages (M-OLs) are depicted in blue [106]. M-OLs use markup schemes to encode knowledge, such as the widely used XML, which have pre-defined presentation semantics.

an ontology markup language aimed at supporting the development of the semantic web. SHOE is a knowledge representation language that allows web pages to be annotated with semantics [8]. By using a combination of DAML and OIL (Ontology Inference Layer) [119, 120] languages, we have the DAML + OIL [121] language. OWL (and subsequent version OWL 2) is another available language, developed by using DAML + OIL languages [122]. It was released and proclaimed as the standard one in 2008; that, still, has not changed [123]. Most effective ontology languages are built on the foundation of Description Logics (DLs) [124]. DLs are a family of knowledge representation languages [125] equipped with a formal, logic-based semantics. DLs reasoning can be applied both in the ontology design phase to enhance quality and in the deployment phase to take advantage of the rich structure of ontologies and ontology-based information. SHIQ Description Logics language, for instance, has formed the basis of several ontology languages, including OIL, DAML+OIL and OWL [126].

3. Ontology development environments (ODEs)

Ontology learning is becoming a sub-area of ontology engineering due to the exponential growth of web documents and the cutting-edge techniques shared by the machine learning, natural language processing, information retrieval, and artificial intelligence communities. Existing ontologies have generally been created manually. This method of creating ontology has traditionally been used by most ontology engineers, even if this procedure is exceedingly error-prone and time-consuming which makes it difficult to maintain and update ontologies. Due of this, researchers have constantly explored new approaches to produce or maintain and update ontologies in both efficient and effective ways [127]. This survey aims to highlight the main contributions on ontology generation, providing descriptions of the most popular and widely used ontology development environments (ODEs), including Protégé [128], Topbraid Composer [26], Ontostudio [27], Fluent Editor [28], VocBench [29], Swoop [30] and Obo-edit [31]. When an ontology is designed for the life sciences, these ODEs can be used as a starting point. Nowadays, Protégé is one of the most used ODE, with its web-based version, WebProtégé. The community includes more than 350,000 registered users [129]. It was developed at Stanford University where ODEs Ontolingua [108] and Chimaera [130] have been defined. A similar choice is the Topbraid Composer [26], even if the tool's free version is no longer available. A few years ago, OntoStudio [27] (now called OntoEdit [131]) was the initial iteration of this widely used application for building and maintaining ontologies. NeOn Toolkit [132], made public in 2006, was founded on it. It provides users with access to design pattern-based processes [133]. Its utility seems to fade currently [133]. OBO-Edit [31], a particular ontology editor for biologists, is another example of an ODE that is currently available. Other examples are VocBench [29], an open-source ODE that is still used by many businesses, and Fluent Editor [28]. Despite being advertised as a straightforward ontology editor on the W3C website, SWOOP [30] still lacks a project website. The following two ODEs emerged that have been crucial in the past: KAON2 [134] and WebODE [135]. We now report on some details of the graphical tools.

3.1. Protégé

The Stanford Center for Biomedical Informatics Research developed Protégé [128], an open-source ontology editor and knowledge store. It is a suitable instrument for editing ontologies since it gives users a thorough understanding of the OWL ontology structure. In the current landscape of ontology editors, it is the most complete and popular option. The National Institute of General Medical Sciences supports it as a national resource for biomedical ontologies and knowledge bases. We list some activity modules below.

- 1 Active ontology: enables the development of a brand-new ontology or the opening of an existing one by specifying the directory that it addresses (ontology IRI) or the URI (Ontology Version IRI). The associated source codes in the RDF/XML and OWL/XML languages can be examined and we can reorganize this data.
- 2 Entities module: it is the most crucial program panel. Without neglecting the hierarchies and values of each entity, it brings together classes of groups, properties and individuals all at once.
- 3 Individuals by class: it lists all the individuals included in the ontology. Individual Annotations and Individual Usage are examined in this section. We can add types or show whether there are distinct or similar individuals in the description. The Property Assertions box also allows the user to report positive or negative assertions for Object Properties and Data Properties.

Several commands are available in Protégé for creating entities; the user can choose from "Data types," "Individuals," "Classes," "Object properties," "Data properties," and "Annotation properties." Axioms must be inserted to produce classes having equivalent properties, including the union of their subclasses, to build equivalence relations. Regarding Exploration and Visualization features, Protégé can explore an ontology simply [25]. Categories such as Classes, Data Properties, Object Properties, Individuals, the Axiom Annotation and Query are used to sort the suggested layout. Protégé can substitute the names of the entities, which are strings that frequently lack any meaning, with labels also created automatically, using all annotations, values, or URI fragments as inspiration. Some of the Plugin and external supports, are listed below:

- 1 OWL Viz: it is an integral part of the standard Protégé 5.0 package, but it requires the installation of GraphViz; it allows one to view the classification of entities in a graph and it is also a interesting support in the study of relationships between the various entities. The resulting navigation chart can be saved and exported as an image file.
- 2 DL Query: it is reasonable to implement queries with DL query to verify the accuracy of the ontology structural integrity. One of the two accessible reasoners (e.g., FaCT++, Pellet) must be active when this support is invoked because, based on the OWL syntax, it can only function if the ontology is already categorized. Once this fundamental requirement has been met, the user can move forward with formulating and executing queries. The plug-in will collect all pertaining class data, properties or individuals into a single construct, or frame, along with hierarchy relationships, in order to locate the selected entity within the ontology.

- 3 Excel Import: this function inserts Excel or CVS files into the ontology and allows the user to build classes that encompass their contents and set arbitrary constraints and limits.
- 4 Cloud Views: this plug-in explains the structure and application of ontologies. The basic idea is simple: it implements a window where the major items are shown; the larger the font used, the higher the linked score. Examples of evaluation criteria are the use frequency of a single term, the position in the hierarchy, and the number of sub-entities.
- 5 Bookmarks: using bookmarks, you can drag classes and properties into a new view. The same entities are also kept as annotations so they can be used again.
- 6 OWLDoc: it comprises two components: OWLDoc View and OWLDoc Export. A dynamic view of any class, property, or selected individual can be created using OWLDoc View. When the presentation is over, OWLDoc Export transfers it to a directory so that a browser can open it. OWLDoc gathers a collection of static HTML pages that can be given to other users or published on the internet.

3.2. TopBraid composer

TopBraid Composer [136] is a visual, all-encompassing Knowledge Graph modeling environment and SPARQL query tool for building and maintaining domain models and ontologies in the Semantic Web standards RDF, RDFS and OWL. The free, standard and maestro versions of this program were all released in 2011. It has a framework that is adaptable and expandable with an API for creating browser-based or client/server semantic applications. All the TopBraid suite components operate within a growing open architectural platform that enables users to deal with files, databases, visual editors with class diagrams, ontology models, RDF data and syntax-directed text entries. The built-in OWL Inference engine, SPARQL query engine and Rules engine provide all testing, consistency checking and debugging. TopBraid Composer makes it simpler for businesses to migrate to Semantic Web standards by integrating legacy models like XML Schemas, UML, RDB Schemas and spreadsheets. It can operate with a database back-end to increase scalability and there are open APIs available. Protégé-like notions include the creation of forms for data collection that are based on schemas. There are several features available, such as a graphical editor, which may be used to easily construct the ontology. Additionally, classes and subclasses can be copied. The TopBraid Suite deployment platform is called TopBraid Live. It is a platform for Semantic Web applications with business service-oriented architecture (SOA) capabilities that enables dynamic, on-demand integration of data from many sources. TopBraid Live [137] offers a complex system to synchronize and cache RDF data between client and server, in addition to a comprehensive library of reusable Flex-based user interface elements, such as maps, calendars, forms, search forms, graphs, tables and trees.

3.3. OntoStudio

OntoStudio 3.2.0 [27] has been the widely used and accepted tool for building and maintaining ontologies a few years ago [106]. Its predecessor version was called OntoEdit [131]. It is a multilingual editor and ontology modeler with a lot of customization options. French, English and German are the languages that can define synonyms. Classes, properties, rules, queries and mapping are all part of it. This editor supports the OWL, RDF and ObjectLogic formats [27]. When starting a project, users have the option to choose their preferred storage type, including internal repositories or a collaborative server. Different file formats, such as xml, html, doc, ppt, pdf and postscript, can store data. With the use of a query builder, it provides the option to get data from databases, such as a SQL database. The availability of Query Builder SQL makes it simple to design queries. An external OntoStudio support tool called OntoBroker uses ObjectLogic and SPARQL to query the graph. SPARQL is a W3C (World Wide Web Consortium) standard and a query language for

RDF (SPARQL: SPARQL Protocol and RDF Query Language). It is possible to convert OWL to object logic, RDF/XML, Turtle, N-triples, n3, ANSI thesaurus, etc., using OntoBroker's ontology creation tools. OntoBroker can also communicate with programs written in .net and java.

3.4. Fluent editor

Fluent Editor [28] is an entirely free and open source. It offers Protégé interoperability by installing the necessary plug-in, allowing users to edit ontologies in natural language and working with the majority of the Semantic Web W3C standard [106]. Using the selected materialization profile, the user can display the ontology as an interactive diagram and immediately analyze the results. Fluent Editor gives the user access to the SWRL Debugger, a tool for creating actual ontologies and expressing intricate conditional relationships as examples. The first iteration of an embedded SWRL debugger has been made available by the developers to assist users while they model with SWRL. It enables the user to see which rules were put into action during materialization and which entities were employed in place of their head clauses and body. The user will concentrate on the real meaning of the ontology being updated (e.g., taxonomy, vocabulary, rule set). Using an R language program to access ontologies (rOntorion), Fluent Editor combines statistics with ontologies to perform quantitative and qualitative data analysis. Ontologies developed using Fluent Editor can be directly accessed through the rOntorion R package, which makes them available for semantic analysis in the R environment.

3.5. VocBench 3

VocBench 3 (VB) is a multilingual, collaborative development platform, with Semantic Web editing tools used to manage OWL ontologies, SKOS(/XL) thesauri, Ontolex-lemon lexicons, generic RDF datasets and linked data environments [29]. VocBench 3, maintained by the ISA2 program of the European Commission [138], is still supported and used by many public organizations, businesses and independent users to keep their thesauri, code lists and authority resources. The European Holocaust Research Infrastructure (EHRI) editorial Board has been used it for the EHRI holocaust thesaurus project [139] and so have the TESEO Italian Senate [140], the UNESCO Thesaurus [141] and the Unified Astronomy Thesaurus (UAT) from Harvard University [142], projects. It is an open-source; according to the official project repository, the most recent updates were made during the previous two years [29]. The platform strength has been its emphasis on collaboration, user job differentiation and workflow management for content validation and publication, especially for companies that need a disseminated but centrally managed publishing environment.

3.6. Swoop

The OWL ontology browser and editor called Swoop [30] was created at the American University of Maryland. The W3C website, described it as a compact and straightforward ontology editor, fully implemented and accessible for OWL. It has a web-oriented approach and includes many common browser features: address bar, history buttons, bookmarks and hypertext navigation. The primary panel of the application, which is divided into two tabs, is where most of the work is done. The Ontology Info panel provides general details about it, such as name, annotations, entity counts, and statistics on entity traits. The Species Validation panel depicts each entity and describes its traits or the nature it assumed in the ontology (class, property or individual). Three different display options are available for the ontology entities. One can create a comprehensive list of all existing entities, arranged alphabetically, or can examine class or attribute hierarchies, proposed with a tree structure. In addition, each phrase is linked to a symbol that designates the category (class, property or individual). Swoop users have the option to build a brand-new ontology from scratch or to use an already-traced

one by "browsing" directories or typing the URL into the navigation bar. In its workspace format, Swoop accounts simultaneously for various ontologies of the same activity and enables the development of projects. Swoop comes with several add-ons. One is the GCI (Generic Concept Illusion

Axiom); it allows the import and modification of ontologies with commands like "Changes/Annotations" and reports the evolution of the changes and related annotations (i.e., Annotation Annotate), or "Editable," allowing one to add, change, and delete data linked to entity characteristics. Data and object are two category properties. Individuals can work on Data type and Object assertions, instances and annotations by using commands planned for them.

Regarding exploration and visualization features, on the main page there is a keyword search field. A new window reports all the compared typed term entities. The content exploration is conducted through a hypertextual approach that transforms navigation into a real browsing. It is also possible to generate a graphical representation of the extracted ontology. Swoop has a "Resource Holder" panel, a window where one saves items. This panel can be useful if one needs to deal quickly with terminological overlaps and distinctions or find commonly used objects. Two reasoners are available for the control and classification of data:

- 1 Pellet: is an essential standard device package plug-in, regarded as more comprehensive. Protégé uses it by default as well. The Pellet Query command is displayed and can be accessed from the Advanced bar. It enables one to create RDQL language queries. The tree structure suggested in the main window is hyperlinked from the findings that are displayed.
- 2 RDFS-like: is based on RDFS semantics. It is lighter and faster compared to Pellet.

Swoop has Blackbox, a method to diagnose entities. Reasoners are tasked to develop a set of domains, in this method; the ontology structure is used to pinpoint a problem's root cause. In case of conflict, the program displays the event reason in the offending entity description field with the use of a semi-formal language and the insertion of some accountable axioms. Here, we can use the Run Debug Tests, Debugging / Explanation, and Repair Ontology commands in the Advanced tab. We can use Version Control to maintain our ontologies appropriately and to verify that their entities are used correctly. This tool offers two sorts of operations, changelogs and checkpoints; it allows users to keep various iterations of the same ontology. Changelogs describe the ontology's evolutionary route and list changes and adjustments in chronological order. Checkpoints provide quick switching between versions, showcasing the range of current models. Swoop supports all RDFS/OWL relational constructs in order to transmit entities to various ontologies. To reuse entities already present in other works, users are thus required to employ links rather than pure importations. Swoop does not currently support trading with shares, but a "simple copy-paste" could provide a workaround.

Swoop uses plug-ins for interpretative processes and reasoner control activities throughout the ontology examination.

- 1 Ontology Renderer Plugin: it specializes in ontology interpretation and can read the expressiveness of the logic description and the number of entities. It can also read existing annotations and identify them. Finally, it can identify the logical structures that establish the place of the ontology within the OWL species.
- 2 Annotea Framework: it accepts the idea of isolating the ontology annotations from the core; it permits the use of a particular RDF vocabulary that can be extended to the annotations and it offers a mechanism for their publication on servers inside the Annotea group. The result is the creation of a new annotation class called the Annotea Scheme. This class records ontology modifications and sends a warning message if anyone of them is not appropriate. The Ontology Graph Visualization Plug-in and Natural Language Entity Renderer are two additional plug-ins, helpful to the software as add-ons.

3.7. OBO-edit

The Gene Ontology (GO) Consortium created the OBO-Edit [31,79] project as a part of its Ontology [34]. Its goal is that of collecting as much terminology as possible from the scientific and biological domains. OBO-Edit deals with three specific topics: biological processes, cellular components, and molecular activities. The version we cover in this paper is the 2.00 beta 49. One completely customizable window contains all the OBO-Edit graphics. New plug-ins can be added to the ones recommended by the application. Users have the option either to upload an existing ontology or to create a new one. It can be got by downloading particular referral packages from the website and specific ontology portions (regularly updated and maintained by developers), or created a new [143]. Information can be heterogeneous and can contain both OWL language traditional ontology elements and content. The hard drive is used only to write or save files; files can just be read from a local disk or a URL address. A variety of editors have been included in the application. The Tree Editor is the program's most crucial panel for content management and navigation. Terms have a graphic symbol and entities are arranged in hierarchical structures. A diagram representing the ontology morphology is proposed by the Graph Editor panel. A user can change entities and relationships directly on the graph. The Text Editor allows the user to change the single characteristic entities and the Parent Editor enables to change the properties of the chosen entity, such as the super-entity, which includes links, words and namespaces. OBO-Edit contains also particular methods like the modification of entity namespaces, IDs, the insertion of roots into ontology paths and the true/false Boolean property in certain properties. OBO-Edit offers two distinct concepts of canceling: the Deleting command, which removes a relationship from the ontology by bringing the associated entity to the obsolete terms; and the Destroying operation, which permanently deletes an existing entity from the ontology. The research section uses two panels:

- 1 The Search Panel, which is used to search terms within the ontology.
- 2 Link Search Panel, which is needed for the investigation of the type of relationships among parent and children's ID entities.

The Viewers bar commands: Term Image Display Panel, Graph, Graphviz and Tree Viewer are dedicated to the hierarchical structure for the visualization of entities. OBO-Edit may generate a narrative about the creation and evolution by placing the actions in chronological order. Every time a user makes a change, the application creates a history, which records all the information pertaining to the action taken, by adding it to a list available in the "History Browser". OBO-Edit offers a different selection of reasoners than Protégé and Swoop, since it is not a plug-in, but an integral part of the code internally calculated by developers. Forward Chaining or Link Pile Reasoners, play a crucial control role in OBO-Edit which is also equipped with the so-called "Semantic Parser Manager" algorithm that has the function of carrying out the purely syntactic analysis of terms. Controls and inspections are also carried out through Ontology Verification, which immediately alerts the user of fatal flaws that could jeopardize accuracy. Checks are performed automatically during very specific stages of the ontology's life, such as when it is loaded, when text fields are changed and while using the Text Editor, that also keeps track of feedback on the actions taken. There are many different control studies, that range from the examination of the name of terms and associated IDs, to glancing for synonyms and disjunction features. If a user needs to manually engage with the program control system he must utilize the Verification Manager. The window contains a list of all kinds of inspection allowed. Finally, we mention the idea of a cross product, also present in Protégé and Swoop, but with a somewhat different meaning in this situation. In OBO-Edit Cross Product represents the intersection or relations among the classes that ensure the identification of all the hierarchical correspondences within the ontology. In Protégé and Swoop,

Cross Product defines a relationship between entities of two separate ontologies.

3.8. Software comparison

We now provide a general ODEs comparison. Table 3 collects general discussed tools features. Table 4 collects tools aspects of Usability, Scalability, Stability, Integration, Documentation and Originality. Each characteristic is evaluated with a score ranging from a minimum of 1 to a maximum of 3 yellow stars.

3.8.1. Usability

Usability concerns access simplicity to software utilization. Software settings and graphical interface have also a significant role in this matter. OntoStudio and TopBraid Composer are available on the market, while Protégé, Fluent Editor, VocBench, Swoop and Obo-edit are open-source applications. Swoop is web based and Protégé, OntoStudio and Swoop have client/server architecture. Protégé, OntoStudio and TopBraid Composer Free Edition (FE) use databases for storing ontologies. Our analyses will be concerned more with open-source tools. Protégé is a complete, all-encompassing software suite in terms of usability. It offers a great graphical interface and enables in-depth ontology study. FluentEditor provides Protégé with interoperability, ontology editing in natural language, support for plug-ins, compatibility with most of the Semantic Web W3C standard and many other features and is free for academic and personal application. VocBench is also another ODE, used and supported by many organizations. It is an open-source; according to the official project repository, its most recent revision dates in the current calendar year. TopBraid Composer is also going strong, mostly profiting from a proprietary version. There was a free version, but it is no longer maintained. A few years ago, OntoStudio was the most widely used commercial program for building and maintaining ontologies. It is noteworthy that the NeON Toolkit was based on it and its predecessor was the widely used ODE OntoEdit. Despite certain content and functionality limitations, Swoop is restricted to individuals who do not need to perform very complicated operations on their ontology, as a result it is considered the most friendly tool. Its programmers created a layout and a complete application that substantially resemble the guiding principles of a web browser. OBO-Edit has also been planned to be friendly, permitting to meet its users' needs. Therefore, it appears to be more than exhaustive for its aims, even though it turns out to be the least advanced software package in terms of usability. In OBO-Edit a user can choose from a variety of ontology access methods, that include downloading definite referral packages, specific ontology components, or completely new ontology [143].

3.8.2. Scalability

Scalability is when applications can integrate software packages or plug-ins. All the programs examined in this study are planned intending to allow additional supports and/or plug-ins to enhance their performance. Protégé is certainly the program with greater potential improvement because of its extensive use; it is the most significant product in the industry; many institutions are working on initiatives to develop plug-ins for this software package. Therefore, it has taken the lead in the research and education fields. The Protégé plugin library contains several items; one of such plugins is SKOSed [144]. Swoop, Fluent Editor and VocBench are also working to develop plug-ins, dealing with large amount of data, while still offering a friendly environment. OBO-Edit is the only program that handles this problem less, as it was designed for a specific ontology rather than for being a generic editor. TopBraid Composer is based on an open plugin structure. Advanced features in this matter are available on its Standard (SE) and Maestro Editions (ME). Fluent editor provides two plug-ins viz., a Protégé interoperability plug-in and an Rplug-in using the ROntorionpackages to organize the ontology content. VocBench3 support plugins which provide additional capabili-

Table 3
Software comparison representation.

Tools features	Protégé	Tobraid composer	Ontostudio	Fluent editor	Vocbench 3	Swoop	Obo-edit
Supporting platform	Windows, Linux, MacOSX, Sun, Solaris, HPLIX, IBM	Windows, Linux, MacOSX	Windows XP, Vista, Windows 7, Windows Server 2003, SUSE, Linux 10.2	Windows platform	Windows platform, MacOSX	Windows, Linux	Windows, Linux, MacOSX
Supporting file formats	RDF/XML, OWL/XML, OWL Functional syntax, Manchester OWL syntax, OBO 1.2 flat file, KRSS2 syntax, Latex and Turtle (Terse RDF Triple Language) Ontograf, OWLviz, OWLGred for UML	OWL, RDF, Turtle, n-triple, XML	OWL, RDF, F-Logic, ObjectLogic	OWL, RDF, RDFS, OWL2	OWL, RDF	OWL, RDF, XML, text, SWOOP, ontology object file	OWL, GO Flat File Format, Java serial files
Graphical representation	DL query	Supported	Supported	Supported	Graph View, Ontotext GraphDB SPARQL	Not available	GraphViz plugin
Query support		SPARQL, SPIN (SPARQL Inferencing Notation)	SPARQL, ObjectLogic query, Query Builder	SPARQL, SKOS, SWRL, ROntorion Controlled Natural Language (OCNL)	SPARQL	Pellet Query	DL query
Reasoner	DL Reason e.g., Fact ++, Hermit, Pellet	Pellet, Jena, TOPSPIN, SwiftOWLIM	Pellet	Hermit	RDF4J reasoner	Pellet, Racer, Fact	OBO-Edit Reasoner (Amina)
Programming languages interface	Protégé API, Protégé Script Tab	HTML, Javascript, Adobe Flex	.net, Java program	R Language	Java program	Java program	Java program
Availability	Open-Source	Licensed version	Licensed version	Open-Source	Open-Source	Open-Source	Open-Source
Multi user support	Supported	Supported	Not Supported	Supported	Supported	Not Supported	Not Supported
Available languages to define synonyms	English	English	English, French, German	English	Multilanguages	English	English
Root node	Thing	Owl:Thing	Owl:Thing	Owl:Thing	Owl:Thing	Owl:Thing	Owl:Thing

Table 4
Tools evaluation table.

FEATURES SOFTWARE COMPARISON	PROTÉGÉ	TOBRAID COMPOSER	ONTOSTUDIO	FLUENT EDITOR	VOCBECH 3	SWOOP	OBO-EDIT
USABILITY	★★	★	★	★★	★★	★★★★	★★
SCALABILITY	★★★★	★★	★★	★★★★	★★★★	★★	★★
STABILITY	★★★★	★★	★★	★★★★	★★	★★★★	★★
INTEGRATION	★★★★	★★	★★	★★★★	★★★★	★★	★
DOCUMENTATION	★★★★	★★	★	★★★★	★★★★	★	★★★★
ORIGINALITY	Entities Panel, Integration systems, e.g., Reasoners	Provides scalable database backends (Allegro-Graph, Jena, RDF4) and multi-user support	The tool allows the user to edit a hierarchy of concepts or classes and is based on an open plug-in structure	Custom plugins, R plugin that uses the ROntorion packages to plot and list the current content of the ontology, Protégé integration	Manages OWL ontologies, SK OS/SK-OS-XL thesauri, OntoLex lexicons and any sort of RDF datasets	Web-oriented Layout, Resource Holder function-ality	Graph Editor panel, Obsolete function

ties, e.g., enable to export data to an FTP server, or introduce a further serialization format, e.g., MADS [145].

3.8.3. Stability

In terms of stability, Protégé is the most complete and superior program so far created. A user can make modifications in total safety. FluentEditor, VocBench and SWOOP are also very reliable and secure programs; SWOOP varies from Protégé mostly in terms of command complexity. Only strictly necessary operational commands have been provided concerning OBO-Edit. Therefore, working data and ontology contents may become unstable and less secure and several actions or even results are incredibly carried out slowly in the application crashing. Swoop is web based and Protégé, OntoStudio and Swoop have client/server architecture. Protégé, OntoStudio and TopBraid Composer (FE) use databases for storing ontologies. TopBraid Composer use a standalone Eclipse plug-in as semantic web architecture, based on an open plug-in structure. TopBraid Composer Standard Edition (SE) includes basic features available in the free version plus graphical viewers, import facilities and plug-in. The Advanced Maestro Edition (ME) includes all features of SE plus support for TopBraid Live, Ensemble as well as SPARQLMotion. Ontostudio supports multilingual development, and the knowledge model is related to frame-based languages. Data model representation can be exported to DAML+OIL, RDF(S), F-Logic and OXML. Additionally, ontologies can be transferred to relational databases via JDBC.

3.8.4. Integration

“Integration” refers to the combination of two or more ontologies into a single one; it gives special consideration to the shared reuse of information. Protégé is the technology that further meets the goals of ontology integration, even though it has a tiny and entirely independent structural footprint. The semantic web has established this as one of its main objectives. Protégé, FluentEditor and VocBench are all

equipped with the capability to shape and combine various ontologies, thus ensuring the potential of developing shared and reusable information archives. Swoop’s predisposition to integration is less clear; it follows different principles. It has commands for the fusion of ontologies, but developers were mainly concerned with operations for the reuse of contents within the same ontology: the Swoop “Resource Holder” tool. OBO-Edit integrations also regard contents in the same ontology. OntoStudio allows the development and maintenance of ontologies by using graphical means, based on client/server architecture. In OntoStudio the ontologies are managed in a central server and users can access and redesign the ontologies. TopBraid Composer (FE) may use in a single user mode working with ontologies stored as files or in a database.

3.8.5. Documentation

The Protégé user manual is relatively straightforward; it contains an introduction that explains all theoretical and conceptual aspects of ontologies and languages for the semantic web. This site has produced good, comprehensive updated work, becoming thus a great resource for all Protégé users. Swoop, OntoStudio and TopBraid Composer (FE) editors provide documentation ontology and ontology libraries. A user’s guide to the program is provided by Fluent Editor, VonBench and OBO-Edit. The huge and well-organized materials are made possible by frameworks with hypertext links.

3.8.6. Originality

Originality, refers to unique software qualities and/or design peculiarity. With Protégé, we refer to the Entity panel as a prominent graphic component that gives access to all different ontology entity types. Ranging with reasoners and operations to combine multiple ontologies, these processes ultimately lead to the application of a variety of plug-ins and external supports, without compromising the operation. The part concerning integration development, starting from reasoners and operations for merging multiple ontologies, that lead to the opportunity to

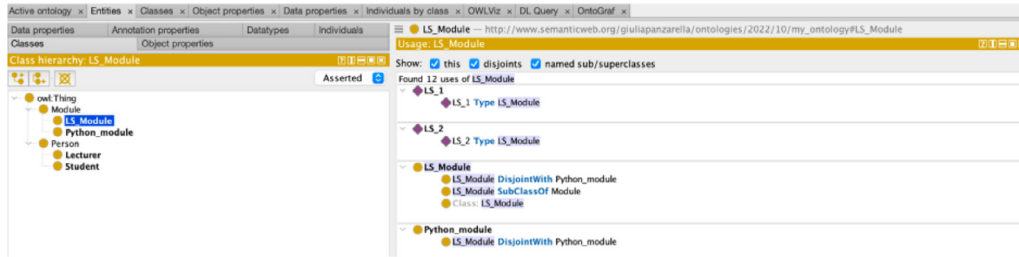


Fig. 3. Creating the ontology classes.

use a wide range of plug-ins and external supports, without risk in the application, is worth mentioning. Swoop allows users to write and share annotation on any ontological entity. Swoop's Web-oriented architecture, which resembles the design of web browsers, can be recognized as a unique layout feature. The second Swoop element highlighted is the Resource Holder, a window where entities a user wants to keep track of due to their usefulness and benefits, can be saved. In relation to OBO-Edit, we refer to the Graph Editor panel. It gives the option to change the information in a graph; that is the most effective method for working on particularly large ontologies, where one can easily get lost. The OBO-Edit Obsolete root is the other feature we highlighted. It allows to collect all considered outdated entities that lack past validity. Any term can be moved to this group thanks to the Deleting command. This is a very original and intuitive idea. The distinction between Deleting and Destroying instructions is another original cue. Deleting moves data to the source of obsolescence; destroying removes permanently an entity from the ontology. In Fluent Editor we want to mention the available of two different interfaces for ontology exploration: a reasoner graph, using an OWL-DL reasoner and a materialized graph, using a Jena-based OWL-RL+ reasoner. By default Fluent Editor loads Hermit [146], but it is possible to implement a simple C interface to add other reasoners to Fluent Editor [147]. In both cases, the user can make OCNL queries to the ontology and the results are displayed in OCNL. The interesting thing about VocBench 3 is that, with very low installation requirements and an as-short-as-possible time-to-use, offers a collaborative environment for development of any kind of RDF dataset, with dedicated facilities for ontologies, thesauri and lexicons [148]. In OntoStudio, ontologies are assessed using the OntoBroker inference engine. By using this engine, OntoStudio takes advantage of F-advantages Logic's and is able to represent expressive rules. Using the OntoBroker Enhancement Collaborative server, OntoStudio offers collaborative ontologies. TopBraid Composer(FE), as well as Protégé and Swoop, have external attached inference engines. TopBraid Composer (FE) uses the Exception Handling.

4. Practical application

Our practical application will regard Protégé Desktop (v. 5.5.0), the most comprehensive and widespread alternative in the panorama of ODEs [15].

4.1. Protégé

First, we specify the ontology IRI and create ontology classes. We want to create an easy ontology describing the people involved in teaching some modules and the people involved in studying the modules. For the applying example we will create two classes, i.e., subclasses of class "thing", e.g., "person" and "module". The subclasses of "module" will be "LS Module" and "Python Module". The subclasses of "person" will be: "student" and "lecturer". Then, we need to make classes disjoint to each other so we can avoid multiple inheritance and overlapping classes or subclasses (Fig. 3). We proceed by creating some object properties and data properties. Object properties describe relationships between two instances or two individuals of classes, whereas data properties describe

relationships between instances and data values. For this ontology example, we have two object properties: "studies" and "teaches" (Fig. 4). "A student studies a module", "a lecturer teaches a module". We need to specify the range and domain for each object property. Regarding "teaches", as shown in Fig. 4, the domain will be "lecturer" and the range will be "module". The domain of the object property "studies" will be "student" and "module" will be the range. Here, all the object properties are sub-properties of "top object property". Data properties describe relationships between instances or individuals and data values. Data properties represent the data values of our classes. First name, surname and student ID number can be "student" data properties. Staff ID can be a "lecturer" data property. In the example, for student ID data property, we set "student" as domain and "integer" as range (Fig. 5). We then create individuals: e.g., Lecturer 1 and Lecturer 2 (Fig. 6).

After creating our classes, our object properties, our data properties, our individuals, object property assertions and data property assertions to our individuals, we can activate the Reasoner; it checks if everything complies with the ontology. You can additionally make available online your ontology so you can access it using sparkle query. For visualizing the developed ontology, we start by searching or selecting a term in the tree of "OntoGraf" section, e.g., module. The graph corresponds to the OWL description (Fig. 7).

Relationships can be explored through incremental expansion of the graph. Nodes 'bubble' automatically to help with exploration. Relationships can be filtered to help reduce graph complexity. We can clean up the graph by removing orphaned nodes and we can narrow our focus by showing just the "neighborhood" of a term. Expansion can be restricted to certain relationship types. Different layout can be applied, e.g., with the spring layout, each expansion reorders the graph. In addition, we can filter all constraints and object properties.

5. Conclusion

The paper goal is to provide a sign of the evolution of ontology engineering over the years. Owing to space restrictions, only some of the major ontology engineering topics could be covered. We hope that the paper can also act as a starting point for a newcomer in the ontology field and a quick reference for experienced academics. Ontologies are being used in a variety of fields, that range from healthcare, to engineering and to finance, because of the past decades research and development activities. The Web evolution is unstoppable; from a rigid and static structure, it is being transformed into an increasingly dynamic and flexible container of information, simulating an eager human intelligence. In this scenario, ontologies and editors make up indispensable bases to plan a new era of "knowledge representation". Ontology has been built to organize knowledge that is being generated from the conduction of systematic reviews, studies, and research. The development of definitions of formal axioms and rules, knowledge acquisition and the documentation processes are important stages throughout the whole ontology building life cycle, as well as the evaluation process. It aims at guaranteeing a higher degree of consistency and robustness of the conceptualization and must be enhanced. Moreover, there aren't many technologies available to assist in knowledge extraction from a

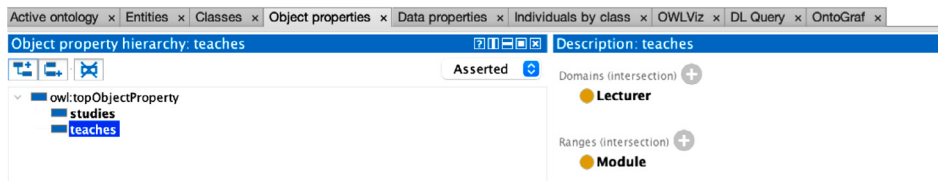


Fig. 4. Adding object properties.

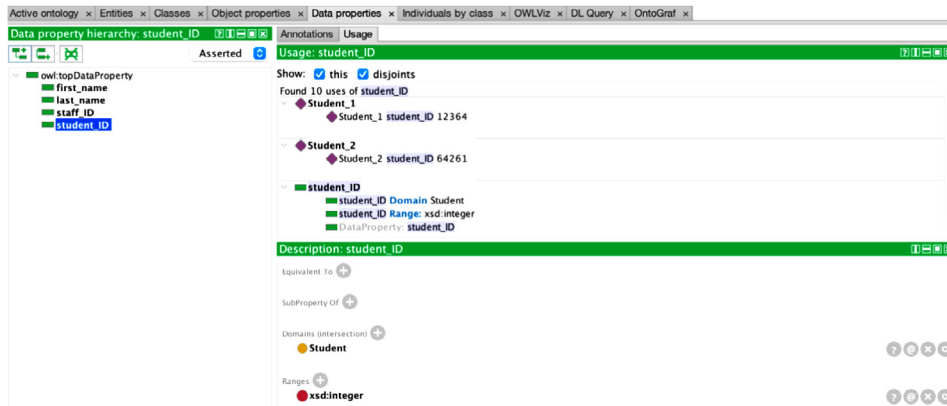


Fig. 5. Adding data properties.

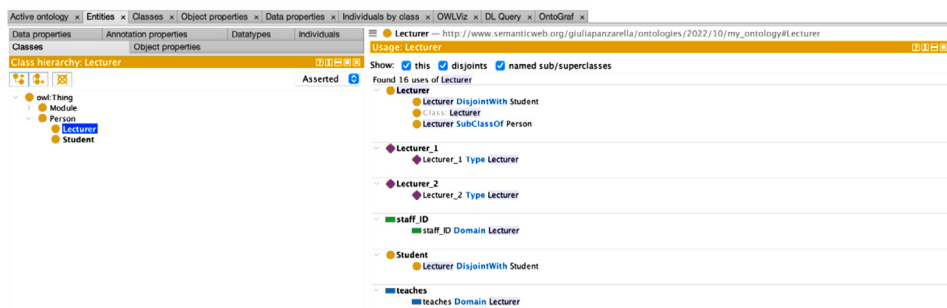


Fig. 6. Adding individuals by class.

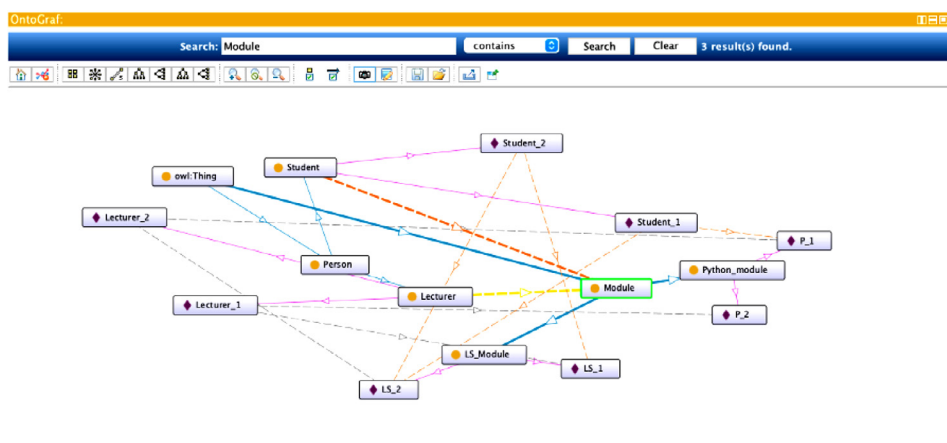


Fig. 7. OntoGraf relationships.

variety of implicit sources, including articles, web papers, and operating procedure documents. Locating and retrieving such knowledge that is unlabeled, dispersed, unorganized, ambiguous, incomplete is difficult [149], but would be extremely interesting for Life Science (LS) applications. Ontology is the core to knowledge retrieval and LS Ontologies can improve the communication of biological, chemical and bioinformatic knowledge, elaborate a conceptual framework for understanding concepts differences and similarities, help academia to set the scope of their research. During a study is extremely important to identify the

general characteristics and common knowledge regarding experimental studies, as well as to allow the communication among users, opening the opportunity to explore e-science concepts into the LS experimentation domain. Potential future research directions would be to automate the process of creating the ontology. Beyond the creation of new ontologies, another challenging area of work in the subject of ontologies is the evolution and mapping of current ontologies. Due to the cost of creation, abstraction, and reusability, this is significant. Different types of ontologies can be derived from a core ontology to suit particular appli-

cation domains. Use of ontologies and semantic web technologies need to become ubiquitous as well as more publishers need to curate literature using ontologies, tackling the problem of semantically generating comprehensive overviews of research findings in a structured and comparable way [150].

Financial and ethical disclosures

This research is neither fully nor partially supported by any funding organization or agency. G.P. acknowledge PON FSC for the PhD grant. P.V. and S.A. acknowledge PON VQA.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

No data was used for the research described in the article.

Acknowledgments

Authors acknowledge Prof. Gregorio Sia for proof reading the article.

References

- [1] Leite ML, de Loiola Costa LS, Cunha VA, Kreniski V, de Oliveira Braga Filho M, da Cunha NB, Costa FF. Artificial intelligence and the future of life sciences. *Drug Discovery Today* 2021;26(11):2515–26.
- [2] Chibucos MC, Mungall CJ, Balakrishnan R, Christie KR, Huntley RP, White O, Blake JA, Lewis SE, Giglio M. Standardized description of scientific evidence using the evidence ontology (eco). *Database* 2014;2014.
- [3] Bard JBL, Rhee SY. Ontologies in biology: design, applications and future challenges. *Nat Rev Genet* 2004;5(3):213–22.
- [4] Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, Lomax J, Mungall C, Neuhaus F, Rector AL, Rosse C. Relations in biomedical ontologies. *Genome Biol* 2005;6(5):1–15.
- [5] Zhao J, Goble C, Stevens R. An identity crisis in the life sciences. In: *International Provenance and Annotation Workshop*. Springer; 2006. p. 254–69.
- [6] Smith B, Köhler J, Kumar A. On the application of formal principles to life science data: a case study in the gene ontology. In: *International Workshop on Data Integration in the Life Sciences*. Springer; 2004. p. 79–94.
- [7] Bajorath J. Artificial intelligence in interdisciplinary life science and drug discovery research. *Fut Sci OA* 2022;8(4):FSO792.
- [8] Tudorache T. Ontology engineering: Current state, challenges, and future directions. *Semant Web* 2020;11(1):125–38.
- [9] Bodenreider O, Burgun A. Biomedical ontologies. In: *Medical Informatics*. Springer; 2005. p. 211–36.
- [10] Poggi A, Lembo D, Calvanese D, De Giacomo G, Lenzerini M, Rosati R. Linking data to ontologies. In: *Journal on data semantics X*. Springer; 2008. p. 133–73.
- [11] Dameron O, Fridman Noy N, Knublauch H, Musen MA. Accessing and manipulating ontologies using web services. *Sws at iswc*; 2004.
- [12] Bateman JA. On the relationship between ontology construction and natural language: a socio-semiotic view. *Int J Hum Comput Stud* 1995;43(5):929–44.
- [13] Schulze-Kremer S. Adding semantics to genome databases: towards an ontology for molecular biology. In: *Ismb*, 5; 1997. p. 272–5.
- [14] Neches R, Fikes RE, Finin T, Gruber T, Patil R, Senator T, Swartout WR. Enabling technology for knowledge sharing. *AI Mag* 1991;12(3) 36–36.
- [15] N. F. Noy, D. L. McGuinness, et al. *Ontology development 101: A guide to creating your first ontology*, 2001.
- [16] Oberle D, Volz R, Staab S, Motik B. An extensible ontology software environment. In: *Handbook on ontologies*. Springer; 2004. p. 299–319.
- [17] Pinto HS, Staab S, Tempich C. Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In: *ECAL*, 16. Citeseer; 2004. p. 393.
- [18] Iqbal R, Azrifah M, Murad A, Mustapha A, Sharef NMohd, et al. An analysis of ontology engineering methodologies: a literature review. *Res J Appl Sci Eng Technol* 2013;6(16):2993–3000.
- [19] Ding Y, Foo S. Ontology research and development. part 1-a review of ontology generation. *J Inf Sci* 2002;28(2):123–36.
- [20] Maniraj V, Sivakumar R. Ontology languages-a review. *Int J Comput Theory Eng* 2010;2(6):887.
- [21] Strömert P, Hunold J, Castro A, Neumann S, Koepler O. Ontologies4chem: the landscape of ontologies in chemistry. *Pure Appl Chem* 2022.
- [22] Alatrish ES. Comparison some of ontology. *J Manage Inf Syst* 2013;8(2):018–24.
- [23] Kumar VK, Ramya D. Ontology editing tools: a comparative perspective. *Int J Comput Sci Eng* 2018.
- [24] Smith B. Ontology (science). *Nature Prec* 2008 1–1.
- [25] Gennari JH, Musen MA, Fergerson RW, Grosso WE, Crubézy M, Eriksson H, Noy NF, Tu SW. The evolution of Protégé: an environment for knowledge-based systems development. *Int J Hum Comput Stud* 2003;58(1):89–123.
- [26] Top Quadrant Inc. *Topbraid composer*, 2007.
- [27] Weiten M. Ontostudio® as a ontology engineering environment. In: *Semantic knowledge management*. Springer; 2009. p. 51–60.
- [28] *Cognitum fluenteditor*, 2012 <http://www.cognitum.eu/semantics/FluentEditor/>.
- [29] Stellato A, Rajbhandari S, Turbati A, Fiorelli M, Caracciolo C, Lorenzetti T, Keizer J, Pazienza MT. Vocbench: a web application for collaborative development of multi-lingual thesauri. In: *European semantic web conference*. Springer; 2015. p. 38–53.
- [30] Kalyanpur A, Parsia B, Sirin E, Grau BCuenca, Hendler J. Swoop: a web ontology editing browser. *J Web Semant* 2006;4(2):144–53.
- [31] Day-Richter J, Harris MA, Haendel M, Ontology Gene. OBO-Edit Working Group, and Suzanna Lewis. Obo-edit—an ontology editor for biologists. *Bioinformatics* 2007;23(16):2198–200.
- [32] Gruber TR. A translation approach to portable ontology specifications. *Knowl Acquisition* 1993;5(2):199–220.
- [33] Gene Ontology ConsortiumThe gene ontology (go) project in 2006. *Nucleic Acids Res* 2006;34(suppl 1):D322–6.
- [34] The gene ontology resource: enriching a gold mine. *Nucleic Acids Res* 2021;49(1):325–34.
- [35] Mi H, Muruganujan A, Ebert D, Huang X, Thomas PD. Panther version 14: more genomes, a new panther go-slim and improvements in enrichment analysis tools. *Nucleic Acids Res* 2019;47(D1):D419–26.
- [36] Carbon S, Ireland A, Mungall CJ, Shu S, Marshall B, Lewis SAMigo hub, web presence working group. AmiGO: online access to ontology and annotation data. *Bioinformatics* 2009;25(2):288–9 2009.
- [37] Gene Ontology ConsortiumGene ontology annotations and resources. *Nucleic Acids Res* 2012;41(D1):D530–5.
- [38] Barrell D, Dimmer E, Huntley RP, Binns D, O'Donovan C, Apweiler R. The goa database in 2009—an integrated gene ontology annotation resource. *Nucleic Acids Res* 2009;37(suppl 1):D396–403.
- [39] Bairoch A, Boeckmann B, Ferro S, Gasteiger E. Swiss-prot: juggling between evolution and stability. *Briefings Bioinf* 2004;5(1):39–55.
- [40] Hubbard TJP, Aken BL, Ayling S, Ballester B, Beal K, Bragin E, Brent S, Chen Y, Clapham P, Clarke L, et al. Ensembl 2009. *Nucleic Acids Res* 2009;37(suppl 1):D690–7.
- [41] Bult CJ, Eppig JT, Kadin JA, Richardson JE, Blake JAMouse Genome Database Group. The mouse genome database (mgd): mouse biology and model systems. *Nucleic Acids Res* 2008;36(suppl 1):D724–8.
- [42] McCarthy FM, Bridges SM, Wang N, Magee GBryce, Williams WP, Luthe DS, Burgess SC. Agbase: a unified resource for functional analysis in agriculture. *Nucleic Acids Res* 2007;35(suppl 1):D599–603.
- [43] Rang Park Y, Hee Park C, Han Kim J. Gochase: correcting errors from gene ontology-based annotations for gene products. *Bioinformatics* 2005;21(6):829–31.
- [44] Doms A, Schroeder Michael. Gopubmed: exploring pubmed with the gene ontology. *Nucleic Acids Res* 2005;33(suppl 2):W783–6.
- [45] Canese K, Weis S. Pubmed: the bibliographic database. *NCBI Handbook* 2013;2(1).
- [46] Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, Goldberg LJ, Eilbeck K, Ireland A, Mungall CJ, et al. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol* 2007;25(11):1251–5.
- [47] Bard J, Rhee SY, Ashburner M. An ontology for cell types. *Genome Biol* 2005;6(2):1–5.
- [48] Haendel M, Gkoutos G, Lewis S, Mungall C. UBERON: towards a comprehensive multi-species anatomy ontology. *Nature precedings*; 2009. 1–1.
- [49] Mungall CJ, Tornia C, Gkoutos GV, Lewis SE, Haendel MA. UBERON, an integrative multi-species anatomy ontology. *Genome Biol* 2012;13(1):1–20.
- [50] Binns D, Dimmer E, Huntley R, Barrell D, O'donovan C, Apweiler Rolf. Quickgo: a web-based tool for gene ontology searching. *Bioinformatics* 2009;25(22):3045–6.
- [51] Rosse C, Kumar A, Mejino JLV Jr, Cook DL, Detwiler LT, Smith B. A strategy for improving and integrating biomedical ontologies. In: *AMIA Annual Symposium proceedings*, 2005. American Medical Informatics Association; 2005. p. 639.
- [52] Côté RG, Jones P, Apweiler R, Hermjakob H. The ontology lookup service, a lightweight cross-platform tool for controlled vocabulary queries. *BMC Bioinf* 2006;7(1):1–7.
- [53] Jupp S, Burdett T, Leroy C, Parkinson HE. A new ontology lookup service at embl-bi. In: *SWAT4LS*, 2; 2015. p. 118–19.
- [54] Côté RG, Jones P, Martens L, Apweiler R, Hermjakob H. The ontology lookup service: more data and better tools for controlled vocabulary queries. *Nucleic Acids Res* 2008;36(suppl 2):W372–6.
- [55] Sioutos N, de Coronado S, Haber MW, Hartel FW, Shaiu Wen-Ling, Wright Lawrence W. Nci thesaurus: a semantic model integrating cancer-related clinical and molecular information. *J Biomed Inform* 2007;40(1):30–43.
- [56] Donnelly K, et al. Snomed-ct: The advanced terminology and coding system for health. *Stud Health Technol Inform* 2006;121:279.
- [57] Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res* 2005;33(suppl 1):D514–17.
- [58] Fenstermacher D, Street C, McSherry T, Nayak V, Overby C, Feldman M. The cancer biomedical informatics grid (cabig tm). In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE; 2006. p. 743–6.
- [59] Hartung M, Kirsten T, Gross A, Rahm E. Onex: Exploring changes in life science ontologies. *BMC Bioinf* 2009;10(1):1–10.

- [60] Chibucos MC, Zweifel AE, Herrera JC, Meza W, Eslamfam S, Uetz P, Siegle DA, Hu JC, Giglio MG. An ontology for microbial phenotypes. *BMC Microbiol* 2014;14(1):1–8.
- [61] Gkoutos GV, Green ECJ, Mallon A-M, Hancock JM, Davidson D. Using ontologies to describe mouse phenotypes. *Genome Biol* 2005;6(1):1–10.
- [62] Gramates LS, Agapite J, Attrill H, Calvi BR, Crosby MA, Dos Santos G, Goodman JL, Goutte-Gattat D, Jenkins VK, Kaufman T, et al. Flybase: A guided tour of high-lighted features. *Genetics* 2022;220(4):iyac035.
- [63] Davis P, Zarowiecki M, Arnaboldi V, Becerra A, Cain S, Chan J, Chen WJ, Cho J, da Veiga Beltrame E, Diamantakis S, Gao S, Grigoriadis D, Grove CA, Harris TW, Kishore R, Le T, Lee RYN, Luypaert M, Müller H-M, Nakamura C, Nuin P, Paulini M, Quinton-Tulloch M, Raciti D, Rodgers Faye H, Russell M, Schindelman G, Singh A, Stickland T, Van Aukun K, Wang Q, Williams G, Wright AJ, Yook K, Berriman M, Howe KL, Schedl T, Stein L, Sternberg PW. WormBase in 2022—data, processes, and tools for analyzing *Caenorhabditis elegans*. *Genetics* 2022;220(4):iyac003 02.
- [64] Müller H-M, Van Aukun KM, Li Y, Sternberg PW. Textpresso central: a customizable platform for searching, text mining, viewing, and curating biomedical literature. *BMC Bioinf* 2018;19(1):1–16.
- [65] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J, Bonino da Silva Santos L, Bourne PE, et al. Addendum: The fair guiding principles for scientific data management and stewardship. *Scient Data* 2019;6(1):1–2.
- [66] github. obophenotype/uberon, 2020.
- [67] Matentzoglou N, Malone J, Mungall C, Stevens R. Miro: guidelines for minimum information for the reporting of an ontology. *J Biomed Semant* 2018;9(1):1–13.
- [68] Petri V, Jayaraman P, Tutaj M, Hayman GT, Smith JR, De Pons J, Laulederkind SJF, Lowry TF, Nigam R, Wang S-J, et al. The pathway ontology—updates and applications. *J Biomed Semant* 2014;5(1):1–12.
- [69] Diehl AD, Meehan TF, Bradford YM, Brush MH, Dahdul WM, Dougall DS, He Y, Osumi-Sutherland D, Ruttenberg A, Sarntinijai S, et al. The cell ontology 2016: enhanced content, modularization, and ontology interoperability. *J Biomed Semant* 2016;7(1):1–10.
- [70] Degtyarenko K, De Matos P, Ennis M, Hastings J, Zbinden M, McNaught A, Alcántara R, Darso M, Guedj M, Ashburner M. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res* 2007;36(suppl 1):D344–50.
- [71] Malone J, Holloway E, Adamusiak T, Kapushesky M, Zheng J, Kolesnikov N, Zhukova A, Brazma A, Parkinson H. Modeling sample variables with an experimental factor ontology. *Bioinformatics* 2010;26(8):1112–18.
- [72] Ochoa D, Hercules A, Carmona M, Suveges D, Gonzalez-Uriarte A, Malangone C, Miranda A, Fumis L, Carvalho-Silva D, Spitzer M, et al. Open targets platform: supporting systematic drug–target identification and prioritisation. *Nucleic Acids Res* 2021;49(D1):D1302–10.
- [73] Hotta K, Dauga D, Manni L. The ontology of the anatomy and development of the solitary ascidian ciona: the swimming larva and its metamorphosis. *Sci Rep* 2020;10(1):1–16.
- [74] Nowotarski SH, Davies EL, Robb SMC, Ross EJ, Matentzoglou N, Doddihall V, Mir M, McClain M, Sánchez Alvarado A. Planarian anatomy ontology: a resource to connect data within and across experimental platforms. *Development* 2021;148(15):dev196097.
- [75] Vasant D, Chanas L, Malone J, Hanauer M, Olry A, Jupp S, Robinson PN, Parkinson H, Rath A. Ordo: an ontology connecting rare disease, epidemiology and genetic data. In: *Proceedings of ISMB*, 30; 2014. researchgate. net.
- [76] Vasilevsky N, Essaid S, Matentzoglou N, Harris NL, Haendel M, Robinson P, Mungall CJ. Mondo disease ontology: harmonizing disease concepts across the world. *CEURWS*, 2807; 2020.
- [77] Bard JBL. The aeo, an ontology of anatomical entities for classifying animal tissues and organs. *Front Genet* 2012;3:18.
- [78] Wishart DS, Gird S, Peters H, Oler E, Jovel J, Budinski Z, Milford R, Lui VW, Sayeeda Z, Mah R, et al. Chemfont: the chemical functional ontology resource. *Nucleic Acids Res* 2023;51(D1):D1220–9.
- [79] Jackson R, Matentzoglou N, Overton JA, Vita R, Balhoff JP, Buttigieg PL, Carbon S, Courtot M, Diehl AD, Dooley DM, et al. Obo foundry in 2021: operationalizing open data principles to evaluate ontologies. *Database* 2021.
- [80] Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, Tudorache T, Musen MA. Biportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic Acids Res* 2011;39(suppl 2):W541–5.
- [81] Hoehndorf R, Slater L, Schofield PN, Gkoutos GV. Aber-owl: a framework for ontology-based data access in biology. *BMC Bioinf* 2015;16(1):1–9.
- [82] Ong E, Xiang Z, Zhao B, Liu Y, Lin Y, Zheng J, Mungall C, Courtot M, Ruttenberg A, He Y. Ontobee: a linked ontology data server to support ontology term dereferencing, linkage, query and integration. *Nucleic Acids Res* 2017;45(D1):D347–52.
- [83] Oliveira JL, Dias G, Oliveira I, Rocha P, Hermosilla I, Vicente J, Spiteri I, Sanchez FM, Sousa Pereira A. Diseasecard: a web-based tool for the collaborative integration of genetic and medical information. In: *International Symposium on Biological and Medical Data Analysis*. Springer; 2004. p. 409–17.
- [84] Dias GS, Oliveira JL, Vicente J, Sanchez FM. Integrating medical and genomic data: a successful example for rare diseases. *Stud Health Technol Inform* 2006;124:125.
- [85] Lopes P, Oliveira JL. An innovative portal for rare genetic diseases research: the semantic diseasecard. *J Biomed Inform* 2013;46(6):1108–15.
- [86] Rappaport N, Nativ N, Stelzer G, Twik M, Golan YG, Stein TI, Bahir I, Belinky F, Morrey CP, Safran M, et al. Malacards: an integrated compendium for diseases and their annotation. *Database* 2013:2013.
- [87] Espe S. Malacards: the human disease database. *J Med Libr Assoc* 2018;106(1):140.
- [88] Safran M, Solomon I, Shmueli O, Lapidot M, ShenOrr S, Adato Avital, Ben-Dor Uri, Esterman N, Rosen N, Peter I, et al. Genecards™ 2002: towards a complete, object-oriented, human gene compendium. *Bioinformatics* 2002;18(11):1542–3.
- [89] Pletscher-Frankild S, Pallegäa A, Tsafou K, Binder JX, Jensen LJ. Diseases: text mining and data integration of disease–gene associations. *Methods* 2015;74:83–9.
- [90] Perfetto L, Briganti L, Calderone A, Cerquone Perpetuini A, Iannuccelli M, Langone F, Licata L, Marinkovic Milica, Mattioni A, Pavlidou T, et al. Signor: a database of causal relationships between biological entities. *Nucleic Acids Res* 2016;44(D1):D548–54.
- [91] Kanehisa M, Goto S. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res* 2000;28(1):27–30.
- [92] Calderone A, Castagnoli L, Cesareni G. Menta: a resource for browsing integrated protein–interaction networks. *Nat Methods* 2013;10(8):690–1.
- [93] Hornbeck PV, Zhang B, Murray B, Kornhauser JM, Latham V, Skrzypek E. Phosphositeplus, 2014: mutations, ptms and recalibrations. *Nucleic Acids Res* 2015;43(D1):D512–20.
- [94] Dinkel H, Chica C, Via A, Gould CM, Jensen LJ, Gibson TJ, Diella F. Phospho. elm: a database of phosphorylation sites—update 2011. *Nucleic Acids Res* 2010;39(suppl 1):D261–7.
- [95] UniProt Consortium. Activities at the universal protein resource (uniprot). *Nucleic Acids Res* 2014;42(D1):D191–8.
- [96] Stenson PD, Mort M, Ball EV, Shaw K, Phillips AD, Cooper DN. The human gene mutation database: building a comprehensive mutation repository for clinical and molecular genetics, diagnostic testing and personalized genomic medicine. *Hum Genet* 2014;133(1):1–9.
- [97] Mattingly CJ, Rosenstein MC, Colby GT, Forrest JN Jr, Boyer JL. The comparative toxicogenomics database (ctd): a resource for comparative toxicological studies. *J Exp Zool Part A* 2006;305(9):689–92.
- [98] Jia J, An Zhongxin, Ming Y, Guo Y, Li W, Li X, Liang Y, Guo D, Tai J, Chen G, et al. Pedam: a database for pediatric disease annotation and medicine. *Nucleic Acids Res* 2018;46(D1):D977–83.
- [99] d'Aquin M, Motta E, Sabou M, Angeletou S, Gridinoc L, Lopez V, Guidi D. Toward a new generation of semantic web applications. *IEEE Intell Syst* 2008;23(3):20–8.
- [100] Feigenbaum EA. The art of artificial intelligence: Themes and case studies of knowledge engineering. In: *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 2, Boston; 1977.
- [101] Motta E. Reusable components for knowledge modelling. *Open University (United Kingdom)*; 1998.
- [102] Schreiber ATH, Schreiber G, Akkermans H, Anjewierden A, Shadbolt N, de Hoog R, Van de Velde W, Wielinga B, Nigel R, et al. Knowledge engineering and management: the CommonKADS methodology. MIT press; 2000.
- [103] Lan G, Liu T, Wang X, Pan X, Huang Z. A semantic web technology index. *Sci Rep* 2022;12(1):1–10.
- [104] Motta E, Sabou M. Next generation semantic web applications. In: *Asian Semantic Web Conference*. Springer; 2006. p. 24–9.
- [105] Karami M, Rahimi A. Semantic web technologies for sharing clinical information in health care systems. *Acta Informatica Medica* 2019;27(1):4.
- [106] Stadnicki A, Pietron F, Burek P. Towards a modern ontology development environment. *Procedia Comput Sci* 2020;176:753–62.
- [107] M. R. Genesereth, R. E. Fikes, et al. Knowledge interchange format-version 3.0: reference manual. 1992.
- [108] Farquhar A, Fikes R, Rice J. The ontolingua server: A tool for collaborative ontology construction. *Int J Hum Comput Stud* 1997;46(6):707–27.
- [109] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies, 1992.
- [110] Angele J, Kifer M, Lausen G. Ontologies in f-logic. In: *Handbook on ontologies*. Springer; 2009. p. 45–70.
- [111] Kifer M, Lausen G, Wu J. Logical foundations of object-oriented and frame-based languages. *J ACM (JACM)* 1995;42(4):741–843.
- [112] Tekli J, Chbeir R, Yetongnon K. An overview on xml similarity: Background, current trends and future directions. *Comput Sci Rev* 2009;3(3):151–73.
- [113] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, Fr. Yergeau, and J. Cowan. Extensible markup language (xml) 1.0, 2000.
- [114] Berners-Lee T. Relational databases and the semantic web (in design issues). *World Wide Web Consortium*; 1998.
- [115] P. D. Karp, V. K. Chaudhri, and J. Thomere. Xol: An xmlbased ontology exchange language, 1999.
- [116] Casellas N. Legal ontology engineering: Methodologies, modelling trends, and the ontology of professional judicial knowledge, 3. Springer Science & Business Media; 2011.
- [117] Hendler J, McGuinness DL, et al. The darpa agent markup language. *IEEE Intell Syst* 2000;15(6):67–73.
- [118] McGuinness DL, Fikes R, Stein LA, Hendler JA. Daml-ont: An ontology language for the semantic web. In: *Spinning the Semantic Web*; 2003. p. 65–93.
- [119] Fensel D, Van Harmelen F, Horrocks I, McGuinness DL, Patel-Schneider PF. Oil: An ontology infrastructure for the semantic web. *IEEE Intell Syst* 2001;16(2):38–45.
- [120] J. Broekstra, M. Klein, D. Fensel, S. Decker, and I. Horrocks. Oil: A case study in extending rdf-schema. *To appear*, 2000.
- [121] Horrocks I, et al. Daml+oil: A description logic for the semantic web. *IEEE Data Eng. Bull.* 2002;25(1):4–9.
- [122] McGuinness DL, Van Harmelen Frank, et al. Owl web ontology language overview. *W3C Recommend* 2004;10(10):2004.
- [123] World Wide Web Consortium et al. Owl 2 web ontology language document overview. 2012.
- [124] Baader F, Horrocks I, Sattler Ulrike. Description logics. *Found Artif Intell* 2008;3:135–79.

- [125] Calvanese D, De Giacomo G, Nardi D, Lenzerini M. Reasoning in expressive description logics. *Handbook Automated Reason* 2001;2:1581–634.
- [126] Baader F, Horrocks I, Sattler U. Description logics. In: *Handbook on ontologies*. Springer; 2004. p. 3–28.
- [127] Hee Hwang C. Incompletely and imprecisely speaking: using dynamic ontologies for representing and retrieving information. In: *KRDB*, 21. Citeseer; 1999. p. 14–20.
- [128] Musen MA. The Protégé project: a look back and a look forward. *AI Matters* 2015;1(4):4–12.
- [129] Tudorache T, Vendetti J, Noy N, Fridman. Webprotege: A lightweight owl ontology editor for the web. In: *OWLED*, 432; 2008. p. 2009.
- [130] McGuinness DL, Fikes R, Rice J, Wilder S. The chimaera ontology environment. *AAAI/IAAI* 2000;2000:1123–4.
- [131] Cardoso J. Semantic Web Services: Theory, Tools and Applications: Theory, Tools and Applications. IGI Global; 2007.
- [132] Erdmann M, Waterfeld W. Overview of the neon toolkit. In: *Ontology Engineering in a Networked World*. Springer; 2012. p. 281–301.
- [133] Singh A, Anand P. State of art in ontology development tools. *Int J* 2013;2(7):96–101.
- [134] Motik B, Studer R. Kaon2—a scalable reasoning tool for the semantic web. In: *Proceedings of the 2nd European Semantic Web Conference (ESWC'05)*, 17, Heraklion, Greece; 2005.
- [135] Arpirez JC, Corcho O, Fernández-López M, Gómez-Pérez A. Webode in a nutshell. *AI Mag* 2003;24(3) 37–37.
- [136] Top Quadrant. Topbraid composer, 2011.
- [137] Pollock JT. *Semantic web for dummies*. John Wiley and Sons; 2009.
- [138] European Commission. *Isa2-interoperability solutions for public administrations, businesses and citizens*, 2017.
- [139] Bryant M, Reijnhoudt L, Speck R, Clerice T, Blanke T. The ehri project-virtual collections revisited. In: *International Conference on Social Informatics*. Springer; 2014. p. 294–303.
- [140] Sagri M-T. Strumenti semantici per l'accesso all'informazione giuridica: Giur-word-net. *Informatica e Diritto* 2002;11(2):185–210.
- [141] Pobukovsky M. The unesco computerized documentation system. *Inf Dev* 1985;1(1):15–18.
- [142] Frey K, Bukovi K. Unified astronomy thesaurus: Usage stats and release cycle. *Res Notes AAS* 2020;4(4):58.
- [143] Botstein D, Cherry JM, Ashburner M, Ball CA, Blake JA, Butler H, Davis AP, Dolinski K, Dwight SS, Eppig JT, et al. Gene ontology: tool for the unification of biology. *Nat Genet* 2000;25(1):25–9.
- [144] Jupp S, Bechhofer S, Stevens R. A flexible api and editor for skos. In: *European Semantic Web Conference*. Springer; 2009. p. 506–20.
- [145] McCrae JP, Bosque-Gil J, Gracia J, Buitelaar P, Cimiano P. The ontolex-lemon model: development and applications. In: *Proceedings of eLex 2017 conference*; 2017. p. 19–21.
- [146] Shearer RDC, Motik B, Horrocks I. Hermit: A highly efficient owl reasoner. In: *OWled*, 432; 2008. p. 91.
- [147] Seganti A, Kaplan'ski P, Zarzycki P. Collaborative editing of ontologies using fluent editor and ontorion. In: *International Experiences and Directions Workshop on OWL*. Springer; 2016. p. 45–55.
- [148] Stellato A, Fiorelli M, Turbati A, Lorenzetti T, Van Gemert W, Dechandon D, Laaboudi-Spoiden C, Gerencs'er A, Waniart A, Costetchi E, et al. Vocbench 3: A collaborative semantic web editor for ontologies, thesauri and lexicons. *Semantic Web* 2020;11(5):855–81.
- [149] Brambilla M, Ceri S, Daniel F, Di Giovanni M, Mauri A, Ramponi G. Iterative knowledge extraction from social networks. In: *Companion Proceedings of the The Web Conference 2018*; 2018. p. 1359–64.
- [150] Vahdati S, Fathalla S, Auer S, Lange C, Vidal ME. Semantic representation of scientific publications. In: *International Conference on Theory and Practice of Digital Libraries*. Springer; 2019. p. 375–9.