Developing Ontological Background Knowledge for Biomedicine

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Anna Elena Beißwanger
aus Stuttgart

Mannheim, 2013

Dekan:	Prof. Dr. Heinz Jürgen Müller
Referent:	Prof. Dr. Heiner Stuckenschmidt
Korreferent:	Prof. Dr. Udo Hahn

Tag der mündlichen Prüfung: 24. Mai 2013

Abstract

Biomedicine is an impressively fast developing, interdisciplinary field of research. To control the growing volumes of biomedical data, ontologies are increasingly used as common organization structures. Biomedical ontologies describe domain knowledge in a formal, computationally accessible way. They serve as controlled vocabularies and background knowledge in applications dealing with the integration, analysis and retrieval of heterogeneous types of data. The development of biomedical ontologies, however, is hampered by specific challenges. They include the lack of quality standards, resulting in very heterogeneous resources, and the decentralized development of biomedical ontologies, causing the increasing fragmentation of domain knowledge across them.

In the first part of this thesis, a life cycle model for biomedical ontologies is developed, which is intended to cope with these challenges. It comprises the stages "requirements analysis", "design and implementation", "evaluation", "documentation and release" and "maintenance". For each stage, associated subtasks and activities are specified. To promote quality standards for biomedical ontology development, an emphasis is set on the evaluation stage. As part of it, comprehensive evaluation procedures are specified, which allow to assess the quality of ontologies on various levels. To tackle the issue of knowledge fragmentation, the life cycle model is extended to also cover ontology alignments. Ontology alignments specify mappings between related elements of different ontologies. By making potential overlaps and similarities between ontologies explicit, they support the integration of ontologies and help reduce the fragmentation of knowledge.

In the second part of this thesis, the life cycle model for biomedical ontologies and alignments is validated by means of five case studies. As a result, they confirm that the model is effective. Four of the case studies demonstrate that it is able to support the development of useful new ontologies and alignments. The latter facilitate novel natural language processing and bioinformatics applications, and in one case constitute the basis of a task of the "BioNLP shared task 2013", an international challenge on biomedical information extraction. The fifth case study shows that the presented evaluation procedures are an effective means to check and improve the quality of ontology alignments. Hence, they support the crucial task of quality assurance of alignments, which are themselves increasingly used as reference standards in evaluations of automatic ontology alignment systems. Both, the presented life cycle model and the ontologies and alignments that have resulted from its validation improve information and knowledge management in biomedicine and thus promote biomedical research.

Zusammenfassung

Die Biomedizin ist ein sich beeindruckend schnell entwickelndes, interdisziplinäres Forschungsgebiet. Um die immer größer werdenden Mengen biomedizinischer Daten besser kontrollieren zu können, werden zunehmend Ontologien als übergreifende Organisationsstrukturen eingesetzt. Sie beschreiben biomedizinisches Fachwissen in einer formalen, automatisch verarbeitbaren Form. Bevorzugt werden sie als kontrollierte Vokabulare und formalisiertes Hintergrundwissen in Anwendungen zur Integration, Analyse und Abfrage von heterogenen Daten verwendet. Die Entwicklung biomedizinischer Ontologien steht jedoch derzeit vor gewissen ungelösten Problemen. Zu diesen zählen der Mangel an Qualitätsstandards, der zu Ergebnissen von sehr unterschiedlicher Qualität führt, und die dezentrale Entwicklung biomedizinischer Ontologien, die eine zunehmende Fragmentierung von formalisiertem Fachwissen zur Folge hat.

Im ersten Teil dieser Arbeit wird ein Lebenszyklusmodell für biomedizinische Ontologien entwickelt, das helfen soll, die genannten Probleme zu lösen. Es umfasst die Stufen "Bedarfsanalyse", "Entwurf und Implementierung ", "Evaluation", "Dokumentation und Veröffentlichung" und "Wartung". Für jede dieser Stufen werden wichtige Teilaufgaben spezifiziert. Um die Rolle von Qualitätsstandards in der biomedizinischen Ontologieentwicklung zu stärken, wird der Schwerpunkt auf die Evaluationsstufe gelegt. Als Bestandteil dieser werden umfassende Evaluationsverfahren etabliert, die es erlauben, die Qualität von Ontologien auf verschiedenen Ebenen zu beurteilen. Um das Problem der Wissensfragmentierung anzugehen, wird das Lebenszyklusmodell auf Ontologie-Alignments ausgeweitet. Alignments bestehen aus Zuordnungen ähnlicher Elemente aus verschiedenen Ontologien. Indem sie Überschneidungen und Ähnlichkeiten zwischen Ontologien explizit machen, helfen sie Ontologien zu verknüpfen und die Fragmentierung von formalisiertem Fachwissen zu reduzieren.

Im zweiten Teil dieser Arbeit wird das Lebenszyklusmodell für biomedizinische Ontologien und Alignments anhand von fünf Fallstudien validiert. Im Ergebnis bestätigen diese die Wirksamkeit des Modells. Vier der Fallstudien zeigen, dass es in der Lage ist, die Entwicklung nützlicher biomedizinischer Ontologien und Alignments zu unterstützen. Letztere ermöglichen neuartige Anwendungen aus den Bereichen der automatischen Sprachverarbeitung und der Bioinformatik und bilden in einem Fall die Grundlage einer Aufgabe der "BioNLP Shared Task 2013", eines internationalen Wettbewerbs in biomedizinischer Informationsextraktion. Die fünfte Fallstudie zeigt, dass die neu etablierten Evaluationsverfahren ein wirksames Mittel zur Überprüfung und Verbesserung von Ontologie-Alignments sind. Sie unterstützen damit die wichtige Aufgabe der Qualitätssicherung von Alignments, welche zunehmend selbst als Referenzstandards in Evaluationen von automatischen Alignmentsystemen eingesetzt werden. Sowohl das Lebenszyklusmodell selbst, als auch die Ontologien und Alignments, die aus dessen Validierung hervorgegangen sind, fördern das Informations- und Wissensmanagement in der Biomedizin und unterstützen damit die biomedizinische Forschung.

Acknowledgements

First, I would like to thank my supervisors Prof. Dr. Heiner Stuckenschmidt (University of Mannheim) and Prof. Dr. Udo Hahn (University of Jena). Heiner Stuckenschmidt I am grateful for his well-directed suggestions and advice, which helped me to focus and streamline this thesis. Udo Hahn I am thankful for introducing me into the worlds of biomedical ontologies and natural language processing and for taking the time to answer my questions (not too few on quality standards applied in these worlds). I further thank him for the opportunity to meet and collaborate with scientists from other places. These cooperations inspired me and contributed substantially to the progress of my work. Finally, I am thankful for the time that he generously granted me for writing this thesis.

My thanks go also to my past and present colleagues at the JULIE Lab, who listened to me and encouraged me, whenever required. Instructive discussions at the office door and enjoyable joint lunch breaks provided for an inspiring and pleasant working atmosphere that helped making long working days worthwhile. Particularly grateful I am to Erik and Johannes, who took much of the project work off my shoulders during the last year, so that I could finish this thesis.

This work would not have been possible without the collaboration with project partners and colleagues. On behalf of all co-authors of the ontologies GRO, MaHCO and BioTop, which were analyzed in this thesis, I would like to thank PhD Vivian Lee, Dr. David S. DeLuca and Prof. Dr. Stefan Schulz for their active collaboration. I would further like to thank Dr. Christian Meilicke from the organizing committee of the OAEI Anatomy track for the fruitful collaboration on the anatomy alignment.

For proofreading this thesis and inviting me to simplify matters and divide page long sentences short, my thanks go to David, Johannes and Nicole. Furthermore, I thank Ekaterina, Erik, Ronny and Sebastian for their feedback on parts of this thesis.

Last but not least, my heartfelt thanks go to my dear family and friends, who gave me invaluable support during writing this thesis. They bore my moods, cheered me up and helped me to put things in perspective—sometimes by holding the rope and allowing me a glance from above. My special thanks go to David, who more than returned the favor in terms of patience and support. Thank you for your incredible intuition and commitment. Many joint plans are waiting to be realized, and I am looking forward to the times ahead.

Jena, February 11, 2013

Contents

List of Figures

List of Tables

Part I

Introduction and Background

CHAPTER 1

Introduction

1.1 Motivation

Knowledge management in biomedicine is increasingly based on tools that provide automatic support. So far, complex tasks such as biocuration were mainly tackled by domain experts [Howe et al., 2008]. To be able to solve them automatically, tools require domain-specific, automatically accessible background knowledge. A sophisticated form thereof is ontological background knowledge. Ontologies represent a domain in terms of classes and relationships between them in a formal language. In contrast to conventional terminologies, the meaning of ontology classes is not only specified by natural language terms (e.g., class labels and verbal definitions) but also by axioms. Axioms are expressed in a formal language. They can automatically be processed and checked for logical consistency. This opens up powerful new opportunities for knowledge-based tools and applications. The development of this advanced form of background knowledge for biomedicine is the focus of this thesis.

Biomedicine is a branch of medical science that studies the molecular and cellular foundations of life, pathological changes and causal therapies. New biomedical knowledge is gained by running experiments or clinical studies, deriving knowledge from the resulting data and publishing it in scientific articles or medical reports. From these—from a computational perspective "unstructured"—documents relevant facts (e.g., verbal mentions of biomedical entities, such as proteins or diseases, and relationships between them, such as protein-protein interactions) are extracted by domain experts

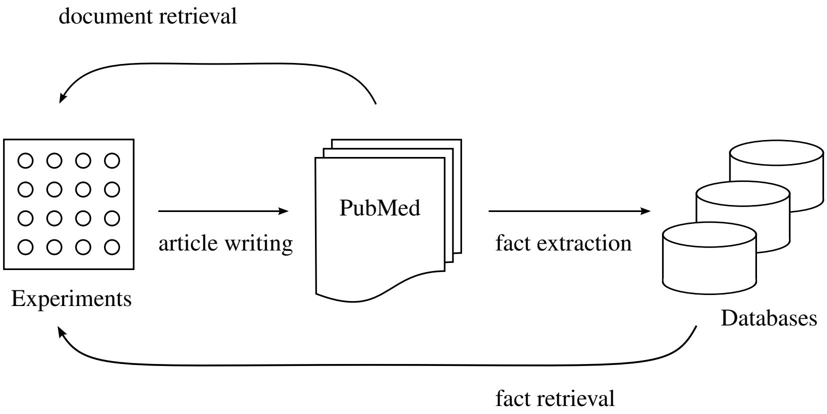


Figure 1.1: Knowledge gaining process in biomedicine.

and put in the computationally easier accessible, "structured" form of databases. This procedure is called "database curation" or "biocuration". Bibliographic and factual databases, in turn, are queried by researchers and clinicians for relevant articles and facts in order to prepare new experiments or clinical studies (figure 1.1).

However, this process is increasingly challenged by the use of data-intensive analysis techniques for biomedical experimentation (e.g., microarrays, high-throughput technologies and medical imaging). The rapid accumulation of experimental data accelerates the publication rate of unstructured documents. Database curators are no longer able to extract facts at the pace at which new documents are published [Baumgartner et al., 2007]. In addition, facts that are extracted are scattered over an increasing number of databases. Also document retrieval systems are challenged by the growing number of available documents. If they retrieve too few or too many documents, researchers may miss facts. The consequence of this development is that an increasing number of facts remains "hidden" in natural language terms or unretrieved from distributed databases. This increases the risk for duplicate research efforts and wrong treatment decisions.

Two impressive figures illustrating the growth in unstructured and structured biomedical data are the number of citations in Medline, a huge bibliographic database for biomedicine and health care and the number of databases in the Molecular Biology Database Collection, a collection of open access databases from molecular biology and biomedicine [Baxevanis, 2000] (see figure 1.2).

A way to prevent that the fast growth of biomedical data hampers research progress and patient treatment, and utilize the benefit of the newly generated data instead, is

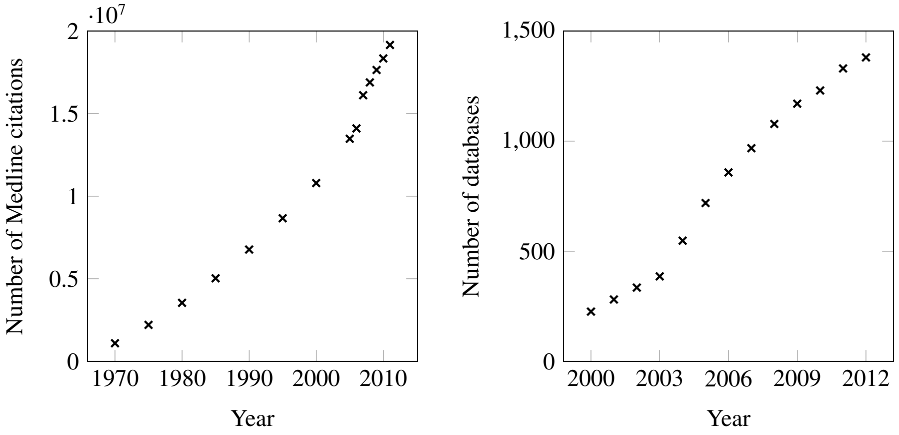


Figure 1.2: Left: Number of Medline citations 1970–2011 [U.S. National Library of Medicine, 2012]. Right: Number of databases in the Molecular Biology Database Collection 2000–2012. Database counts were taken from the yearly published Database issues of the journal Nucleic Acids Research [Oxford Journals, 2012].

the provision of tools that provide automatic support. Researchers, database curators and clinicians would particularly benefit from tools for tightly focused document retrieval, fact extraction from text and knowledge integration. However, these are very complex tasks. In order to solve them, domain experts heavily use their ability to analyze natural language documents, background knowledge and reasoning skills. A strategy to approximate human-generated results is therefore to provide tools also with natural language processing (NLP) competencies, domain knowledge and the ability of reasoning. While NLP competencies may be achieved by incorporating existing NLP approaches, possibly after a domain adaption [see, e.g., Tomanek et al., 2007], domain knowledge and the ability to reason on it may be obtained by incorporating appropriate terminological and conceptual resources as background knowledge.

The use of comparatively informal terminological resources as background knowledge has proven to be sufficient for solving parts of the mentioned tasks. For example, large parts of the retrieval power of the Medline database rely on the Medical Subject Headings (MeSH) as indexing vocabulary [Funk and Reid, 1983], a conventional thesaurus (see page 41). Furthermore, many named entity recognition (NER) tools, which are used to detect mentions of certain types of biomedical entities in text, rely on plain listings of biomedical categories. The latter are either used as dictionary (as in the rule-based protein and gene entity recognizer ProMiner [Hanisch et al., 2005]) or as annotation vocabulary for the creation of training data (as in the machine learning-based gene name normalizer Geno [Wermter et al., 2009]). NER tools contributed already successfully to the manual [Dowell et al., 2009] and automatic fact extraction from text [Buyko et al., 2011].

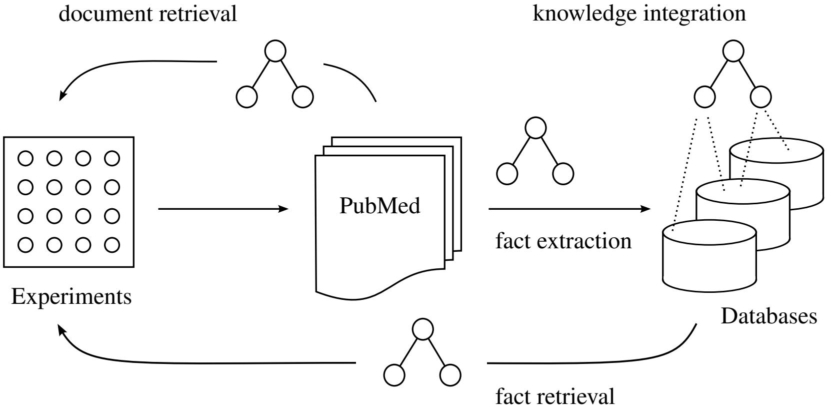


Figure 1.3: Use cases of biomedical ontologies.

However, the requirements that more complex tasks pose on background knowledge cannot be answered by conventional terminologies, but require the use of biomedical ontologies instead. An example is the automatic fact extraction from text. It is challenged by the high variability of natural language terms that are used to express facts, and the circumstance that some verbal mentions of facts are nested or spread over several sentences. Supervised machine learning-based approaches, which were backed with ontology-based training data, have shown to be able to cope with the highly variable biomedical language [Buyko et al., 2011]. In addition, rule-based approaches, which rely on expressive biomedical ontologies that facilitate automatic reasoning, have shown to be able to detect even nested facts and facts that can only be concluded from distributed evidence [Kim and Rebholz-Schuhmann, 2011]. Further tasks that have shown to profit from biomedical ontologies as background knowledge include knowledge integration and fact retrieval across databases [see, e.g., Ashburner et al., 2000] and subject-specific document retrieval [see, e.g., Doms and Schroeder, 2005]. The above-mentioned exemplary use cases of biomedical ontologies are outlined in figure 1.3.

Since the first biomedical ontologies were developed roughly a decade ago, pioneered by the Gene Ontology (GO) [Ashburner et al., 2000], their number has continuously grown. Today, up to a few hundred biomedical ontologies are available in umbrella systems such as the OBO library or the NCBO BioPortal (see page 17). While, at first, most biomedical ontologies (including GO) were expressed in a rather informal language, today many of them are represented in the formal and—depending on the sublanguage—expressive Web Ontology Language (OWL) [Bechhofer et al., 2004]. OWL has originally been introduced to represent ontologies in the context of the Semantic Web. It aims at making the implicit semantics of Web contents explicit and computationally accessible by introducing ontology-backed markup [Berners-Lee et al., 2001]. OWL is also used as reference formalism in the context of this thesis.

This thesis deals with the development of ontological background knowledge for biomedicine. The number of already available biomedical ontologies may raise concerns about the relevance of developing additional ones. However, there are two strong arguments to resolve possible concerns. First, an important characteristic of biomedicine is the pace at which new knowledge is generated. It can be expected to result in a continuous need for new ontologies that cover the most recent knowledge. Second, it should be noticed that the development of tools for the complex tasks of tightly focused document retrieval and automatic biocuration has just started. Only recently, the biomedical NLP community shifted its focus from simple NER to more advanced forms of information extraction for biomedicine, which must be solved for automated biocuration. This shift is indicated by a series of international competitions, called the "BioNLP Shared Tasks".[1] Experience has shown that new use cases impose new requirements on the coverage and expressiveness of ontologies. Accordingly, the current development of new approaches and tools can be expected to result in the need for new biomedical ontologies. In the context of this thesis, several important subareas of biomedicine are identified that are not yet covered by existing biomedical ontologies and application requirements are derived that are currently unmet. Subsequently, resources are developed intended to fill these gaps.

The potential of ontologies for biomedical research makes a thorough analysis of the building process of biomedical ontologies worthwhile. There are different strategies for building ontologies (see chapter 3). The most widespread one, which is also in the focus of this thesis, is their manual development. It is a rather complex task that demands a broad spectrum of knowledge and skills from developers [Neuhaus et al., 2011]. Accordingly, it has a strong creative component [Noy and McGuinness, 2001]. Since biomedical ontologies are not created as ends in themselves, but as components of often complex tools and applications, it is important to restrain this creative component and instead standardize their development process to increase their reliability. For this purpose, ontology development tools, life cycle models and guidelines, up to entire methodologies for ontology development have been proposed. However, meanwhile the plurality of available proposals raises the question for the most appropriate life cycle model, guidelines, and so on, for a given ontology project. For developers of biomedical ontologies, who deal with an especially demanding field of application (see section 4.1) this question is of particular relevance. What they need is a life cycle model for ontologies that is able to cope with the challenges that biomedicine poses as field of application. Since existing life cycle models for ontologies lack any domain adaption, in the context of this thesis a life cycle model is compiled that is tailored to biomedical ontologies.

[1] The first two Shared Tasks were held in 2009 [Kim et al., 2009] and 2011 [Pyysalo et al., 2012] and a third one is scheduled for 2013, see `http://bionlp-st.org/` – access date 2012-11-30.

Each individual biomedical ontology has a limited coverage. Since tools, as they are currently build in assistance of biomedical researchers and clinicians, use fact extraction patterns and search queries that easily transcend the conceptual coverage of individual ontologies, missing links between ontologies hamper effective information extraction and search. The process of establishing such links between related elements of different ontologies is called ontology matching and the result an ontology alignment [Euzenat and Shvaiko, 2007]. An alignment bridges its input ontologies to one larger background knowledge resource. The more biomedical ontologies are available, the more important it becomes to consider ontology matching as an alternative to ontology development from scratch. It has not only the potential to save duplicate work, but also to avoid redundancy. Thus, in the context of this thesis not only the development of new ontologies is considered but also the linkage of existing ones.

The effectiveness of ontologies and ontology alignments in applications and the strength of analysis and evaluation results based on them strongly depend on their quality. For this reason, a crucial step in the development of ontologies and ontology alignments is a thorough evaluation. Since both are complex artifacts, their quality depends on multiple aspects. However, most existing evaluation approaches consider individual aspects only (see section 3.6 and 3.7). In the context of this thesis, comprehensive evaluation approaches are compiled that assess the quality of ontologies and ontology alignments regarding various aspects.

The overall success of an ontology or ontology alignment project depends on various factors. Two important ones have already been mentioned above, *viz.*, the choice of an appropriate life cycle model and the choice of a suitable evaluation approach. Additional ones are difficult to access because they are scattered over various articles, guides and tutorials or were not reported at all, as yet. Since it would be beneficial to explicitly know about success factors for ontology and ontology alignment development, parts of this thesis deal with compiling them.

1.2 Objectives and Contributions

The main contributions of this thesis are centered around six objectives, of which each targets gaps in previous work on the development of ontological background knowledge for biomedicine.

1. Appropriate strategies to obtain ontological background knowledge for biomedicine are investigated and an approach is framed that combines the popular strategy of manual ontology development with the currently less widespread strategy of ontology matching.

2. The challenges that biomedicine poses as field of application of ontological background knowledge are studied and a life cycle model for biomedical ontologies and ontology alignments is compiled that is intended to cope with these challenges. This life cycle model forms the core of the above-mentioned approach to building ontological background knowledge for biomedicine. The effectiveness of the approach is tested in five practical case studies.

3. As part of the life cycle model, comprehensive evaluation approaches are established that address different aspects of ontology and ontology alignment quality. For ontologies a focus is set on checking their compliance with design and implementation guidelines and for ontology alignments on checking basic aspects of their validity and reusability. It will be demonstrated that these evaluation approaches facilitate the correction and extension of newly developed ontologies and ontology alignments as well as the improvement of existing ontology alignments and hence the strengthening of evaluation results that are based on them.

4. In the context of this thesis, three important subdomains of biomedicine are identified that have not yet been represented in form of ontological background knowledge. These gaps are tackled by appropriate resources. The Gene Regulation Ontology (GRO) addresses the field of gene regulation, the Major Histocompatibility Complex Ontology (MaHCO) different aspects of the major histocompatibility complex (MHC) and the Protein alignment—bridging parts of the protein database UniProtKB and the MeSH thesaurus— the hierarchical organization of proteins.[2] In addition to these resources on specific subdomains of biomedicine BioTop is developed, a top domain ontology for molecular biology and biomedicine.[3]

5. Furthermore, in the context of this thesis it is demonstrated that very diverse applications profit from ontological background knowledge for biomedicine, where different applications utilize different features of it. On the one hand, it is shown that the requirements of complex NLP applications on background knowledge are higher than those of standard applications of biomedical ontologies, such as database annotation. For example, it is shown that automatic fact extraction requires particularly expressive biomedical ontologies as background knowledge that provide the possibility for automatic reasoning. On the other hand, an explanation is given why even comparatively simple tasks, such as the semantic annotation of text corpora, will benefit from the use of ontological background knowledge, compared to simpler forms thereof.

[2] The fact that the Protein alignment is no ontological background knowledge in the strict sense of this term (see page 169) is deliberately neglected at this point.

[3] The ontologies GRO, MaHCO and BioTop have been developed collaboratively in the context of different research projects (see below). The author of this thesis substantially contributed to the development process of each of the three ontologies (for BioTop mainly up to version 2008-02-19, described in Beisswanger et al. [2008c]).

6. Finally, success factors for the creation and curation of ontological background knowledge are compiled, based on practical experience collected during the development and evaluation of different biomedical ontologies and ontology alignments.

Some aspects of the mentioned contributions were the subject of previous publications of the author of this thesis. An earlier version of the domain ontology GRO has been presented in Beisswanger et al. [2008a] and of MaHCO in Beisswanger et al. [2007] and DeLuca et al. [2009]. The early development of BioTop is described in Schulz et al. [2006a] and Schulz et al. [2006b] and interfaces of BioTop and selected biomedical domain ontologies in Beisswanger et al. [2008c]. Finally, the creation of the Protein alignment has been reported in Beisswanger et al. [2010] and the evaluation approach proposed for ontology reference alignments in Beisswanger and Hahn [2012]. These previous publications are cited again in the respective parts of this thesis.

1.3 Structure of this Thesis

This thesis is structured into four parts. Part I introduces our work and supplies basic background information. After the present introductory chapter, in chapter 2 the basic notions around which this thesis is centered are introduced, such as “ontology” and “ontology alignment”. Furthermore, the syntactic representation and semantics of ontologies and ontology alignments are explained. Finally, some notational conventions are established.

Part II deals with the methods developed in the context of this thesis. First, in chapter 3 related work on building and evaluating ontological background knowledge is presented and the ambiguous use of the term “biomedical ontology” is explained. In chapter 4 an approach to building ontological background knowledge for biomedicine is introduced that forms the core of this thesis. It consists of a five staged life cycle model for biomedical ontologies with a variant for ontology alignments. A distinctive feature of the approach are the comprehensive evaluation approaches for biomedical ontologies and ontology alignments that it includes, described in section 4.5. Chapter 5 features an extensive discussion on the proposed approach to building ontological background knowledge for biomedicine. It highlights that the approach positively sticks out from existing ones in terms of scope, granularity and coverage.

Part III comprises five case studies on the proposed approach to building ontological background knowledge for biomedicine. The first and the second case study deal with the development of the domain ontologies GRO (chapter 6) and MaHCO (chapter 7), the third with the development of the top domain ontology BioTop (chapter 8) and the fourth with the creation of the Protein alignment (chapter 9). For each of the resources practical use cases are specified. Related work for GRO is specified in section 6.1, for MaHCO in section 7.1, for BioTop in section 8.1 and for the Protein alignment in section 9.1, in respective subsections named "Related knowledge resources". The fifth case study deals with the evaluation of the Anatomy, the Lod and the Bridge alignments (chapter 10), three existing ontology alignment datasets that have already been used as reference standard for the evaluation of ontology matching systems. In chapter 11, the five case studies are discussed. The discussion points out that in form of GRO, MaHCO, BioTop and the Protein alignment the proposed approach to building ontological background knowledge for biomedicine supported the development of four novel and useful resources. The expressiveness of GRO, MaHCO and BioTop is assessed by comparing them to existing biomedical domain and top domain ontologies. The discussion further highlights that the proposed approach also supported the consolidation and refinement of three existing ontology alignment datasets and hence helped to strengthen the evaluation results based on them.

Part IV that comprises chapter 12 concludes this work. The contributions are summarized, conclusions are drawn and future perspectives are specified. The success factors compiled for the creation and curation of ontological background knowledge in the context of this thesis are summarized in appendix A, in terms of suggestions for biomedical ontology developers.

CHAPTER 2

Background

This thesis deals with the development of ontological background knowledge for biomedicine. For this reason, below the basic notions of “ontology” (section 2.1) and “ontology alignment” (section 2.5) are introduced. Furthermore, the differences between top level, top domain and domain ontologies are clarified (section 2.2). The ontology language OWL is introduced (section 2.3) and the syntax and semantics of OWL DL ontologies are outlined (section 2.4). Finally, some conventions are specified that apply to the remainder of this thesis (section 2.6).

2.1 Ontologies

The word “ontology” is derived from the Greek words “ontos” (“of that which is”) and “logos” (“word”). In its original meaning, it refers to a branch of metaphysics studying the nature of being. However, in contemporary philosophy there is also a countable reading of the word, referring to “a particular system of categories accounting for a certain vision of the world” [Guarino, 1998, page 4]. There is a distinct field of research of philosophy, called “formal ontology”, in which philosophers aim at the formal, domain and application-independent description of the categories of the world in terms of precise logical statements. Basic ontological distinctions are introduced, and categories are provided with definitions in the Aristotelian tradition of specifying genus and differentiae (i.e., a supertype or family of a category and conditions that distinguish it from other categories with the same genus).

With the rise of artificial intelligence, the notion of ontologies was adopted by computer scientists, who redefined it as engineering artifacts that represent a machine processable abstraction of a domain. A computer science ontology consists of a vocabulary, referring to the classes and relations of a certain domain, and a set of explicit assumptions about their intended meaning [Guarino, 1998]. The assumptions are an attempt to translate the conceptualization, which the classes and relations are intended to capture, into a computational representation that is explicit and unambiguous, i.e., independent from reader and context. The most common ontology definition in computer science is "An ontology is an explicit specification of a conceptualization." [Gruber, 1993, page 1], where the notion of "conceptualization" is further discussed in Guarino [1998]. The vocabulary and assumptions, of which a computer science ontology consists, are usually expressed in a formal ontology language (i.e., a language with strict semantics). For this reason, computer science ontologies are sometimes called "formal" ontologies. It is important to notice that "formal" in this context does not necessarily mean that the principles of formal ontology are adhered to, such as the commitment to basic ontological distinctions or the explicit specification of the semantics of classes and relations (although the formality of the language *can* help in making ontological distinctions [Bodenreider and Stevens, 2006]). Consequently, computer science ontologies may represent different, not necessarily compatible views on a given reality, just as the same reality may be described by different ontologies, varying in the vocabulary used [Guarino, 1998]. Furthermore, computer science ontologies may be simple class hierarchies that rely on strict subclass relationships, but mostly lack formal class definitions (they are called "taxonomies" or "lightweight" ontologies, respectively), as well as highly expressive ontologies, in which the meaning of classes is specified explicitly in terms of axioms expressed in the underlying ontology language, or anything in between. A major benefit of computer science ontologies is that they make domain knowledge explicit and accessible not only to human users but also to machines. For example, they allow to separate domain knowledge from programming code, facilitating maintenance, flexible extension, sharing and reuse of domain knowledge [Guarino, 1998]. Since computer science ontologies are usually expressed even in a *decidable* formal ontology language, they provide the additional advantage that they can automatically be classified and checked for logical consistency, using appropriate reasoning tools.

In biology and biomedicine, in turn, the term "ontology" is used to refer to various types of artifacts that serve the purpose of knowledge organization [Bodenreider and Stevens, 2006; Rubin et al., 2008]. A growing number of artifacts denoted as biomedical ontologies is implemented in a formal ontology language, and hence complies with the computer science reading of ontologies. However, the remaining ones are in fact either terminologies (e.g., controlled vocabularies and thesauri), which are typically strongly natural language-oriented and lack formal rigor, or data models or data exchange standards, which typically come with a rather restricted, application-dependent view on a domain (see section 3.8).

In the context of this thesis, “formal ontologies” are developed in the computer science reading of the term, i.e., ontologies expressed in a formal language. Nevertheless, an additional objective is to make the ontologies compliant with the principles of formal ontology by adopting basic ontological distinctions from existing top level ontologies and explicitly specifying the intended meaning of ontology elements in terms of axioms. Furthermore, an objective is to provide the ontologies with natural language annotations to tie in with the strength of traditional knowledge organization systems in biology and medicine that, despite being sometimes named “biomedical ontologies”, are mostly classical terminologies.

2.2 Levels of Generality

Ontologies can be distinguished into top level, top domain and domain ontologies, depending on their level of generality and scope.[1] How the three types of ontologies relate to each other is depicted in figure 2.1.

2.2.1 Top level Ontologies

Top level (also “upper level” or “foundational”) ontologies cover basic domain and application-independent classes (e.g., ‘object’, ‘space’, ‘time’ and ‘event’) [Guarino, 1998] and relations (e.g., ‘part-of’ and ‘participates-in’) and explicitly capture their meaning on the logic level in terms of axioms. Distinct top level ontologies differ in the ontological position they take. For example, the Basic Formal Ontology (BFO) is intended to represent basic entities and relationships that exist in reality, independently from human conception [Grenon et al., 2004]. In contrast, the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) is intended to capture human common sense, reflecting a cognitive bias [Masolo et al., 2003].

However, there are some generally accepted ontological distinctions, such as the mutually exclusive division between classes and individuals (which is linked to the distinction between the ‘is-a’ and the ‘instance-of’ relation, the first one linking classes to superclasses, the second one individuals to classes). A further generally accepted distinction is that between continuants (entities that have no temporal parts and persist through time, maintaining their identity—“things”) and occurrents (entities that have temporal parts and happen or develop through time—“processes”). Continuants can

[1] The strict separation into top level, top domain and domain ontologies is for descriptive purposes only. In practice, some resources span the three levels. An example is the Suggested Upper Merged Ontology (SUMO) with its domain ontologies, see `http://www.ontologyportal.org/` – access date 2012-10-23.

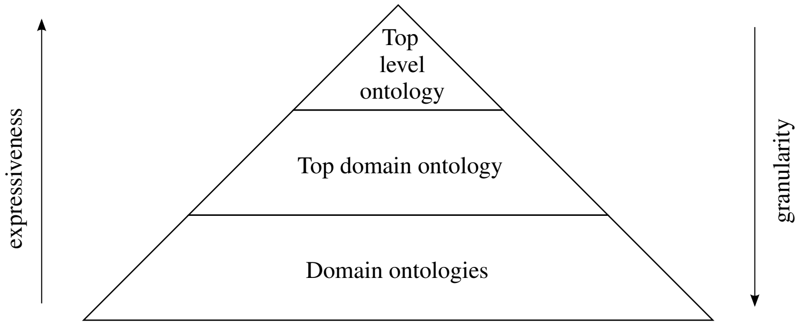


Figure 2.1: Stack of top level, top domain and domain ontologies.

participate in occurrents, and occurrents can have continuants as participants. Continuants are further subdivided into dependent continuants (e.g., roles or functions, which depend on their bearers) and independent continuants (e.g., role or function bearers), and physical continuants (objects with spatial dimension) and *non*-physical continuants (abstract objects). Physical continuants in turn are subdivided into material continuants (physical objects with mass) and immaterial continuants (physical objects without mass, e.g., cavities or wholes). The above-mentioned distinctions refer to pairs of mutually exclusive (also called "disjoint") classes, depicted in figure 2.2.[2]

A top level ontology for relations has been proposed in terms of the OBO Relation Ontology (RO) [Smith et al., 2005]. Although it was designed as an ontology of core relations for the use in biomedical ontologies, the original version of RO (that is in the focus of this thesis) can be taken as a general top level ontology for relations, covering ten binary class level relations ('is-a', 'part-of', 'located-in', 'contained-in','adjacent-to', 'transformation-of', 'derives-from', 'preceded-by', 'has-participant' and subrelation 'has-agent'). They are provided with strict definitions, which rely on the corresponding instance level relations, and specifications on whether the relations are transitive, reflexive, or antisymmetric [Smith et al., 2005, table 3].

2.2.2 Top Domain Ontologies

Top domain (also "upper domain") ontologies cover the basic classes of a domain (for biomedicine, e.g., 'organism', 'tissue', 'cell', 'gene', 'protein', 'drug', 'disease', or 'diagnostic procedure') and relationships between them. Similarly to top level ontologies, they are intended to capture the meaning of classes by means of axioms. Top

[2] Class disjointness in OWL, used as reference formalism in this thesis, is introduced in section 2.4.3.

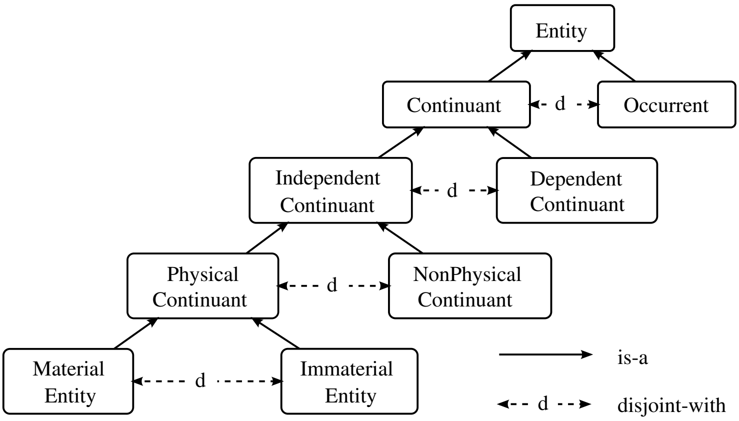


Figure 2.2: Basic ontological distinctions (selection).

domain ontologies constitute an intermediate layer between top level ontologies and domain ontologies (figure 2.1). From a top level ontology viewpoint they play the role of ordinary domain ontologies that further specialize domain-independent classes and relations. From a domain ontology viewpoint they act as top level ontologies that constitute a semantic umbrella and provide formal rigor. The benefit of such an intermediate layer is that it provides a place for formal definitions of basic though domain-specific classes. The latter do not fit into domain-independent top level ontologies, but they are required to clarify and unify the semantics of more specific classes, as they are available in great numbers in lightweight domain ontologies.

Top domain ontologies for biology and biomedicine, beyond BioTop that (in its early stage) has been developed in the context of this thesis, are presented in subsection “Related knowledge resources” of section 8.1.

2.2.3 Domain Ontologies

Domain ontologies cover domain knowledge in terms of domain-specific classes and relationships between them. Common characteristics of domain ontologies are that they excel in a high coverage and granularity, provide a rich set of natural language annotations, but are mostly lightweight (i.e., contain a low proportion of formally defined classes and disjointness relationships between classes).

Collections of biological and biomedical domain ontologies are available in terms of the Open Biomedical Ontologies (OBO) library[3] and the NCBO BioPortal[4]. Existing biomedical ontologies that cover related domains like the domain ontologies GRO (chapter 6) and MaHCO (chapter 7) include the Gene Ontology (GO) [Ashburner et al., 2000], the Sequence Ontology (SO) [Eilbeck et al., 2005], the Cell Ontology (CL) [Bard et al., 2005], the Protein Ontology (PRO) [Natale et al., 2011], the INOH Molecule Role Ontology (IMR) [Yamamoto et al., 2004], the Ontology of Chemical Entities of Biological Interest (ChEBI) [Degtyarenko et al., 2008] and the Foundational Model of Anatomy (FMA) [Rosse and Mejino, 2003]. GO is by far the most popular of these ontologies. It consists of three branches, covering biological processes, molecular functions and cellular components. SO formally represents features and properties of biological sequences and relationships between them. CL classifies cell types of different species along multiple criteria, such as function, histology, and lineage. PRO represents proteins of human and different model organisms and organizes them according to evolutionary relatedness and gene locus. ChEBI classifies natural and synthetic molecular entities that occur or intervene in processes in living organisms according to structure, biological role (e.g., “antibiotic” or “hormone”) and use or application (such as “pesticide” or “drug”). Finally, the FMA is a formal and expressive ontology on the human anatomy.

2.3 Ontology Languages

For the syntactic representation of ontologies different ontology languages have been proposed. For the representation of biomedical ontologies, the OBO flat file format is still widely used, due to its human readability, extensibility and ease of parsing [Day-Richter, 2006]. However, it is an *informal* language, lacking a clear syntax and semantics. The *formal* ontology languages that have been proposed vary with respect to expressive power and decidability, between which a well known trade-off exists. Highly expressive languages, which provide a rich inventory of logical constructs, but are undecidable (e.g., first-order and higher-order logics), are typically used for the representation of ontologies in philosophy that are intended to describe the world in terms of precise logical statements, whereas computability is a secondary concern. In contrast, languages that provide more restricted sets of logical constructs, but are decidable, are typically used for the representation of computer science ontologies that are intended to establish a shared understanding of a domain amongst humans and machines, which requires the possibility for automatic reasoning.

For the representation of computer science ontologies, OWL has been established as a de facto standard [Bechhofer et al., 2004]. OWL has been developed by the Web Ontology Working Group for the representation of Semantic Web ontologies. It is a W3C

[3] `http://obofoundry.org/` – access date 2012-03-15.

[4] `http://bioportal.bioontology.org/ontologies` – access date 2012-03-01.

recommendation since 2004. OWL ties in with previous knowledge representation standards. It extents RDF [Klyne and Carroll, 2004] and RDF Schema [Brickley and Guha, 2004] and its design has strongly been influenced by description logics (DLs), a family of formal knowledge representation languages that rely on decidable fragments of first order logic with slight extensions [Baader et al., 2003]. The original release of OWL comprises the three sublanguages OWL Full, OWL DL and OWL Lite, in the order of decreasing expressiveness. The latter two are decidable, where OWL DL is much more expressive than OWL Lite (e.g., it allows to specify class disjointness and form the complement, union and intersection of classes).

In the work presented here, OWL DL is used as reference formalism because as a sublanguage of OWL it belongs to the current de facto standard for the representation of computer science ontologies, and amongst the three sublanguages of OWL it best balances expressiveness and decidability [Bechhofer et al., 2004]. Furthermore, a mapping between OWL and the OBO flat file format has been proposed that allows to integrate OWL-based work with results achieved by the biomedical ontology community [Golbreich et al., 2007]. OWL 2, a new extended version of OWL [W3C OWL Working Group, 2009] has not been considered because it was not released until 2009.

2.4 OWL DL Ontologies

In this section, a brief introduction to ontologies represented in OWL DL is given. For a formal introduction to description logics the reader is referred to the description logics literature [e.g., Baader et al., 2003, 2007], for an exhaustive description of OWL DL to the OWL Web Ontology Language Reference [Bechhofer et al., 2004] and for an in-depth explanation of the relationship between OWL DL and the equivalent description logic $\mathcal{SHOIN}(\mathbf{D})$ to Horrocks and Patel-Schneider [2004] and Horrocks et al. [2007].

2.4.1 Basic Elements

The basic elements of OWL ontologies are classes, properties and individuals, where properties are further subdivided into object and datatype properties.

- Individuals represent the concrete and countable entities in the world.
- A class is interpreted as a set of individuals. This set is called the “extension” of the class and the individuals therein its “instances”.
- An object property is interpreted as set of pairs of individuals. The set is called the extension and the pairs therein the instances of the property.

- A datatype property is analogously interpreted as a set of pairs of individuals and data values.

The extension of a class or property is related, but not the same as its intension (i.e., its intended meaning). Potential classes of a biomedical ontology are ‘Cell’, ‘Gene’ and ‘Protein’. The class ‘Cell’, for example, would have all individual cells as instances. Relations such as ‘part-of’ and ‘has-agent’, which link pairs of classes (or instances thereof, respectively), are represented as object properties in OWL (the ‘is-a’, ‘equivalent-to’ and ‘disjoint-with’ relations are exception, see below). Relations such as ‘has-age’ and ‘has-name’, which link classes to data values, are represented as datatype properties.

In fact, when talking about classes, properties and individuals of OWL ontologies, usually *named* classes, properties and individuals are meant (in contrast to anonymous ones, see section 2.4.2). They are atomic, i.e., not further reducible and constitute those elements of an ontology that are depicted when the ontology is represented as a graph or that are displayed when the ontology is edited in an ontology editor. A distinctive feature of named ontology elements is that they are provided with a Uniform Resource Identifier (URI) as unique name, consisting of a namespace part and a local name. The local name must be unique in the given namespace, and the namespace part may be abbreviated by a prefix. In the exemplary URI “`http://www.bootstrep.eu/ontology/GRO#Gene`”, the namespace part ends with “#” and the local name is “`Gene`”. Given the prefix “`gro`”, the URI reduces to “`gro:Gene`”. Prefix-namespace mappings used in this thesis are summarized in table D.1.

There are two predefined named classes in OWL. The first one is the top class ‘$\top$’ (‘owl:Thing’) that has all individuals as extension. The second one is the bottom class ‘$\bot$’ (‘owl:Nothing’) that has the empty set as extension.

2.4.2 Constructors

OWL DL ontologies may additionally contain anonymous classes, properties and individuals. They are not directly visible in an ontology graph or when viewing the ontology in an ontology editor. However, we will see that they play a crucial role in assigning explicit meaning to named ontology elements (hence, they occur particularly numerous in expressive ontologies). As their name suggests, anonymous ontology elements lack a name assignment, i.e., they are not provided with a URI. Another difference to named ontology elements is that they are complex, i.e., they are composed of other (atomic or complex) elements by different constructors, in a theoretically arbitrarily deep nesting.

The constructors available in OWL DL are specified in section C.1. For example, the intersection class constructor (see formula C.1) may be used to compose the two classes ‘DNA’ and ‘RNA’ to an anonymous intersection class

$$\text{‘DNA’} \sqcap \text{‘RNA’}. \tag{2.1}$$

It has all individuals as instances that are instances of both constituent classes. Furthermore, the union class constructor (formula C.2) may be used to compose the two classes ‘Gene’ and ‘Protein’ to an anonymous union class

$$\text{‘Gene’} \sqcup \text{‘Protein’}. \tag{2.2}$$

It has all individuals as instances that are instances of at least one of the constituent classes. In addition, the existential and the universal restriction constructors (formulae C.5 and C.6) may be used to specify the anonymous restriction classes

$$\exists \text{‘part-of’}.\text{‘Protein’ and} \tag{2.3}$$

$$\forall \text{‘part-of’}.\text{‘Protein’}. \tag{2.4}$$

The first one is an existential restriction class that comprises all individuals as instances that are related to *some* instance of the class ‘Protein’ by a ‘part-of’ relationship. The second one is a universal restriction class that has all individuals as instances that are *only* related to instances of the class ‘Protein’ by a ‘part-of’ relationship. We will see that both constructors involved play a major role in specifying relationships between classes that rely on other relations than ‘is-a’, ‘equivalent-to’ and ‘disjoint-with’.

2.4.3 Axioms

Facts about a domain are in expressed in form of axioms in OWL ontologies. The types of axioms available in OWL DL are specified in section C.2. Axioms of type class subsumption (formula C.16), class equivalence (formula C.17) and class disjointness (formula C.18), in OWL denoted as ‘rdfs:subClassOf’, ‘owl:equivalentClass’ and ‘owl:disjointWith’, are used to express ‘is-a’, ‘equivalent-to’ and ‘disjoint-with’ relationships between classes.[5] For example, the fact that the class ‘Enzyme’ is a subclass of the class ‘Protein’ is expressed via the class subsumption axiom

$$\text{‘Enzyme’} \sqsubseteq \text{‘Protein’}. \tag{2.5}$$

It implies that the extension of the class ‘Enzyme’ is a subset of the extension of the class ‘Protein’, i.e., every instance of ‘Enzyme’ is also an instance of ‘Protein’. Furthermore, the fact that the class ‘Cell’ is equivalent to the class ‘Zelle’ is expressed via

[5] In fact, class equivalence and class disjointness are reducible to class subsumption [Baader et al., 2003, proposition 2.12].

the class equivalence axiom

$$\text{`Cell'} \equiv \text{`Zelle'}. \tag{2.6}$$

It implies that the classes ‘Cell’ and ‘Zelle’ have the same extension. The fact that the class ‘DNA’ is disjoint from the class ‘RNA’ is expressed via a class disjointness axiom

$$\text{`DNA'} \sqcap \text{`RNA'} \sqsubseteq \bot. \tag{2.7}$$

It implies that the intersection of the extensions of the classes ‘DNA’ and ‘RNA’ is empty, i.e., the two classes do not have common instances.

The use of class equivalence and class subsumption axioms for the specification of formal class definitions will be dealt with in section 2.4.4. For object and datatype properties, appropriate axioms may be specified to state domain and range restrictions, as well as some further characteristics as specified below. Domain and range restrictions of a property (see formulae C.29–C.32) imply that all pairs of individuals (or individuals and data values, respectively) linked by the property are from the specified domain and range, even if this is not explicitly asserted. Object properties may further be specified as being transitive, symmetric, functional or inverse functional (see the formulae C.23 and C.33–C.35). Datatype properties may only further be specified as being functional (formula C.36) [Bechhofer et al., 2004].

2.4.4 Formal Class Definitions

Innately, named ontology classes, properties and individuals are described by their URI only. To explicitly specify the intended meaning of named classes, axioms must be stated that relate them to other named atomic or anonymous complex classes. A class equivalence axiom (see formula C.17) with a named class on the left hand side and an arbitrary class on the right hand side is called a “definition” of the named class, or “formal definition” to distinguish it from a verbal definition. A definition determines necessary and sufficient conditions for the belonging of individuals to the extension of the named class. Named classes with at least one definition are called “defined”, all remaining ones “primitive”. An example for a definition is

$$\text{`Enzyme'} \equiv \text{`Protein'} \sqcap \exists\text{`has-function'}.\text{`CatalyticActivity'} \tag{2.8}$$

It states that the class ‘Enzyme’ comprises exactly those individuals that are an instance of the class ‘Protein’ and that are related by a ‘has-function’ relationship to at least one instance of the class ‘CatalyticActivity’.

In the context of this thesis, a subsumption axiom (see formula C.16) with a named class on the left hand side and an arbitrary class on the right hand side is analogously called a “partial definition” (or “partial formal definition”) of the named class. A partial definition determines necessary, but not sufficient conditions for the belonging of individuals to the extension of the named class. Named classes with at least one partial definition are called “partially defined”. Examples for partial definitions are

$$\text{`Polymerase'} \sqsubseteq \text{`Enzyme'} \quad \text{and} \tag{2.9}$$

$$\text{`ProteinDomain'} \sqsubseteq \exists\text{`part-of'}.\text{`Protein'}. \tag{2.10}$$

The first example states that each instance of ‘Polymerase’ is also an instance of ‘Enzyme’. The second example states that the class ‘ProteinDomain’ comprises a (not further specified) subset of individuals that are related by a ‘part-of’ relationship to at least one instance of the class ‘Protein’.

In OWL DL ontologies, full and partial formal class definitions that involve appropriate restriction classes on their right hand side are the means by which relationships between (instances of) classes are expressed that rely on object properties (such as ‘has-function’ and ‘part-of’). Before a new relationship that is based on an object property is stated in an OWL DL ontology, the developers must be clear on the precise nature of the relationship and choose the type of restriction class accordingly [see, e.g., Stevens et al., 2007].

2.4.5 Reasoning

From the axioms stated in an OWL DL ontology logical consequences can be computed. This process is called automatic “reasoning”, “inferencing” or “classification”. Typical reasoning problems include checking the satisfiability of classes, checking ‘is-a’, ‘equivalent-to’ and ‘disjoint-with’ relationships of pairs of classes, checking class instantiation and checking the logical consistency of ontologies.

To be able to understand the main characteristics of automatic reasoning, the model theoretic semantics behind OWL DL ontologies must be considered. It is centered around the notion of *interpretations* [Baader et al., 2003, section 2.2.1]. An interpretation consists of a non-empty set denoting the domain of the interpretation and an interpretation function. The interpretation function assigns a subset of the domain of interpretation to every class, a binary relation on the domain of interpretation to every property, and an element of the domain of interpretation to every individual. The axioms, of which an ontology consists, act as constraints on interpretations. An interpretation that satisfies all axioms of an ontology is called a “model” of it [Baader et al., 2003, section 2.2.2].

A class of an ontology is satisfiable if there is a model of the ontology in which the class denotes a non-empty set. A class subsumes another class of an ontology if for all models of the ontology the subsumption holds (analogously for class equivalence and disjointness). An individual is an instance of a class if for all models of the ontology the instantiation holds. An ontology is consistent if it has at least one model [Baader et al., 2003, 2007].

For OWL different reasoning tools have been proposed [see, e.g., Motik et al., 2009; Sirin et al., 2007; Tsarkov and Horrocks, 2006]. Reasoning is useful for different purposes. Using a reasoner during ontology development and maintenance fosters the creation of correct ontologies [Rector et al., 2004]. Using a reasoner for the automatic classification of large, multi-hierarchical ontologies after updating them helps to reduce error-prone manual maintenance work [Rector, 2003]. In addition, reasoners can support querying ontologies by computing the subclasses, superclasses and instances of a class.

However, when automatic reasoning is employed, it is important to be aware of the fact that OWL DL ontologies rely on the open world assumption (OWA), while the unique name assumption (UNA) is *not* adhered to. According to the OWA, absent information indicates the lack of knowledge, rather than negative information (the latter is the case for the “closed world assumption”, on which, e.g., classical databases rely). To prevent unwanted side effects of “open world” reasoning, the intended meaning of named ontology elements should explicitly be stated in terms of axioms. This is achieved, e.g., by introducing explicit class disjointness axioms (formula C.18), formal definitions (section 2.4.4), the complementation of existential restrictions (formula C.5) with universal restrictions (formula C.6) and vice versa, where necessary [Rector et al., 2004]. A universal restriction that complements an existential restriction is sometimes called a “closure restriction” or “closure axiom” [Rector et al., 2004]. The complementation of universal with existential restrictions is to avoid trivial satisfiability. Not adhering to the UNA means that two named ontology elements with different URIs are not necessarily different until their difference is explicitly stated. For example, two differently named individuals are assumed to be the same or distinct, until an explicit individual equality (formula C.24) or inequality axiom (formula C.25) is stated.

2.4.6 Annotations

Named classes, properties and individuals of OWL DL ontologies may be provided with natural language annotations (e.g., labels or verbal definitions) using annotation properties as known from RDF. Annotation properties are designed in a way that does not change the formal semantics of an ontology. They must not be used in property axioms (e.g., it is not allowed to specify a domain or range for annotation properties) and they remain uninterpreted by OWL reasoners. Some annotation properties are predefined in OWL (e.g., ‘rdfs:label’ and ‘rdfs:comment’) or elsewhere (e.g., ‘skos:prefLabel’, ‘skos:altLabel’ and ‘skos:definition’ are annotation properties specified in the SKOS vocabulary [Miles and Bechhofer, 2009]). However, also custom properties may be defined.

Natural language annotations are intended to communicate the meaning and status of ontology elements to human users of ontologies. In addition, they support the automatic detection of related elements in different ontologies and related knowledge resources and they help to identify verbal mentions of ontology elements in natural language documents.

2.5 Ontology Alignments

Different ontologies may contain related or even overlapping contents. An ontology alignment is a means to make ‘equivalent-to’, but also other types of relationships between elements or axioms from different ontologies explicit. It consists of a set of correspondences (“mappings”) that specify which elements or axioms in different input ontologies are related, and which type of relationship holds between them. The process of detecting such correspondences and generating the ontology alignment is called “ontology matching” [Euzenat and Shvaiko, 2007, chapter 2.4]. It may be carried out manually, semi or fully automatically (see section 3.5).

In the context of this thesis, the standard case is considered in which an alignment links *pairs* of ontologies that record relationships between *named* ontology elements. In such an alignment, a correspondence consists of a pair of named ontology elements (e.g., classes or individuals), one from each input ontology, the type of relationship that has been detected between them (e.g., ‘equivalent-to’, ‘is-a’ or ‘related-to’) and optionally a confidence value. The number of correspondences, in which the elements of the input ontologies are involved across the alignment relations, determines the cardinality of an alignment. For example, if each element of the first input ontology is mapped to exactly one element of the second input ontology, and vice versa, the cardinality is “one-to-one”, and if each element of the first input ontology is mapped to zero, one or more elements of the second input ontology, and vice versa, it is “many-to-many” .

To facilitate the automatic analysis and easy reuse of ontology alignments, a uniform, automatically processable format for the syntactic representation of ontology alignments is required. Currently, there are two widespread approaches. The first one is to use the RDF-based format that Euzenat [2004] proposed for the representation of alignments between pairs of ontologies. Using this format, correspondences are represented as “cells” that consist of a pair of aligned ontology elements, a relation type, and a confidence value. The format has been provided together with the Alignment API[6], a Java API for creating, editing, comparing, loading and saving alignments expressed in

[6] `http://alignapi.gforge.inria.fr/` – access date 2012-09-07.

it. A disadvantage of the format is that it does not rely on a formal ontology language and hence lacks strict semantics, which would allow to directly reason on alignments expressed in this format.

The second approach is to express ontology alignments in terms of axioms of an existing formal ontology language, such as OWL. Choosing OWL, ‘equivalent-to’ or ‘is-a’ relations between pairs of classes from different ontologies could be expressed in terms of class equivalence (formula C.17) or class subsumption axioms (formula C.16). Using this approach, an alignment can easily be integrated with OWL-based input ontologies. This facilitates logical consistency checking on the *merger* of alignment and input ontologies, which is a huge advantage. However, a difficulty with this approach is that it requires very strong commitments (possibly stronger ones than a particular ontology matching approach supports). For example, all correspondences must be stated concretely, i.e., there is no possibility to consider confidence values. Furthermore, for correspondences based on other relations than ‘equivalent-to’, ‘is-a’ and ‘disjoint-with’, a decision must be made about how to represent the detected relationship (possibly by means of an anonymous existential restriction class, see formula C.5).

2.6 Conventions

In this thesis, the class hierarchy of an ontology that is directly stated in terms of class subsumption axioms is called “asserted” class hierarchy, and the class hierarchy computed by a reasoner “inferred” class hierarchy.

As mentioned in section 2.4.1, in OWL DL ontologies named ontology elements are syntactically represented by URI references, where a URI consists of a namespace part and a local name. In cases in which the namespace part of URIs is either known from the context or not critical in the respective context (the latter concerns, e.g., mentions of classes in examples), it will be omitted in the remainder of this thesis for the sake of readability.

Throughout this thesis, the term “relation” will be used in the sense of relation type. If an instance of a relation is meant, the term “relationship” will be used. The subsumption, equivalence and disjointness relation will be called ‘is-a’, ‘equivalent-to’ and ‘disjoint-with’ relation, respectively. Relations that are represented as object properties in OWL (e.g., ‘part-of’ or ‘participates-in’) will sometimes be called “conceptual relations”, and relations that are represented as datatype properties (e.g., ‘has-age’) “data relations”.

A common practice to review an ontology and assess its expressiveness and relevance for a given purpose is to check which top level classes and relations occur, and count the number of classes and relationships between classes, itemized by relation type. Counting ‘is-a’, ‘equivalent-to’, and ‘disjoint-with’ relationships in OWL DL ontologies is straightforward, because they are stated directly between classes by means of class subsumption (formula C.16), class equivalence (formula C.17) and class disjointness axioms (formula C.18), respectively. However, this is different with conceptual relationships. They must be stated between instances of classes, by means of anonymous restriction classes (see the formulae C.5–C.9) that might be nested in arbitrarily complex anonymous classes. Hence, for counting conceptual relationships a counting mode must be determined.

In the context of this thesis, a rather simple approach is used to achieve relationship counts for conceptual relations of ontologies. Given an ontology, all distinct, ordered triples (C_i, R, C_j) of named classes C_i, C_j and a named conceptual relation R are computed, for which at least one of the following axioms is contained in the ontology

$$C_i \sqsubseteq \exists R.C_j \sqcap \ldots \tag{2.11}$$

$$C_i \sqsubseteq \forall R.C_j \sqcap \ldots \tag{2.12}$$

$$C_i \equiv \exists R.C_j \sqcap \ldots \tag{2.13}$$

$$C_i \equiv \forall R.C_j \sqcap \ldots, \tag{2.14}$$

where “$\sqcap \ldots$” denotes that the preceding anonymous restriction class is allowed to occur as constituent of an (arbitrarily deeply nested) intersection class. This simple approach has been chosen because it covers the most plausible cases of relationships between instances of classes and avoids unintuitive cases, such as negated relationships or relationships that only potentially apply (as in the case when union classes would be involved).

In OWL ontologies conceptual relationships can only be specified between instances of classes, and not between classes themselves. However, given that in an ontology two classes C_i, C_j and a conceptual relation R are linked by at least one axiom of the types 2.11–2.14, for the sake of readability, it will sometimes be stated in this thesis that C_i and C_j are related by a conceptual relationship of type R, or that a conceptual relationship of type R holds between the classes C_i and C_j, respectively.

In the context of this thesis, annotations that represent natural language names of ontology elements will be called “labels” instead of “names” in order to distinguish them from both, URIs (which are sometimes called “names” of ontology elements) and local names (which are part of URIs).

The term “ontological background knowledge” will be used to collectively refer to ontologies and ontology alignments.

Part II

Methods

Chapter 3

Related Work

In this chapter, different strategies for building ontological background knowledge are presented. Ontologies can be developed manually—from scratch or reusing existing knowledge resources—(section 3.2), automatically by utilizing ontology learning techniques (section 3.3), or semi-automatically based on patterns (section 3.4). Alternatively, if domain knowledge is already formally represented but distributed over different ontologies the latter may be bridged by creating an ontology alignment (section 3.5). In addition, life cycle models (section 3.1) and evaluation approaches (sections 3.6 and 3.7) for ontologies and ontology alignment are presented, the latter being dedicated to assuring the quality of ontologies and ontology alignments. Finally, an overview of knowledge resources is given that are commonly referred to as biomedical ontologies (section 3.8).

3.1 Life Cycle Models

Ontologies and ontology alignments are artifacts and artifacts have a life cycle. However, neither for ontologies nor for ontology alignments there is a consensus about a definite life cycle model, as yet. In fact, for ontologies different life cycle models have been proposed. Most of them have their origin in the neighboring field of software engineering, where the study of life cycle models has a long tradition. Probably the most prominent model from software engineering is the basic "waterfall" model, in which life cycle stages are processed in strictly consecutive order [Royce, 1970]. However, already in the original publication the model has been contrasted to more flexible ones. Examples for more flexible models are "iterative and incremental" models, "evolutionary prototyping" and "spiral" models. In the case of "iterative and incremental" models life cycle stages are run in several iterations and the software is developed incrementally [Larman and Basili, 2003]. In case of "evolutionary prototyping" an incomplete version of the software is created and constantly refined, allowing for early user involvement and feedback. "Spiral" models combine the ideas of controlled, iterative and incremental modeling with risk assessment and evolutionary prototyping [Boehm, 1986]. An overview of software life cycle models is given by Davis et al. [1988].

A waterfall-like life cycle model for ontologies has been proposed by Uschold and King [1995]. It comprises the stages "identify purpose", "ontology building" (comprising ontology capture, ontology coding, and the integration of existing ontologies), "evaluation", and "documentation". A very similar, also staged model has been proposed by Stevens et al. [2000], in an article about ontology-based knowledge representation for bioinformatics. In contrast to these strictly staged models, Fernandez et al. [1997] have proposed a life cycle model for ontologies called "evolving prototype", which resembles the "evolutionary prototyping" model from software engineering. The model comprises the stages "specification", "conceptualization", "formalization", "integration", "implementation" and "maintenance". Furthermore, it covers the support activities "knowledge acquisition", "documentation" and "evaluation", which concern all of its stages. A main characteristic of this model is that it allows the ontology developer to go back and forth between stages anytime in the ontology development process. An overview of life cycle models for ontologies is given, e.g., by Mizoguchi [2003] and Suárez-Figueroa and Gómez-Pérez [2008].

Given that biomedicine is one of the major fields of application of ontologies and as such comes with specific characteristics and challenges, it is remarkable that available life cycle models for ontologies are mostly domain-independent. In addition, particularly those models that derive from the field of software engineering remain rather abstract, i.e., they miss a detailed description that could help ontology developers to realize the individual life cycle stages in practice.

In contrast to ontologies, the study of the life cycle of ontology alignments has just started. Probably the first and so far only life cycle model for ontology alignments is the five-staged model by Euzenat et al. [2008]. It comprises an ontology matching stage, followed by an iterative loop through an evaluation and an enhancement stage, a communication stage, in which the ontology is released, and a terminal exploitation stage, in which the alignment is actually used [Euzenat et al., 2008, section 3.2]. A planning and a maintenance stage are missing. Provided that the impact of ontology alignments as bridge between ontologies used in real world applications is continuously growing, there is a need to catch up as far as comprehensive life cycle models are concerned.

3.2 Manual Ontology Development

Manual ontology development is probably the most widely-used strategy to obtain new ontologies. Since it has a strong creative component [Noy and McGuinness, 2001], the structure and contents of a manually created ontology very much depend on the skills of the ontology developer. However, there are many attempts to standardize the development process. Inspired by the field of software engineering, life cycle models have been introduced for ontologies that structure the development process into different stages with associated subtasks (see section 3.1). In addition, ontology development methodologies have been proposed. They consist of sets of guidelines that specify how to tackle the tasks associated with individual life cycle stages. For example, the Methontology approach by Fernandez et al. [1997] deals with ontology construction from scratch.

Guidelines are recommendations that are intended to guide and standardize a procedure (in this case ontology development) by determining which activities should be carried out and how. They usually represent best practices that are based on years of experience. Particularly comprehensive guidelines have been specified for the design and implementation stage of ontology development. They give detailed instructions on how to conceptualize a domain and implement the conceptualization in terms of an ontology. The usual form of distribution for such guidelines are guides and tutorials, such as the popular guide to ontology development by Noy and McGuinness [2001] or the guide on practical aspects of ontology engineering by Mizoguchi [2003]. A proposal targeting the implementation of modular ontologies has been made by Rector [2003]. As a complement to design and implementation guidelines also common ontology development mistakes have been recorded and were published together with guidelines how to avoid them. For example, based on practical experience from courses on developing OWL DL ontologies, Rector et al. [2004] compiled a set of common pitfalls and presented them together with guidelines for avoiding them. Furthermore, based on a critical review on an ISO standard for data integration, sharing and exchange, Smith [2006] proposed a set of "general principles which a good ontology should satisfy" [Smith, 2006, page 16]. Also Pease [2011] has compiled ontology development guidelines and pitfalls in the context of developing SUMO (see page 15).

For the collaborative development of biomedical ontologies best practices have been proposed in terms of the OBO Foundry principles [OBO Foundry, 2012]. They deal with various ontology management issues such as ontology documentation, maintenance and release, exceeding the scope of mere design and implementation guidelines by far. The principles rely on community consensus and are subject to change. They are actively developed by the OBO Foundry, an initiative of developers of different biomedical ontologies (the "OBO Foundry ontologies"), who have committed themselves to the quality assurance of collaborative ontology development in the field of biomedicine [Smith et al., 2007].

Manual ontology development is increasingly supported by software tools. Many new tools are developed and existing tools are improved and get more powerful with respect to functionality and performance. Most importantly, ontology editors enable the creation of ontologies in a formal language using a graphical user interface (GUI). An example is the free and open source ontology editor Protégé[1]. It comes with an extension for the construction, manipulation and visualization of OWL ontologies, called Protégé-OWL [Knublauch et al., 2004]. How to use Protégé-OWL for building OWL ontologies is described in detail in the Protégé tutorial by Horridge [2011]. Protégé is free and open source. Due to its plug-in architecture, it can easily be extended. The rather recent variant Collaborative Protégé, with the lightweight ontology editor WebProtégé as Web-client, allows multiple developers to edit the same ontology at the same time [Tudorache et al., 2008]. Furthermore, it supports ontology maintenance by enabling the annotation of ontology elements and changes and voting on changes. A workflow supporting the whole life cycle of biomedical ontologies involving Protégé is underway [Noy et al., 2010].

As an alternative to GUI-based ontology development, APIs that support the reading, processing, writing and querying of OWL ontologies by programming have been proposed. Two prominent examples are the Java-based Jena framework[2] and the OWL API[3] [Horridge and Bechhofer, 2011]. Both with ontology editors and the mentioned APIs, tools for automatic reasoning (called "reasoners" or "classifiers") have been integrated to automatically classify and check the logical consistency of ontologies. Popular reasoners for OWL DL ontologies include Fact++[4] [Tsarkov and Horrocks, 2006], Pellet[5] [Sirin et al., 2007] and HermiT[6] [Motik et al., 2009], amongst others. They differ with respect to the algorithms they use and the language constructs they are able to process. Both factors have a major impact on run time. Currently, "tableau"-based approaches are the most prominent ones [Baader et al., 2007], where tableau-based reasoners represent highly optimized implementations of tableau decision procedures. HermiT, which relies on a novel "hypertableau" calculus, currently stands out with respect to performance [Glimm et al., 2010; Motik et al., 2009]. Tools that are able to check ontologies for mistakes that raise *no* logical inconsistencies are just emerging. An example is the ontology pitfall scanner Oops! [Poveda-Villalon et al., 2012].

Also for the maintenance of ontologies tool support is available. First of all, version control systems, which have originally been designed for software development, may be used to track and administer the changes made to ontology files. They enable the collaboration of developers who work asynchronously or remote from each other. An

[1] `http://protege.stanford.edu/` – access date 2012-09-07.

[2] `http://jena.apache.org/` – access date 2012-09-07.

[3] `http://owlapi.sourceforge.net/` – access date 2012-09-07.

[4] `http://owl.man.ac.uk/factplusplus/` – access date 2012-09-07.

[5] `http://clarkparsia.com/pellet/` – access date 2012-09-07.

[6] `http://hermit-reasoner.com/` – access date 2012-09-07.

example is the open source version control system Apache Subversion[7], which relies on a client-server architecture [Pilato, 2004]. Furthermore, dedicated ontology maintenance software has been published that provides sophisticated features such as advanced change tracking (allowing to annotate, negotiate, accept or reject individual changes, browse changes, and view the history of changes), rights control, and embedded consistency checking. An example is the system developed by Noy et al. [2006]. It consists of the Change and Annotation Ontology (CHAO) and two specific plug-ins for Protégé.

To sum up, ontology developers are supported by increasingly powerful tools for the creation, modification, consistency checking and visualization of ontologies. Furthermore, they have access to a growing number of life cycle models for ontologies, guidelines for ontology design and implementation and entire methodologies for ontology development. It is therefore increasingly becoming a challenge to select the most appropriate life cycle model and guidelines for a specific ontology project and to strictly adhere to them in practice.

3.3 Automatic Ontology Learning

An alternative strategy to manual ontology development is automatic ontology learning (OL). The most prominent types of OL approaches deal with the construction of ontologies based on natural language text documents. A typical text-based OL system extracts relevant terms and associated term variants from text, groups them to concepts (“classes”), and subsequently identifies ‘is-a’ relationships between them [Cimiano, 2006]. OL approaches dealing with “ontology population” additionally extract class instances from text, such as the approach underlying the Sofie framework [Suchanek et al., 2009]. It extracts facts from free-text parts of Wikipedia articles to extend ontologies with instance data. An overview of text-based OL approaches and systems is given by Cimiano [2006].

A quite recent development are approaches to learn more expressive ontologies, e.g., ontologies that contain additional semantic relation types or formal class definitions. For example, Ciaramita et al. [2005] have proposed an unsupervised approach to learn semantic relations from text in the field of molecular biology. Furthermore, Völker et al. [2008] have proposed the LExO (“Learning Expressive Ontologies”) approach. Based on appropriate definitory sentences, LExO can generate formal definitions for previously generated classes. A comprehensive overview of this new subfield of OL is given by Völker [2009] in her doctoral thesis on “Learning expressive ontologies”.

An additional type of OL approaches support the semi-automatic development of on-

[7] `http://subversion.apache.org/` – access date 2012-09-07.

tologies. These approaches are intended to assist, rather than replace the ontology developer [Fortuna et al., 2007; Wächter and Schroeder, 2010]. For example, primed with key words, the Dog4Dag system by Wächter and Schroeder [2010] utilizes NLP techniques to extract new terms, classes and relations from suitable text corpora and proposes them to the ontology developer for further assessment. It is available as plug-in for Protégé, amongst others.

The potential of OL lies in the automatic creation of (possibly large-scale) ontologies for arbitrary domains, independently of whether a domain expert is available for ontology development. However, automatically learned ontologies are still inferior to manually created ones with respect to expressiveness. In addition, due to their automatic creation they must be checked for contradictory contents before they are used in practice. This might change when recent approaches dealing with learning expressive ontologies and techniques to derive improved logically consistent ontologies from automatically generated ones [see, e.g., Haase and Völker, 2008] have further matured.

3.4 Pattern-Based Ontology Development

A rather recent strategy to build ontological background knowledge is the semi-automatic development of ontologies based on design patterns. Pattern-based development adds automatic aid to manual ontology development. A design pattern describes a common design problem and proposes a solution to it on a level of abstraction that makes it fit in many similar situations. Originating from the architectural domain, the use of design patterns has quickly been established in other application areas, including software engineering and more recently ontology development. Ontology design patterns are usually created by expert ontology engineers. They are intended to enable even non-experts to successfully solve complex knowledge engineering problems by assisting them in using established best practices [Blomqvist, 2009]. In addition, using the same patterns for the construction of different ontologies fosters ontology integration [Gangemi, 2005].

The wide spectrum of already available ontology design patterns (ODPs) is illustrated by a typology for ODPs, proposed by Blomqvist [2010]. An introduction to ODPs and an overview of existing patterns is given by Blomqvist [2009] in her doctoral thesis on semi-automatic ontology construction based on patterns. Two public catalogs on ODPs are provided in terms of the Semantic Web portal “OntologyDesignPatterns.org”[8] and the “Ontology Design Patterns Public Catalog”[9]. A recent series of workshops on ODPs indicates that pattern-based ontology development is gaining momentum.[10]

[8] `http://ontologydesignpatterns.org/` – access date 2012-12-05.

[9] `http://odps.sourceforge.net/odp/html/` – access date 2012-12-05.

[10] Workshops on ontology patterns (WOP) took place in conjunction with ISWC 2009, 2010, and 2012.

As a complement to design patterns, also anti-patterns have been studied, i.e., "worst practices" that should be avoided when developing ontologies. Examples include the set of common OWL modeling mistakes compiled by Rector et al. [2004] and the modeling pitfalls cataloged by Poveda-Villalon et al. [2012].

The idea of improving the results of manual ontology development by using ODPs, especially if it is carried out by non-experts, is appealing. However, patterns beyond the most established ones (e.g., "closure axioms" and "value partitions", which are both described in the user guide to Protégé [Horridge, 2011, pages 63 and 67]), should be used with care. The reason is that there is evidence that many patterns are published without being sufficiently studied and evaluated before [Hammar and Sandkuhl, 2010]. Furthermore, ODP-based ontology development is currently hampered by the fact that only few ODPs are integrated with popular ontology editors (e.g., Protégé) although this integration seems to be crucial for the successful adoption of ODPs by developers of biomedical ontologies [Mortensen et al., 2012].

3.5 Ontology Matching

Ontology matching denotes the process of finding semantic relationships between related elements from different ontologies. The matching process results in an ontology alignment (see section 2.5). It may serve as bridge between the input ontologies. Ontology matching can be carried out manually, semi-automatically or automatically. An example for a manual matching approach is the one that has been used to match the top domain ontology BioTop with the UMLS Semantic Network [Schulz et al., 2009a]. An example for a combined manual and automatic approach is the work by Bodenreider et al. [2005]. They match two anatomy ontologies using an automatic matching approach and add a manual curation step. A revised version of the resulting alignment is used as reference alignment in the Anatomy track of the Ontology Alignment Evaluation Initiative (OAEI) evaluation campaigns (see below).

Since manual ontology matching is a labor-expensive and time-consuming task, various automatic approaches have been proposed. They can roughly be classified into terminological, structural, extensional and semantic approaches, depending on which kind of data they use as input [Euzenat and Shvaiko, 2007, chapter 3]. While terminological approaches mainly utilize the natural language annotations of ontology elements, structural approaches focus on the position of ontology elements in the labeled graph structures that ontologies represent, extensional approaches on the individuals of ontology elements and semantic approaches on the semantic interpretations of the latter, respectively. In most ontology matching systems, several types of approaches are combined. An overview of ontology matching systems is given by Euzenat and Shvaiko [2007, chapter 6].

To evaluate and monitor the performance of automatic ontology matching systems, annual international evaluation campaigns have been carried out since 2005, organized by the OAEI [Euzenat et al., 2011b]. The OAEI evaluation campaign 2012 had seven tracks. Amongst them two tracks addressed the matching of biomedical ontologies (the "Anatomy track" and the "Large BioMed track") and one with the matching of particularly expressive ontologies on conference organization (the "Conference track"). In the Anatomy track, a manually curated alignment was used as reference standard and in the Conference track fully manually created alignments. In the Large BioMed track, the evaluation relied on mappings derived from the UMLS Metathesaurus (see page 41). The results of the campaign are available in Aguirre et al. [2012] and at the OAEI website[11].

Ontology matching qualifies as a strategy for building ontological background knowledge in cases in which the required knowledge is already represented in a formal ontology language, though distributed over different ontologies. However, the still growing number of available biomedical ontologies and the low number of available alignments between them reveal that ontology matching is not widely-recognized as a strategy for building ontological background knowledge for biomedicine, as yet.

3.6 Evaluation of Ontologies

Various approaches have been proposed for the evaluation of ontologies. Surveys are given, e.g., by Hartmann et al. [2004], Brank et al. [2005], Gangemi et al. [2005a,b] and more recently, Vrandečić [2010]. Brank et al. [2005] classify ontology evaluation approaches on two axes. The first axis deals with the level of ontologies that an approach targets at. A distinction is made between the level of classes, instances, the vocabulary used to represent classes and instances, taxonomy, additional semantic relations, syntax, architecture and design and context or application. The second axis deals with the standard of comparison that an approach uses. Here a distinction is made between gold standard, human assessment, data, and application-based approaches. An alternative classification is proposed by Vrandečić [2010] in his doctoral thesis on ontology evaluation. He introduces a framework for ontology evaluation that classifies approaches also on two axes. On the first axis the aspect of ontology that an approach targets is considered. A distinction is made between the vocabulary (comprising URIs and literals), syntax, structure, semantics, representation and the context or applications of ontologies. On the second axis, the quality criterion that is used by an approach is considered. Here a distinction is made between accuracy, completeness, consistency, computational efficiency, conciseness, clarity, adaptability and organizational fitness.

A family of particularly popular ontology evaluation approaches relies on graph struc-

[11] http://oaei.ontologymatching.org/2012/ – access date 2012-12-06.

ture-based measures that target the (asserted or inferred) class hierarchy of ontologies. Graph structure-based measures check, for example, if the class hierarchy represents a tree or a forest (i.e., a disjoint union of trees), if it is non-cyclic, its maximum or average depth or breadth, its "tangledness" (i.e., the ratio between all classes and classes with at least two super classes in the class hierarchy) or the "fan-outness" of its leaf classes (i.e., the ratio between leaf classes and all classes) [Gangemi et al., 2005a]. The popularity of structural measures results from the fact that they can be applied fully automatically. Furthermore, they result in numbers that can easily be compared, visualized, and matched against constraints to distinguish "high quality" from "low quality" ontologies. Additional automatic evaluation approaches include the formal competency questions approach proposed by Grüninger and Fox [1995] and data-driven approaches. The latter use datasets from databases or text corpora as comparison standard. Data driven-approaches are popular for the evaluation of ontologies on the instance level. An example is the approach by Netzer et al. [2009].

Another popular approach to ontology evaluation is OntoClean [Guarino and Welty, 2002]. It is directed at the formal analysis of 'is-a' relationships between classes, using a set of meta-properties (originally "identity", "unity", "rigidity", and "dependence"). In the first step of the approach, meta-properties are assigned to ontology classes. The meta-properties assigned to a class are passed on to its subclasses. In the second step, the ontology is checked against predefined constraints on allowed combinations of meta-properties. Constraint violations indicate mistakes in the class hierarchy. However, an obstacle regarding this approach is that the manual assignment of meta-properties to classes is very laborious. Tools that have been proposed for automating this task [see, e.g., Völker et al., 2005] are not widely used, as yet.

Each of the mentioned approaches targets at the evaluation of one particular level of ontology only, such as the class hierarchy or the instance level. However, the validity and (re)usability of ontologies always depends on various different levels and aspects. Indeed, most of them are meanwhile target of ontology design and implementation guidelines and best practices. Yet, only few evaluation approaches are available, so far, that check the compliance of ontologies with guidelines. One of them is the work by Zhang and Bodenreider [2006], who present several ontology modeling principles and check if the FMA (see page 18) is compliant with them. Another example is the work by Poveda-Villalon et al. [2012], who developed a tool for screening OWL ontologies for common modeling pitfalls.

3.7 Evaluation of Ontology Alignments

The evaluation of ontology alignments has so far primarily been addressed from the perspective of assessing the quality of automatic ontology matching systems. Different approaches have been proposed to evaluate the performance of automatic matching systems and the alignments they produce. These include the manual analysis of correspondences in automatically generated alignments [see, e.g., van Hage et al., 2005], the comparison of alignments to reference alignments [see, e.g., Jain et al., 2010; Mascardi et al., 2009], measuring the extent to which alignments preserve the structural properties of the input ontologies [Joslyn et al., 2009], checking the coherence of alignments with respect to the input ontologies [see, e.g., Meilicke and Stuckenschmidt, 2009] and evaluating alignments within an application [see, e.g., van Hage et al., 2007]. Amongst these approaches, the evaluation of ontology alignments by comparison to a reference alignment is by far the most common one. It has been used in international ontology alignment evaluation campaigns (see page 37) for many years [Euzenat et al., 2011b]. Meanwhile even special precision and recall measures have been proposed that in contrast to standard precision and recall, as known from information retrieval, reflect the fact that hierarchical structures are being evaluated [Ehrig and Euzenat, 2005; Euzenat, 2007].

For the evaluation of automatic ontology matching systems preferably manually created (or at least curated) ontology alignments are used as reference standard because of their expected precision. Despite the fact that their quality is of paramount importance for the credibility of evaluation results based on them, the evaluation of such alignments themselves has been addressed only fragmentary, so far. Ceusters [2006] introduced a metric for measuring the quality of the input ontologies of an alignment and the ontology resulting from the merger of the input ontologies and the alignment. Meilicke et al. [2009] proposed a Web-based tool that supports human alignment curators in detecting and solving conflicts in alignments, capitalizing on the outcome of logical reasoning processes. As a follow-up, Meilicke and Stuckenschmidt [2009] proposed an automatic approach for analyzing the coherence of alignments with respect to their input ontologies. Both approaches have been used to improve the manually created reference alignments of the OAEI Conference track. Furthermore, Joslyn et al. [2009] presented an approach for checking the preservation of structural properties of the input ontologies of alignments and applied it to the anatomy alignment used as comparison standard in the OAEI Anatomy track. However, aspects of ontology alignments beyond their structure and logical consistency currently remain unevaluated, due to the lack of appropriate evaluation approaches. An inspection of several manually created or curated ontology reference alignments showed that this is a real deficiency [Beisswanger and Hahn, 2012]. Despite the enormous efforts that certainly have gone into their development, they turned out to suffer from both, content-specific shortcomings that restrict their validity and technical deficiencies (down to the accessibility and formatting issues) that hamper their reuse.

3.8 Biomedical Ontologies

Various biomedical ontologies have already been created and made publicly available. However, it is important to notice that in biology and biomedicine the term "ontology" is used to refer to various types of artifacts that serve the purpose of knowledge organization, independent from the fact whether they are expressed in a formal language or not [Bodenreider and Stevens, 2006; Rubin et al., 2008]. Accordingly, not every artifact that is called a biomedical ontology qualifies as ontological background knowledge for biomedicine in the previously introduced reading of the term (see section 2.6). In fact, so-called biomedical ontologies may be classified into terminologies (in a rather broad sense), information and data models, and ontologies in a stricter sense [Rubin et al., 2008].[12] Below the three groups are described in more detail.

So-called biomedical ontologies that belong to the heterogeneous group of terminologies have in common that they are strongly natural language-oriented. They are usually designed for a well-defined purpose (e.g., document annotation, document indexing, document retrieval, database annotation, or in the medical field disease classification and billing). In some cases they are even named after the purpose they are used for (e.g., "indexing vocabulary" or "classification scheme"). The simplest form of terminologies that are sometimes called ontologies are glossaries. A glossary consists of a set of terms provided with natural language glosses that informally specify their meaning. The next stage of complexity is constituted by controlled vocabularies (CVs). In a CV, terms are provided with unambiguous natural language definitions instead of glosses, each sense of ambiguous terms is specified by a differently named instance, and a preferred term is specified if several terms refer to the same concept. CVs may additionally be hierarchically organized, where different hierarchy-forming relations are possible (e.g., 'broader-than'/'narrower-than'). Finally, there are thesauri, hierarchically organized CVs that contain additional, non-hierarchy-forming relation types (e.g., 'equivalent-to', 'associated-with' or 'related-to').

The largest terminology system for biomedicine is the UMLS Metathesaurus[13]. It integrates more than 100 individual source vocabularies (controlled vocabularies, thesauri and a few biomedical ontologies), of which the MeSH thesaurus[14], the NCI Thesaurus[15] [Sioutos et al., 2007] and the NCBI Taxonomy[16] (amongst others) play a role in our work.[17] The general strength of terminologies is the provision of natural lan-

[12] It should be noticed that the strict grouping is for description purposes only. In reality, there are no strict divisions between the groups, especially not between the group of information models and ontologies in a stricter sense [see, e.g., Strömbäck et al., 2007].

[13] `http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html` – access date 2012-02-24.

[14] `http://www.nlm.nih.gov/mesh/` – access date 2012-03-03.

[15] `http://ncit.nci.nih.gov/` – access date 2012-12-04.

[16] `http://www.ncbi.nlm.nih.gov/Taxonomy/` – access date 2012-03-02.

[17] Parts of MeSH have been used as input resource for the Protein alignment (chapter 9). The anatomy part of the NCI Thesaurus is one of the input resources for the Anatomy alignment that is analyzed in chapter 10.

guage annotations. A weakness is their missing formal rigor. If concepts are available, they are constituted by groups of synonymous or related terms, which are possibly provided with natural language-based definitions, but no formal descriptions. If concepts are hierarchically organized, the hierarchy usually relies on other relations than the strict 'is-a' relation (e.g., 'broader-than'/'narrower-than'). Furthermore, if additional semantic relationships are specified, they are based on relations expressed in terms of informal natural language predicates.

So-called biomedical ontologies that belong to the group of information models have in common that they serve as organizing structure for domain-specific data. For example, they facilitate the grouping of similar information or data within or across knowledge resources. A usually abstract information model may be mapped to several concrete data models, the latter representing an implementation of the first. Prominent types of data models are object models, entity relationship models and XML schemata. Besides data models that underly individual biomedical databases, several bioinformatics-specific XML standards have been developed that are intended to link data across knowledge resources [Strömbäck et al., 2007]. An example is the BioPAX format[18]. It has been designed as a standard language for the integration, exchange, visualization and analysis of biological pathway data across different databases, whose original formats are not necessarily compatible. Some information models are implemented in formal ontology languages (e.g., BioPAX has been implemented in OWL), illustrating their close relationship to ontologies. However, in contrast to ontologies the focus of most information models is on the organization of data and knowledge in existing resources, rather than the representation of domain knowledge itself. Hence, they are generally less granular than terminologies and ontologies and tend to be application-dependent.

Biomedical ontologies in a stricter sense comprise artifacts that are expressed in a formal ontology language and aim at the explicit specification of the meaning of their contents in terms of axioms. This kind of artifacts came up only recently. One of the first representatives has been the FMA (see page 18). However, their number is continuously increasing. New biomedical ontologies are created, but also artifacts that belong to the groups of terminologies and information models are formalized by converting them into a formal ontology language and adding formal class definitions. For example, GO was converted to OWL [Wroe et al., 2003] and efforts have been made to add new formal class definitions to GO, SO, CL and PRO (see page 18), where the definitions link classes within but also across these ontologies [Meehan et al., 2011; Mungall et al., 2011a,b; Natale et al., 2011]. Naturally, the conversion of weakly formalized knowledge resources into a formal ontology language, such as OWL, requires the thorough adaption of contents. For example, the conversion of underspecified and ambiguous conceptual relationships, which are stated on the class level, into OWL re-

Furthermore, identifiers from the NCBI Taxonomy were used for organism disambiguation in the context of the creation of the PROTEIN alignment (see page 143).

[18] `http://www.biopax.org/` – access date 2012-12-18.

quires a differentiated reflection on the nature of the relationships in order to clarify into which kind of instance level relationships they should be converted (see page 23).

The OBO library and the NCBO BioPortal are two umbrella systems that provide access to most available biomedical ontologies (see page 17). Today, the OBO library records more than 100 biomedical ontologies dealing with complementary subdomains of biology and biomedicine. Each of the OBO ontologies is either directly available in OWL, or an automatically generated OWL version is provided. The NCBO BioPortal, in turn, records more than 300 biomedical ontologies, including the OBO ontologies. Biomedical ontologies published at individual websites may be retrieved using a Semantic Web search engine, such as Swoogle[19] [Ding et al., 2004]. From the perspective of our work, important OBO ontologies are GO, SO, CL, PRO, ChEBI and the FMA (see page 18).[20]

Although an increasing number of biomedical ontologies is specified in a formal ontology language and recently efforts have been made to increase the expressiveness of biomedical ontologies, most of them are still rather lightweight. For example, at present even comparatively expressive biomedical ontologies, such as CL, hardly contain any 'disjoint-with' relationships between classes (see table 11.1, page 165). This is in line with the fact that biomedical ontologies are primarily used as controlled vocabularies for the consistent annotation of factual data across databases. A prominent example is the use of labels of GO classes for the functional annotation of genes and gene products in different model organism and protein databases [Ashburner et al., 2000]. Furthermore, SO is used for the annotation of biological sequence data [Eilbeck et al., 2005] and CL for the annotation of phenotypic and gene expression data with cell type information [Bard et al., 2005]. Standardized meta-data makes databases interoperable and facilitates cross-database data integration, querying and inferencing, even on levels that are not inherently computationally accessible, such as gene product functions. For the annotation of databases the controlled vocabulary aspect of ontologies is important, while advanced features, such as a high expressiveness or the possibility for automatic reasoning, are less deciding. The use of biomedical ontologies in applications that pose stronger demands on ontological background knowledge and that would profit, e.g., from the specification of formal class definitions and disjointness relationships between classes that enable particularly effective reasoning on domain knowledge are just emerging.

[19] `http://swoogle.umbc.edu/` – access date 2012-05-31.

[20] These ontologies have been used as comparison standard for the assessment of the expressiveness of the domain ontologies GRO and MaHCO (see section 11.4).

Chapter 4

An Approach to Building Ontological Background Knowledge

In this chapter, an approach to building ontological background knowledge for biomedicine is presented that consists of a five-staged life cycle model for biomedical ontologies and ontology alignments. Generally, the life cycle of an artifact denotes a structure subdividing its development process into stages through which it passes between planning and disposal. Each life cycle stage refers to particular developmental subtasks and is typically associated with principles and guidelines assisting developers in coping with these tasks. When developing an artifact the choice of an appropriate life cycle model is an important factor for success.

Biomedicine as application domain of ontologies and ontology alignments poses specific challenges. A life cycle model for biomedical ontologies and ontology alignments should be able to cope with these challenges. Since existing life cycle models are mostly domain-independent (section 3.1), below the requirements of biomedicine as application domain are analyzed and a life cycle model for ontologies and ontology alignments is framed that is intended to meet them.

4.1 Biomedicine as Application Domain

There are several reasons why biomedicine is a particularly challenging application domain of ontologies and ontology alignments. First of all, it is one of the few areas in which ontologies are effectively used on a large scale in practice. Since different applications impose rather heterogeneous requirements on ontological background knowledge, it is important to carry out a thorough requirements analysis before a new resource is developed. The compilation of requirements regarding a broad range of aspects can help initiate and guide the development process. In addition, it allows to check existing resources against the stated requirements in order to avoid redundant work.

Biomedical knowledge is increasingly scattered over various knowledge resources. Amongst others, the number of databases and ontologies for biomedicine is continuously growing (see figure 1.2 and section 3.8). To foster the consolidation and standardization of domain knowledge and at the same time avoid further redundancy and fragmentation, it is important to start the acquisition of knowledge for a new ontology by a thorough analysis of existing resources and reuse knowledge as much as possible.

Biomedicine involves many context dependent facts. For example, many relationships between molecular entities (such as interaction between proteins) hold under some conditions but not under others. Accordingly, when conceptualizing a subdomain of biomedicine, it is important to draw a clear distinction between “stable” knowledge that fits in an ontology and constitutes an appropriate basis for deductive reasoning and context dependent facts, which rather should populate a fact database.

Biomedicine is a highly multidisciplinary field that lies at the junction of cell biology, molecular biology, biochemistry, medicine and pharmaceutical sciences, amongst others. The multidisciplinarity promises an exhaustive view on domain-specific problems. However, it also entails an increased risk of terminological and ontological problems. Different communities often use different terms to denote the same concepts, the same terms to denote different concepts, view the same entities from different facets or conceptualize a domain differently. A way to promote a shared understanding of a domain is the implementation of expressive ontologies. In an expressive ontology the intended meaning of classes and relations is made explicit in terms of axioms, which help to preclude semantic ambiguities.

Biomedical ontologies are increasingly used in real-world applications. They are made publicly available, shared and interlinked. For this purpose their validity and reusability is vital. Thus, it is important to thoroughly evaluate ontologies before they are released. Besides checking the logical consistency and the correctness of represented domain knowledge, the guideline compliance of ontologies should be checked to assure that best practices are adhered to.

Biomedicine is a data and fact intensive domain. A way to cope with the increasing amounts of data and facts is to share, reuse and integrate them appropriately, using ontologies as common organization structures. To enhance the visibility and accessibility of biomedical ontologies it is important to publicly release them. Furthermore, a thorough documentation is important to prevent misconceptions when biomedical ontologies are reused, possibly for different applications than they were developed for.

Finally, biomedicine is a very dynamic domain in which continuously new knowledge is generated and existing knowledge may change. To keep biomedical ontologies up to date and effective with respect to associated applications, it is important that they are continuously maintained.

4.2 Life Cycle Model

Based on the above-mentioned considerations, in the context of this thesis a five-staged life cycle model for biomedical ontologies has been developed. It consists of the four developmental stages “requirements analysis”, “design and implementation”, “evaluation” and “documentation and release”, and a maintenance stage (figure 4.1). The stages may briefly be described as follows:

1. In the requirements analysis stage requirements for purpose and scope of the ontology are derived and existing resources are checked whether they already fulfill the requirements. If this is not the case, the development is continued.
2. In the design and implementation stage contents are compiled (knowledge acquisition step), structured into classes, relationships between classes and class instances (conceptualization step) and the resulting conceptualization is expressed in a formal ontology language (implementation step).
3. In the evaluation stage the quality of the ontology is assessed.
4. In the documentation and release stage the ontology is documented and made publicly available.
5. In the cyclic maintenance stage the ontology is corrected, improved and extended to adapt it to changes in the underlying domain, cross-linked resources and applications. A maintenance cycle consists of the steps “collection of change requests”, “ontology update”, “validation” and “documentation and release”.

Each of the five stages has a strong impact on the respective ontology, influencing amongst others its appropriateness for intended use cases, validity, and (re)usability.

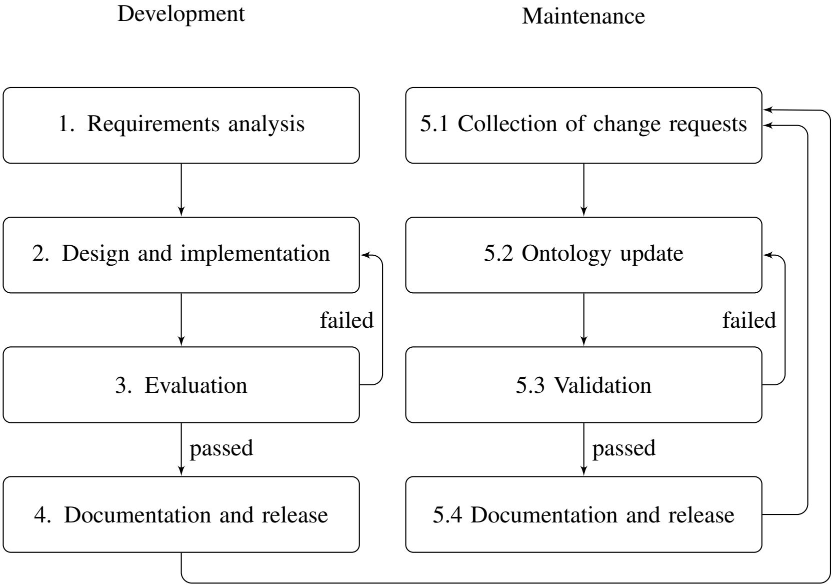

Development
Maintenance
1. Requirements analysis
2. Design and implementation
3. Evaluation
4. Documentation and release
failed
passed
5.1 Collection of change requests
5.2 Ontology update
5.3 Validation
5.4 Documentation and release
failed
passed

Figure 4.1: Five-staged life cycle model for biomedical ontologies.

The proposed life cycle model has primarily been framed for biomedical domain ontologies. However, it is also applicable to top domain ontologies, if it is considered that both, the evaluation and the maintenance of the latter must be carried out with particular care. The reason for this is that any errors in a top domain ontology that are not detected at development time are immediately propagated to attached domain ontologies. Similarly, any changes applied to a top domain ontology during maintenance have an immediate effect on attached domain ontologies. They can even raise inconsistencies. For each new release of a top domain ontology therefore changes should be clearly announced. To enable the “roll back” to a previous version in case of an inconsistency, version control should be taken seriously. Fortunately, basic ontological distinctions do not change and high level domain knowledge that is represented in top domain ontologies is less dynamic than, e.g., highly specific domain knowledge. Accordingly, updates of top domain ontologies are less frequently required than updates of domain ontologies.

Given the need for the ontological representation of some domain, it depends on the kind of ontological background knowledge required and the availability of existing ontologies if the development of a new ontology or the alignment of existing ones is the more appropriate strategy. The following cases occur and corresponding actions should be taken:

1. No appropriate ontology already exists: Create a new ontology from scratch.

2. An ontology exists that is very similar but does not match the required one: Redesign and extend the existing ontology.

3. Two or more ontologies exist each of which covers some parts of the required ontology, but no pair of them covers the latter completely: Create a new ontology reusing elements from the existing ones.

4. Two ontologies exist that together cover the required one: Create an alignment that bridges the two ontologies.

For the fourth case a life cycle model for ontology alignments is required. Certain adaptations are necessary to make the five-staged life cycle model for biomedical ontologies applicable to ontology alignments. Most importantly, the second stage must be rededicated from ontology design and implementation to ontology matching. In the ontology matching stage, correspondences between semantically related elements from different input ontologies are compiled and assembled in an alignment. The remaining stages require only minor modifications. For example, compared to ontologies different aspects need to be considered for the requirements analysis stage. As another example, different approaches need to be applied in the evaluation stage.

Below, the stages of the proposed life cycle model are described in more detail. By means of a small example ontology, dealing with portions of gene regulation (see page 89), it will be illustrated how the stages could be realized in practice. The fully developed example ontology will consist of six classes, linked by five 'is-a' relationships, one 'has-agent' relationship and one 'has-patient' relationship (figure 4.2).

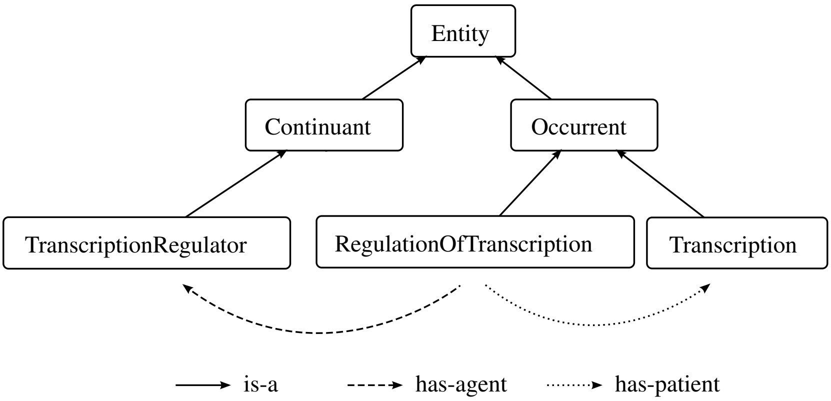


Figure 4.2: Example ontology on transcriptional regulation of gene expression.

Aspect	Description
Ontologies	
Purpose	application or use case of the ontology
Domain	the domain to be represented
Coverage and granularity	extent and detailedness of the domain description
Expressive power and computational demands	available ontology language constructs, facilities for automatic classification and consistency checking
User group	humans (experts or laymen), machines, or both
Tool support	required tools, such as editors, reasoners, etc.
Ontology alignments	
Input	the input resources of the alignment (specify a version)
Target elements	the types of ontology elements to be aligned
Alignment relations	the types of relationships to be considered
Cardinality	the intended cardinality of the alignment
Approach	the matching approach to be used

Table 4.1: Aspects of ontologies and ontology alignments that are to be considered in the requirements analysis stage of their development.

4.3 Requirements Analysis

In the first stage of building an ontology or ontology alignment requirements for different aspects of the respective resource should be compiled (see table 4.1) and put down in written form. For ontologies key aspects to be considered include the intended purpose, the domain to be covered, the coverage and granularity of the domain description, the expressive power, computational demands, the intended user group and the required tool support. Aspects that depend on each other or that are affected by the same parameters should be analyzed together. The purpose of an ontology should be specified first, because it has a strong impact on requirements for most other aspects.

For ontology alignments key aspects to be considered include the input ontologies, the types of ontology elements to be aligned (e.g., classes, relations, or class instances), the types of relations to be considered between these elements (e.g., ‘equivalent-to’ or ‘is-a’), the cardinality of the alignment (see page 25) and the ontology matching approach. For each of the input ontologies, a version number or release date should be specified to avoid ambiguities.

Suppose that a requirement analysis would be carried out for the mentioned example ontology. Possible requirements would then be that the ontology should serve as conceptual backbone for the rule-based extraction of facts on the transcriptional regulation of gene expression, cover basic classes and relations of transcriptional regulation, specify explicit relationships between process classes and classes representing their participants, provide the possibility for logical consistency checking and be useful for domain experts and computers.

After the requirements have been identified it is important to thoroughly examine if an existing knowledge resource already fulfills the stated requirements. Only if this is not the case, the development process should be continued.

4.4 Design and Implementation

The design and implementation stage of ontologies falls into a knowledge acquisition, a conceptualization and an implementation step that may be carried out either manually or automatically using OL techniques (see section 3.3). Most automatic approaches generate potentially large, but little expressive ontologies [Völker et al., 2008]. Hence, for the development of expressive ontologies a manual approach should be preferred, while for the creation of lightweight ontologies automatic approaches may be considered. In the latter case, approaches should be preferred that draw on structured knowledge resources (e.g., databases and thesauri) rather than unstructured ones (e.g., text corpora). The reason for this is that biomedicine is rich in structured knowledge resources and it is usually easier to extract knowledge from structured resources than from unstructured ones.

Below, the design and implementation of ontologies is described, as it is carried out in manual ontology development. Some of the stipulated guidelines have been derived from existing work. They are marked with appropriate citations. Others are based on our own experience with the development of ontologies.

4.4.1 Knowledge Acquisition

In the knowledge acquisition step of ontology development, the contents for a new ontology are compiled. To advance the integration and standardization of domain knowledge, improve the interoperability between knowledge-based systems and avoid the creation of overlapping or even redundant resources, knowledge from existing resources should be reused as far as possible [Noy and McGuinness, 2001; Pease, 2011; Smith, 2006]. Primarily “structured” knowledge resources, such as already existing ontologies, thesauri, controlled vocabularies and databases, should be screened for relevant contents, because they are comparatively easy to access automatically. Top level ontologies are available in terms of BFO, DOLCE and RO, amongst others (see section 2.2.1). Biomedical domain ontologies are available from the OBO library and the NCBO BioPortal and further terminological resources—mostly thesauri and controlled vocabularies with a medical focus—from the UMLS Metathesaurus (see section 3.8). In addition, domain-specific databases are available from the Molecular Biology Database Collection [Baxevanis, 2000]. Also "unstructured" knowledge resources may be searched for relevant contents such as text books or scientific publications. However, it is important to notice that the detection, extraction and integration of contents from unstructured resources usually requires extensive manual work or the use of sophisticated NLP techniques, respectively. As an additional source for domain knowledge, experts may be asked to contribute their expertise.

Suppose that knowledge for building the example ontology would be acquired. Then top level classes would be compiled from the top level ontology BFO (page 15), relations from RO (page 16) and process classes from the biological process branch of GO (page 18), as far as possible. In addition, domain experts would be asked to contribute their domain knowledge to prepare classes that are not yet covered by existing knowledge resources (e.g., 'TranscriptionRegulator').

4.4.2 Conceptualization

In the conceptualization step, based on the previously compiled contents, domain-specific classes are framed, hierarchically organized, linked by conceptual relationships and provided with class instances, if applicable. The previously formulated requirements should serve as guidance. Ideally, the conceptualization is carried out by ontology developers who are domain experts *and* experienced knowledge engineers. Basic ontological distinctions (e.g., between continuants and occurrents, see section 2.2.1) should be adopted from theoretically well founded top level ontologies, such as BFO or DOLCE and basic relation types (e.g., 'part-of' and 'participates-in') from RO, whenever possible. Given synonymous terms, no separate classes should be introduced, but one class with the two terms as preferred and alternative class label [Noy and McGuinness, 2001; Pease, 2011]. Generally, a clear distinction should be made between the 'is-a' and other relation types [Mizoguchi, 2003; Pease, 2011; Smith, 2006]. The class hierarchy of an ontology should exclusively rely on 'is-a' relationships between classes.

To form the class hierarchy of an ontology, "middle-out" approaches have proven effective in practice. In a middle-out approach, mid-level classes are framed first, which are subsequently generalized and specialized appropriately [Uschold and Grüninger, 1996]. The class hierarchy of an ontology must not contain cycles [Noy and McGuinness, 2001]. Furthermore, each class that is not intended as root class should be provided with a superclass to avoid the fragmentation of the ontology. Non-terminal classes should preferably be provided with more than one but not overly many direct subclasses. While a single subclass is usually taken as a hint for missing sibling classes or the immoderate fine-graininess of the ontology, a large number of subclasses is considered as a hint for missing intermediate classes [Noy and McGuinness, 2001].

Suppose that a conceptualization for the example ontology would be framed. Following a middle-out approach first the central classes ‘TranscriptionRegulator’, ‘Transcription’ and ‘RegulationOfTranscription’ would be created, the latter two on the basis of biological process classes of GO, before the top level classes ‘Continuant’, ‘Occurrent’ and ‘Entity’ would be added on the basis of BFO. Subsequently, these classes would be linked by ‘is-a’ relationships: The class ‘TranscriptionRegulator’ that represents a particular type of proteins would be made a subclass of ‘Continuant’ and the process classes ‘Transcription’ and ‘RegulationOfTranscription’ subclasses of ‘Occurrent’. The classes ‘Continuant’ and ‘Occurrent’ themselves would be made subclasses of the topmost class ‘Entity’. Next, the conceptual relations ‘has-agent’ and ‘has-patient’ would be introduced on the basis of RO. A ‘has-agent’ relationship would be established that links the classes ‘RegulationOfTranscription’ and ‘TranscriptionRegulator’ and a ‘has-patient’ relationship that links the classes ‘RegulationOfTranscription’ and ‘Transcription’ (see figure 4.2).

4.4.3 Implementation

In the implementation step, the previously developed conceptualization is implemented in a formal ontology language. The language should be chosen in agreement with the previously stated requirements for the expressive power and computability of the ontology, as well as the needed tool support. An advantage of using a standard language is the possibility to share and reuse software tools [OBO Foundry, 2012, principle FP02]. For example, for the development of OWL ontologies powerful ontology editors (e.g., Protégé-OWL), Java-based APIs for the ontology language (e.g., the Jena framework and the OWL API) and reasoning tools (e.g., HermiT) are available (see section 3.2). They support the GUI and the programming-based implementation of ontologies and their automatic classification and logical consistency checking.

In OWL ontologies classes should be represented using the corresponding built-in construct ‘owl:Class’. Furthermore, ‘is-a’, ‘equivalent-to’ and ‘disjoint-with’ relationships between classes should be expressed using the respective built-in properties ‘rdfs:subClassOf’, ‘owl:equivalentClass’ and ‘owl:disjointWith’. Further semantic relation types should be represented as OWL object (or datatype) properties (see section 2.4 and [Bechhofer et al., 2004]). In addition, each class and relation must be provided with a URI as unique identifier. The namespace to be used and the formatting rules to be applied to local names should be defined by an ontology-wide URI policy.

To make the intended meaning of classes (including conceptual relationships to other classes) computationally accessible formal class definitions should be created [Pease, 2011; Rector et al., 2004]. For top level and top domain ontologies that are intended as source of formal rigor, for less expressive domain ontologies exhaustive formal class definitions are of particular importance. To assure that no logical inconsistencies have been introduced with the new formal class definitions, the logical consistency of ontologies should repeatedly be checked (see section 4.5.1) during the implementation process [Pease, 2011; Rector et al., 2004]. Again, this applies to top level and top domain ontologies in particular, because they are usually provided with exceedingly many, often rather complex formal class definitions.

To prevent unintended side effects of “open world reasoning” (see section 2.4.5), when implementing ontologies in OWL certain design patterns should be adhered to. For example, “closure axioms” should be used (see page 24), trivially satisfiable restrictions avoided by complementing universal with existential restrictions, existential restrictions used as default (instead of universal ones) and the union and intersection of classes strictly distinguished in restriction classes [Horridge, 2011; Rector et al., 2004]. Furthermore, equivalent and disjoint classes should be marked as such by specifying explicit ‘equivalent-to’ and ‘disjoint-with’ relationships. For conceptual relations domain and range, properties such as being transitive or functional and inverse relations should be specified, if applicable. If ontologies comprise class instances, these should mutually be marked as being the same or distinct. Since OWL does not allow to specify formal definitions for conceptual relations, a conceptual relation must be reified (i.e., transformed into a class) before it can be described in more detail.

Ontology classes that are not intended as root classes should either be provided with a direct superclass in the asserted class hierarchy of the ontology, or with a formal definition that lets reasoners automatically compute a superclass in the inferred class hierarchy. Furthermore, the ontology normalization approach by Rector [2003] should be adhered to. It targets at the construction of modular ontologies. First, a class hierarchy is asserted that consists of disjoint trees (the “modules”). In these trees each class has only one superclass and sibling classes are disjoint. Second, restrictions are imposed on classes that implicitly encode further subclass relationships. The latter may cross the limits of modules. Third, a reasoner is run to make these additional subclass relationships explicit. A major advantage of modular ontologies is that they are easier to maintain than complex multihierarchies, because error-prone human maintenance is avoided and the classification task is left to the reasoner [Rector, 2003].

Regarding natural language annotations each ontology class should be provided with a preferred label, a verbal definition and, optionally, alternative labels. The preferred labels and verbal definitions of classes should be intelligible, unambiguous and descriptive, helping human users to capture their intended meanings [Köhler et al., 2006]. Class labels should comply with the terminology used by the prospective user group of an ontology [Smith, 2006] and follow established naming conventions [Noy and McGuinness, 2001; Schober et al., 2009]. The verbal definition of a class should be non-circular [Köhler et al., 2006] and ideally specify the genus and differentiae of the respective class (see page 13) [OBO Foundry, 2012, principle FP06]. Incomplete verbal definitions, such as listings of subclasses or individuals, should strictly be avoided [Pease, 2011; Smith, 2006].

In OWL ontologies natural language annotations are represented using annotation properties (see section 2.4.6). To promote the consistent representation and facilitate the unambiguous automatic access of natural language annotations, strict annotation guidelines should be defined. For each type of annotation they should specify for which type of ontology elements annotations should be provided, whether they are mandatory or optional, which annotation property should be used to represent them, whether special formatting rules apply, and whether a language tag should be provided, amongst others. Generally, an annotation property should be used to represent annotations of one type only and annotations of one type should be represented using the same annotation property throughout the ontology.

To adopt and reuse elements from existing ontologies, either whole ontologies may be imported (for OWL ontologies, the 'owl:imports' construct is provided for this purpose [Bechhofer et al., 2004]) or individual ontology elements may be recreated. Generally, the origin of elements should be made transparent. Elements from imported ontologies are immediately recognizable as such by their URI, which is preserved during the import procedure. The same applies to recreated elements if they are provided with their original URI. However, if recreated elements are provided with new URIs their origin must otherwise be specified, such as by natural language annotations that specify the original URIs or alternative identifiers of the corresponding source elements.

Suppose that the example ontology would be implemented in OWL DL using Protégé. Then its six classes and five 'is-a' relationships between them would be implemented as a hierarchy of OWL ontology classes and the two relations 'has-agent' and 'has-patient' as OWL object properties. To express the 'has-agent' relationship between the classes 'RegulationOfTranscription' and 'TranscriptionRegulator' and the 'has-patient' relationship between the classes 'RegulationOfTranscription' and 'Transcription', the class 'RegulationOfTranscription' would be provided with two anonymous restriction classes as superclasses that specify that each of its instances is linked to at least one instance of the class 'TranscriptionRegulator' by a 'has-agent' relationship and to at least one instance of the class 'Transcription' by a 'has-patient' relationship. The resulting partial definitions of the class 'RegulationOfTranscription' would be specified as

$$\text{`RegulationOfTranscription'} \sqsubseteq \exists\text{`has-agent'}.\text{`TranscriptionRegulator'} \quad \text{and}$$

$$\text{`RegulationOfTranscription'} \sqsubseteq \exists\text{`has-patient'}.\text{`Transcription'}.$$

As URI policy, the use of CamelCase notation for local names and an arbitrary but constant namespace (e.g., "`http://www.semanticweb.org/gro-small.owl#`") would be stipulated. As annotation policy, the provision of classes with natural language labels and verbal definitions would be stipulated, represented as 'rdfs:label' and 'rdfs: comment' annotations, respectively. For example, the class 'TranscriptionRegulator' would be provided with an 'rdfs:label' annotation "transcription regulator" as label and an 'rdfs:comment' annotation "A protein that has transcription regulator activity." as verbal definition.

4.4.4 Ontology Matching

For ontology alignments “design and implementation” means matching the selected input ontologies using the previously chosen matching approach and generating the alignment. The spectrum of available matching approaches ranges from manual over semi-automatic to fully automatic approaches (see section 3.5). For the automatic creation of alignments between biomedical ontologies language-based matching approaches should be considered in particular. There are two major reasons for this. First, in the biomedical domain there is a plurality of publicly available terminological resources that may be utilized, providing domain-specific terms, sets of synonymous terms and even term variants. These resources include the source vocabularies of the UMLS Metathesaurus (see page 41), the UMLS Specialist lexicon[1], biomedical databases in the Molecular Biology Database Collection (see page 4) and biomedical ontologies in the OBO library and the NCBO BioPortal (see page 17), amongst others. Second, most biomedical ontologies and related resources, which constitute the possible input of biomedical alignments, excel in natural language annotations rather than a pronounced structure, formal rigor or the provision of instance data. This makes them an appropriate basis for language-based matching approaches, rather than structural, semantics or instance-based ones.

4.5 Evaluation

The goal of the evaluation stage is to measure the quality of an ontology or ontology alignment. It should be measured both intrinsically and extrinsically, the latter with respect to the usefulness and impact of the respective resource on intended applications. The focus of this thesis is on the intrinsic evaluation of ontologies and ontology alignments. Since both kinds of resources constitute complex artifacts, quality measurements may address different levels and aspects and use different quality criteria [Gangemi et al., 2005b].

Below, selected intrinsic evaluation approaches are described in more detail. For ontologies three approaches are presented. The first one checks their logical consistency (section 4.5.1), the second one the correctness and the coverage of their contents (section 4.5.2) and the third one their compliance with design and implementation guidelines (section 4.5.3). While the logical consistency and correctness and coverage of contents are crucial for the validity and usefulness of ontologies, their compliance with design and implementation guidelines is favorable because guidelines concentrate years of experience in ontology engineering.

[1] http://www.nlm.nih.gov/pubs/factsheets/umlslex.html – access date 2012-12-10.

For ontology alignments two approaches are presented. The first one deals with checking the correctness and coverage of contents of alignments (section 4.5.4). The second one comprises basic quality checks that address both content and technical aspects (e.g., formatting and representation issues) of alignments (section 4.5.5). While the correctness and coverage of contents are important factors for the validity and usefulness of ontology alignments, their correct formatting and representation is vital for their (re)usability.

4.5.1 Logical Consistency of Ontologies

Checking the logical consistency of OWL ontologies—from the perspective of tool users—has become an easy task, because various reasoning tools have been made available (see page 34). They are either invoked via the GUI of an ontology editor (e.g., Protégé), the command line or program code. If a reasoner has detected an inconsistency, the respective ontology should be revised and the reasoner rerun, until no further inconsistencies are detected. Only then the ontology development should be continued. Besides in the evaluation stage of ontology development, logical consistency checks should also be run in the implementation step of the design and implementation stage and the validation step of maintenance cycles.

Suppose that the example ontology would be checked for logical consistency using the ontology editor Protégé. Protégé would be opened, in the menu “Reasoner” a reasoner would be chosen (e.g., “HermiT”, see page 34) and the reasoning process would be started. Upon completion the inferred class hierarchy would be checked for unsatisfiable classes (they would be displayed as subclasses of the predefined class ‘owl:Nothing’ in the “Classes” tab of Protégé). Since in our small example ontology there are no unsatisfiable classes, there would be no need to revise the ontology and invoke the reasoner again.

4.5.2 Contents of Ontologies

A straightforward way to evaluate the contents of an ontology is to compare the latter to an appropriate gold standard ontology. For this purpose, modified (also called “relaxed”) variants of standard precision and recall may be used as evaluation measures that reflect the fact that hierarchically organized structures are compared [Ehrig and Euzenat, 2005].

Two major advantages of gold standard-based evaluations are that they can be run fully automatically and they can easily be repeated. However, ontologies are typically developed for domains in which ontological background knowledge is sparse and hence gold standard ontologies are missing. An alternative way to automatically evaluate the contents of an ontology is to answer formal competency questions [see Grüninger and Fox, 1995]. If neither a gold standard ontology is available, nor formal competency questions have been specified beforehand, the contents of an ontology can be evaluated by comparing them to appropriate knowledge resources, such as databases or text corpora, or by resorting to expert judgment.

Suppose that the contents of the example ontology would be evaluated. Since no gold standard ontology is available to which it could be compared and also no formal competency questions have been specified, a domain expert who was not involved in building the ontology would be asked to judge the correctness and relevance of the classes and relationships contained. Subsequently, rejected classes and relationships would be analyzed and revised, if applicable.

4.5.3 Guideline Compliance of Ontologies

For the evaluation of the guideline compliance of ontologies comprehensive approaches are so far missing. Hence, in the context of this thesis the following three-step procedure has been developed:

Step 1: Answer selected guideline control questions.

Step 2: Check the ontology for modeling mistakes running an OWL pitfall scanner.

Step 3: Check the compliance of the ontology with the OBO Foundry principles.

The first step relies on a set of 30 "guideline control questions" presented in table 4.2. The corresponding guidelines form parts of the previously introduced approach to ontology design and implementation (see section 4.4). As mentioned above, some of them were derived from existing work (details are given in section 4.4), while others are based on our own experience with ontology development. As basis for the control questions such guidelines have been selected that together address a broad spectrum of levels and aspects of ontologies, are generic enough to apply to various different ontologies and in case of a violation have serious effects on the validity or usability of the respective ontology. From the selected set of 30 questions (Q), the questions Q01-Q07 address the class and taxonomy level of ontologies, Q08-Q15 the level of formal semantics, Q16-Q23 the annotation level, Q24-Q27 technical aspects, and Q28-Q30 rather global aspects of ontology reuse and usability (see table 4.2). The questions were implemented in Java, as far as possible, utilizing the OWL API (see page 34). For certain control questions the implementation allows to specify parameters to adapt the question to the respective ontology to be evaluated. For example, for questions

ID	Guideline control question
Class and taxonomy level	
Q01	Are basic ontological distinctions adhered to?
Q02	Is the 'is-a' relation clearly distinguished from other relation types?
Q03	Is the class hierarchy non-cyclic (asserted)?
Q04	Is each class provided with at least one superclass (inferred)?
Q05	Is each class provided with at most one superclass (asserted)?
Q06	Are classes with only one direct subclass avoided (inferred)?
Q07	Are classes with many direct subclasses avoided (inferred)?
Level of formal semantics	
Q08	Is the equivalence of classes with the same meaning explicitly specified?
Q09	Is the disjointness of classes explicitly specified?
Q10	Are domain, range and further properties of relations specified?
Q11	Are formal class definitions provided?
Q12	Are existential restrictions used as default?
Q13	Are existential restrictions complemented with universal restrictions?
Q14	Are universal restrictions complemented with existential restrictions?
Q15	Are class intersections avoided as fillers of universal restrictions?
Annotation level	
Q16	Are compulsory annotations provided (e.g., preferred labels)?
Q17	Are optional annotations provided (e.g., alternative labels)?
Q18	Are annotation type-specific guidelines adhered to?
Q19	Is each annotation property used for one type of annotation only?
Q20	Is each type of annotation represented using one annotation property only?
Q21	Do class labels follow approved naming conventions?
Q22	Are duplicate class labels and duplicate verbal definitions avoided?
Q23	Are verbal definitions non-circular?
"Technical" level	
Q24	Do local names adhere to a consistent naming policy?
Q25	Are redundant ontology elements avoided?
Q26	Are unused object, datatype and annotation properties avoided?
Q27	Are artifacts from tool-use (e.g., empty annotations) avoided?
Knowledge reuse and usability	
Q28	Is the ontology provided with meta data annotations?
Q29	Have ontology elements or annotations been reused?
Q30	Are reused contents up to date and are their sources referenced?

Table 4.2: Guideline control questions for the evaluation of ontologies. The additions "(asserted)" and "(inferred)" specify whether the scope of a control question is the asserted or inferred class hierarchy of the respective ontology.

Q16 (Are compulsory annotations provided?) and Q17 (Are optional annotations provided?) annotation properties can be specified that are used in a particular ontology for the representation of compulsory and optional annotations.

For the second step, the ontology pitfall scanner Oops! may be used [Poveda-Villalon et al., 2012]. According to its authors it is able to automatically detect instances of 21 common modeling pitfalls in OWL ontologies (see table 4.3). The pitfalls represent worst practices or “adverse” guidelines of ontology implementation. Oops! is available as Web application. In order to run it on a particular ontology, the ontology URI needs to be specified or the content of the respective ontology document pasted in the input field at the Oops! website[2].

The third step relies on the OBO Foundry principles (see page 33), which represent best practices for the collaborative development and maintenance of biomedical ontologies developed by the OBO Foundry [Smith et al., 2007]. The significance of these principles lies in the fact that in contrast to ontology development guides with a strict focus on ontology design and implementation, they also address ontology management issues, such as the documentation and maintenance of ontologies. Furthermore, they represent a community effort and hence enjoy a broad acceptance in the bio-ontology community. The principles are available at the OBO Foundry Principles website[3]. A compressed version is presented in table 4.4.

Suppose that the guideline compliance of the example ontology would be checked using the proposed three-step procedure. Then answering the 30 guideline control questions would reveal that the class ‘Continuant’ possibly misses further subclasses, the class pairs ‘Continuant’ and ‘Occurrent’ and ‘Transcription’ and ‘RegulationOfTranscription’ miss explicit disjointness relationships, the conceptual relations ‘has-agent’ and ‘has-patient’ possibly miss domain and range restrictions, all classes possibly miss full formal definitions and alternative labels, the ‘has-agent’ and ‘has-patient’ relationships possibly lack closure axioms, and reused ontology elements (such as the classes ‘Continuant’ and ‘Occurrent’, derived from BFO) miss a reference of their origin.

Running the pitfall scanner Oops! would result in additional complaints about the lack of disjointness relationships in the ontology and missing domain and range restrictions of the relations ‘has-agent’ and ‘has-patient’, as well as new complaints about the lack of inverse relations, labels and verbal definitions of these relations. Checking the adherence to the OBO Foundry principles would reveal that the small example ontology is neither interlinked with OBO Foundry ontologies, nor developed in collaboration with OBO Foundry ontology projects. In addition, it has no real users and lacks a contact person who is also responsible for its maintenance.

[2] http://www.oeg-upm.net/oops – access date 2012-03-21.

[3] http://obofoundry.org/crit.shtml – access date 2012-02-11.

ID	Pitfall
P02	creating synonyms as classes
P03	creating the relation “is”
P04	creating unconnected ontology elements
P05	defining wrong inverse relations
P06	including cycles in the hierarchy
P07	merging different concepts in the same class
P08	missing annotations
P10	missing disjointness
P11	missing domain or range in properties
P12	missing equivalent properties
P13	missing inverse relations
P19	swapping intersection and union
P20	swapping label and comment
P21	using a miscellaneous class
P22	using different naming criteria in the ontology
P24	using recursive definition
P25	defining a relation inverse to itself
P26	defining inverse for symmetric relation
P27	defining wrong equivalent relation
P28	defining symmetric relations with different domain and range
P29	defining transitive relations with different domain and range

Table 4.3: Pitfalls detectable by Oops!—with original numbering [Poveda-Villalon et al., 2012].

ID	OBO Foundry principle (compressed)
FP01	The ontology must be open and available to be used by all.
FP02	The ontology is (or can be expressed) in a common shared syntax.
FP03	Each class and relation in the ontology must have a unique URI.
FP04	Ontology versions must be clearly distinguished and metadata should be provided for changes.
FP05	The ontology has a clearly specified and delineated content, provides coherent verbal definitions of top level classes and incorporates explicit links to other OBO Foundry ontologies.
FP06	The ontology includes verbal definitions for classes.
FP07	The ontology uses relations in the style of the OBO Relation Ontology.
FP08	The ontology is well-documented (e.g., in a published paper or manual).
FP09	The ontology has a plurality of mutually independent users.
FP10	The ontology is developed collaboratively with other OBO Foundry members.
FP11	There should be a contact person, who is also responsible for maintenance.
FP12	The ontology adheres to the OBO naming conventions.
FP16	The ontology should be continuously maintained.

Table 4.4: OBO Foundry principles with status “accepted” [OBO Foundry, 2012].

While obviously the third step of the three-step procedure would essentially disclose the nature of the evaluated resource as an artificial example ontology, the first two steps would result in valuable hints for its further development and improvement.

4.5.4 Contents of Ontology Alignments

In order to evaluate the contents of an ontology alignment, domain experts may be asked to check the correctness and coverage of the correspondences that it consists of. However, if an appropriate gold standard alignment is available (also called "reference alignment"), the evaluation may be carried out automatically by comparing the correspondences of the alignment to that of the reference alignment.

As mentioned above, "hierarchy-aware" precision and recall measures are particularly suitable for this purpose. Ehrig and Euzenat [2005] define "overlap proximity" as basis for such measures. Given an alignment A between the ontologies O and O', a corresponding gold standard alignment G and a proximity function σ, overlap proximity ω is defined as

$$\omega(A, G) := \sum_{(a,g)\in M(A,G)} \sigma(a, g), \tag{4.1}$$

where M denotes a matching between correspondences a of A and g of G [see Ehrig and Euzenat, 2005]. Overlap proximity measures the proximity of the sets of correspondences of A and G, rather than their strict overlap as standard precision and recall metrics do. Based on this overlap proximity, relaxed precision P_ω and recall R_ω are defined as

$$P_\omega(A, G) := \frac{\omega(A, G)}{|A|} \text{ and } R_\omega(A, G) := \frac{\omega(A, G)}{|G|} . \tag{4.2}$$

Relaxed precision and recall values depend on the fact which proximity function σ is chosen. Let "equ" denote the 'equivalent-to' relation and "is" the *direct* 'is-a' relation. Then a simple example of a proximity function is

$$\sigma(a, g) := \begin{cases} 1 & \text{if } \ \mathrm{equ}(c_a, c_g) \land \mathrm{equ}(c'_a, c'_g) \\ 0.5 & \text{if } \ (\mathrm{equ}(c_a, c_g) \land (\mathrm{is}(c'_a, c'_g) \lor \mathrm{is}(c'_g, c'_a))) \lor \\ & \quad\ (\mathrm{equ}(c'_a, c'_g) \land (\mathrm{is}(c_a, c_g) \lor \mathrm{is}(c_g, c_a))) \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

where $a = (c_a, c'_a)$ and $g = (c_g, c'_g)$ denote correspondences contained in A and G as before, c_a and c_g elements of O and c'_a and c'_g elements of O'. Given this proximity function, correspondences contained in A that are also contained in G score 1 (as in case of standard precision and recall). Correspondences in A that link an element from one ontology to a slightly more general or more specific element in the other ontology, compared to the one that is the correct match according to G, score 0.5. All remaining correspondences of A score 0. Hence, in contrast to standard precision and recall, this proximity function pays tribute to the fact that slightly "imprecise" correspondences are still of some value, in contrast to complete mismatches or missing correspondences. This example and further proximity functions are described in detail in Ehrig and Euzenat [2005].

Using these relaxed precision and recall measures, F-scores are computed as

$$F_\beta := \frac{(1+\beta^2)P_\omega R_\omega}{\beta^2 P_\omega + R_\omega}, \ \beta \in \mathbb{R}^+, \tag{4.4}$$

where the parameter β determines whether an emphasis is put on recall or precision or neither of the two. The value of β should be chosen dependent on the intended application of the alignment. If the application is more seriously affected by missing correspondences than by incorrect ones, $\beta > 1$ should be chosen to emphasize recall. If the contrary applies, $\beta < 1$ should be chosen to emphasize precision. If missing and incorrect correspondences have a similarly serious impact, $\beta = 1$ should be chosen that balances recall and precision.

4.5.5 Validity and Reusability of Ontology Alignments

Approaches to evaluate the validity and technical consistency of ontology alignments were hitherto scarce. Hence, in the context of this thesis basic quality checks have been compiled for this purpose [Beisswanger and Hahn, 2012]. They are presented in figure 4.3 and described in more detail below.

Checks 1–5 focus on the (re)usability of an alignment as reference for the evaluation of matching systems. Checks 1 and 2a test whether the correspondences contained in an alignment can be found at all by a matching system depending on the available release versions of the input ontologies. (Imagine that a matching system is provided with a more recent version of an input ontology than it was used for the creation of the reference alignment. If in the more recent version several classes have been deleted, any correspondences in the reference alignment referencing one of the deleted classes cannot be reproduced anymore by the system.) Check 2b tests for class label changes. It is targeted at the tacit evolution of the meaning of classes. In particular for light-weight ontologies, lacking thorough formal class definitions, verbal labels virtually carry the entire meaning of a class and, hence, a new label might indicate a subtle or even severe change of the meaning of an ontology class requiring further scrutiny. If check 1 is positive, check 2 can be skipped. Check 3 addresses the accessibility of an alignment, while check 4 tests whether the references to classes are unique. (If, for example, local names or class labels are given as class references, the references are potentially

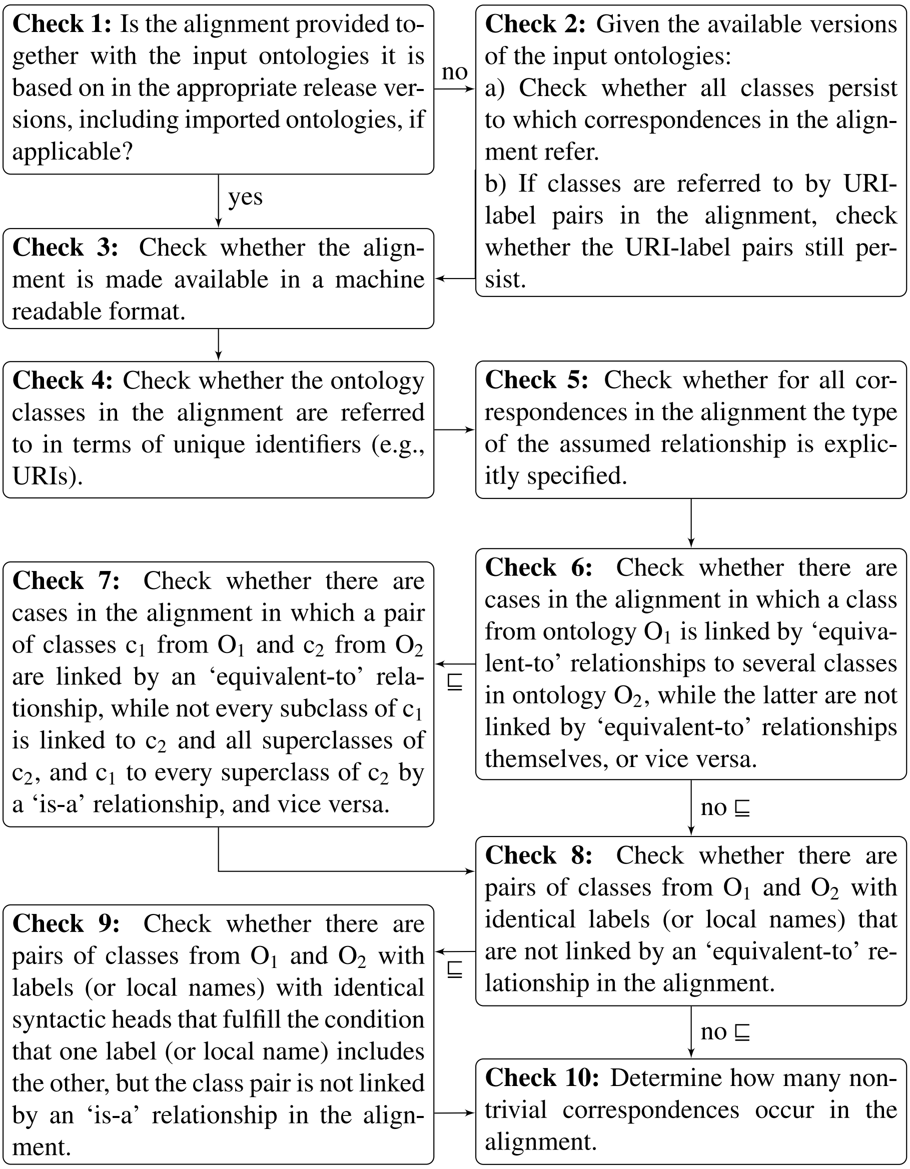

Check 1: Is the alignment provided together with the input ontologies it is based on in the appropriate release versions, including imported ontologies, if applicable?
no
Check 2: Given the available versions of the input ontologies:
a) Check whether all classes persist to which correspondences in the alignment refer.
b) If classes are referred to by URI-label pairs in the alignment, check whether the URI-label pairs still persist.
yes
Check 3: Check whether the alignment is made available in a machine readable format.
Check 4: Check whether the ontology classes in the alignment are referred to in terms of unique identifiers (e.g., URIs).
Check 5: Check whether for all correspondences in the alignment the type of the assumed relationship is explicitly specified.
Check 6: Check whether there are cases in the alignment in which a class from ontology O_1 is linked by 'equivalent-to' relationships to several classes in ontology O_2, while the latter are not linked by 'equivalent-to' relationships themselves, or vice versa.
$\sqsubseteq$
Check 7: Check whether there are cases in the alignment in which a pair of classes c_1 from O_1 and c_2 from O_2 are linked by an 'equivalent-to' relationship, while not every subclass of c_1 is linked to c_2 and all superclasses of c_2, and c_1 to every superclass of c_2 by a 'is-a' relationship, and vice versa.
no $\sqsubseteq$
Check 8: Check whether there are pairs of classes from O_1 and O_2 with identical labels (or local names) that are not linked by an 'equivalent-to' relationship in the alignment.
$\sqsubseteq$
Check 9: Check whether there are pairs of classes from O_1 and O_2 with labels (or local names) with identical syntactic heads that fulfill the condition that one label (or local name) includes the other, but the class pair is not linked by an 'is-a' relationship in the alignment.
no $\sqsubseteq$
Check 10: Determine how many non-trivial correspondences occur in the alignment.

Figure 4.3: Basic quality checks for ontology alignments and the proposed order of execution. O_1 and O_2 denote the input ontologies of the alignment. For alignments supposed to incorporate 'is-a'-based correspondences after check 6 and check 8 the arrows marked with "$\sqsubseteq$" should be followed, otherwise those with "no $\sqsubseteq$".

ambiguous.) Check 5 is meant to test whether explicit semantic relation types were specified by the alignment creators for the relationships they asserted between pairs of classes.

Checks 6–9 address the completeness and check 10 the non-triviality of the alignment. Checks 6 and 7 address the *structural* level utilizing the class hierarchy of the input ontologies to find evidence for possibly missing or erroneous correspondences. Since in an alignment a class from one ontology should be mapped to at most one class in the other ontology by an ‘equivalent-to’ relationship (or, if it links to several classes, these should be linked by ‘equivalent-to’ relationships themselves), check 6 may provide hints for redundant or even mistaken correspondences in an alignment, but also for implicit class equivalences in the input ontologies. Check 7 utilizes the fact that stating an ‘equivalent-to’ relationship between a pair of classes logically entails ‘is-a’ relationships between all subclasses of one class and all superclasses of the other. The check may thus help to identify missing ‘is-a’-based correspondences, but it may also provide hints for erroneous ‘equivalent-to’-based correspondences in the alignment or modeling errors in the input ontologies. Checks 8 and 9 address the same concern, but target the *language* level instead, utilizing class labels. They reflect the observation that when two ontologies are aligned that show a strong conceptual overlap, label identity between classes provides strong evidence for an ‘equivalent-to’ relationship between them, whereas labels with identical syntactic heads that fulfill the condition that one label includes the other are a strong indicator for an ‘is-a’ relationship. Examples include the label pairs “blood cell” and “cell”, and “membrane of cell” and “membrane” (with rightmost and leftmost heads, respectively). Both checks may help in detecting missing correspondences in an existing alignment. Checks 7 and 9 may be skipped if regarding the alignment under scrutiny ‘is-a’-based correspondences are out of scope.

Finally, check 10 allows for a stricter evaluation of the capabilities of an ontology matching system by distinguishing between relaxed and tight test conditions. In the relaxed mode, the determination of lots of trivial correspondences may overestimate the true potential of a matching system, simply because exact (sub)string matching is entirely sufficient for finding trivial correspondences. “Trivial” correspondences in this context are either ‘equivalent-to’-based correspondences that can be detected via class label (or local name) identity, or ‘is-a’-based correspondences that can be detected via mere syntactic head analysis of class labels (or local names) after applying a simple term normalization procedure. In the strict mode, however, only non-trivial correspondences are taken into account rendering evidence for the true sophistication of the alignment finding procedure. Certainly, a large proportion of trivial correspondences in an alignment (an indication of strongly overlapping input ontologies) decreases its value as reference alignment, although trivial correspondences do play a certain role as anchors for advanced matching strategies [see, e.g., Jiménez-Ruiz and Grau, 2011].

4.6 Documentation and Release

Ontologies and ontology alignments should be documented in terms of research papers or technical reports [OBO Foundry, 2012, principle FP08] that describe their purpose, structure and contents. This type of documentation is intended to help potential new users to assess if the ontology or ontology alignment is appropriate for their application or use case and prevent misconceptions. Ontologies should additionally be documented by means of meta data annotations in the header of the ontology document (in OWL ontologies called "ontology annotations"). Ontology annotations may carry the title of an ontology, a brief description of contents, links to external knowledge resources from which knowledge has been derived, the names of ontology creators and license information, amongst others. These annotations are intended to support the retrieval of ontologies from specialized repositories or from the Web using search engines such as Swoogle (see page 42). For the representation of meta data annotations in OWL ontologies, annotation properties from the Dublin Core Metadata Element Set[4] (e.g., 'dc: title' or 'dc:subject') or the OWL vocabulary[5] (e.g., 'owl:versionInfo') may be used.

Ontologies and ontology alignments should be made publicly available [OBO Foundry, 2012, principle FP01]. There are at least three good reasons for this. First, the public release of knowledge resources enables knowledge sharing and reuse. Only a publicly released knowledge resource can be picked up, reused and extended by external users, saving duplicate work and avoiding the creation of redundant resources. Second, the public release and reuse of knowledge resources ensure their continuity even beyond project run times and funding periods. A publicly released ontology may be reused and extended, even after the original authors have left the ontology project or the project has terminated. Third, the more often knowledge resources are used and in the more different contexts, the more feedback can be expected, including error reports or suggestions for improvements. User feedback is an important basis for the maintenance and improvement of knowledge resources [Pease, 2011].

A biomedical ontology or ontology alignment may be publicly released by putting it on a freely accessible website or submitting it to a special repository (e.g., the OBO library or the NCBO BioPortal, see page 17). The second variant comes with the advantage that it may increase the visibility of the knowledge resource for a certain community. In addition, it allows the use of the infrastructure of the repository. For example, the NCBO BioPortal allows to search, browse and compare ontologies, and it provides access to different ontology versions, amongst others [Noy et al., 2009]. The NCBO BioPortal also accepts ontology alignments in the form of mappings between classes of BioPortal ontologies. For the release of ontology alignments it is of great importance to include the original input ontologies in the release [Beisswanger and Hahn, 2012],

[4] `http://dublincore.org/documents/dces/` – access date 2012-12-10.

[5] `http://www.w3.org/TR/owl-ref/` – access date 2012-12-10.

especially if they are intended as reference standard for evaluations.

Suppose that the development of the example ontology has been completed and the next step would be its documentation and release. Apart from the limited meaningfulness of such a step for a didactic example ontology, a technical report, conference or journal paper would have to be written to document the contents, structure and use cases of the ontology. In addition, meta data annotations would be added to the corresponding ontology document. Suitable annotations would be a ‘dc:title’ annotation “Transcriptional Regulation Ontology”, a ‘dc:description’ annotation “This is an example ontology that is intended for didactic purposes only.” and a ‘owl:versionInfo’ annotation “version 1.0”. To make the ontology publicly available, it would possibly be submitted to the NCBO BioPortal, using the “Submit New Ontology” dialog on the corresponding website (see page 17).

4.7 Maintenance

Ontologies and ontology alignments should continuously be maintained to correct them, if required, and keep them up to date. For ontologies the maintenance procedure should comprise at least the steps “collection of change requests”, “ontology update”, “ontology validation” and “documentation and release”, in analogy to the four developmental stages of the proposed ontology life cycle (see figure 4.1). These steps should consecutively be executed and then cyclically repeated. The first maintenance cycle should be started directly after the first release of the ontology.

The step “collection of change requests” deals with the compilation of change requests from ontology users. Reasons for change requests are manifold. They include the detection of mistakes in the ontology, the emergence of new domain knowledge that should be reflected and the changing of application requirements. To administer change requests and changes, an issue tracking system may be used. An example is Trac[6], as known from software engineering.

The step “ontology update” involves the decision about the acceptance or rejection of change requests and the implementation of accepted changes. The person(s) entrusted with the task should possess appropriate domain expertise and knowledge engineering skills. Updates may be performed manually using an ontology editor or via programming using appropriate APIs.

The step “ontology validation” deals with the reconfirmation of the quality of the updated ontology. For ontology validation some of the analyses carried out during the

[6] `http://trac.edgewall.org/` – access date 2012-02-16.

evaluation stage of ontology development should be repeated. Especially the logical consistency (see section 4.5.1) and guideline compliance (see sections 4.5.3) of the updated ontology should be checked again. Additionally, automatic persistence checks should be run to approve that no contents were lost during the update of the ontology (e.g., that no classes or relations have mistakenly been deleted) and a manual inspection should be carried our to assure the empirical plausibility of the updated ontology. In case that a check failed, the ontology needs to be corrected and validated again, until all checks have been passed successfully.

The step "documentation and release" deals with the documentation of the updated ontology and its release. The new ontology version should be assigned a distinct version number and release date in terms of meta data annotations in the header of the ontology document. These specifications enable a clear distinction between the new ontology version and previous versions, which is an important factor for the reproducibility of ontology-based work. In addition, release notes documenting the changes executed on the ontology should be compiled and published together with the new ontology version. After the release of the new ontology version, the next maintenance cycle should be started. Tool support for tracking changes and controlling different ontology versions is available in terms of general version control systems and dedicated ontology maintenance software (see page 34).

For the maintenance of ontology alignments basically the same four-step procedure is applicable as for ontologies. However, it must additionally be considered that any changes in the input ontologies of the alignment potentially require the adaption of the alignment [Euzenat et al., 2008, section 5.3]. The co-evolution of ontologies and alignments is an intricate task, for which tool support is virtually missing, so far. However, there are tools that allow to compare different versions of the same ontology and create an alignment of them (e.g., the Prompt plug-in for Protégé [Noy and Musen, 2003]). The latter may be useful to lift an ontology alignment from out-dated to more recent versions of its input ontologies [see Euzenat et al., 2008, figure 6.4].

Suppose that the task would be to make the example ontology subject to maintenance. Then the ontology would first be put under version control using, e.g., Subversion (page 34). Next, the first maintenance cycle would be started by compiling change requests and error reports and checking the domain represented in the ontology as well as ontology applications for changes (in this example case, of course, both are fictive activities). Remember that the example ontology was intended as background knowledge for rule-based fact extraction. Now suppose that a new requirement of the fact extraction system would be to extract no longer only facts on *transcriptional* regulation of gene expression, but now also on *translational* regulation (for the biological background, see figure 6.1).

To support also this new task, the example ontology would need to be extended. At least three new classes (*viz.*, 'Translation', 'RegulationOfTranslation' and 'Transla-

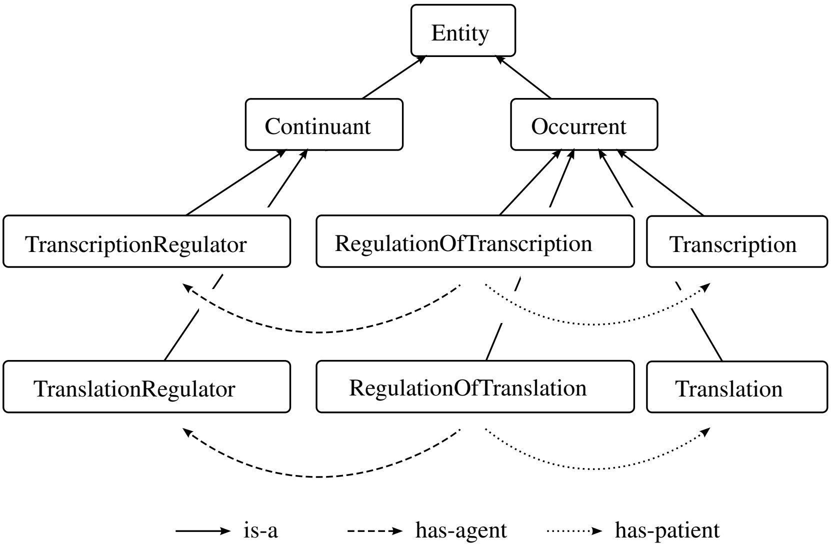


Figure 4.4: Extended example ontology. In addition to transcriptional regulation it addresses the translational regulation of gene expression.

tionRegulator'), three new 'is-a' relationships (*viz.*, between the classes 'TranslationRegulator' and 'Continuant', 'Translation' and 'Occurrent', and 'RegulationOfTranslation' and 'Occurrent'), one 'has-agent' relationship (*viz.*, between the classes 'RegulationOfTranslation' and 'TranslationRegulator') and one 'has-patient' relationship (*viz.*, between the classes 'RegulationOfTranslation' and 'Translation') would have to be added. Given that the required changes would have been accepted, they would be implemented using Protégé. Thereafter, the example ontology would look like the one depicted in figure 4.4. The logical consistency of the updated ontology would be assured using an OWL reasoner. In this example case, there would be no unsatisfiable classes that would need to be fixed. Finally, the new version of the ontology would be released, together with release notes documenting the changes made to the ontology (in this case the inclusion of three new classes and five new relationships). Then a new maintenance cycle would be started. The classes and relations contained in the example ontology and the extended version thereof are summarized in appendix B.

CHAPTER 5

Discussion of the Approach

In this chapter, the previously proposed approach to building ontological background knowledge for biomedicine is discussed. If necessary, cross references to part III of this thesis are provided that deals with the practical implementation of this approach.

5.1 Life Cycle Model

The approach to building ontological background knowledge for biomedicine proposed in chapter 4 of this thesis is based on a five-staged life cycle model for biomedical ontologies and ontology alignments (see section 4.2). A special feature of this life cycle model is that its stages have been selected in response to the specific characteristics of biomedicine as field of application, described in section 4.1. Stevens et al. [2000] and Noy et al. [2010] also address the life cycle of biomedical ontologies. However, Stevens et al. [2000] present a general life cycle model, which is not tailored to biomedicine as field of application, and Noy et al. [2010] do not present an explicit life cycle model at all, but address the topic from a tooling perspective.

A second distinguishing feature of the model proposed in this thesis is that it is rather comprehensive. Considering the first four stages only, it corresponds to the life cycle models for ontologies proposed by Uschold and King [1995] and Stevens et al. [2000].[1] However, in contrast to the models by Uschold and King [1995] and Stevens et al. [2000], the model proposed in this thesis contains an additional maintenance stage. As argued in section 4.1, for biomedical ontologies this stage is of particular importance, because it allows to adapt them to the frequently changing domain knowledge and application requirements. An existing life cycle model that *does* contain a maintenance stage is the "evolving prototype" model by Fernandez et al. [1997]. However, the authors do not further specify how maintenance (and the remaining stages of their model) should be realized in practice, beyond the statement that the whole ontology life cycle is target of the support activities knowledge acquisition, documentation and evaluation. In contrast, the model presented in this thesis comes with an explicit specification of subtasks and activities associated to each life cycle stage, facilitating its implementation in practice.

A third distinguishing feature of the proposed life cycle model is that it comes with a variant for ontology alignments. Besides one model by Euzenat et al. [2008], this variant seems to be the first life cycle model for ontology alignments that has been proposed so far. Both models include a matching stage, an evaluation stage and a documentation and release stage (called "communication stage" in the model by Euzenat et al. [2008]). However, the model proposed in this thesis is far more extensive. In contrast to the model by Euzenat et al. [2008] it does not only comprise an additional requirements analysis stage at the beginning of the life cycle, but also a maintenance stage at the end. As argued in section 4.1, for biomedical ontology alignments (as for biomedical ontologies) the requirements analysis stage is important to successfully cope with the heterogeneous requirements that applications pose on ontological background knowledge. Furthermore, the continuous maintenance of ontology alignments is important to adapt them to both, changing domain and application requirements and changes in the respective input ontologies.

Last but not least, the fact that the life cycle model proposed in this thesis has been tested in five different case studies makes it positively stand out from most existing models ("Although there exist ontology development methodologies there is still a lack of mature and well-tested life cycle models." [Blomqvist, 2009, page 41]).

[1] The stage "requirements analysis" corresponds to the stages "identify purpose" [Uschold and King, 1995] and "identify purpose and scope" [Stevens et al., 2000]. The stage "design and implementation" corresponds to the stage "ontology building" (incorporating ontology capture, coding and the integration of existing ontologies) [Uschold and King, 1995] and the stages "knowledge acquisition", "conceptualization", "integration" and "encoding" [Stevens et al., 2000]. All three models share the stages "evaluation" and "documentation".

5.2 Requirements Analysis

To start the requirements analysis of ontologies with the purpose aspect (as proposed in section 4.3) is in line with the recommendation by Stevens et al. [2000] to start the development of ontologies by identifying their purpose and scope. The proposal to analyze requirements with respect to purpose, domain, coverage, granularity, expressive power and intended user group of ontologies, as made in section 4.3, is related to the approaches by Uschold and Grüninger [1996] and Noy and McGuinness [2001]. [Uschold and Grüninger, 1996] propose to derive motivating scenarios from applications and formulate competency questions based on them that represent requirements for the coverage and expressiveness of the future ontology. Noy and McGuinness [2001], in turn, propose to answer several basic questions, such as "For what we are going to use the ontology?" [Noy and McGuinness, 2001, page 5] and "Who will use and maintain the ontology?" [Noy and McGuinness, 2001, page 5], and compile a list of key notions of the chosen domain that later might become labels of ontology classes, relations or individuals.

While the requirements analysis approach that has been proposed for ontologies in the context of this thesis is related to different existing approaches, the approach proposed for ontology alignments is probably the first of its kind in this explicit form.

5.3 Design and Implementation

For the design and implementation of ontologies a guideline-based approach has been proposed in section 4.4. From the practical case studies presented in part III of this thesis there is evidence that guidelines effectively stimulate the creation of expressive, well annotated, and consistently formatted ontologies. The retrospective evaluations of GRO (see section 6.3), MaHCO (see section 7.3) and BioTop (see section 8.3) confirmed that overall they comply well with the guidelines according to which they were developed: They adhere to basic ontological distinctions, the meaning of ontology classes is expressed in terms of axioms (to a varying extent, though), classes are provided with natural language annotations, and knowledge from existing ontologies and other terminological resources has been reused. Furthermore, the evaluations revealed that in GRO and MaHCO, for which strict annotation guidelines have been specified, natural language annotations are represented more consistently than in BioTop, for which explicit annotation guidelines were lacking at that time. Finally, the statistics on early versions of GRO (see table 6.4) illustrate that the successive extension and tightening of ontology design and implementation guidelines imposed on GRO led to an increasingly expressive and well annotated ontology. While the first pre-release of GRO lacked conceptual relations, 'disjoint-with' relationships between classes, formal class definitions and natural language annotations, these missing pieces of information were gradually added in subsequent ontology releases, in response to the tightening of guidelines.

A distinctive feature of the ontology implementation approach proposed in this thesis is the promotion of strict annotation guidelines. While existing ontology development guides point at the importance of particular types of annotation, such as verbal definitions [OBO Foundry, 2012, principle FP06], and propose guidelines for class labels [Noy and McGuinness, 2001; Schober et al., 2009], the annotation guidelines promoted in the context of this thesis go further. They clearly identify mandatory and optional annotations for different types of ontology elements and give precise instructions for their representation, amongst others. The relevance of annotation guidelines lies in the fact that in most practical applications of ontologies the availability of proper natural language annotations is crucial. They are not only a means to communicate the intended meaning of ontology elements to human users, but they also enable machines to automatically access ontology elements and find correspondences between them. For example, many concept recognition and ontology matching systems depend on natural language labels of ontology classes. The first ones use them for the detection of verbal class mentions in natural language documents [see, e.g., Aronson, 2001; Jonquet et al., 2009]. The second ones require them for the detection of relationships between classes of different ontologies [see, e.g., Cruz et al., 2009; Jain et al., 2010; Jiménez-Ruiz and Grau, 2011; Mascardi et al., 2009]. Obviously, the lack of class labels or their limited automatic accessibility, due to their inconsistent representation, would lower the performance of both types of systems. In contrast, strict annotation guidelines, as promoted in the context of this thesis, foster the creation of unambiguous, automatically processable annotations. Eventually, they can help to increase the usefulness of ontologies for practical applications.

The guidelines-based approach to ontology design and implementation, as it is proposed in section 4.4, is closely related to ODP-based ontology development (see section 3.4). Virtually each of the ontology design and implementation guidelines proposed in this thesis could be expressed in terms of an ODP. Guidelines for which a pattern already exists include the guideline to create closure axioms for existential restriction and the guideline to develop ontologies adhering to the ontology normalization approach by Rector [2003].[2]

The proposal to construct expressive ontologies manually and to consider for the development of large, rather lightweight ontologies automatic approaches, made in section 4.4, reflects the prevailing view that manual ontology development is inherently precision oriented, whereas automatic OL approaches tend to deliver large, but rather

[2] The corresponding patterns are available in the "Ontology Design Patterns Public Catalog" (see `http://odps.sourceforge.net/odp/html/Closure.html`) and the Semantic Web portal "OntologyDesignPatterns.org" (see `http://ontologydesignpatterns.org/wiki/Submissions:Normalization`), respectively. Both websites were accessed on 2012-12-20.

imprecise and inexpressive ontologies. Two statements pinpointing this view on OL are: “It is inherent in the ontology learning process that the acquired ontologies represent uncertain and possibly contradicting knowledge.” [Haase and Völker, 2008, page 366] and “The current state-of-the-art in lexical ontology learning is able to generate ontologies that are largely informal or lightweight ontologies in the sense that they are limited in their expressiveness and often only consist of concepts organized in a hierarchy.” [Völker et al., 2008, page 2]. However, it must be acknowledged that the field of OL is rapidly evolving. Meanwhile, techniques have been proposed to derive improved, logically consistent ontologies from automatically generated, possibly inconsistent ones [Haase and Völker, 2008]. In addition, first approaches for learning expressive ontologies have been proposed [Völker, 2009; Völker et al., 2008], as well as semi-automatic OL systems to assist rather than replace the ontology developer [Fortuna et al., 2007; Wächter and Schroeder, 2010]. Certainly, for some development tasks manual approaches are still the most appropriate ones. An example is the creation of top level and top domain ontologies, for which profound domain knowledge and knowledge engineering skills are required. However, given further progress in the field of OL, it can be expected that ontology engineers will increasingly profit from automatic support. Semi-automatic OL systems could accelerate the manual ontology development process by proposing new classes, class labels and verbal definitions. Furthermore, approaches to learn expressive ontologies could support ontology developers in creating formal class definitions.

A typical example for an automatically generated biomedical ontology is MaHCO HLA, an extension of the MaHCO ontology introduced in chapter 7. It is lightweight, relies on knowledge extracted from domain-specific nomenclatures and databases and has been created without using classical text-based OL. Another typical example is the Cell Cycle Ontology (CCO) [Antezana et al., 2009]. This external ontology has been created and is continuously rebuilt fully automatically, also without using text-based OL. CCO relies on various biomedical ontologies and domain-specific databases, from which knowledge has been extracted and integrated using Semantic Web technologies. A major advantage of generating ontologies based on structured instead of unstructured resources is the saving of efforts caused by the easier accessibility of structured compared to unstructured data. Another advantage is the possibility to interlink the resulting ontology with the knowledge resources it depends on, enlarging the amount of domain knowledge that is automatically accessible and interlinked, and hence may be used in knowledge-based applications.

An important aspect of the development of biomedical ontologies is the reuse of knowledge from existing ontologies and other structured and unstructured knowledge resources, as emphasized in section 4.4.1. The recommendation to develop ontologies by reusing contents from existing knowledge resources, as far as possible, coincides with existing ontology development guides that promote knowledge reuse as best practice [see, e.g., Noy and McGuinness, 2001; Pease, 2011; Rector et al., 2006b; Smith, 2006]. However, in the context of this thesis it became apparent that besides the reuse of knowledge itself, also the way how it is realized in practice is of great importance.

Two strategies to reference the origin of reused ontology elements have been tested in practice in the context of developing GRO, MaHCO and BioTop. The first strategy is the preservation of the original URIs of reused classes and relations. When one ontology imports another one (such as MaHCO core imports the MaHCO HLA extension, see page 114), it is the default. An advantage of this strategy is that ontology elements do not have to be duplicated in the newly developed ontology. A disadvantage is that a dependency on external knowledge resources is introduced, which can have serious implications. Given that one ontology imports another one from the Web, then changing the imported ontology may cause logical inconsistencies in the importing ontology. Furthermore, if the imported ontology is moved or deleted, the import fails completely. The second strategy that was tested is the recreation of reused classes and relations, in conjunction with the creation of reference annotations that cite their origin (see the 'gro:reference' and 'mhc:reference' annotations introduced for GRO and MaHCO on pages 96 and 116). It does not suffer from the mentioned dependency on external ontologies. However, in addition to the duplication of ontology elements, a disadvantage of this second strategy is that the origin of reused ontology elements is specified in a rather informal manner, which is hard to access automatically. This hampers, for example, Semantic Web applications as they are increasingly widespread in the field of biomedicine. Using this referencing strategy, it is not possible to specify which relationship exists between a referenced ontology element and the recreated variant of it (it may, but need not be an 'equivalent-to' relationship, because the variant may have been revised). Furthermore, in practice it turned out that the informal nature of this strategy results in many missing and improperly formatted reference annotations. This prevents that the benefits of knowledge reuse can take effect. To sum up, both mentioned strategies come with advantages and disadvantages. However, for future work the first approach should be preferred for its effectiveness, transparency and compatibility with the Semantic Web approach.

For the creation of alignments between biomedical ontologies the proposal to focus on string and language-based matching approaches has been made (see page 56). Evidence for the effectiveness of language-based approaches for biomedical matching tasks has been collected in the case study on the Protein alignment (see chapter 9). The matching approach that was used to create the Protein alignment relies on the comparison of natural language labels of the entries of its input resources and associated terms derived from cross-linked databases. In a gold standard-based evaluation it outperformed an also language-based baseline approach (see table 9.3). The advantage of the proposed matching approach compared to the baseline approach resulted primarily from the incorporation of the terms harvested from cross-linked databases. In this exemplary case, parts of a database and of a thesaurus have been matched instead of two ontologies. However, given that most available biomedical ontologies are rather lightweight (see section 3.8) and tend to provide a rich set of annotations, setting the focus on string and language-based matching approaches instead of structure or semantics-based ones can be supposed to be similarly appropriate for biomedical ontologies.

5.4 Evaluation of Ontologies

In the application-focused literature on ontology evaluation it is a prevailing view that the successful extrinsic evaluation of ontologies is most important, because it guarantees their successful application. For example, Obrst et al. [2007] state that "The ultimate evaluation of an ontology is in terms of its adoption and successful use, rather than its consistency or coverage." [Obrst et al., 2007, page 153]. They argue that for example GO is extremely successful, although it is still impoverished in many representational aspects. Along the same lines, Noy and McGuinness [2001] conclude their guide to ontology development "we can assess the quality of our ontology only by using it in applications for which we designed it" [Noy and McGuinness, 2001, page 23]. In fact, ontologies are only useful when they are accepted by intended users and perform well in respective applications. However, in spite of the current focus on extrinsic ontology evaluation in practice, previous work has shown that intrinsic evaluations are still required in addition to extrinsic ones, in order to ensure that ontology-based work is valid and reliable. For example, the intrinsic evaluation of GO by Smith et al. [2003] helped to correct and substantially improve GO, at a time at which it was already accepted and successfully used. For this reason, several intrinsic evaluation approaches for ontologies and ontology alignments have been included in the approach to building ontological background knowledge for biomedicine, as proposed in this thesis (see section 4.5).

An important form of intrinsic evaluation is checking the logical consistency of ontologies, as proposed in section 4.5.1. For the evaluation of OWL ontologies it has become a standard approach. The reason for this is that reasoning tools have been made available as standalone tools or plug-ins for established ontology editors that allow ontology developers to classify their ontologies and check them for logical consistency without requiring programming skills. Using the high-performance reasoner HermiT (see page 34) meanwhile even ontologies that are large or contain complex formal class definitions (such as the top domain ontology BioTop, see chapter 8) can be classified without difficulty. The inclusion of logical consistency checks in a life cycle model for ontologies is crucial, because of the serious implications that undiscovered logical inconsistencies would have on ontology-based work.

Another important form of intrinsic evaluation is checking the correctness and the coverage of contents of ontologies. As an alternative to approaches based on gold standard ontologies or formal competency questions, which are usually rare, the assessment of contents by domain experts has been proposed in section 4.5.2. Expert judgment-based approaches are known to be laborious and innately suffer from missing reproducibility. However, due to the domain experts involved, they are generally expected to deliver precise results. The practical test of an expert-based approach in the case study on GRO revealed that it is effective in practice. It led to the detection of 'is-a' relationships between classes, formal class definitions and class labels that needed corrections (see section 6.3). However, the practical test also revealed that certain requirements must be fulfilled so that the expert involved—who is usually not an ontology engineer— makes correct decisions. A major requirement turned out to be the provision of a precise and complete description of the evaluation task. As a negative example, in the evaluation of GRO, the task description to check ontology classes for their relevance for the field of gene regulation turned out to be too vague. Based on this task description, the domain expert who carried out the evaluation rejected the top level classes of GRO with the argument that they are not directly relevant for the domain. However, they *are* relevant for the hierarchical organization of domain knowledge and thus should not be rejected. Another major requirement turned out to be the provision of precise labels and verbal definitions of ontology classes. In the practical test, incomplete and imprecise class labels led to wrong expert decisions, because they allowed for unintended interpretations of classes. The practical test further indicated that for domain experts formal class definitions do not compensate inadequate natural language annotations, and verbal definitions and scope descriptions of the ontology do not compensate inadequate class labels. To sum up, the practical test confirmed the effectiveness of expert judgment-based ontology evaluation and helped to identify requirements that must be met in order to run such evaluations successfully.

A third important form of intrinsic evaluation is checking the guideline compliance of ontologies. For this purpose a three-step procedure has been proposed in section 4.5.3. Its first step is based on 30 guideline control questions (see table 4.2). The practical test of these questions by applying them to GRO (see section 6.3), MaHCO (see section 7.3) and BioTop (see section 8.3) has shown that they are an effective means to improve the quality of ontologies with respect to various levels and aspects. To some extent, this guideline control question-based approach is similar to the approach by Zhang and Bodenreider [2006]. They present 15 ontology modeling principles and check if the FMA (see page 18) is compliant with them. Some of the 15 principles match guidelines targeted by the guideline control questions. For example, the principle H_1 by Zhang and Bodenreider [2006], stating that no 'is-a' hierarchical cycles are allowed, matches the guideline control question Q03 (Is the asserted class hierarchy non-cyclic?). However, the scope of the proposed guideline control questions is much broader than that of the 15 principles by Zhang and Bodenreider [2006]. For example, the guideline control questions address the annotation level of ontologies and the aspect of knowledge reuse, of which neither is addressed by the 15 principles. Furthermore, the guideline control questions are domain-independent and hence applicable to a broad spectrum of ontologies, while some of the 15 principles are not. For example, the principle D_1 states "'Concept Subdivision of x' (or 'Organ component of x') does not exist unless concept 'x' exists." [Zhang and Bodenreider, 2006, page 682]. Due to its focus on anatomy it is suitable for the evaluation of anatomy ontologies only. An additional advantage of the guideline control question-based approach is that it has been implemented in a configurable way (see page 58). Hence, the implementation can be customized according to characteristics of particular input ontologies. The benefit of a configurable implementation can be illustrated by the following example. Given an ontology in which class labels are represented using the annotation property ‘skos:prefLabel’ instead of ‘rdfs:label’, which is standard. Asked to check the ontology for class labels, a non-configurable tool (such as the pitfall scanner Oops!, see page 60) would vainly search for ‘rdfs:label’ annotations and report that no class labels are available. In contrast, the configurable implementation of the guideline control questions would be provided with the URI ‘skos:prefLabel’ as parameter value, which would enable it to correctly identify and report the class labels. To sum up, the proposed guideline control questions-based evaluation approach is the first one that checks the guideline compliance of ontologies in this broad scope. Due to its domain-independence it is widely applicable. In addition, due to its implementation in a configurable way it is flexible enough to cope with specific characteristics of particular input ontologies.

The second step of the proposed procedure to check the guideline compliance of ontologies relies on the automatic detection of common modeling mistakes in ontologies using available software. In the context of this thesis, the pitfall scanning tool Oops! (see page 60) has been tested in practice by running it on GRO, MaHCO and BioTop. Oops! was able to detect different types of modeling mistakes in all three ontologies (see table D.2), which indicates that it is effective in practice. However, the practical test also revealed limitations of the tool. It turned out that in some cases it behaves less strict than appropriate and overly strict in others. The authors of Oops! are aware of this fact and state that the tool detects “potential” pitfalls [Poveda-Villalon et al., 2012]. For example, it handles each class with a local name that contains the conjunction “and” or “or” as instance of pitfall P07 (merging different concepts in the same class). It is not able to make an exception for classes with local names such as “ComplexOfProtein-AndDNA”, which contain a conjunction, but still denote a single concept (linguistically a syntactic ambiguity or scope resolution problem). In addition, local names, but no class labels are checked for conjunctions. As a consequence, this pitfall check cannot deliver results for ontologies with numerical local names that provide class labels in terms of annotations. As another example, the tool handles classes and conceptual and data relations that lack an ‘rdfs:label’ or ‘rdfs:comment’ annotation as instance of pitfall P08 (missing annotations). It is not able to identify labels or verbal definitions that are represented by means of alternative annotation properties, such as ‘skos:prefLabel’ and ‘skos:definition’. A particularly serious limitation of the tool is that it accepts an ontology as *not* being an instance of pitfall P10 (missing disjointness) if it contains at least one explicitly stated ‘disjoint-with’ relationship between classes, irrespective of the fact that many additional explicit ‘disjoint-with’ relationships might still be missing. To sum up, the practical test of the automatic pitfall scanner Oops! confirmed its effectiveness. However, it also revealed that the tool would strongly benefit from algorithmic extensions on the one hand (e.g., the inclusion of an approach to automatically detect missing explicit ‘disjoint-with’ relationships [see, e.g., Meilicke et al., 2008]) and a redesign that makes it configurable and hence able to reflect individual ontology design and implementation guidelines.

The third step of the proposed procedure to check the guideline compliance of ontologies relies on the OBO Foundry principles (see page 60). The use of the OBO Foundry principles outside their original context is so far uncommon. Usually, OBO Foundry custodians watch over the adherence of OBO Foundry member ontologies to the principles and work with developers of OBO Foundry candidate ontologies to ensure that their ontologies conform to the principles before they become new member ontologies. Since the principles have not been compiled for the external use, the applicability of some of them outside the OBO Foundry is limited. An example is the principle FP10, which requires that an ontology is developed in close collaboration with other OBO Foundry member ontologies [OBO Foundry, 2012, principle FP10]. This circumstance must be considered when the results of evaluations based on these principles are interpreted. However, despite this constraint the third step of the proposed evaluation procedure has proven to be effective in practice. Checking GRO, MaHCO and BioTop for their adherence to the OBO Foundry principles yielded valuable suggestions for improvement of the three ontologies (see table D.3). Two special features of the OBO Foundry principles make them worthwhile for the evaluation of ontologies. First, in contrast to ordinary ontology development guidelines, they address ontology management issues, such as ontology versioning, documentation and maintenance. Second, they have the direct support of the biomedical ontology community, from the center of which they arose. Both facts distinguish the third step of the proposed evaluation procedure from previously existing evaluation approaches for ontologies.

To sum up, the proposed three-step procedure for checking the guideline compliance of ontologies is novel in the breadth and granularity of guidelines and best practice principles that are considered. In addition, according to three different case studies it is effective in practice. The mutual comparison of the three steps of the proposed procedure revealed that the 30 guideline control questions, the pitfall catalog and the OBO Foundry principles largely complement each other. However, there *are* certain overlaps. For example, question Q03 asks if the asserted class hierarchy is non-cyclic and pitfall P06 deals with cycles in the class hierarchy. As another example, question Q21 asks if class labels follow approved naming conventions and the OBO Foundry principle FP12 stipulates that ontologies should adhere to the OBO naming conventions. Accordingly, the proposed three-step procedure could further be streamlined and the mentioned overlaps resolved by merging the three steps of the procedure into one, taking the converse of each pitfall as guideline.

The importance of automatic guideline checks for ontologies is increasingly acknowledged. For example, de Coronado et al. [2009] state in an article on the quality assurance of the NCI Thesaurus “even when editorial guidelines are well documented, editors do not always apply those guidelines systematically” [de Coronado et al., 2009, page 537]. As another example, Vrandečić concludes his doctoral thesis on ontology evaluation with the statement that the most useful evaluation paradigm to improve the quality of ontologies and increase their benefit for applications would be to check if ontologies are defective, and if so, in which way [Vrandečić, 2010, page 197f]. The need for guideline checking procedures that is embodied in these exemplary statements underlines the relevance of our three-step approach for checking the guideline compliance of ontologies.

5.5 Evaluation of Ontology Alignments

To evaluate automatically generated alignments (or the systems that generated them, respectively) by comparing them to appropriate reference alignments, as proposed in section 4.5.4, has become a standard approach. However, the use of relaxed precision and recall measures [see Ehrig and Euzenat, 2005] and precision or recall oriented F-scores for this purpose, as further proposed, is so far rather uncommon. In the 2011 edition of the international OAEI campaigns on ontology alignment evaluation relaxed precision and recall measures were only used in the Benchmark track and precision or recall-oriented F-scores only in the Conference track [Euzenat et al., 2011b]. Hence, the proposed evaluation procedure promotes the supplementation of a standard evaluation approach by sophisticated evaluation measures that are tailored to the underlying evaluation task.

To evaluate the validity and (re)usability of ontology reference alignments, ten basic quality checks have been proposed in section 4.5.5. They basically search for evidence of missing and erroneous correspondences in alignments by utilizing basic structural and linguistic features of the alignments and their input ontologies (check 6–10) and low technical aspects of ontology alignments (check 1–5). The application of these checks to three different sample datasets (see chapter 10) confirmed that they are effective in practice. The results of this practical test showed that already such very basic checks are quite effective and can help increase the quality of alignments. Check 1–5 helped increase the (re)usability of the evaluated alignments. Check 6–10, in turn, helped improve their validity by identifying incorrect correspondences that should be removed and missing correspondences that should be added. In addition, the checks revealed shortcomings in the input ontologies of the evaluated alignments, such as missing or invalid relationships between classes.

Since manually created reference alignments are used as ground truth in evaluations of automatic ontology matching systems, the quality of the alignments themselves is of topmost importance for the credibility of the evaluations based on them. However, the manual creation of ontology alignments is known to be challenging and inherently error-prone ("humans are not usually very good at matching ontologies manually" [Euzenat and Shvaiko, 2007, page 202]). Thus, approaches to scrutinize the quality of manually created ontology alignments are required. The quality checks for ontology alignments that are proposed in this thesis supplement the few existing approaches (see section 3.7) in a significant way. They assess the quality of ontology alignments with respect to aspects that are not in the focus of existing approaches, but with regard to which deficiencies have shown to strongly affect the validity and reusability of alignments in practice (see section 10.2). Examples for such aspects include the format and representation of alignments. Compiling the checks, aspects of the quality of alignments that have already been addressed elsewhere [see, e.g., Joslyn et al., 2009; Meilicke and Stuckenschmidt, 2009; Meilicke et al., 2009] have deliberately been excluded. Accordingly, the quality checks presented in this thesis are by no means intended to be exhaustive for checking the quality of ontology alignments. Instead, they have been designed as a simple, yet effective first step of an intended multi-step procedure of extensively checking the quality of an ontology alignment before it is used as reference standard in evaluations. Subsequent steps should include a check of the logical consistency of alignments [see, e.g., Meilicke and Stuckenschmidt, 2009; Meilicke et al., 2009], an analysis of the preservation of structural properties of the respective input ontologies of alignments [Joslyn et al., 2009] and extrinsic, i.e., application-based evaluations.

5.6 Documentation and Release

The proposal to document ontologies and ontology alignments in terms of research papers or technical reports, made in section 4.6, is in line with existing ontology development guides, such as the OBO Foundry principles [OBO Foundry, 2012, principle FP08]. The same applies to the recommendation to publicly release ontologies and ontology alignments [OBO Foundry, 2012, principle FP01]. In contrast, the proposal to provide ontologies with meta-data annotations as a complementary form of documentation, also made in section 4.6, has so far been neglected by most existing ontology development guides. The major reason why it deserves attention is that it supports the automatic retrieval of resources from the Web. In brief, for the documentation and release of ontologies and ontology alignments two standard approaches have been proposed, in the context of this thesis, and a complementary, less widespread approach for the documentation of ontologies.

5.7 Maintenance

For the maintenance of ontologies and ontology alignments, in section 4.7 of this thesis, a cyclic procedure for maintenance has been proposed. For each of the four steps it contains associated subtasks and activities have been specified. Compared to most existing life cycle models, this constitutes a rather elaborate approach to maintenance. Existing life cycle models for ontologies either mention maintenance as a stage, but fail to give specifications for its implementation in practice [see, e.g., Fernandez et al., 1997] or they do not contain a maintenance stage at all [see, e.g., Stevens et al., 2000; Uschold and King, 1995]. The latter also applies to the one existing life cycle model for ontology alignments [Euzenat et al., 2008, section 3.2]. If the need arises to extend the proposed maintenance approach, in the closely related field of ontology evolution more complicated approaches are available that could be considered as prototype [see, e.g., Stojanovic, 2004]. Hence, with regard to maintenance the life cycle model proposed in the context of this thesis exceeds most existing life cycle models for ontologies. For ontology alignments it seems to be the first one that covers maintenance at all.

5.8 Summary

To sum up, the approach to building ontological background knowledge for biomedicine, presented in chapter 4 of this thesis, positively sticks out from existing approaches in multiple respects. The first distinctive feature of the approach is that the stages of the underlying life cycle model for ontologies and ontology alignments have been compiled with respect to biomedicine as application domain. This enables the model to cope with the characteristics of this particularly important, but also particularly challenging application domain. With regard to ontology alignments the proposed approach constitutes one of the first life cycle models that have been proposed at all.

The second distinctive feature is that it covers ontology development and ontology matching as two complementary strategies of building ontological background knowledge. In biomedicine the virtue of ontology development is accepted at least since the rise of GO. Furthermore, the acceptance of ontology matching is growing with the number of biomedical ontologies available. However, the combination of the two strategies in a single approach to building ontological background knowledge is novel and makes the approach more flexible than existing ones.

Third, the proposed approach excels in being comprehensive. In contrast to most existing approaches to ontology development, which focus on the design and implementation of ontologies only, and most existing approaches to ontology matching, which mainly focus on the actual matching step, it covers the whole life cycle of ontologies and ontology alignments. This breadth is advantageous to the effect that important tasks beyond the mere construction of ontologies and ontology alignments are addressed, such as their evaluation and maintenance. Both are crucial for assuring the validity, up-to-dateness and reusability of ontologies and ontology alignments. In addition, the proposed approach describes the life cycle stages that it covers in a detailedness as it is known from practical guides to ontology development. However, the latter are usually restricted to ontology design and implementation. An advantage of this detailedness is that in contrast to most existing life cycle models for ontologies and ontology alignment, which are rather abstract, it facilitates the realization of the proposed approach in practice.

Fourth, a particular strength of the proposed approach is that it incorporates comprehensive evaluation approaches for ontologies and ontology alignments. By focusing on the guideline compliance of ontologies and basic aspects of validity and reusability of ontology alignments, these evaluation approaches have the potential to substantially enhance the quality of ontologies and ontology alignments on various levels and regarding various aspects.

As a general remark, given the need for ontological background knowledge, an alternative strategy to newly developing it would be searching the Web or a specialized repository for candidate resources and appropriately rank the retrieved candidates. Though ontology searching and ranking lies outside the scope of this thesis, it could be useful in the requirements analysis stage of the proposed life cycle model to detect the knowledge resources to be checked against the specified requirements (see section 4.3).

The approach to building ontological background knowledge for biomedicine presented in this thesis has been described using OWL DL as reference formalism. However, large parts of the approach do not depend on a specific ontology language. An example is the set of basic quality checks for ontology reference alignments, proposed in section 4.5.5. Although the checks have been implemented for OWL ontologies, the basic idea behind each check is independent from the ontology language being used.

Part III

Case Studies

CHAPTER 6

The Gene Regulation Ontology

The Gene Regulation Ontology (GRO) is an ontology about gene regulation. It describes basic processes of gene regulation, their participants, and the relationships between them. An early version of GRO has been described in Beisswanger et al. [2008a]. The focus of the current chapter is on the life cycle of GRO, including the evaluation and maintenance of the latter.

6.1 Requirements Analysis

GRO has been developed in the context of the BOOTStrep project[1]. The general goal of the project was the creation of biomedical resources and resource-building text analysis tools. A three-layered biomedical knowledge repository should be created, centered on gene regulation of the bacterial model organism *Escherichia coli* (*E.coli*). It should consist of a bio-lexicon, a bio-ontology and a bio-fact store, the latter integrating facts that have been automatically extracted from the biomedical literature, existing factual databases, ontologies, and related terminological resources. As core of the bio-ontology and conceptual backbone of the bio-lexicon, the bio-fact store and automatic fact extraction approaches a comprehensive formal knowledge resource on gene regulation was required. In particular, it should serve as a well-defined vocabulary for the semantic annotation of gene regulatory processes in text documents and

[1] `http://www.bootstrep.org/` – access date 2012-02-20.

Aspect	Description of requirements
Purpose	vocabulary for the semantic annotation of text corpora, conceptual basis for rule-based fact extraction, interface to related terminological resources
Domain	regulation of gene expression (with a focus on *E.coli*)
Coverage and granularity	gene regulatory processes and their participants, elementary classes and relations
Expressive power and computational demands	formal class definitions, automatic classification and consistency checking
User group	domain experts and machines
Tool support	ontology editor, reasoner

Table 6.1: Results of the requirements analysis of GRO.

basis of domain-specific inference rules for automatic fact extraction. The resource should cover the basic categories of gene regulatory processes and their participants (i.e., physical entities, such as genes, regulatory regions of genes or proteins, or other processes). It should *not* provide fine-grained hierarchies of classes representing certain aspects of gene regulation in an overly detailed manner. Furthermore, it should express explicitly how processes and their participants relate to each other. The knowledge resource should be represented in a formal, machine-processable format. Classes should be provided with both, natural language labels and definitions that facilitate the work of human users (e.g., annotators of biomedical documents), and formal definitions that facilitate automatic classification and consistency checking. The results of the requirements analysis of GRO are summarized in table 6.1.

Below, the field of gene regulation is introduced, before existing knowledge resource on gene regulation and neighboring fields are checked if they already satisfy the stated requirements.

Regulation of Gene Expression

Living cells store their hereditary information in the form of deoxyribonucleic acid (DNA), consisting of a sequence of nucleotides of four different types [Alberts et al., 2002, page 4]. A gene is a DNA segment that encodes the construction plan of a gene product. A gene product is either a protein, or a ribonucleic acid (RNA) molecule with a catalytic or structural function [Alberts et al., 2002, page 9]. The process of synthesizing gene products using the information encoded in genes is called gene expression. The expression of protein coding genes falls into the two major steps transcription and translation [Alberts et al., 2002, page 6, figure 1.4].

During transcription, the enzyme RNA polymerase transcribes the gene into an RNA sequence, called messenger RNA (mRNA) [Alberts et al., 2002, page 303-304]. Special sequences of nucleotides in the DNA, called promoter and terminator regions, signal the enzyme where to start and stop the transcription [Alberts et al., 2002, page 306]. While in bacteria, the detachable σ-factor subunit of the RNA polymerase is needed to initiate transcription [Alberts et al., 2002, page 306], in eukaryotic cells additional proteins are required for this purpose, called general transcription factors (TFs) [Alberts et al., 2002, page 310]. Amongst others, they help to position the polymerase enzyme correctly at the promoter.

During translation, the nucleotide sequence of a gene is translated into the amino acid sequence of a protein, i.e., mRNAs are used as template for the synthesis of proteins [Alberts et al., 2002, page 336]. The rules determining which triplets of nucleotides ("codons") code for which amino acids are referred to as the genetic code [Alberts et al., 2002, page 336]. Translation depends on transfer RNAs (tRNAs) as adapter molecules that can bind an amino acid at the one end and a codon at the other one [Alberts et al., 2002, page 337]. The amino acid sequence is synthesized in ribosomes, complex structures consisting of ribosomal RNAs (rRNAs) and proteins [Alberts et al., 2002, page 342]. Most proteins pass through different maturation steps before they become functional in the cell, e.g., they are folded into a three dimensional structure, assembled with further protein subunits to a protein complex, or modified by the attachment of a functional group.

There are various cellular mechanisms that control the amounts of gene products that are synthesized in a cell. They are collectively referred to as "regulation of gene expression", or briefly "gene regulation". Regulatory processes occur on all steps of gene expression [Alberts et al., 2002, page 379], see figure 6.1. The most common target of control mechanisms is the initiation of transcription. Gene regulatory proteins with specific binding domains bind to regulatory regions, i.e., specific binding sites in the DNA [Alberts et al., 2002, page 383], acting as activator or repressor of gene expression. Genes transcribed from the same promoter ("operons") are subject to common control mechanisms [Alberts et al., 2002, page 395]. The importance of gene regulation lies in the fact that it enables cells to control their structure and function and adapt to different intra- and extracellular conditions. If gene regulation is disordered, this may cause disease. A prominent example is cancer.

Related Knowledge Resources

Knowledge resources on gene regulation and neighboring fields include biomedical ontologies, data models of domain-specific databases and data exchange formats. First,

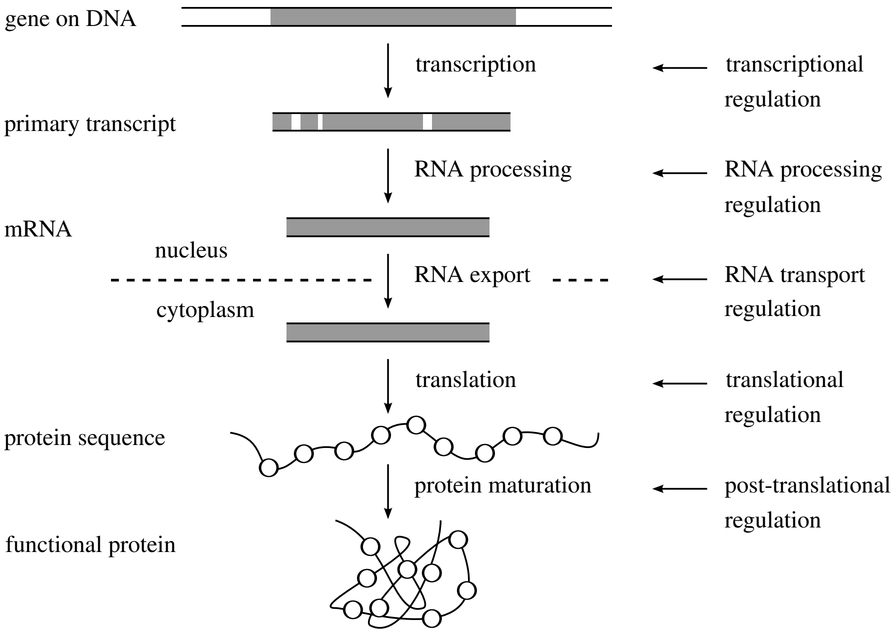


Figure 6.1: Levels of gene expression and the regulation of gene expression. This figure was inspired by the Wikimedia Commons file “Gene_expression_control.png” [Wikimedia Commons, 2013].

the domain-specific databases EcoCyc[2], RegulonDB[3] and Transfac[4] were examined. EcoCyc is a comprehensive, manually curated database for the model organism *E. coli*, strain K-12 MG1655. It provides functional annotations for gene products of *E. coli*, information on the regulation of gene products at the transcriptional, post-transcriptional and protein level, and on their organization into operons, complexes and pathways. RegulonDB is another database on gene regulation in *E. coli*. It provides curated information on the organization of genes in transcription units, operons and regulons, and on the complex regulation of transcription initiation and regulatory networks. Transfac is a database on eukaryotic gene regulation. It provides data on eukaryotic TFs, TF binding sites, regulated genes and regulatory DNA regions. While the EcoCyc data is stored in a frame-based knowledge representation system, using an object-oriented data model, RegulonDB and Transfac rely on relational models. None of the data models matched the previously stated requirements for coverage, granularity and, in particular, expressive power.

Subsequently, the data exchange format BioPAX (see page 41) has been examined.

[2] `http://ecocyc.org/`, – access date 2012-11-07.

[3] `http://regulondb.ccg.unam.mx/` – access date 2012-11-10.

[4] `http://www.gene-regulation.com/pub/databases.html` – access date 2012-11-07.

It has become a standard format for the representation of biological pathways at the molecular and cellular level. BioPAX has been developed in a collaborative effort of data providers, users, and tool developers. By design, it contains the key elements of data models of different established pathway databases to foster integration and interpretation across databases. However, the analysis revealed that BioPAX Level 2, which has been examined, did not support the representation of gene regulation and genetic interactions. Although this has changed with BioPAX Level 3 (released in 2010), the representation of gene regulatory processes and their participants in this new version is still too coarse-grained to satisfy the previously stated requirements.

Finally, GO, SO and ChEBI were examined as three biomedical ontologies from the OBO library (see page 18), and MeSH as thesaurus from the UMLS Metathesaurus (see page 41). GO was found to cover a large spectrum of gene regulatory processes, cellular locations in which they take place, and functions of participating proteins. However, they are dealt with in separate ontology branches, which at examination time were isolated from each other, since GO at that time lacked any explicit relationships between classes in different ontology branches. In addition, GO was found to describe some relevant aspects of gene regulation, but misses others (e.g., genes, proteins, regulatory regions of genes and binding sites of proteins). Analogously, the other mentioned ontologies and terminologies were found to provide sub-hierarchies or single classes relevant for the construction of a gene regulation ontology (for example, the SO provides classes representing genes, transcription factor binding sites, and other sequence features, and ChEBI nucleic acids, proteins, their constituents nucleotides and amino acids, amongst others). However, none of the resources qualified as computational model of the entire gene regulation, either due to the fragmented coverage of the field or the missing expressive power, indicated by the lack of formal class definitions and explicit relationships between classes, as in the case of MeSH, which is a thesaurus and not an ontology.

Since none of the existing knowledge resources fully satisfied the specified requirements (which is not surprising since the processing requirements of fact extraction are rather sophisticated, compared to those of data annotation or document retrieval, as typical applications of existing knowledge resources covering aspects of gene regulation) the decision was made to develop a new ontology, GRO.

Resource name	Relevant contents for GRO
BFO	top level classes
RO	basic relation types
GO	molecular functions, biological processes, cellular components
SO	sequence regions and attributes thereof
ChEBI	chemical entities
IMR	transcription factors and their functional domains
NCBI Taxonomy	eukaryotes, prokaryotes
MeSH Thesaurus	verbal class definitions
InterPro	transcription factors and their functional domains
Transfac	transcription factors and their functional domains

Table 6.2: Conceptual and terminological resources utilized for the construction of GRO.

6.2 Design and Implementation

Knowledge Acquisition

For the creation of GRO, top level classes were derived from the top level ontology BFO and basic relation types from RO (see section 2.2.1). Domain-specific classes were derived from biomedical ontologies from the OBO library (e.g., GO, SO, IMR and ChEBI, see page 18), knowledge resources from the UMLS Metathesaurus (e.g., the NCBI Taxonomy and the MeSH thesaurus, see page 41) and domain-specific databases (e.g., InterPro[5] and Transfac). Verbal definition were compiled based on GO, SO, MeSH, WordNet[6] and various Web resources (e.g., the English Wikipedia[7] and domain-specific glossaries). Some important knowledge resources for the construction of GRO are summarized in table 6.2. In preparation for additional classes and relations, domain knowledge was manually extracted from a comprehensive textbook on molecular biology [Alberts et al., 2002] and 150 Medline abstracts on gene regulation retrieved by the search engine PubMed[8]. Prior to the manual analysis, the abstracts were automatically tokenized, part-of-speech-tagged and chunked, in order to identify noun phrases as candidates of verbal mentions of ontology classes. Finally, domain experts contributed their background knowledge as an additional, complementary knowledge source.

[5] http://www.ebi.ac.uk/interpro/ – access date 2012-11-07.

[6] http://wordnet.princeton.edu/ – access date 2012-11-07.

[7] http://en.wikipedia.org/ – access date 2012-11-24.

[8] http://www.ncbi.nlm.nih.gov/pubmed – access date 2012-11-29.

Conceptualization

The conceptualization underlying GRO has been developed fully manually, in a collaborative effort of domain experts and knowledge engineers. First, fundamental gene regulatory processes and their participants were represented as classes. Next, the classes were hierarchically organized in a continuants and an occurrents branch (figure 6.2 A and B). The first one covers "things", such as molecular entities and molecular functions, the second one processes. Continuants were further subdivided into physical continuants, such as genes or proteins, and non-physical continuants, such as protein functions. The resulting GRO represents a directed acyclic graph (DAG), in which classes can have more than one superclass. The ontology was further provided with domain-independent top level classes derived from BFO, and domain-specific classes derived from existing biomedical ontologies and domain-specific databases. No class instances were added to GRO.

Next, conceptual and data relations were added to the ontology (table 6.3). Below, inverse relations are given in brackets. In GRO 0.5, the conceptual relation 'partOf' ('hasPart') has been provided to link spatial or temporal parts to their whole, such as 'ProteinDomain' to 'Protein' (spatial) or 'TranscriptionInitiation' to 'Transcription' (temporal). The relation 'participatesIn' ('hasParticipant') and its subrelations 'agentOf' ('hasAgent') and 'patientOf' ('hasPatient') have been provided to link continuants and occurrents to the processes they are involved in. While "agent" refers to an active participant, which drives a process, "patient" denotes a passive participant, on which the process has a certain impact. The relation 'encodes' ('encodedIn') links genes to gene products, and 'resultsIn' ('resultsFrom') processes to their outcomes, 'hasQuality' entities to qualities that inhere in them, and 'functionOf' ('hasFunction') functions to their bearers. The relation 'locatedIn' ('locationOf'), a subrelation of 'spatiallyRelated', is used to link a physical entity or process to the place where it is located. The relations 'startsIn' and 'endsIn', subrelations of 'temporallyRelated', are used to link processes to the location where they started or terminated, respectively. The 'precedes' ('precededBy') relation is used to link consecutive processes, such as transcription and translation. The relation 'fromSpecies' links species-specific classes to the species they belong to, such as 'BacterialRNAPolymerasePromoter' to 'Bacterium'. The possibility to express species-specificity is essential for the realization of a cross-species ontology such as GRO. Finally, the data relation 'hasPolarity' specifies the "polarity" of a process. For example, an activation process has positive polarity and an inhibition process negative polarity. Based on the mentioned conceptual relations, non-taxonomic relationships were specified between GRO classes (or their instances, respectively), within and across the ontology branches.

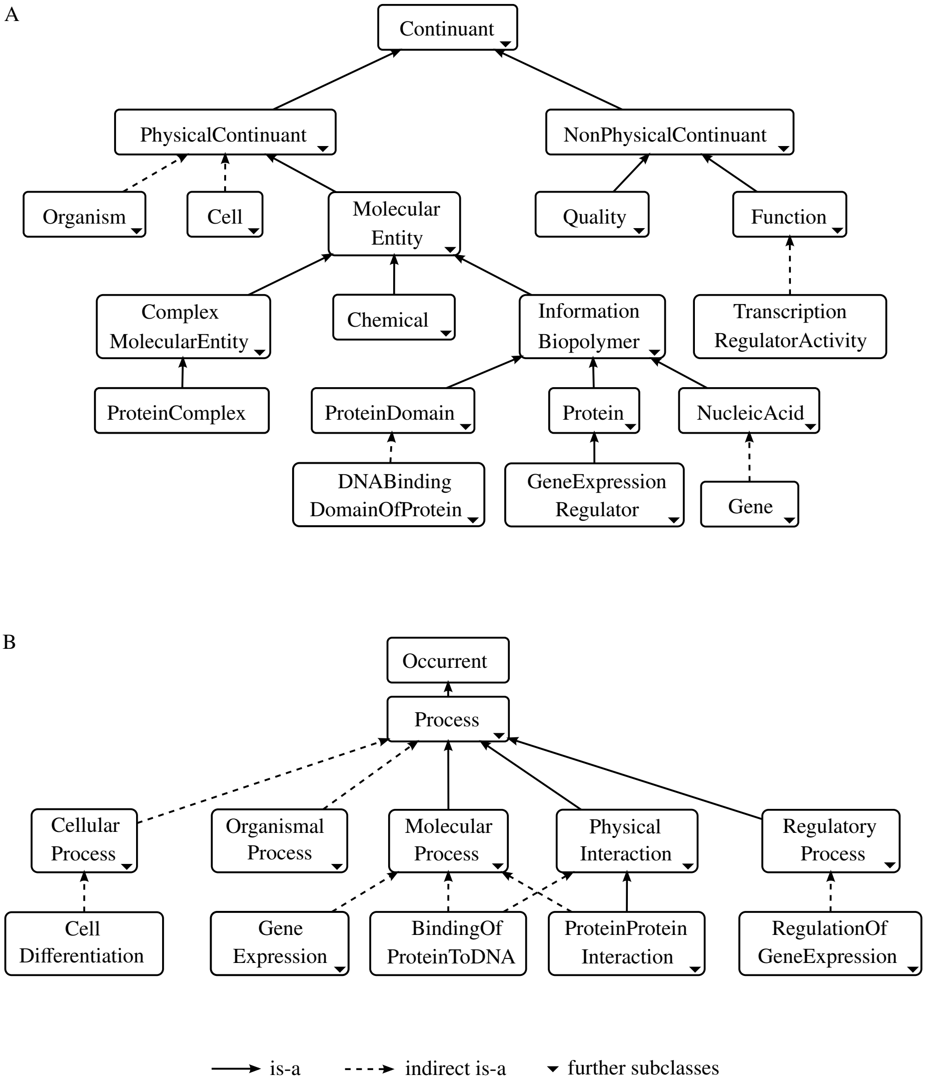


Figure 6.2: A: Continuant branch of GRO . B: Occurrent branch of GRO.

Relation	Domain	Range
‘partOf’ (‘hasPart’)	-	-
‘participatesIn’ (‘hasParticipant’)	-	-
‘agentOf’ (‘hasAgent’)	-	-
‘patientOf’ (‘hasPatient’)	-	-
‘encodes’ (‘encodedIn’)	‘NucleicAcid’	‘GeneProduct’
‘resultsIn’ (‘resultsFrom’)	-	-
‘hasQuality’	-	‘NonPhysicalContinuant’
‘hasFunction’ (‘functionOf’)	-	‘Function’
‘spatiallyRelated’	-	-
‘locatedIn’ (‘locationOf’)	-	‘PhysicalContinuant’
‘startsIn’	‘Process’	‘PhysicalContinuant’
‘endsIn’	‘Process’	‘PhysicalContinuant’
‘temporallyRelated’	-	-
‘precedes’ (‘precededBy’)	-	-
‘fromSpecies’	-	‘Organism’
‘hasPolarity’	-	{positive, negative, positive and negative, unknown}

Table 6.3: Relation types used in GRO 0.5. For inverse relations, given in brackets, inverted domain and range restrictions apply.

Implementation

For the implementation of GRO, OWL DL was chosen as ontology language because it fulfills the previously stated requirements for expressive power, computational demands, and required tool support. On the one hand it is expressive enough to allow for the creation of formal class definitions. On the other hand, it retains computational completeness and decidability, enabling automatic classification and consistency checking. In addition, ontology editors, reasoners, and APIs to create, edit and classify OWL DL ontologies are available.

To large parts, GRO was implemented manually, using the ontology editor Protégé. First, the class hierarchy was implemented. Next, the conceptual and the data relations were implemented as OWL object and datatype properties. For half of the conceptual relations and the only data relation domain or range have been specified. In addition, the data relation was defined as being a functional relation. The conceptual and data relations were used to create formal class definitions of GRO classes. These definitions express, amongst others, non-taxonomic relationships between GRO classes.

The following URI policy has been specified for GRO: GRO classes and relations should be provided with URIs that consist of the namespace “`http://www.bootstrep.eu/ontology/GRO#`” (abbreviated with “`gro`”) and a local name that is unique in

GRO. Local names in GRO should start with an upper case letter, and use CamelCase notation.

Each GRO class, except for the root classes ‘Continuant’ and ‘Occurrent’, has explicitly been linked to at least one superclass, or provided with a formal definition that allows a reasoner to classify it as subclass of at least one superclass. An example for the second case is the class ‘GeneProduct’. It has no direct super class in the asserted class hierarchy, but is defined as the union of the classes ‘RNA’ and ‘Protein’. Since both classes are either a direct or an indirect subclass of the class ‘InformationBiopolymer’, running a reasoner, the class ‘GeneProduct’ itself is classified as subclass of ‘InformationBiopolymer’. To check GRO for logical consistency during implementation, the OWL reasoner Pellet was run repeatedly. To enable more restrictive consistency checks, explicit ‘disjoint-with’ relationships were introduced between some GRO classes, such as the top level classes ‘Continuant’ and ‘Occurrent’ and the classes ‘DNA’ and ‘RNA’. Some relation types have been reified in GRO in order to be able to provide them with a formal definition. An example is the relation ‘positively-regulates’. It has been transformed into the class ‘PositiveRegulation’ and was defined as a regulatory process with polarity “positive”.

Next, classes were provided with natural language annotations. The following annotation guidelines have been specified for GRO: Each class should be provided with a preferred class label that is represented as ‘rdfs:label’ annotation and unique within GRO. Optional alternative class labels should be represented as ‘gro:synonym’ annotations. In addition, each class should be provided with a verbal definition, represented as ‘gro:definition’ annotation. Verbal definitions adopted from existing terminological resources should reference their origin appropriately. Finally, classes derived from existing knowledge resources should be provided with ‘gro:reference’ annotations that specify the unique identifier and label of the original class or entry. For example, the ‘gro:reference’ annotation “GO:0003723 RNA binding” of the GRO class ‘BindingToRNA’ expresses that the class has been derived from the equivalent GO class ‘GO:0003723’, labeled “RNA binding”. Annotations were manually compiled and added to the ontology either manually, using Protégé, or by programming, using, e.g., the Jena framework.

Several pre-release versions of GRO have been created (three of them are referred to as GRO I, II and III), before the first versions were project internally (GRO 0.1) and publicly released (GRO 0.2). Statistics on these early versions of GRO are presented in table 6.4. The numbers illustrate the evolution of GRO. While GRO I consists of a pure class hierarchy, GRO II already contains class labels, the first conceptual relations, formal class definitions, and ‘disjoint-with’ relationships between classes. GRO III contains additional classes, conceptual relations, formal class definitions, and ‘disjoint-with’ relationships. GRO 0.1 contains the first ontology annotations, including a version number. GRO 0.2 contains the first verbal class definitions, alternative class labels, and reference annotations for classes that have been adopted from external

Feature	GRO I	GRO II	GRO III	GRO 0.1	GRO 0.2
Classes	184	238	352	419	419
Conceptual relations used	-	6	15	15	15
Defined classes	-	2%	16%	14%	14%
‘is-a’ relationships	194	248	314	386	384
‘disjoint-with’ relationships	-	53	89	94	96
Conceptual relationships	-	140	281	309	309
Annotation properties used	1	2	2	4	7
Ontology annotations	-	-	-	3	3
Classes with label	-	100%	99%	100%	100%
Classes with alt. label	-	-	-	-	5%
Classes with verbal def.	-	-	-	-	50%
Classes with reference	-	-	-	-	32%

Table 6.4: Statistics on early versions of GRO, up to the first public release. The numbers refer to the asserted class hierarchy of the respective ontology. Percentages have been rounded. Relationships were counted as explained on page 26. Conceptual relations and annotation properties were only counted if they are used in at least one formal class definition or annotation. Only non empty annotations were considered.

resources. With GRO 0.2 the implementation of GRO was completed and maintenance began. Changes made to GRO across different ontology versions were tracked using the version control system Subversion.

6.3 Evaluation

All GRO pre-release and release versions have been classified and checked for logical consistency using the OWL reasoner Pellet [Sirin et al., 2007]. Consistency checks were run during the development and before the release of a new version of GRO. Each unsatisfiable class was fixed before the development was continued or the ontology was released.

Evaluation of Contents

To evaluate the correctness of contents of GRO 0.5, a domain expert who had not been involved in the construction of GRO before was asked to judge the relevance of GRO classes for the field of gene regulation (first task), to check the correctness of direct ‘is-a’ relationships between pairs of classes in the inferred class hierarchy (second task) and the correctness of formal class definitions (third task). For each task, the expert was provided with a table specifying the contents to be evaluated. In the first table, the preferred label and verbal definition of each class were specified, if available. In the second table, the preferred labels of pairs of classes were specified that are related by a direct 'is-a' relationship. In the third table, triples consisting of a class label, a relation type and a formal class definition were specified, the latter paraphrased in natural language terms, as proposed by Rector et al. [2004]. In case of a partial formal definition, the relation type is 'is-a', in case of a full formal definition 'equivalent-to'. Overall, the domain expert judged almost 97% of the classes as being relevant, over 95% of the 'is-a' relationships and almost 98% of the formal definitions as being correct (see table 6.5).

Since the evaluation relied on the authority of a domain expert who was no ontology engineer, all cases of rejections were subsequently reviewed. In this post-processing analysis, 11 out of 17 rejected classes were found to represent organisms (e.g., 'Organism', 'Microorganism', or 'Bacterium'), or upper level classes (e.g., 'Occurrent' or 'Quality'). In fact, opposed to the decisions taken by the expert, both types of classes are required in GRO and must not be deleted. While organism classes are needed for the representation of species-specific knowledge on gene regulation, the upper level classes are required because they introduce basic ontological distinctions that structure GRO. For the remaining six rejected classes, further scrutiny is required.

Regarding the rejected 'is-a' relationships between classes it turned out that 16 out of 27 rejected relationships are in fact correct. However, an imprecise or misleading class label allowed for an unintended interpretation of one of the classes concerned that is incompatible with the 'is-a' relationship. It turned out that in seven cases, the misinterpreted class was provided with a verbal definition that specified its correct meaning. Obviously, it had not been considered by the domain expert. An example is the rejected 'is-a' relationship between the classes 'Localization' and 'Process'. Probably, it was rejected because of the misleading local name "Localization". While in general language, the expression refers to a site, in molecular biology it denotes a process. The class 'Localization' of GRO refers to the molecular biology reading of the term, as its verbal definition ("Any process by which a cell, a substance, or a cellular entity, such as a protein complex or organelle, is transported to, and/or maintained in a specific location."), borrowed from GO class 'GO:0051179', clearly specifies. In another nine cases even considering the verbal definitions of the misinterpreted classes would not have clarified their meaning. For example, the expert had rejected the subclass relationship between the classes 'TranslationElongation' and 'IntraCellularProcess', probably because of the existence of *in vitro*-translation, an experimental procedure that is carried out in a reaction tube, i.e., outside a cell. In fact, GRO is intended to cover *in vivo* processes only, as a 'dc:description' annotation in the header of the ontology document specifies ("The Gene Regulation Ontology (GRO) covers gene regulatory processes occurring on the *intracellular level* and molecular entities participating in these processes."). However, the expert was not explicitly pointed to this annotation and missed it. Thus, it did not help to rule out the unwanted interpretation of the class

Feature	Total	Accepted (expert)	Accepted (final)
Classes	506	489 (96.64%)	500 (98,81%)
Direct ‘is-a’ relationships	568	541 (95.25%)	557 (98,06%)
Formal class definitions	350	342 (97.71%)	342 (97.71%)

Table 6.5: Results of evaluating the precision of contents of GRO. The numbers on ‘is-a’ relationships and formal class definitions refer to the inferred class hierarchy. The addition “expert” marks results of the expert evaluation and “final” of the expert evaluation combined with the post-processing analysis.

‘TranslationElongation’. The remaining 11 rejected ‘is-a’ relationships turned out to be in fact mistakes or require further scrutiny. For example, the ‘is-a’ relationship between the classes ‘ProteinCodingGene’ and ‘ProteinCodingDNARegion’ that was rejected by the expert is indeed incorrect because of the existence of RNA genes, i.e., genes of viruses that have RNA instead of DNA as genetic material. RNA genes encode proteins, but constitute no protein coding DNA region.

The rejected formal definitions turned out to be imprecise or deficient indeed. For example,

$$\text{‘Gene’} \sqsubseteq \exists\text{‘partOf’}.\text{‘DNA’}$$

was rightly rejected by the expert again due to the existence of RNA genes. They instantiate the class ‘Gene’, but are no segment of a DNA sequence.

As a result of the expert evaluation combined with the post-processing analysis six GRO classes (1%), 11 ‘is-a’ relationships between GRO classes (2%), 16 imprecise labels (3%), nine incomplete verbal definitions (2%) and eight formal definitions of GRO classes (2%) should be considered for revision (see table 6.5).

Evaluation of Guideline Compliance

To evaluate the guideline compliance of GRO 0.5, the previously proposed three-step procedure was applied (see section 4.5.3). In the first step, the adherence of GRO to selected design and implementation guidelines was checked, answering 30 guideline control questions. Questions Q05 (Is each class provided with at most one superclass in the asserted class hierarchy?), Q06 (Are classes with only one direct subclass avoided in the inferred class hierarchy?) and Q7 (Are classes with many direct subclasses avoided in the inferred class hierarchy?) revealed on the class and taxonomy level that in the asserted class hierarchy 30 classes of GRO (6%) had more than one direct superclass, and in the inferred class hierarchy 65 classes (13%, or 36% of all non-terminal classes) have only one direct *sub*class, while two have overly many subclasses, applying a threshold of 12, as proposed by Noy and McGuinness [2001].

Questions Q09 (Is the disjointness of classes explicitly specified?) and Q10 (Are domain, range and further properties of relations specified?) revealed on the level of formal semantics that only 91 class pairs were linked by an explicit 'disjoint-with' relationships. Furthermore, for eleven conceptual and one data relation no domain, for six conceptual relations no range and for all 14 conceptual relations no additional properties (e.g., being functional or transitive) have been specified, considering only relations that are involved in at least one full or partial formal class definition. Questions Q11 (Are formal class definitions provided?) and Q13 (Are existential restrictions complemented with universal restrictions?) revealed that for only 65 classes (13%) full formal class definitions and for only 54 existential restrictions (less than 20%) closure axioms have been specified.

Questions Q16 (Are compulsory annotations provided?) and Q17 (Are optional annotations provided?) revealed on the annotation level that 8% of the classes of GRO miss a verbal definition and more than 94% alternative class labels. Question Q18 (Are annotation type-specific guidelines adhered to?) revealed 13 cases of incorrectly formatted reference annotations. They either miss whitespace, include additional whitespace or contain more than one reference at once. Question Q20 (Is each type of annotation represented using one annotation property only?) revealed that two verbal definitions have mistakenly been provided in terms of 'rdfs:comment' instead of 'gro:definition' annotations. Question Q21 (Do class labels follow approved naming conventions?) revealed that in disagreement with established naming conventions, from a total of 539 distinct preferred and alternative labels of GRO classes 17% start with a capital letter. In addition, 27% consist of or contain an acronym or abbreviation, such as "ORF" (open reading frame), "S-phase" (synthesis phase of the cell cycle) or "tRNA" (transfer ribonucleic acid). The two acronyms "RNA" and "DNA" are included in 95 labels of GRO classes. Question Q22 (Are duplicate class labels and duplicate verbal definitions avoided?) helped to identify two duplicate class labels and three duplicate verbal definitions. Question Q23 (Are verbal definitions non-circular?) revealed for 60% of the verbal definitions of GRO classes circularity index values [see Köhler et al., 2006] greater than zero. Further scrutiny would be required to confirm or reject the circularity concerns.

Question Q24 (Do local names adhere to a consistent naming policy?) revealed on a technical level that 6% of the local names of GRO classes violate the URI policy of GRO, which requires a capital first letter and the strict adherence to CamelCase notation. They either start with a lower case letter or an underscore, or contain an underscore or hyphen as word delimiter. Question Q25 (Are redundant ontology elements avoided?) revealed that the annotation properties 'gro:synonym', 'gro:definition' and 'gro:reference' have mistakenly been specified also as data relations. Questions Q26 (Are unused object, datatype and annotation properties avoided?) and Q27 (Are artifacts from tool-use avoided?) revealed that ten conceptual relations, seven data relations, and two annotation properties of GRO are currently "unused", i.e., not involved in any formal class definition or annotation. Furthermore, three unwanted individuals and eight empty annotations were detected in GRO. Four of the unused data relations, three of the unused annotation properties, and two individuals contained in GRO seem to be artifacts from using the ontology editor Protégé and should be removed. Question Q30 (Are reused contents up to date and are their sources referenced?) revealed regarding the aspects of knowledge reuse and usability of the ontology that 38% of the classes of GRO are provided with a 'gro:reference' annotation. Furthermore, 58% of the verbal class definitions contain an external source identifier, indicating that the classes and annotations have been adopted from external resources. However, further scrutiny would be required to reveal if *all* reused classes, relations and annotations have been provided with appropriate reference annotations and citations.

In the second step, GRO was checked for common modeling pitfalls using the ontology pitfall scanner Oops!. The tool identified seven types of potential errors. It identified the classes 'Continuant' and 'Occurrent' as cases of pitfall P04 (creating unconnected ontology elements) and the classes 'ComplexOfProteinAndDNA', 'ComplexOfProteinAndRNA' and 'CellularComponentOrganizationAndBiogenesis' as cases of pitfall P07 (merging different concepts in the same class). In addition, it identified 536 classes and relations of GRO that lack an 'rdfs:label' or 'rdfs:comment' annotation as cases of pitfall P08 (missing annotations), 28 conceptual or data relations that lack a domain or range as cases of pitfall P11 (missing domain or range in properties) and eight conceptual relations that lack an inverse relation as cases of pitfall P13 (missing inverse relations). The tool further detected that different naming criteria have been used for local names of classes and relations in GRO, a case of pitfall P22 (using different naming criteria in the ontology). Finally, it identified the recursive definition of the class 'PocketDomain' as case of pitfall P24 (using recursive definition).

In the third step, the adherence of GRO to accepted OBO Foundry principles was checked. The checks revealed that three of the principles are not adhered to by GRO, and another three are only partially adhered to. GRO has mainly been used by BOOTStrep members. Hence, it does not adhere to the principles FP09 (The ontology has a plurality of mutually independent users.) and FP10 (The ontology is developed collaboratively with other OBO Foundry members.). Furthermore, maintenance of GRO has officially been terminated in spring 2009, when the BOOTStrep project ended. Hence, GRO does not adhere to principle FP16 (The ontology should be continuously maintained.). The contents of GRO are clearly specified, and most top level classes are provided with appropriate verbal definitions. However, GRO by design overlaps with other OBO ontologies and explicit links to OBO Foundry ontologies are missing, indicating a partial adherence to FP05 (The ontology has a clearly specified and delineated content, provides coherent verbal definitions of top level classes and incorporates explicit links to other OBO Foundry ontologies.). GRO relations such as 'partOf', 'locatedIn', 'precededBy', and 'hasParticipant' with the subrelation 'hasAgent' were derived from RO. However, some of them are used less strictly than in RO. For example, 'partOf' is defined as a reflexive, anti-symmetric, and transitive relation in RO, but not in GRO, indicating a partial adherence to principle FP07 (The ontology uses relations in the style of RO.). Finally, GRO to a large extent complies with the OBO naming conventions. However, certain labels of GRO classes start with a capital letter or contain an acronym or abbreviation, indicating only a partial adherence to principle FP12 (The ontology adheres to the OBO naming conventions.).

6.4 Documentation and Release

The first public release of GRO has been documented in terms of a proceedings paper [Beisswanger et al., 2008a]. In the ontology document representing GRO, meta-data annotations are provided as an alternative way of documentation. For GRO 0.5, an 'owl:versionInfo', a 'dc:date', a 'dc:title', a 'dc:description' and several 'dc:creator' annotations are provided that specify the version number, the release date, the name, a short summary of contents and the names of the developers of the ontology. The public release versions of GRO (GRO 0.2–0.5) are available at the GRO website, hosted at the European Bioinformatics Institute.[9] GRO is additionally available at the NCBO BioPortal and the OBO library (see page 17).

6.5 Maintenance

GRO has been actively developed and maintained between spring 2006 and spring 2009, the runtime of the BOOTStrep project. In this period, several pre-releases, one project internal release and three public releases of GRO were created. An additional public release (GRO 0.5) was created retrospectively, in spring 2010. Maintenance of GRO was carried out according to requirements. The first maintenance cycle started after the first public release of GRO. New requirements and suggestions for corrections and extensions resulted primarily from use cases of GRO, such as the semantic corpus annotation projects by Buyko et al. [2010] and Thompson et al. [2009] and the work on rule-based relation extraction by Kim and Rebholz-Schuhmann [2011]. For manual maintenance work the ontology editor Protégé-OWL was used and for programming-based subtasks the Jena framework and the OWL API. Changes were tracked and different ontology versions administered using the version control system Subversion. Maintenance has been carried out by a team of knowledge engineers and domain experts. Each team member worked on a local copy of the ontology and uploaded changes to the central Subversion repository. To avoid conflicts, edits were performed in mutual consultation, whenever possible. Before a new version of GRO was released, it was checked for logical consistency using an appropriate OWL reasoner, potential

[9] `http://www.ebi.ac.uk/Rebholz-srv/GRO/GRO.html` – access date 2012-02-27.

Feature	GRO 0.2	GRO 0.3	GRO 0.4	GRO 0.5
Classes	419	433	439	506
Conceptual relations used	15	15	15	14
Defined classes	14%	15%	15%	13%
‘is-a’ relationships	384	390	397	473
‘disjoint-with’ relationships	96	91	91	91
Conceptual relationships	309	322	327	326
Annotation properties used	7	7	10	10
Ontology annotations	3	3	6	8
Classes with label	100%	100%	100%	100%
Classes with alternative label	5%	5%	6%	6%
Classes with verbal definition	50%	94%	94%	92%
Classes with reference	32%	35%	36%	38%

Table 6.6: Statistics on public release versions of GRO. The numbers refer to the asserted class hierarchy of the respective ontology. Percentages have been rounded. Conceptual and data relations and annotation properties were only counted if they are used in at least one formal class definition or annotation. Relationships were counted as explained on page 26. Only non-empty annotations were counted.

errors were fixed, and a version number was provided in terms of an ontology annotation in the header of the respective ontology document. For the most recent GRO releases, brief release notes have been published at the GRO website, in the “Latest news” section. Statistics on the four public release versions of GRO are presented in table 6.6. Compared to GRO 0.2, GRO 0.3 contains many new verbal class definitions. GRO 0.4 contains additional classes, alternative class labels and ontology annotations, and GRO 0.5 additional classes and a new data relation.

6.6 Use Cases

Semantic Annotation of Text Documents

GRO has been used in support of the semantic annotation of three different domain-specific corpora. The GeneReg corpus consists of 314 Medline abstracts dealing with gene regulation in the model organism *E. coli* [Buyko et al., 2010]. The abstracts have been annotated by domain experts with domain-specific entities (primarily genes and transcription factors, as participants of gene regulatory processes), pairwise relationships between regulators and regulated genes, and so-called “event triggers”. The latter are primarily verbs that are essential for the description of gene expression and gene regulation events. In this context, “event” denotes a fact describing state changes of certain entities or their properties, or sequences of such changes. For all three annotation levels the annotation vocabulary was taken from GRO. While the vocabulary for the named entity annotation has been extracted from the continuant branch of GRO, the one for the annotation of gene regulatory processes was extracted from the occurrent branch. The GeneReg corpus served as training data for a high performance event extraction system based on supervised machine learning [Buyko et al., 2011]. The GREC corpus consists of 240 Medline abstracts in which domain experts have annotated sentence-bound events relating to gene expression and its regulation [Thompson et al., 2009]. The annotation scheme used is centered on verbs and nominalized verbs. For each instance of an event, all participants (on the linguistic level "arguments") in the same sentence were identified and subsequently assigned both a semantic role and a biological concept type. The biological concept types were matched to classes in GRO. Besides core relationships between entities, in GREC also the location or environmental conditions in the context in which an event took place were annotated. GREC has been used for the supervised acquisition of semantic event frames, which are an essential linguistic resource for the automatic extraction of information from the biological literature [Sasaki et al., 2008]. Finally, Kim and Rebholz-Schuhmann [2011] compiled a corpus of 209 Medline abstracts about gene regulation in the model organism *E. coli* and let domain experts annotate verbal mentions of three types of events based on GRO: regulation of gene transcription, regulation of gene expression, and binding of transcription factors to gene regulatory regions. The corpus was used in support of a rule-based information extraction system (see page 104).

Creation of Formal Class Definitions

GRO has further been used as conceptual backbone for the creation of formal class definitions for GO [Ashburner et al., 2000]. GO has become a powerful tool for the functional annotation of gene products represented in biomedical databases. However, in the field of biomedical NLP, the use of GO is hampered by the often long and complex class labels of GO classes [Hirschman et al., 2005] and missing formal class definitions [Kim et al., 2010]. Hence, Kim et al. [2010] proposed a method to create formal definitions for GO classes, which is based on GRO. It runs as follows: First, verbal mentions of genes, proteins and classes from selected biomedical ontologies (e.g., SO [Eilbeck et al., 2005] and ChEBI [Degtyarenko et al., 2008]) are identified in labels of GO classes and the most closely matching GRO class is assigned to them. Next, a syntactic parser is run on the class labels. The resulting predicate-argument structures are converted in dependency structures, including the GRO assignments. Finally, manually constructed patterns are matched to the dependency structures to reveal the semantic structure of the labels of GO classes. The GRO-based definitions rely, amongst others, on the 'has-agent' and 'has-patient' relations specified in GRO. They turned out to be particularly well-suited for biomedical information extraction and representation because they are event-independent, i.e., not restricted to a particular type of event [Kim et al., 2010]. According to Kim et al. [2010], the GRO-based approach provided full formal class definitions for 75% of the GO classes representing gene regulatory processes, while an alternative approach by Mungall et al. [2011a] achieved definitions for only 15% of these classes.

Rule-Based Information Extraction

GRO has additionally been used as basis for rule-based information extraction. In biomedical NLP, information extraction approaches are explored to support human database curators who manually extract relevant facts from the biomedical literature and incorporate them in factual databases (see section 1.1. Since the natural language terms used to express facts show a high variability and, e.g., event descriptions are often nested ("complex" events) or spread over several sentences (implicitly mentioned events), in-depth domain knowledge and reasoning skills are required to detect or even conclude events from distributed evidence. This is why the *automatic* extraction of events is challenging and requires an in-depth semantic analysis and automatic inferencing, in particular if complex or implicit event descriptions are concerned. Kim and Rebholz-Schuhmann [2011] developed a system to automatically deduce implicit, possibly complex events from explicitly expressed ones. It is based on manually created rules that capture sophisticated forms of domain knowledge. The rules are based on the explicit specification of gene regulatory processes, their participants, and the relationships holding between them in GRO.

The system by Kim and Rebholz-Schuhmann [2011] first recognizes mentions of entities in text, using a dictionary-based approach, and associates them to classes in the continuant branch of GRO (e.g., verbal mentions of genes are associated to the class 'Gene', and verbal mentions of transcription regulator proteins to the class 'TranscriptionRegulator'). Next, it identifies mentions of processes in text, based on keywords and linguistic pattern matching, and associates them to classes in the occurrent branch of GRO (e.g., "regulate" is taken as keyword for mentions of the class 'RegulatoryProcess'). Finally, it deduces complex events utilizing the manually encoded rules.

Three exemplary rules are presented in table 6.7. The first one states that if the system detects a verbal mention of the class 'GeneExpression' and another one of the class 'RegulatoryProcess' within one sentence and identifies the first as patient of the latter via syntax analysis, it will deduce that the mention of the regulatory process in fact refers to the more specific class 'RegulationOfGeneExpression'.

The second rule states that if the system detects a verbal mention of the class 'Protein' and another one of the class 'RegulationOfTranscription' within one sentence and identifies the first as agent of the latter via syntax analysis, it will deduce that the mention of the protein in fact refers to the more specific protein class 'TranscriptionRegulator'.

No.	Condition(s)	Conclusion

1	{‘RegulatoryProcess’
	{‘hasPatient’ ‘GeneExpression’}
	}
		{‘RegulationOfGeneExpression’
		{‘hasPatient’ ‘GeneExpression’}
		}

2	{‘RegulationOfTranscription’
	{‘hasAgent’ protein}
	}
		{‘RegulationOfTranscription’
		{‘hasAgent’ ‘TranscriptionRegulator’}
		}

3	{‘RegulatoryProcess’
	{‘hasPolarity’ polarity_i}
	{‘hasAgent’ ‘RegulatoryProcess’
	{‘hasPatient’ patient}
	{‘hasPolarity’ polarity_j}
	}
	}
		{‘RegulatoryProcess’
		{‘hasAgent’ patient}
		{‘hasPolarity’ XNOR(polarity_i, polarity_j)}
		}

Table 6.7: Three exemplary rules based on GRO, used for rule-based event extraction. “XNOR” denotes the Boolean function “inverted exclusive OR”.

The third rule states that if the system detects two verbal mentions of regulatory processes P_1, P_2 in a text, not necessarily within one sentence, of which both have a polarity that is either “positive” or “negative” and it identifies that P_1 is the agent of P_2 via syntax analysis, it will deduce that an only implicitly mentioned third regulatory process exists that has the patient of the nested process P_1 as agent and positive polarity if P_1 and P_2 share the same polarity and negative polarity otherwise.

The system was tested in three different settings. The first one involved a manually annotated corpus and the remaining two curated databases. According to Kim and Rebholz-Schuhmann [2011], in each setting the rule-based inferencing step substantially improved the extraction results. Hahn et al. [2009] further report that the same rule-based system outperformed a more general, machine learning-based system in an evaluation against real world data.

Chapter 7

The Major Histocompatibility Complex Ontology

The Major Histocompatibility Complex Ontology (MaHCO) is an ontology on the MHC of multiple species. Its structure, contents and use cases have already been described in Beisswanger et al. [2007] and DeLuca et al. [2009]. The focus of the current chapter is on the life cycle of MaHCO, including the evaluation of the latter.

7.1 Requirements Analysis

MaHCO has been developed in the context of the StemNet project[1]. The project aimed at the development of an infrastructure supporting donor search for hematopoietic stem cell transplantation. The infrastructure should enable the integrated semantic access to the scientific literature and domain-specific databases, as two hitherto disconnected knowledge resources. It should comprise a semantic document retrieval system and specialized bioinformatics tools for browsing data relevant for donor search. As computationally accessible background knowledge for the retrieval system and the mentioned tools, vocabulary for the semantic annotation of text documents and semantic mediator between textual and factual data a knowledge resource was required that describes the major histocompatibility complex (MHC) of human and other species relevant for research, such as mouse and dog. It should cover MHC molecules, chains,

[1] `http://www.stemnet.de/` – access date 2012-11-26.

Aspect	Description of requirements
Purpose	conceptual backbone for domain-specific browsing and search applications, vocabulary for the semantic annotation of text corpora
Domain	MHC of human, mouse and dog
Coverage and granularity	MHC genes, alleles and molecules of human, mouse and dog; fundamental classes and relations and fine-grained class hierarchies
Expressive power and computational demands	formal class definitions of top level classes, automatic classification and consistency checking
User group	domain experts and machines
Tool support	ontology editor, reasoner

Table 7.1: Results of the requirements analysis of MaHCO.

genes and alleles and relationships between them. In particular, it should render a consistent view on human leukocyte antigen (HLA) alleles (see below), handling intricacies of the HLA nomenclature existing at that time. The granularity of the knowledge resource should range from basic classes with exact textual and formal definitions to fine-grained subclass hierarchies. While the basic classes were intended as annotation vocabulary to be used by human annotators, the fine-grained class hierarchies were intended as background knowledge to be used by machines in browsing and retrieval settings. The resource should be implemented in a formal, machine processable language that allows for automatic classification and consistency checking, and for which editors and reasoners are available. The results of the requirements analysis are summarized in table 7.1.

Below, an introduction to the MHC is given, before existing domain-specific databases and biomedical ontologies are checked whether they already satisfy the stated requirements.

The Major Histocompatibility Complex

MHC molecules take a central position in the immune system. As cell surface receptors that bind antigen fragments and present them to crucial cells of the innate and adaptive immune system [Paul, 2003, page 572] they are significantly involved in immune recognition, histocompatibility and autoimmunity, amongst others. MHC molecules are mainly distinguished into class I and class II molecules. MHC I molecules occur on the surface of nearly every nucleated body cell and present fragments of proteins that have been synthesized by the cell to cytotoxic T-cells. MHC II molecules, in turn, occur on the surface of professional antigen presenting cells (e.g., B-cells and macrophages) and present fragments of proteins that have been ingested by the cell to T-helper cells [Paul, 2003, page 577]. While MHC class I molecules consist of a single MHC chain, called α chain, and a non-MHC chain, MHC class II molecules consist of two MHC chains, called α and β chain [Paul, 2003, page 576].

The genes in which MHC chains are encoded are called MHC genes, and the genomic area of vertebrates that encloses MHC genes simply the MHC. The MHC of human is called human leukocyte antigen (HLA) system, that of dog dog leukocyte antigen (DLA) system and that of mouse H-2 [Paul, 2003, page 572]. Besides MHC genes, the MHC also encloses some non-MHC genes, such as the MIC genes located in the MHC class I region (hence called “class I-similar”), the TAP genes located in the MHC class II region and so-called MHC class III genes. The latter encode plasma proteins, of which some are involved in the complement system (e.g., the complement factors C2 and C4) [Paul, 2003, page 576].

In the medical field, a chance to cure patients with leukemia and other malignant hematological tumors is to carry out a hematopoietic stem cell transplantation (HSCT) [Paul, 2003, page 1537]. The goal is to trigger the elimination of tumor cells using multipotent hematopoeitic stem cells from an allogeneic (i.e., genetically different) donor. This desired therapeutic effect is known as graft versus leukemia (GvL) [Paul, 2003, page 1537]. However, due to the complex genetic differences between stem cell donor and recipient, the transplantation involves high risks for unintended immunological side effects, such as graft versus host disease (GvHD), an immune response of donor T-cells against host cells [Paul, 2003, page 1537]. To ensure the compatibility between donor and recipient and control the risk of GvHD, without increasing the risk of relapse after the transplantation, a complex, interactive analysis of numerous parameters is required. It includes typing of HLA alleles of donor and recipient, because as determinants of histocompatibility, MHC class I and II molecules have a strong impact on transplant compatibility.

HLA-typing is a complex, data intensive task that heavily relies on computers for storage, organization and interpretation of information. It can be performed with various levels of precision [Little, 2007], leading to a hierarchical categorization of HLA alleles. The traditional method of HLA typing is via serological testing. Antibodies are used to recognize particular structural domains of HLA proteins classifying the proteins into serological groups. For a more precise grading into serological splits, antibodies of higher specificity are used that react only with subsets of HLA proteins of a serological group. However, the precision of serological typing is limited by the fact that several HLA proteins share relevant structural domains. This is different with sequencing-based HLA typing. It relies on DNA sequencing to determine which HLA allele is present. Sequencing-based typing is also performed with different levels of precision, resulting in so-called low, medium or high resolution results [Little, 2007].

There is a nomenclature for HLA alleles that makes typing level and classification of alleles apparent from their names. When MaHCO was developed, the HLA nomenclature relied on a two-to-eight digit code [Marsh, 2003]. According to this code, allele names sharing the first two digits refer to alleles that encode proteins belonging to the same serological group (e.g. "A2") and hence are considered a "low resolution match". Allele names sharing the first four digits refer to alleles that encode the same protein. Allele names that share the first six digits refer to alleles with identical coding sequence (i.e., without synonymous mutations). Finally, allele names that share all eight digits refer to alleles that are even identical in non-coding parts. However, given that the number of known HLA alleles is continuously increasing, a serious limitation of this naming policy is that only 99 alleles can be distinguished on each typing level. In the groups A*02 and B*15 the number of HLA alleles encoding different proteins already exceeded 99. To cope with this issue, the WHO Nomenclature Committee for Factors of the HLA System introduced the "rollover" groups A*92 and B*95 for the hundredth and all further A*02 and B*15 alleles [Marsh et al., 2002]. [2]

Related Knowledge Resources

Existing knowledge resources on the MHC comprise domain-specific databases, biomedical ontologies and broad-coverage thesauri. First appropriate databases were examined. The IMGT/HLA database[3] was found to contain HLA allele entries relevant for the above-stated purpose, the international ImMunoGeneTics (IMGT) information system[4] mouse MHC gene entries, the IPD-MHC database[5] (providing access to the DLA Nomenclature Reports) DLA gene entries, the Immune Epitope Database (IEDB)[6] MHC allele entries for different species and UniProtKB[7] MHC chain entries for different species. While the IMGT information system and the IPD-MHC database provide simple listings of mouse and dog MHC genes and alleles, the IMGT/HLA database and UniProtKB rely on relational database models that support typical database queries. IEDB additionally facilitates browsing of epitope data along organisms and viruses and a hierarchical representation of MHC alleles.

Next, appropriate biomedical ontologies were examined. Though dealing with aspects of immunology, the IMGT ontology [Giudicelli and Lefranc, 1999] and the IEDB ontology, which has meanwhile been converted into the Ontology of Immune Epitopes (ONTIE) [Greenbaum et al., 2010], were found to contain only few classes di-

[2] Meanwhile, the "rollover" groups have been disposed again, because they increasingly caused confusion. In spring 2010, a completely overhauled version of the HLA nomenclature was released in which colon-delimited allele names are used (e.g., "A*9201" became "A*02:101") that make rollover groups superfluous [Marsh et al., 2010].

[3] http://www.ebi.ac.uk/ipd/imgt/hla/ – access date 2012-11-26.

[4] http://www.imgt.org/ – access date 2012-11-26.

[5] http://www.ebi.ac.uk/ipd/mhc/dla/index.html – access date 2012-11-26.

[6] http://www.iedb.org/ – access date 2012-11-26.

[7] http://www.uniprot.org/ – access date 2012-11-26.

rectly relevant for the above-stated purpose (e.g., the IMGT ontology classes 'gene' and 'MH-Chain', with subclasses). As organization schema for large databases, the IMGT ontology and ONTIE both provide high and intermediate level classes, but no detailed classes on the MHC. Furthermore, SO [Eilbeck et al., 2005], IMR [Yamamoto et al., 2004] and PRO [Natale et al., 2011] were investigated. They were found to contain some relevant mid-level classes (including the SO classes labeled "gene", "allele", "pseudogene" and "polypeptide", the IMR classes labeled "MHC", "MHC class I molecule", "MHC class II molecule", "MHC class I alpha chain", "MHC class II alpha chain", "MHC class II beta chain", and the PRO classes labeled "MHC class I histocompatibility antigen alpha chain", "MHC class II histocompatibility antigen alpha chain", "MHC class II histocompatibility antigen beta chain" and their species-specific subclasses). MHC chain classes and MHC molecule classes in IMR were further found to be explicitly linked by 'part_of' relationships, and amino acid sequence classes and corresponding MHC chain classes by 'sequence_of' relationships. However, SO lacks any MHC-specific classes and IMR and PRO deal with proteins and chemicals only, i.e., they do not cover classes representing MHC genes and their alleles. In addition, PRO has been examined retrospectively only, since it was in very premature state when the MaHCO project started.

Finally, two large-coverage thesauri were examined. The MeSH thesaurus (see page 41) was found to contain some relevant headings (e.g., 'Major Histocompatibility Complex' with subheadings 'MHC Class I Genes' and 'MHC Class II Genes', the headings 'Histocompatibility Antigens Class I' and 'Histocompatibility Antigens Class II', and the species-specific headings 'H-2 Antigens' and 'HLA Antigens' with subheadings). However, for the above-stated purpose it misses some important distinctions. For example, the heading 'MHC Class I Genes' conflates the species independent with the species dependent reading (indicated by the narrower entry terms "HLA Class I Genes" and "H-2 Class I Genes" that refer to human and mouse MHC class I genes). As another example, the heading 'HLA-A1 Antigen' conflates HLA-A surface antigens that contain alpha chains and the alpha chains themselves (the latter is indicated by a related entry term of the heading, called "HLA Class I Histocompatibility Antigen, A-1 alpha Chain"). Additional limitations of MeSH include the lack of distinct headings for MHC alleles, the low coverage of the mouse and especially the dog MHC, and the lack of explicit relationships between species-specific headings and corresponding species headings (e.g., the heading 'H-2 Antigens' is provided with the natural language annotations "mouse only", instead of being explicitly linked to the heading 'Mice'), and between MHC genes and encoded gene products (e.g., the heading 'Histocompatibility Antigens Class I' is not linked to the heading 'MHC Class I Genes').

The NCI Thesaurus [Sioutos et al., 2007], in turn, was found to contain relevant entries (e.g., the classes labeled "Major Histocompatibility Complex Gene", "MHC Class I Protein" and "MHC Class II Protein" with subclasses, such as "MHC Class I Gene" and "MHC Class II Gene"). Regarding (non-hierarchical) semantic relationships between entries, it clearly exceeds MeSH. For example, explicit relationships are stated between species-specific gene and gene product classes and the corresponding species classes (e.g., the classes labeled “HLA-A Gene” and “Human” are linked by a ‘Gene_Found _In_Organism’ relationship), and between gene product and the corresponding gene classes (e.g., the classes labeled “MHC Class I Protein” and “MHC Class I Gene” are linked by a ‘Gene_Product_Encoded_By_Gene’ relationship). However, a limitation of the NCI Thesaurus includes that it covers MHC alleles only fragmentary. Furthermore, the MHC of other species than human, such as mouse or dog, is hardly covered at all.

To sum up, each of the examined knowledge resources was found to fulfill some, but none of them all of the stated requirements. Hence, the decision was made to create MaHCO as a new ontology on the MHC. It should reuse knowledge from existing knowledge resources as far as possible and extend it appropriately.

7.2 Design and Implementation

Knowledge Acquisition

For the creation of high level and intermediate level classes of MaHCO, domain experts contributed their domain knowledge on molecular biology and immunogenetics. For the creation of conceptual relations, RO (see page 16) has been consulted. For the creation of species-specific classes, knowledge was reused from appropriate databases. Knowledge on HLA alleles and chains was compiled from the IMGT/HLA database, serological definitions were adopted from the HLA Dictionary [Holdsworth et al., 2009] and definitions of serological splits from the HLA Informatics Group at the Anthony Nolan Trust[8]. Knowledge on DLA genes and alleles was compiled from the DLA Nomenclature Reports provided by the IPD-MHC database, and on murine MHC genes from the IMGT information system.

Conceptualization

The top level of MaHCO was created manually by a team of domain experts and knowledge engineers. First, elementary classes were created (e.g., ‘MHC_Protein’, ‘MHC _Chain’, ‘MHC_Gene’ and ‘MHC_Allele’). Next, more general classes (e.g., ‘Gene’ and ‘Allele’) and more specific classes (e.g., ‘MHC_Class_I_Allele’ and ‘HLA_Class_I _Allele’) were framed, paying particular attention to the provision of interface classes to existing ontologies and terminologies (see table 7.2). The classes were organized in a multi-hierarchy. Hence, the resulting MaHCO constitutes a DAG. MaHCO is based

[8] http://www.anthonynolan.org/hig/ – access date 2012-02-19.

MaHCO class	External class	Label of external class
‘Gene’	‘SO:0000704’	gene
‘Allele’	‘NCI:C16277’	allele
‘Allele’	‘SO:0001023’	allele
‘Pseudogene’	‘SO:0000336’	pseudogene
‘Protein’	‘CHEBI:36080’	proteins
‘Polypeptide’	‘SO:0000104’	polypeptide
‘Chain’	‘CHEBI:16541’	protein polypeptide chains
‘Chain’	‘SO:0001063’	immature peptide region
‘Jawed_Vertebrates’	‘NCBITaxon:7776’	Gnathostomata
‘Human’	‘NCBITaxon:9606’	Homo sapiens
‘Dog’	‘NCBITaxon:9615’	Canis lupus familiaris
‘Mouse’	‘NCBITaxon:10090’	Mus musculus
‘Organism’	‘NCI:C14250’	organism
‘MHC_Protein’	‘IMGT:major_histocompatibility’	major histocompatibility
‘MHC_Multi_Chain _Protein’	‘GO:0042611’ (from ONTIE)	MHC protein complex
‘MHC_Chain’	‘IMGT:MH-chain’	MH-Chain
‘MHC_CassI_Alpha’	‘IMGT:MH1-Alpha-Chain’	MH1-Alpha-Chain

Table 7.2: Interface classes of MaHCO to related knowledge resources.

on strictly defined ‘is-a’ relationships between classes. The hierarchy was further extended by classes derived from species-specific databases, as mentioned above. No class instances were added to MaHCO.

MaHCO consists of four branches, describing proteins, polypeptides, nucleotide sequences and species (figure 7.1 A). The protein branch primarily contains classes representing MHC molecules, the polypeptide branch MHC chains and the nucleotide sequence branch MHC genes and alleles. The top level classes of these three branches are species-independent. However, each of the branches contains species-specific subhierarchies. Classes representing species are covered by the species branch.

To enable a clear distinction between classes representing MHC class I and class II alleles and classes representing alleles of other genes located in the MHC class I or II region (see page 109), the class ‘MHC_Allele_Encoding_Peptide_Presenting_Protein’ was introduced as new subclass of the class ‘MHC_Allele’ and union of the classes ‘MHC _Class_I_Allele’ and ‘MHC_Class_II_Allele’ (figure 7.1 B). Furthermore, to enable the grouping of classes that represent human MHC alleles encoded in the same region of the MHC, regardless of the gene product they encode, the classes ‘Human_MHC_Class _I_Region_Allele’ and ‘Human_MHC_Class_II_Region_Allele’ were introduced as new subclasses of the class ‘Human_MHC_Allele’ and direct superclasses of the classes ‘HLA_Class_I_Allele’ and ‘MIC_Allele’, and ‘HLA_Class_II_Allele’and ‘TAP_Allele’,

Relation	Domain	Range
‘encodes’ (‘encoded_in’)	‘Nucleotide_Sequence’	‘Polypeptide’, ‘Protein’
‘part_of’ (‘has_part’)	-	-
‘variant_of’ (‘has_variant’)	‘Allele’	‘Gene’
‘from_species’	‘Nucleotide_Sequence’, ‘Polypeptide’, ‘Protein’	‘Organism’

Table 7.3: Relation types used in MaHCO 1.0.1. For inverse relations, given in brackets, inverted domain and range restrictions apply.

respectively (figure 7.1 C).

A particular focus of MaHCO is on the human MHC, i.e., the HLA system. Classes representing HLA alleles were organized according to serological groups and serological splits, and in parallel according to the HLA nomenclature, whose two-to-eight digit codes reflect different resolutions of sequencing-based HLA typing. Classes that represent serological groups of HLA alleles (e.g., ‘A2’) and the classes representing the corresponding two digit groups (e.g., ‘A*02’) were represented as siblings. Furthermore, the decision was made not to represent the auxiliary rollover groups A*92 and B*95 (see page 110) in the ontology, but to represent alleles comprised by them as subclasses of the MaHCO classes ‘A*02’ and ‘B*15’.

To enable the explicit specification of non-hierarchical semantic relationships between MaHCO classes within and across ontology branches, conceptual relations were introduced (see table 7.3). Below, inverse relations are given in brackets. The relation ‘encodes’ (‘encoded_in’) was introduced to relate gene and allele classes to the respective gene product classes residing in the peptide and protein branches of MaHCO (e.g., ‘MHC_Allele’ to ‘MHC_Chain’). The relation ‘part_of’ (‘has_part’) was introduced to relate parts and wholes (e.g., MHC chains and MHC protein). The relation ‘variant_of’ (‘has_variant’) was introduced to link allele classes to the respective gene classes, and the relation ‘from_species’ to link species-specific classes to the corresponding species classes (e.g., ‘HLA_Class_I_Allele’ to ‘Human’).

Implementation

For the implementation of MaHCO, OWL DL has been chosen as ontology language, because it fulfills the previously stated requirements regarding the expressive power, computability and tool support. The top level of MaHCO was implemented manually using the ontology editor Protégé. The species-specific subbranches of the class hierarchy were implemented by programming using the Jena framework.

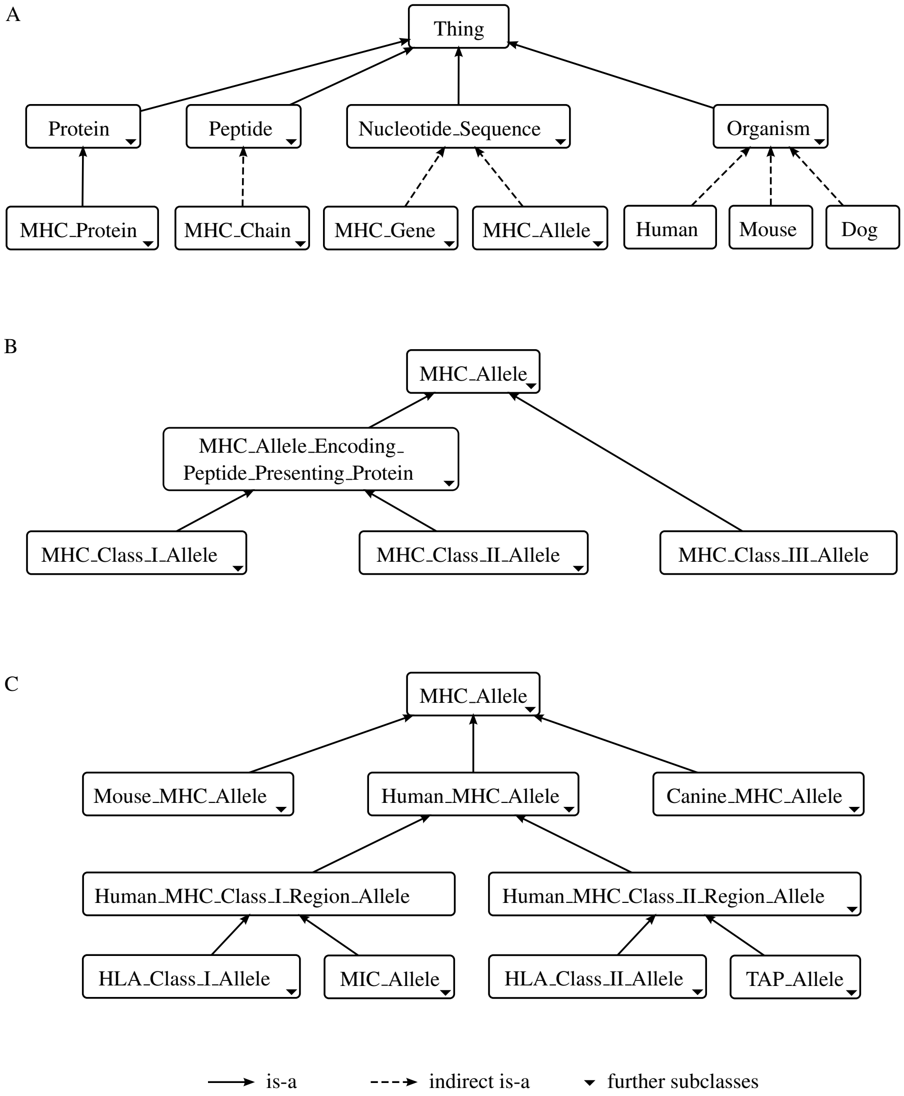


Figure 7.1: A: Top level of MaHCO. B: Separation of MHC alleles encoding peptide presenting proteins. C: Classification of human MHC alleles according to MHC regions.

Since the vast majority of species-specific classes in MaHCO concerns the HLA system and was automatically derived from the IMGT/HLA database, they were put in a separate ontology, mainly for maintenance reasons. The HLA ontology (“MaHCO HLA”) is imported by the core ontology (“MaHCO core”) by means of an ‘owl:imports’ statement in the header of the ontology document. Below, “MaHCO” refers to the entire MaHCO, i.e., MaHCO core importing MaHCO HLA.

Conceptual relations of MaHCO were manually implemented as OWL object properties. All but one relation were provided with domain and range restrictions. The relation ‘from_species’ was additionally specified as being functional. The relations were used to create formal class definitions that specify the meaning of classes by making relationships to other classes explicit. An example is the full formal definition

$$\text{‘Human\_MHC\_Allele’} \equiv \text{‘MHC\_Allele’} \sqcap \exists\text{‘from\_species’}.\text{‘Human’}.$$

It defines the class ‘Human_MHC_Allele’ as the intersection of ‘MHC_Allele’ and the (anonymous) class of individuals that are linked by a ‘from_species’ relationship to at least one individual of the class ‘Human’.

Each MaHCO class, except for the root classes ‘Protein’, ‘Polypeptide’, ‘Nucleotide_Sequence’ and ‘Organism’ has been explicitly linked to at least one superclass or was provided with a formal definition that allows a reasoner to classify it as subclass of at least one other MaHCO class. Furthermore, some MaHCO classes have been linked by ‘disjoint-with’ relationships. For example, the classes ‘MHC_Class_I_Allele’, ‘MHC_Class_II_Allele’ and ‘MHC_Class_III_Allele’ were specified as being mutually disjoint.

The following URI policy for MaHCO has been specified: Classes and conceptual relations of MaHCO should be provided with URIs that consist of the MaHCO core namespace “`http://purl.org/stemnet/MHC#`”, abbreviated “`MHC`” or the MaHCO HLA namespace “`http://purl.org/stemnet/HLA#`”, abbreviated “`HLA`”, and a local name that is unique in the respective namespace. Local names should start with an upper case letter and use underscores as word delimiter. Furthermore, the following annotation guidelines have been specified for MaHCO: Each class should be provided with a preferred class label, represented as a ‘rdfs:label’ annotation, and a verbal definition, represented as ‘mhc:definition’ annotation. Classes may additionally be provided with alternative class labels, represented as ‘mhc:synonym’ annotations. Ontology elements derived from external knowledge resources should be provided with ‘mhc:reference’ annotations that reference their origin. For example, the ‘mhc:reference’ annotation “SO:0001023 allele” of the MaHCO class ‘Allele’ indicates that the specified MaHCO class was derived from the SO class ‘SO:0001023’.

7.3 Evaluation

MaHCO 1.0.1 was checked for logical consistency running the OWL reasoner Pellet [Sirin et al., 2007]. No unsatisfiable classes were detected.

To evaluate the guideline compliance of the current version MaHCO 1.0.1, the previously proposed three-step procedure was applied (see section 4.5.3). In the first step, the adherence of MaHCO to selected design and implementation guidelines was checked, answering 30 guideline control questions. Questions Q05 (Is each class provided with at most one superclass in the asserted class hierarchy?), Q06 (Are classes with only one direct subclass avoided in the inferred class hierarchy?) and Q7 (Are classes with many direct subclasses avoided in the inferred class hierarchy?) revealed on the class and taxonomy level that in the asserted class hierarchy 2,359 classes of MaHCO (30%) have more than one direct superclass, and in the inferred class hierarchy 120 classes (2%, or 13% of all non-terminal classes) have only one direct *sub*class, whereas 197 classes (2%) have overly many subclasses, based on a threshold of 12.

Questions Q09 (Is the disjointness of classes explicitly specified?) and Q10 (Are domain, range and further properties of relations specified?) revealed on the level of formal semantics that for 127 class pairs of MaHCO core explicit ‘disjoint-with’ relationships have been specified, and for four conceptual relations (out of six that are used in at least one formal class definition) no domain and for two no range were specified. Only one conceptual relation has been specified as being functional. Questions Q11 (Are formal class definitions provided?) and Q13 (Are existential restrictions complemented with universal restrictions?) revealed that for only 16 classes of MaHCO (less than 1%) full formal class definitions and for none of the existential restrictions closure axioms were specified.

Questions Q16 (Are compulsory annotations provided?) and Q17 (Are optional annotations provided?) revealed on the annotation level that six classes of MaHCO lack a class label, and only very few classes have been provided with a verbal definition and optional alternative class labels. Questions Q19 (Is each annotation property used for one type of annotation only?) and Q20 (Is each type of annotation represented using one annotation property only?) revealed that by mistake three alternative class labels were represented as ‘rdfs:label’ instead of ‘mhc:synonym’ annotations and two verbal definitions as ‘rdfs:comment’ instead of ‘mhc:definition’ annotations. Question Q21 (Do class labels follow approved naming conventions?) revealed that MaHCO deviates from established naming conventions in two respects. It provides class labels that contain acronyms (e.g., “MHC” or “HLA”), and 99% of all distinct class labels start with a capital letter. Question Q22 (Are duplicate class labels and duplicate verbal definitions avoided?) revealed that 66% of the distinct class labels occur more than once in MaHCO. Question Q23 (Are verbal definitions non-circular?) revealed for 50% of the verbal definitions of MaHCO classes a circularity index value greater than zero [see

Köhler et al., 2006]. However, further scrutiny would be required to confirm or reject circularity concerns.

Question Q24 (Do local names adhere to a consistent naming policy?) revealed on a technical level that a few local class names in MaHCO (less than 1%) contradict the MaHCO URI policy. They either contain a hyphen as word delimiter, use CamelCase notation, or contain a word that starts with a lower case letter. Question Q25 (Are redundant ontology elements avoided?) revealed that the annotation properties 'mhc:synonym', 'mhc:definition' and 'mhc:reference' have mistakenly been specified also as data relations. In addition, the conceptual relation 'encodes' has mistakenly been defined twice, once in MaHCO core, and once in MaHCO HLA. Questions Q26 (Are unused object, datatype and annotation properties avoided?) and Q27 (Are artifacts from tool-use avoided?) revealed that three data relations and four annotation properties in MaHCO are currently "unused", i.e., not involved in any formal class definition or annotation, and two annotations are empty. Question Q30 (Are reused contents up to date and are their sources referenced?) revealed regarding the aspects knowledge reuse and usability that only few MaHCO classes have been provided with 'mhc:reference' annotations, although the majority of them have been derived from existing knowledge resources, primarily domain-specific databases.

In the second step, the avoidance of common modeling pitfalls of MaHCO was checked using the ontology pitfall scanner Oops!. The tool identified five types of potential errors in MaHCO. It detected 7,935 MaHCO classes and relations that lack an 'rdfs:label' or 'rdfs:comment' annotation as cases of pitfall P08 (missing annotations), five conceptual relations that miss domain or range as cases of pitfall P11 (missing domain or range in properties). Furthermore, it detected four conceptual relations that lack an inverse relation as cases of pitfall P13 (missing inverse relations) and suspected the rather short 'rdfs:comment' annotation "Pseudogene?" of class 'DLA_DQB2_Allele' as case of pitfall P20 (swapping label and comment). Finally, it detected that different naming criteria have been used for local names of classes and relations in MaHCO, a case of pitfall P22 (using different naming criteria in the ontology).

In the third step, the adherence of MaHCO to accepted OBO Foundry principles was checked. The checks revealed that two principles are not adhered to by MaHCO, and five principles are only partially adhered two. MaHCO has mainly been used by StemNet members. Hence, it does not adhere to principle FP09 (The ontology has a plurality of mutually independent users.). Furthermore, maintenance of MaHCO has been terminated in 2009, when the StemNet project ended. Hence, it does not adhere to principle FP16 (The ontology should be continuously maintained.). The contents of MaHCO are clearly specified. However, by design it overlaps to some extent with other OBO ontologies. In addition, it misses explicit links to OBO Foundry ontologies, indicating a partial adherence to principle FP05 (The ontology has a clearly specified and delineated content, provides coherent verbal definitions of top level classes and incorporates explicit links to other OBO Foundry ontologies.). Furthermore, only few MaHCO classes are provided with a verbal definition, denoting a limited adherence to principle FP06 (The ontology includes verbal definitions for classes.). The ‘part_of’ relation of MaHCO has been derived from RO. However, it is used less strictly than in RO (e.g., it is not specified as being reflexive, anti-symmetric, and transitive as in RO), indicating a partial adherence to principle FP07 (The ontology uses relations in the style of RO.). For some time, MaHCO has been developed in active collaboration with developers of SO (for example, David S. DeLuca, a co-author of MaHCO, was involved in SO Immunology Workshop focusing on the representation of immunological features, genes, alleles, and HLA in SO, in June 2007). However, SO is currently an OBO Foundry *candidate* ontology only, and the collaboration has meanwhile been paused, indicating a partial adherence to principle FP10 (The ontology is developed collaboratively with other OBO Foundry members.). Finally, MaHCO complies with the OBO naming conventions apart from two exceptions. Certain class labels contain acronyms, and the vast majority of labels start with a capital letter (see above), indicating a partial adherence to principle FP12 (The ontology adheres to the OBO naming conventions.).

7.4 Documentation and Release

MaHCO has been documented in a proceedings paper [Beisswanger et al., 2007] and a journal article [DeLuca et al., 2009]. Furthermore, meta-data annotations have been specified in the ontology document itself. For MaHCO 1.0.1, two ‘dc:creator’, one ‘dc:publisher’, one ‘dc:title’, one ‘dc:subject’ and one ‘dc:date’ annotation are provided, which specify the names and affiliations of the ontology developers, the title, a brief content description and the release date of the ontology. The current release version MaHCO 1.0.1 in OWL DL is publicly available at the MaHCO website[9]. For MaHCO HLA, also a non-OWL XML representation is provided, and a variant in which allele classes are organized along the two-to-eight digit nomenclature instead of the serological one. While the first one might be practical for bioinformaticians and programmers who strive to avoid OWL-parsing libraries, the second one might be interesting for applications that do not deal with serological HLA typing, which is more and more becoming a legacy technology. MaHCO is additionally available at the NCBO BioPortal, where it can be searched and browsed using BioPortal tools.

7.5 Maintenance

MaHCO has been actively developed and maintained during the run time of the Stem-Net project (2006–2009). To track changes and administer different ontology ver-

[9] `http://www.bioinformatics.org/mahco/` – access date 2012-02-28.

Feature	MaHCO core	MaHCO HLA	MaHCO
Classes	105	7,823	7,928
Conceptual relations used	5	1	6
Defined classes	13%	-	<1%
‘is-a’ relationships	101	10,158	10,270
‘disjoint-with’ relationships	127	-	127
Conceptual relationships	27	3,332	3,360
Annotation properties used	12	3	12
Ontology annotations	7	-	7
Classes with label	94%	100%	100 %
Classes with alternative label	3%	-	<1%
Classes with verbal definition	38%	-	1%
Classes with reference	12%	-	<1%

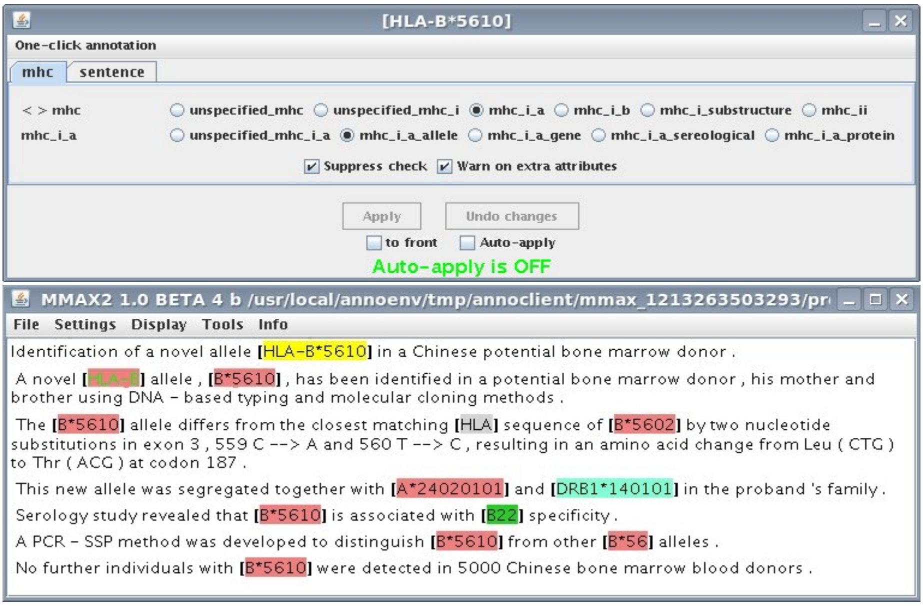
Table 7.4: Statistics on MaHCO 1.0.1. The numbers refer to the asserted class hierarchy of the respective ontology. Percentages were rounded. Relationships were counted as explained on page 26. Only non-empty annotations were counted.

sions, the version control system Subversion was used. Statistics on the latest release, MaHCO 1.0.1, are given in table 7.4.

7.6 Use Cases

Semantic Annotation of Text Documents

In the context of the StemNet project, MaHCO has been used for the semantic annotation of a corpus of Medline abstracts dealing with immunology. More precisely, a subset of MaHCO core classes served as annotation vocabulary for the manual annotation of verbal mentions of MHC class I and II genes, alleles, and gene products, across different species. A screenshot of the annotation environment that was used is given in figure 7.2. In total, about 300,000 words were annotated. Subsequently, they were used as training data for machine learning-based NER tool for the MHC. After training, the tool should be able to automatically recognize verbal mentions of MHC genes, alleles and proteins. For example, given the sentence “Serological study revealed that B*5610 is associated with B22 specificity.” from the Medline abstract with PMID 12694576, it would be expected to recognize the text strings “B*5610” and “B22” as mentions of an MHC class I region allele, and a HLA serological group, respectively. To evaluate the quality of the automatic annotation, around 300 abstracts were manually annotated as gold standard. Evaluated against this gold standard the trained MHC tagger achieved an F-score of 82.8% (83.1% precision, 82.5% recall), which is a good result given the



< > mhc ○ unspecified_mhc ○ unspecified_mhc_i ● mhc_i_a ○ mhc_i_b ○ mhc_i_substructure ○ mhc_ii

mhc_i_a ○ unspecified_mhc_i_a ● mhc_i_a_allele ○ mhc_i_a_gene ○ mhc_i_a_sereological ○ mhc_i_a_protein

☑ Suppress check ☑ Warn on extra attributes

□ to front □ Auto-apply

Auto-apply is OFF

MMAX2 1.0 BETA 4 b /usr/local/annoenv/tmp/annoclient/mmax_1213263503293/pr
File
Settings
Display
Tools
Info

Identification of a novel allele [HLA-B*5610] in a Chinese potential bone marrow donor .

A novel [HLA-B] allele , [B*5610] , has been identified in a potential bone marrow donor , his mother and brother using DNA - based typing and molecular cloning methods .

The [B*5610] allele differs from the closest matching [HLA] sequence of [B*5602] by two nucleotide substitutions in exon 3 , 559 C --> A and 560 T --> C , resulting in an amino acid change from Leu (CTG) to Thr (ACG) at codon 187 .

This new allele was segregated together with [A*24020101] and [DRB1*140101] in the proband 's family .

Serology study revealed that [B*5610] is associated with [B22] specificity .

A PCR - SSP method was developed to distinguish [B*5610] from other [B*56] alleles .

No further individuals with [B*5610] were detected in 5000 Chinese bone marrow blood donors .

Figure 7.2: Semantic annotation of text documents using a MaHCO-based annotation vocabulary. The annotation of the text span "HLA-B*5610" as MHC class Ia allele is shown.

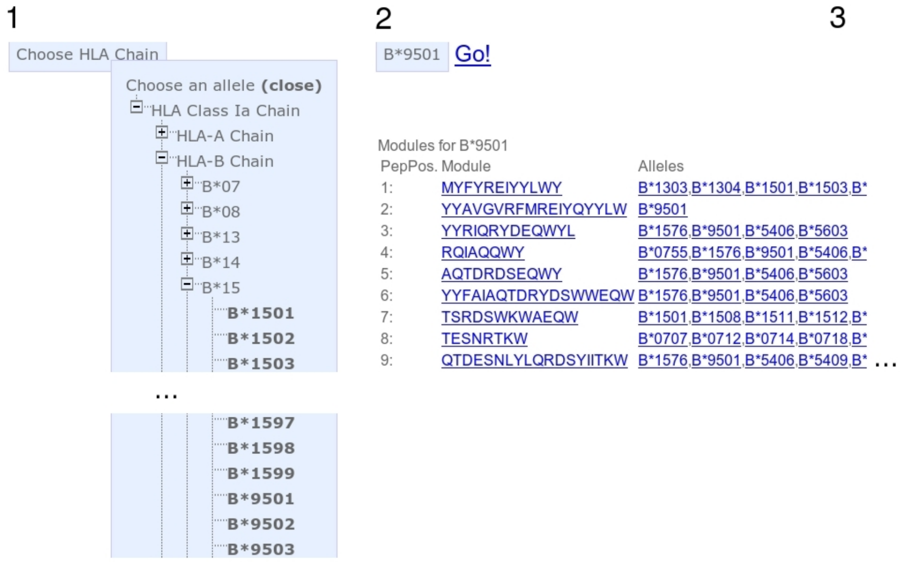
state-of-the-art performance of biomedical NER tools at that time [Hirschman et al., 2007], and the inherent complexity of the MHC annotation and recognition task.

Browsing of HLA Data

MaHCO HLA has further been used as conceptual backbone for Web-based tools for computational immunology, which have been developed at the Institute for Transfusion Medicine at Hannover Medical School (MHH), in the context of the StemNet project. An example is the "HLA Module Explorer". It is a component of the MHH's PeptideCheck tool[10], dedicated to HLA-peptide binding prediction. The latter plays an important role in finding donor/recipient matches for HSCTs, through which the desired GvL effect is achieved, without the development of GvHD. The HLA Module Explorer allows to analyze and compare the peptide binding characteristics of HLA alleles on the level of peptide modules (figure 7.3, step 3).

Starting the HLA Module Explorer, a Web-based dialog appears that allows the user to select the HLA allele of interest by browsing the MaHCO HLA allele hierarchy

[10] http://www.peptidecheck.org/ – access date 2012-02-29.



Modules for B*9501

PepPos.	Module	Alleles
1:	MYFYREIYYLWY	B*1303,B*1304,B*1501,B*1503,B*
2:	YYAVGVRFMREIYQYYLW	B*9501
3:	YYRIQRYDEQWYL	B*1576,B*9501,B*5406,B*5603
4:	RQIAQQWY	B*0755,B*1576,B*9501,B*5406,B*
5:	AQTDRDSEQWY	B*1576,B*9501,B*5406,B*5603
6:	YYFAIAQTDRYDSWWEQW	B*1576,B*9501,B*5406,B*5603
7:	TSRDSWKWAEQW	B*1501,B*1508,B*1511,B*1512,B*
8:	TESNRTKW	B*0707,B*0712,B*0714,B*0718,B*
9:	QTDESNLYLQRDSYIITKW	B*1576,B*9501,B*5406,B*5409,B* ...

Figure 7.3: User interface of the HLA Module Explorer. In step 1, the user clicks the “Choose HLA Chain” button to reveal a MaHCO HLA-based tree structure in which subclasses can be opened. In step 2, the user has selected the desired HLA chain. Clicking the “Go!” button reveals in step 3 the modules of the chosen MHC chain and alleles that encode MHC chains that share these modules.

(figure 7.3, step 1). Obviously, this is much more convenient than selecting the respective allele from a long flat list of alphabetically sorted allele names. In principle, basing the dialog on a hierarchical representation of HLA alleles even allows the user to select and perform actions on whole groups of alleles. Another advantage results from using an *ontology*-driven dialog, compared to one relying on a hierarchical representation of alleles that simply copies the HLA nomenclature. If, for example, the user selects “B*15” in the ontology-driven dialog, alleles of which the names start with B*15 or B*95 will be selected. The reason is that in MaHCO HLA both are encoded as belonging to B*15, because alleles are strictly classified according to their characteristics, while auxiliary rollover groups, such as B*95 (see page 110), are dispensed with. Finally, the browsing-based approach has also performance benefits with respect to the time it takes to load a website. With each click in the dialog, only the relevant information is loaded by communicating with the server in the background. This saves significant lag time compared to loading the entire ontology into the browser, especially when dealing with large ontologies. The MaHCO HLA-based dialog is available at the previously mentioned PeptideCheck website at MHH, following the links “PeptideCheck” and “HLA Module Explorer”.

Chapter 8

BioTop

BioTop is a top domain ontology for molecular biology and biomedicine. Its origin as a redesign of an existing top domain ontology has been described in Schulz et al. [2006a,b]. BioTop as potential semantic top level for different biomedical ontologies has been presented in Beisswanger et al. [2008c]. This chapter is focused on the life cycle of BioTop, including the evaluation of the latter.

8.1 Requirements Analysis

The development of BioTop started in the context of the BOOTStrep project (see page 87). As mentioned above, a major goal of BOOTStrep was the creation of a three-layered biomedical knowledge repository, intended to support advanced biomedical NLP. The repository should consist of a bio-lexicon, a bio-ontology and a bio-fact store. As top level for the bio-ontology and semantic umbrella that integrates the three layers of the knowledge repository and enables their interlinkage with external knowledge resources a semantic top level was required. It should represent foundational classes and relations of molecular biology and biomedicine and allow to bridge conceptual gaps between existing knowledge resources. The meaning of the classes contained should be specified in terms of formal definitions, which allow to reason on them, as well as verbal definitions, intended for human ontology users. The top level should be processable by established ontology editors and reasoners. The results of the

Aspect	Description of requirements
Purpose	formal semantic top level for the integration of heterogeneous biomedical knowledge resources
Domain	molecular biology, biomedicine
Coverage and granularity	basic classes and relation types (physical and non-physical continuants and processes and relationships between them)
Expressive power and computational demands	rich formal class definitions, automatic classification and consistency checking
User group	humans and computers
Tool support	ontology editor, reasoner

Table 8.1: Results of the requirements analysis of BioTop.

requirements analysis are summarized in table 8.1.

Related Knowledge Resources

After compiling these requirements, existing top domain ontologies for biology and biomedicine and related knowledge resources were checked whether they already fulfill these requirements. In contrast to biomedical domain ontologies, only few top domain ontologies for biology and biomedicine have been proposed so far. The Ontology of Biomedical Reality (OBR) by Rosse et al. [2005] is a framework that has been introduced to integrate domain ontologies from anatomy, physiology and pathology, applying principles from the domain-independent top level ontology BFO to the field of medicine and biomedicine. Furthermore, the Simple Bio Upper Ontology (SBUO) by Rector et al. [2006a] provides basic distinctions in terms of foundational relations and classes expressing constraints on their use, where most classes represent continuants (occurrents are represented in terms of the generic top classes 'Ocurrent_entity' and 'Biological_physical_process' only). SBUO is intended to support the creation of more specialized ontologies. OBR and SBUO come with a focus on the medical domain.

In contrast, GFO-Bio is a top domain ontology for biology that is based on the top level ontology GFO [Hoehndorf et al., 2008]. At the time it was examined, it was still under construction. However, meanwhile it has matured and is now used in support of a semantic Wiki, amongst others [Hoehndorf et al., 2008]. The GENIA ontology is another top domain ontology for biology [Kim et al., 2003]. It is a pure taxonomy composed of classes that represent biochemical substances and their natural locations. It has been created as annotation vocabulary for the GENIA term corpus [Kim et al., 2003]. The latter constitutes a collection of semantically annotated biomedical documents that has become a de facto standard for biomedical NLP applications that target information extraction and text mining. Finally, there is the UMLS Semantic Network [McCray, 2003], which can be regarded as the archetype of a top domain ontology for biology and biomedicine. Its main purpose is the categorization of concepts in the UMLS Metathesaurus. However, it is lacking formal rigor [Schulze-Kremer et al., 2004].

To sum up, at the time of the examination, none of the examined resources satisfied all stated requirements. They either lacked formal rigor (such as the UMLS Semantic Network and the GENIA ontology), their coverage was limited in certain respects (e.g., the GENIA ontology described continuants only and SBUO focused on continuants, representing occurrents in terms of generic top classes only) or they were still in a premature or proposal state (such as the OBR and GFO-Bio). Hence, the decision was taken to create BioTop as a new top domain ontology. However, instead of building it from scratch, the decision was taken to redesign, adapt and extend an existing resource, until it fully complies with the stated requirements. Due to its relevance for biomedical NLP, the GENIA ontology was chosen for this purpose, in the version distributed with version 3.01 of the GENIA term corpus.

In the first step of the redesign process, the GENIA ontology was further analyzed. The analysis revealed that it is not only rather small but also quite fragmentary. In addition, it lacks formal rigor and proper natural language annotations, leaving the intended meaning of GENIA classes poorly specified, both, on the logic and the natural language level. For example, no commitment to a formal top level ontology is made by the GENIA ontology, resulting in the lack of a clear ontological structure. Furthermore, no formal class definitions and non-taxonomic relationships between classes are stated explicitly. All classes are provided with a class label and many classes are informally described by verbal scope notes. However, certain labels contain non-standard terminology that may lead to confusion. In addition, not for every class a scope note is provided, and many existing scope notes are in fact incomplete [Schulz et al., 2006a]. As outlined below and described in more detail in Schulz et al. [2006a] and Schulz et al. [2006b], subsequent steps of the redesign process target at the avoidance of these shortcomings in BioTop. Generally, redesign decisions were taken with care, knowing that they might influence knowledge resources that would subsequently be attached to BioTop as semantic top level.

8.2 Design and Implementation

Knowledge Acquisition

BioTop originated as a redesign of the GENIA ontology. Hence, compiling contents for subsequent ontology construction for BioTop primarily meant to analyze and revise classes from the GENIA ontology. They were complemented with upper level classes from the top level ontology BFO [Grenon et al., 2004], conceptual relations from RO [Smith et al., 2005] and elementary domain-specific classes from different biomedical domain ontologies available in the OBO library (e.g., GO, CL and ChEBI, see page 18).

Conceptualization

The redesign of the original GENIA ontology and its integration with the top level ontology BFO was carried out manually by a team of domain experts and knowledge engineers. Underspecified classes of the GENIA ontology were refined, problematic classes redefined or removed, and the class hierarchy was rearranged and based on strictly defined 'is-a' relationships [Schulz et al., 2006a]. In addition, new general and basic domain-specific classes were added, and even whole new ontology branches were introduced [Schulz et al., 2006a]. Instead of reusing the top level distinction between 'Source' and 'Substance' of the GENIA ontology, BioTop has been integrated with the top level ontology BFO. Amongst others, 'Entity', 'Function', 'Role', 'Quality' and 'Process' have been introduced as direct interface classes to BFO (see figure 8.1 A). Furthermore, basic domain-specific classes were added as interfaces to existing biomedical ontologies. Examples include the classes 'BiologicalProcess', 'MolecularFunction' and 'CellularComponent' (see figure 8.1 A) that constitute interfaces to GO. The majority of BioTop classes represent material entities. However, in contrast to the GENIA ontology, the material entity branch of BioTop additionally covers collectives (see figure 8.1 B), i.e., entities that are defined in terms of their constituting uniform entities. An example for a collective is a population of organisms. The material entity branch is further complemented by branches describing dependent continuants and processes (see figure 8.1 A). The dependent continuant branch in turn is further subdivided into subbranches covering quality classes (such as 'PhysicalMass'), function classes (such as 'MolecularFunction') and role classes (such as 'DrugRole'), see figure 8.1 A. Some BioTop classes have more than one superclass. Hence, in contrast to the GENIA ontologies that represents a tree, BioTop constitutes a DAG.

Subsequently, conceptual relations were introduced as a basis for non-taxonomic relationships between classes (see table 8.2). Most of them were adopted from RO. Examples include the relation 'hasPart' with subrelation 'hasProperPart', both with inverse

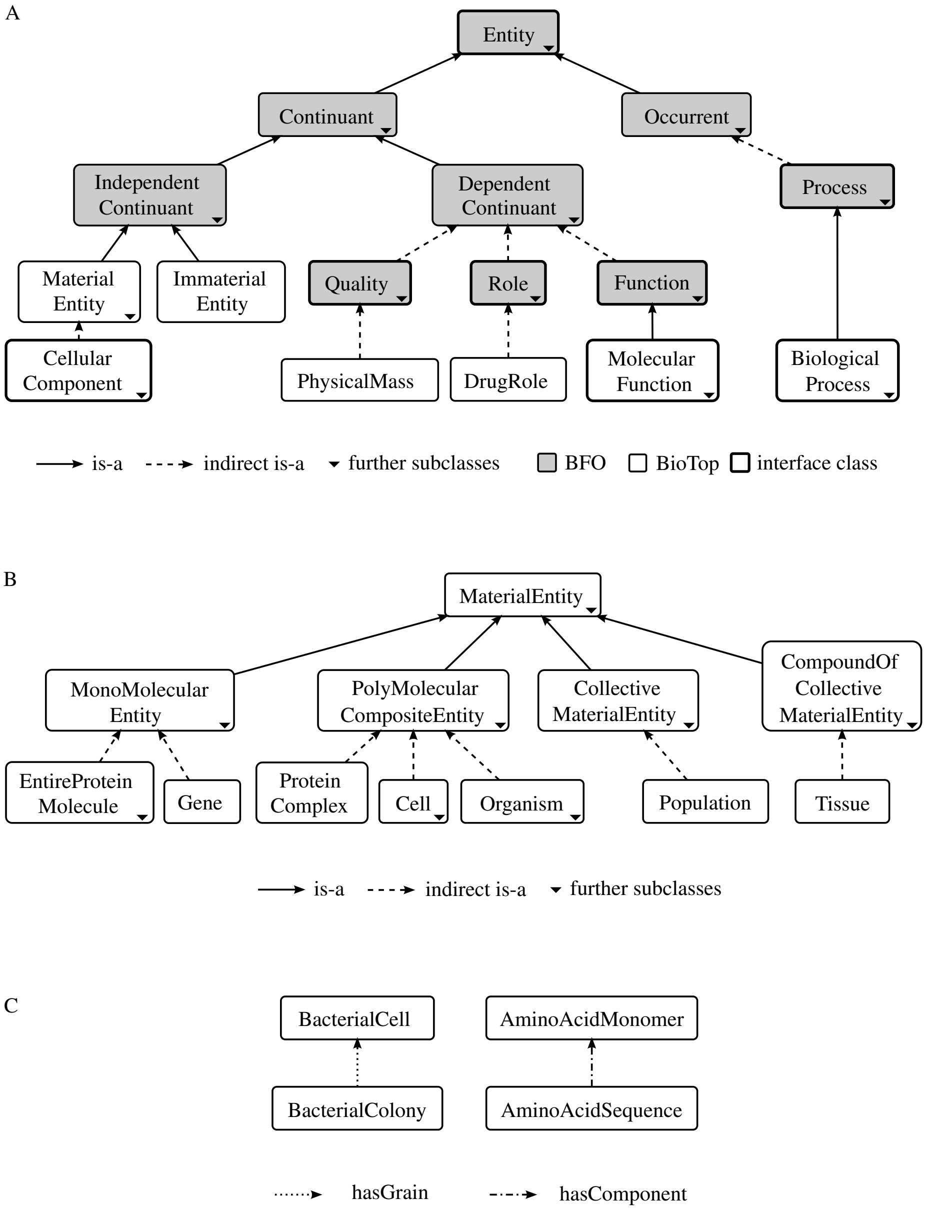


Figure 8.1: A: BFO-based top level of BioTop. Interface classes to BFO and GO are framed in bold. B: Selected subclasses of the BioTop class ‘MaterialEntity’. C: Exemplary relationships of type ‘hasGrain’ and ‘hasComponent’.

Relation	Properties	In RO
‘hasPart’ (‘partOf’)	transitive	yes
‘hasProperPartf’ (‘properPartO’)	transitive	yes
‘hasComponent’ (‘componentOf’)		no
‘hasGrain’		no
‘locatedIn’		yes
‘derivesFrom’	transitive	yes
‘hasParticipant’		yes
‘hasAgent’		yes
‘inheresIn’ (‘hasInherence’)		no

Table 8.2: Relation types used in BioTop 2008-02-19. Inverse relations are given in brackets.

relations, the relations ‘locatedIn’ and ‘derivesFrom’, and the relation ‘hasParticipant’ with subrelation ‘hasAgent’. Relations that are not covered by RO were newly created. An example is the relation ‘hasInherence’. It links physical objects and their inherent (biological) functions, qualities or roles. Further examples are the relations ‘hasGrain’ (first proposed by Rector et al. [2006a]) and ‘componentOf’ that were created as non-transitive subrelations of the ‘hasPart’ relation. The relation ‘hasGrain’ has been introduced to enable the definition of collectives as mass entities that are composed of their constituent singletons (such as a bacterial colony is composed of bacterial cells, see figure 8.1 C, left). The relation ‘hasComponent’ has been introduced to enable the definition of compounds as non-overlapping and exhaustive partitions of their components (such as an amino acid sequence is composed of particular amino acid monomers, see figure 8.1 C, right). While collectives remain the same when singletons are added or removed, compounds change their identity when components are added or removed.

Implementation

For the implementation of BioTop, OWL DL was chosen as ontology language as a compromise regarding the expressive power of BioTop, computational demands and required tool support (a more expressive ontology language, such as full first order logic, would have allowed for richer formal class definitions, though at the expense of computability). The implementation of BioTop was carried out manually, using the ontology editor Protégé. First, the class hierarchy was implemented. Next, RO was imported to make conceptual relations available in BioTop. The relations ‘derivesFrom’, ‘hasPart’ and ‘hasProperPart’ with inverse relations have been specified as being transitive. Below, “BioTop” refers to BioTop from February 19, 2008 (henceforth abbreviated “BioTop 2008-02-19”), if not otherwise specified.

The following URI policy has been specified for BioTop: Classes and conceptual relations should be provided with URIs that consist of the namespace "`http://purl.org/biotop/core/dev#`", abbreviated "`biotop`", and a local name that is unique in BioTop. Local names should start with an upper case letter and use CamelCase notation.

To specify the meaning of BioTop classes on the natural language level, classes were provided with a class label, optional alternative class labels and a verbal definition. To specify the meaning of classes also on the logic level and enable restrictive consistency checks, formal class definitions, closure axioms for conceptual relationships and 'disjoint-with' relationships between classes were established. For example, the class 'Deoxyribonucleotide' has been provided with the formal definition

$$\begin{aligned}&\text{`Deoxyribonucleotide'} \equiv \exists\text{`hasComponent'}.\text{`Deoxyribose'}\\ &\quad\sqcap \exists\text{`hasComponent'}.\text{`HeterocyclicBase'}\\ &\quad\sqcap \exists\text{`hasComponent'}.\text{`Phosphate'}\\ &\quad\sqcap \forall\text{`hasComponent'}.(\text{`Deoxyribose'} \sqcup \text{`HeterocyclicBase'} \sqcup \text{`Phosphate'}),\end{aligned}$$

specifying its relationship to further BioTop classes. Closure axioms have been specified, for example, for the 'hasGrain' relationship between the classes 'BacterialColony' and 'BacterialCell' and the 'hasComponent' relationship between the classes 'AminoAcidSequence' and 'AminoAcidMonomer'. The formal definition of the class 'BacterialColony' is

$$\begin{aligned}&\text{`BacterialColony'} \equiv \text{`BiologicalColony'} \sqcap \exists\text{`hasGrain'}.\text{`BacterialCell'}\\ &\quad\sqcap \forall\text{`hasGrain'}.\text{`BacterialCell'}\end{aligned}$$

and that of the class 'AminoAcidSequence'

$$\begin{aligned}&\text{`AminoAcidSequence'} \equiv \exists\text{`hasComponent'}.\text{`AminoAcidMonomer'}\\ &\quad\sqcap \forall\text{`hasComponent'}.\text{`AminoAcidMonomer'}.\end{aligned}$$

Class pairs linked by a 'disjoint-with' relationship include the top level classes 'Continuant' and 'Occurrent' and the domain-specific classes 'OrganicMolecularEntity' and 'InorganicMolecularEntity', and 'DNA' and 'RNA', amongst others.

The integration of BioTop with the top level ontology BFO has been implemented in terms of a "bridge" ontology. It includes 'owl:imports' statements in the header of the ontology document that trigger the import of BioTop and BFO. It further contains relationship specifications that link BFO classes to appropriate interface classes of BioTop, establishing a seamless integration of the two resources. For example, 'equivalent-to' relationships are specified that link the BFO class 'Process' to the BioTop class 'Process', and the BFO class 'Role' to the BioTop class 'Role'.

8.3 Evaluation

During the development of BioTop, across different versions, it was repeatedly classified and checked for logical consistency using different OWL reasoners. Each unsatisfiable class was fixed before the development was continued. BioTop 2008-02-19 has again been checked for logical consistency running the OWL reasoner HermiT. No unsatisfiable classes were found.

To evaluate the compliance of BioTop 2008-02-19 with established ontology design and implementation guidelines, the previously proposed three-step procedure was applied (see section 4.5.3). In the first step, the adherence of BioTop to selected design and implementation guidelines was checked, answering the 30 control questions presented in table 4.2. Questions Q05 (Is each class provided with at most one superclass in the asserted class hierarchy?), Q06 (Are classes with only one direct subclass avoided in the inferred class hierarchy?) and Q7 (Are classes with many direct subclasses avoided in the inferred class hierarchy?) revealed on the class and taxonomy level that in the asserted class hierarchy two BioTop classes have more than one direct superclass, and in the inferred class hierarchy 18 BioTop classes (10%, or 28% of all non-terminal classes) have only one direct *sub*class, while one class has overly many subclasses, applying a threshold of 12.

Questions Q09 (Is the disjointness of classes explicitly specified?) and Q10 (Are domain, range and further properties of relations specified?) revealed on the level of formal semantics that for 102 class pairs in BioTop explicit 'disjoint-with' relationships have been specified. Furthermore, for none of the conceptual relations involved in at least one full or partial formal class definition in BioTop a domain or range and for seven of them no further properties (e.g., being functional or transitive) have been specified. Questions Q11 (Are formal class definitions provided?), Q13 (Are existential restrictions complemented with universal restrictions?) and Q14 (Are universal restrictions complemented with existential restrictions?) revealed that for 45 BioTop classes (26%) full formal class definitions have been specified, for half of the existential restrictions in BioTop closure axioms have been provided and for about 70% of the universal restrictions complementing existential restrictions.

Questions Q16 (Are compulsory annotations provided?) and Q17 (Are optional annotations provided?) revealed on the annotation level that 7% of the BioTop classes miss a class label and even more classes a verbal definition and optional alternative class labels. However, no precise numbers can be given for the latter two, since BioTop lacks a clear policy for their representation. Taking all 'rdfs:comment' annotations as verbal definitions that either contain the prefix "Definition:" or lack a prefix, 76% of the classes in BioTop are verbally defined. Furthermore, considering all additional 'rdfs:label' annotations and 'rdfs:comment' annotation with the prefix "Synonym" as alternative class labels, 7% of the classes in BioTop come with alternative class labels.

Questions Q19 (Is each annotation property used for one type of annotation only?) and Q20 (Is each type of annotation represented using one annotation property only?) revealed that BioTop lacks strict annotation guidelines. As a consequence, annotation properties are used inconsistently. For example, 'rdfs:label' is used to represent class labels, but also alternative class labels (three cases) and comments (two cases). As another example, 'rdfs:comment' is used to represent verbal definitions, comments, and alternative class labels, amongst others, as the prefixes "Definition:" (48 cases), "Comment:" (19 cases) and "Synonym:" (3 cases) of 'rdfs:comment' annotations indicate. In addition, annotations are represented inconsistently. For example, some alternative class labels are represented as 'rdfs:label' and others as 'rdfs:comment' annotations. Question Q21 (Do class labels follow approved naming conventions?) revealed that BioTop deviates from established naming conventions in at least three ways: 10% of the 169 distinct preferred and alternative class labels contain an acronym or abbreviation, such as "RNA" or "DNA", 6% of the class labels start with a capital letter and very few class labels (e.g., "sign or symptom role") contain a conjunction. Question Q23 (Are verbal definitions non-circular?) revealed for over 45% of all 'rdfs:comment' annotations, which at this point were taken as verbal definitions, circularity index values greater than zero [see Köhler et al., 2006]. Further scrutiny would be required to confirm or reject circularity concerns.

Question Q24 (Do local names adhere to a consistent naming policy?) revealed on a technical level that four local names of BioTop classes do not follow the URI policy specified for BioTop. Three of them start with a lower case letter and one contains a hyphen as token delimiter. Questions Q26 (Are unused object, datatype and annotation properties avoided?) and Q27 (Are artifacts from tool-use (e.g., empty annotations) avoided?) revealed that four conceptual relations and one annotation properties are currently "unused", i.e., not involved in any formal class definition or annotation, and four annotations are empty.

In the second step, the avoidance of BioTop 2008-02-19 of common modeling pitfalls was checked using the ontology pitfall scanner Oops!. The tool identified seven types of potential errors in BioTop. It identified the class 'Entity' as case of pitfall P04 (creating unconnected ontology elements) and the classes 'SignOrSymptomRole' and 'OligoOrPolymer' as cases of pitfall P07 (merging different concepts in the same class). Furthermore, it identified 47 BioTop classes and conceptual relations that lack an 'rdfs:label' or 'rdfs:comment' as cases of pitfall P08 (missing annotations), and 32 conceptual relations that miss domain or range and an inverse relation as cases of pitfall P11 (missing domain or range in properties) and pitfall P13 (missing inverse relations). In addition, the tool identified the rather short 'rdfs:comment' annotations "Synonym: AminoAcidSequence" of the class 'AminoAcidSequence' and "(OBI 295)" of the class 'NonRealizableInformationEntity' as cases of pitfall P20 (swapping label and comment), and a recursive partial formal definition of the class 'Cell' as case of pitfall P24 (using recursive definition).

In the third step, the adherence of BioTop 2008-02-19 to accepted OBO Foundry principles was checked. The checks revealed that one of the principles is not adhered to by BioTop and another three are only partially adhered to. BioTop is mainly used by members of the projects that funded its development. Hence, it does not adhere to the principles FP09 (The ontology has a plurality of mutually independent users.). The contents of BioTop are clearly specified and most top level classes are provided with appropriate verbal definitions. However, as a top domain ontology BioTop *per se* does not fit to the principle of "orthogonality" with domain ontologies, and it also does not provide explicit links to them, indicating a partial adherence to FP05 (The ontology has a clearly specified and delineated content, provides coherent verbal definitions of top level classes and incorporates explicit links to other OBO Foundry ontologies.). Furthermore, approximately three-fourth of the BioTop classes are provided with a verbal definition, indicating a large but not complete agreement with principle FP06 (The ontology includes verbal definitions for classes.). Finally, BioTop has been developed and is further improved and maintained in close collaboration with the OBO community. However, the collaboration is rather focused on the intended top level of the OBO ontologies, BFO, indicating a limited agreement with principle FP10 (The ontology is developed collaboratively with other OBO Foundry members.).

8.4 Documentation and Release

Early versions of BioTop have been documented in the proceedings papers Schulz et al. [2006b] and Schulz et al. [2006a] and BioTop 2008-02-19 in the journal article Beisswanger et al. [2008c]. As an alternative form of documentation, meta-data annotations have been specified in the ontology document describing BioTop 2008-02-19. A 'dc:title', a 'dc:identifier', a 'dc:language', several 'dc:creator', a 'dc:publisher', several 'dc:source', a 'dc:format' and an 'owl:versionInfo' annotation are provided. They specify the title of the ontology, the URI, the language in which natural language annotations are expressed, the names and affiliations of its developers, articles in which it has been described, the content type of the ontology document itself and the status of the ontology as development version, respectively.

BioTop in its latest version is available at the BioTop website[1], which also provides links to previous release versions of the ontology. BioTop is further available at the NCBO BioPortal.

[1] http://purl.org/biotop/ – access date 2012-02-20.

Feature	BioTop	BioTop-BFO
Classes	175	188
Conceptual relations used	13	13
Defined classes	26 %	26 %
‘is-a’ relationships	142	159
‘disjoint-with’ relationships	102	107
Conceptual relationships	98	98
Annotation properties used	10	10
Ontology annotations	14	14
Classes with label	93 %	94 %
Classes with comment	81 %	82 %

Table 8.3: Statistics on BioTop 2008-02-19 and the bridge ontology integrating it with BFO and RO. The numbers refer to the asserted class hierarchy of the respective ontology. Percentages have been rounded. Relationships were counted as explained on page 26. Only non-empty annotations were counted.

8.5 Maintenance

BioTop has first been developed and initially maintained in the context of the BOOTStrep project. Statistics on BioTop 2008-02-19, a version that has been developed during the runtime of BOOTStrep, are given in table 8.3. After BOOTStrep terminated in spring 2009, BioTop has further been developed and maintained by the BioTop founder and co-author Stefan Schulz and his research group in the context of different research projects, first at the University of Freiburg, Germany, and since 2010 also at the University of Graz, Austria. Technically, BioTop has been developed and is maintained as an open source project. Ontology changes and different ontology versions and variants are administered using a version control system. Furthermore, an issue tracker and a discussion group have been established to report implementation issues and debate topics relating to the theoretical background of BioTop.

BioTop has been created as an experimental ontology to study the design principles of formal semantic top levels for biomedicine and the description of fundamental classes in this field by precise formal definitions. Accordingly, it is subject to continuous change. Below, some of the most important changes are described that BioTop has undergone since the release of version 2008-02-19, which is in the focus of the current case study. To resolve the imbalance of BioTop in favor of chemical entities, shortly after the release of BioTop 2008-02-19, BioTop has been modularized into “BioTop core” and an extension called “ChemTop” [Stenzhorn et al., 2008], to which various BioTop classes representing chemical entities have been moved. While BioTop core has a focus on biomedicine, ChemTop focuses on biochemistry. At the BioTop website, mentioned above, two bridge ontologies are provided that integrate ChemTop with BioTop. The first one links ChemTop to BioTop only. The second one additionally embeds the two ontologies with BFO and RO.

In the context of the modularization, the alignment of BioTop with BFO and RO has been improved and adapted to BioTop core. Issues that arose with regard to granularity-dependent classes are discussed by Schulz et al. [2009b]. In addition, an alternative alignment with the top level ontologies DOLCE and RO has been created. At the BioTop website, bridge ontologies are available that integrate BioTop with the respective top level ontologies. Another major change of BioTop was triggered by the creation of the alignment of BioTop and the UMLS Semantic Network. Amongst others, it required a substantial extension of the original role hierarchy in BioTop and the addition of new process classes that reflect relations of the UMLS Semantic Network [Schulz et al., 2009a]. The alignment is also available at the BioTop website. Finally, in 2011 "BioTop-Lite" was created as a variant of BioTop that contains important top level distinctions only. It is intended for the use in applications for which these top level distinctions are sufficient. BioTop lite is available at the BioTop website, too.

8.6 Use Case

BioTop has been designed as a bridge for non-overlapping domain ontologies (figure 8.2 A), conceptual basis for new domain ontologies and source of formal rigor for light-weight existing domain ontologies (figure 8.2 B). Furthermore, it is intended as a cleansing tool to reveal potential modeling errors in domain ontologies (figure 8.2 C), as well as mistakes in alignments of domain ontologies (figure 8.2 D). A selection of practical applications of BioTop, in which it acted in at least one of the ways mentioned above, is presented below.

Bridging of Biomedical Domain Ontologies

A compilation of more than 100 biomedical domain ontologies is covered by the OBO library (see page 17). The OBO ontologies deal with various subdomains of biomedicine, such as anatomy, cell types, molecular functions, biological processes, molecular sequences, or chemicals of biological interest. Although these subdomains are neighboring or even overlapping, at the time when BioTop was created, the ontologies were rather isolated, lacking any deeper form of conceptual integration. BioTop has been equipped with interface classes to particularly relevant OBO ontologies, such as GO, CL and ChEBI [Beisswanger et al., 2008c]. For example, the three BioTop classes 'BiologicalProcess', 'MolecularFunction' and 'CellularComponent' constitute interfaces to the three branches of GO. These interface classes have been used to bridge the

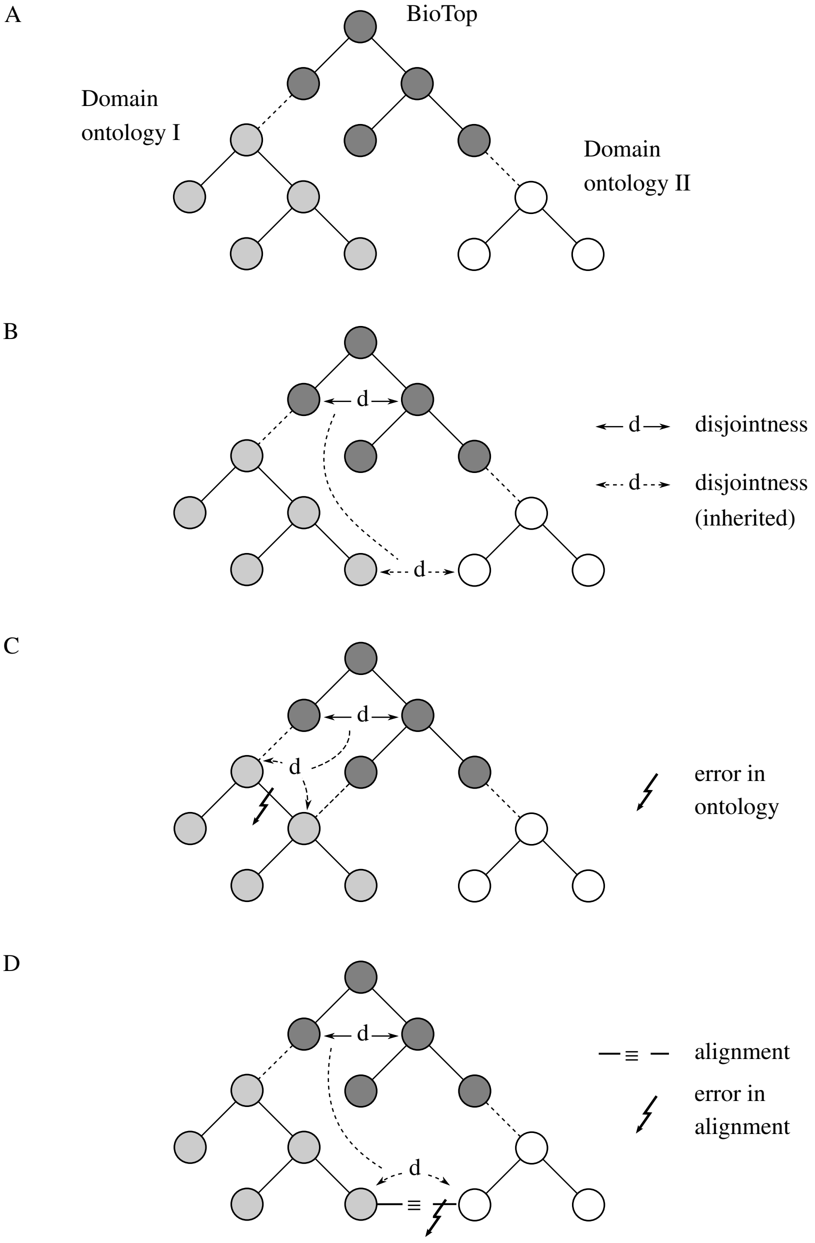


Figure 8.2: Use cases of BioTop. A: Semantic bridge of non-overlapping domain ontologies. B: Source of formal semantics for domain ontologies. C: Cleansing tool for domain ontologies. D: Cleansing tool for ontology alignments.

mentioned ontologies and provide them with access to common semantics. A systematic study on the integration of the OBO ontologies with BioTop is pending, though.

Formalization of the UMLS Semantic Network

The task of attaching domain ontologies to BioTop as common top level can get laborious if a large number of biomedical ontologies is concerned. As an alternative way of establishing a connection between domain-specific knowledge resources and BioTop, BioTop has been linked to the UMLS Semantic Network (SN). As mentioned above, the SN has been designed as semantic upper-level framework for the consistent categorization of UMLS Metathesaurus concepts (see page 41). It consists of a tree of over hundred semantic types, organized in an entity and an event branch, and a hierarchy of semantic relation types with domain and range restrictions that are used to specify non-taxonomic relationships between semantic types. Each individual concept of each of the over hundred Metathesaurus source vocabularies is linked to at least one semantic type in the SN. However, a major shortcoming of the SN is its missing formal rigor. In contrast to BioTop, it is neither founded upon strict ontological principles, nor is it expressed in a formal ontology language. Instead, it suffers from arbitrary divisions, vague and ambiguous descriptions of semantic types and relations and a rather low granularity [Schulze-Kremer et al., 2004]. The integration of BioTop and the SN was realized by creating an alignment between the two resources [see Schulz et al., 2009a].

The alignment allows to capitalize on both the linkage of the SN with domain-specific knowledge resources and the expressiveness and sound structure of BioTop. It covers mappings between semantic types of the SN and BioTop classes, anonymous classes representing restricted BioTop classes and anonymous classes representing the union of several BioTop classes. It further contains mappings between relation types of the SN and BioTop classes representing reified relations. The latter were provided with existential and universal restrictions that represent the original domain and range restrictions of the corresponding SN relation types.

The usefulness of the alignment has been analyzed in two separate evaluation studies. The first study confirmed it as an effective means for the recognition of inconsistent combinations of semantic types used to categorize Metathesaurus concepts. Based on the alignment, 130 out of 400 distinct combinations of at least two jointly assigned semantic types could be rejected as inconsistent, affecting the categorization of over 6,000 UMLS Metathesaurus concepts. The second study revealed that the alignment needs to be formulated even more strictly in order to make it a useful basis for the decision on relationships between Metathesaurus concepts of which verbal mentions co-occur in sentences from Medline abstracts. Independently from the usefulness of the alignment as artifact and resource the matching process itself has proven beneficial. It helped to improve BioTop by identifying errors (e.g., faulty 'disjoint-with' relationships or unrecognized ambiguities) and limitations (e.g., conceptual gaps and missing relationships), which thereupon could be fixed.

Representation of Biological Taxa

BioTop has been used as basis for an ontology that deals with the representation of biological taxa. Many biomedical domain ontologies deal with species-dependent classes. However, so far a standard way of introducing taxon information into ontologies is missing. Schulz et al. [2008] discuss different approaches on how to represent biological taxa in ontologies. Amongst others, they propose a novel approach in which taxon information is taken as quality that inheres in biological organisms, parts of organisms and populations. The approach has been implemented based on BioTop. BioTop is suited for this purpose because it does not only cover material entities, but also abstract independent continuants and dependent continuants. In particular, the BioTop classes 'Organism' and 'Population' (both indirect subclasses of 'MaterialEntity'), 'Quality' (indirect subclass of 'DependentContinuant'), and 'Region' (meanwhile 'ValueRegion', a top level class available in recent versions of BioTop) have been used to attach hierarchies of classes representing organisms, populations of organisms, their taxon qualities and taxon (value) regions [Schulz et al., 2008]. A sample ontology called "taxdemo", which demonstrates how biological taxa can be specified based on BioTop, is available at the BioTop website mentioned above.

Representation of Antibiotics Resistance Patterns

BioTop has further been used in the European DebugIT project[2] as conceptual basis for the DebugIT Core Ontology (DCO), implemented in OWL DL [Schober et al., 2010]. The goal of DebugIT has been to develop a platform to monitor and analyze distributed clinical data to detect fast emerging antibiotic resistances among pathogens and the overuse of antibiotics in European hospitals. The DCO is intended as the center of this platform. It covers basic categories and relations of medicine of infectious diseases, and serves as a hub for the integration and exchange of heterogeneous data streams, ranging from data on pathogens and antibiotics therapies to real patient data.

[2] http://www.debugit.eu/ – access date 2012-02-20.

Chapter 9

The Protein Alignment

The Protein alignment links parts of the protein database UniProtKB (see page 110) with protein-centered portions of the MeSH thesaurus (see page 41). The creation and evaluation of the alignment have already been described in Beisswanger et al. [2010]. This chapter is focused on the life cycle of the Protein alignment.

9.1 Requirements Analysis

The Protein alignment has been created in support of a semantic document retrieval system for the life sciences, called Semedico[1]. Semedico supports faceted search [Hearst, 2006]. A central facet of Semedico is the “Genes and Proteins” facet, which enables the gene and protein-centered retrieval of documents. Currently, it is based on an almost flat list of protein entries derived from the protein sequence database UniProtKB. The facet is operative in “as-is” state. However, the spectrum of search strategies that it supports could be extended by powerful taxonomic strategies, if the protein entries it is based on were more strongly hierarchically organized. To achieve a consistently comprehensive, but stronger hierarchically structured “Genes and Proteins” facet, a knowledge resource would be required that provides fine-grained, but at the same time hierarchically organized protein knowledge. Various existing knowledge resources dealing with proteins were examined in order to find out whether any

[1] `http://www.semedico.org/` – access date 2012-11-28.

of them provide the required knowledge. The results of the examination for exemplary resources, *viz.*, a database, a thesaurus and two ontologies, are presented below.

Related Knowledge Resources

UniProtKB is an authoritative resource for protein sequence data, linked to associated gene data. The curated part of it, UniProtKB/Swiss-Prot, constitutes a comprehensive, high-quality resource that covers over 500,000 protein entries from various species (state of June, 2012). Each entry of UniProtKB/Swiss-Prot is provided with a unique database identifier, a recommended name, optional alternative names, a gene name and gene name synonyms, amongst others. However, the protein entries are represented in tables and lack any explicit form of hierarchical organization.

The MeSH thesaurus is an indexing vocabulary used for the categorization of abstracts in the Medline database (see page 4). MeSH contains about 26,500 entries, organized in 16 hierarchically structured branches (state of June, 2012). The branches rely on thesaurus-style relations (e.g., 'broader' and 'narrower'). Each MeSH entry consists of a "main heading" (also called "descriptor"), a unique identifier and optional "entry" terms, amongst others. The latter are related but not necessarily synonymous to the main heading. MeSH has a broad coverage, ranging from various subdomains of biology and medicine to further health care-related fields. Protein families, groups and complexes are covered in the "Chemicals and Drugs" branch. However, no individual protein classes are represented by MeSH. This is different with an extension of the MeSH, called MeSH Supplementary Concept Records (SCR). It is a separate, database-style resource that mainly deals with chemicals and proteins. Each SCR entry is provided with a unique identifier, a preferred name ("name of substance"), optional alternative names and a link to at least one (usually broader) MeSH entry.

Two ontologies dealing with proteins are IMR and PRO (see page 18). IMR has been designed as a vocabulary for the annotation of molecules in descriptions of signal transduction pathways [Yamamoto et al., 2004]. It consists of a protein and a chemical branch. PRO, in turn, is intended as a formal representation of proteins of various species, covering protein isoforms, variants, modified forms, and recently also complexes [Natale et al., 2011]. For the above-stated purpose, an interesting feature of IMR and PRO is that on the one hand, they cover specific proteins, similar to UniProtKB, while on the other hand they cover protein families or groups, similar to MeSH. However, in terms of coverage, neither of the ontologies reaches UniProtKB/Swiss-Prot. IMR comprises 1,000 classes (state of April, 2012) and PRO contained less than 700 classes when it was first examined. Meanwhile the latter has considerably been extended, up to 28,000 classes (state of April, 2012). However, with 900 classes representing specific human proteins it still lags far behind UniProtKB/Swiss-Prot, which contains 20,000 human protein entries (state of July, 2012). Furthermore, compared to

Aspect	Description of requirements
Input	UniProtKB (RDF version, November 2008) MeSH (2009 release)
Target elements	UniProtKB entries, MeSH headings
Alignment relations	‘related’
Cardinality	many-to-many
Approach	automatic, language-based, customized to input

Table 9.1: Results of the requirements analysis of the Protein alignment.

MeSH, PRO seems to miss certain generic protein classes. For example, it misses a class similar to the MeSH heading “Heat-Shock Proteins”.

Generally, protein databases were found to represent large amounts of specific proteins without providing explicit hierarchical superstructures, while the examined thesauri and ontologies were found to represent fewer specific proteins (or none at all), but therefore cover hierarchically organized groups or families of proteins. Each of the examined knowledge resources qualified to some extent, but none of them fully as basis for the intended hierarchically structured gene and protein facet. Hence, the decision was made to create a new knowledge resource. To avoid redundant work, it should link complementing parts of existing resources instead of being created from scratch. As knowledge resources, UniProtKB was chosen for its coverage and MeSH for its hierarchical representation of protein groups, families and complexes. The two resources were target of integration efforts before [Mottaz et al., 2008]. However, while this previous study dealt with the mapping of mentions of diseases in UniProtKB entries to the MeSH disease terminology, the goal of the Protein alignment project is to map UniProtKB entries themselves to MeSH headings. In contrast to previous integration efforts on protein knowledge, which mostly dealt with the integration of different protein databases (e.g., UniProtKB resulted from the integration of the databases Swiss-Prot, TrEMBL and the Protein Information Resource [Apweiler et al., 2004]), the goal of the Protein alignment project is the integration of database knowledge and parts of a thesaurus.

The linkage of UniProtKB and relevant portions of the MeSH (in versions specified below) required to solve a typical matching task, *viz.*, to find the most closely related MeSH heading (henceforth “MeSH entry”) for each UniProtKB entry, where in some cases there might be none, and in others there might be several most closely related MeSH entries. Since UniProtKB and MeSH denote large knowledge resources that excel in the provision of natural language designators for their entries, an automatic, language-based matching approach was chosen. The results of the requirements analysis of the Protein alignment are summarized in table 9.1.

9.2 Design and Implementation

Input Resources

UniProtKB was downloaded in terms of the UniProtKB RDF release of November, 2008. As input for the alignment, all entries were considered that represent human proteins, are contained in the curated part of of the database, i.e., UniProtKB/Swiss-Prot, and do not contain the phrase “uncharacterized protein” in their recommended name. This resulted in 19,052 UniProtKB entries.

For MeSH, the 2009 release was downloaded, including the extension MeSH SCR. As input for the alignment, all MeSH entries were considered that belong to at least one of the sub-hierarchies labeled “Amino Acids, Peptides, and Proteins”, “Biological Factors”, “Enzymes and Coenzymes”, “Genetic Structures”, “Glycopeptides”, “Glycoproteins”, “Hormones, Hormone Substitutes, and Hormone Antagonists”, “Macromolecular Substances” and “Nucleic Acids, Nucleotides, and Nucleosides”, *viz.*, entries representing proteins and genes, in the broadest sense, though. This resulted in 5,097 MeSH entries. For MeSH SCR, which in the matching procedure was used as intermediate mapping target only, all 183,030 entries were considered.

Extraction of Labels

For each UniProtKB entry, the recommended protein name, optional alternative protein names in their long form and corresponding gene names were gathered as labels. As additional labels, associated enzyme and protein family names were compiled from three additional knowledge resources. For entries provided with an Enzyme Commission (EC) number, additional enzyme names were extracted from the enzyme nomenclature database ENZYME[2]. For example, for the UniProtKB entry ‘EYA2_HUMAN’, which comes with the EC number “EC 3.1.3.48” the enzyme name “Protein-tyrosine-phosphatase” was extracted from the corresponding ENZYME entry ‘EC 3.1.3.48’. Furthermore, for UniProtKB entries provided with “Similarity Annotation” fields, protein family names were extracted using simple regular expressions. For example, for the UniProtKB entry ‘IRAK3_HUMAN’ the protein family names “protein kinase”, “TKL Ser/Thr protein kinase” and “Pelle” were extracted from the annotation “Belongs to the protein kinase superfamily. TKL Ser/Thr protein kinase family. Pelle subfamily.” Finally, for UniProtKB entries linked to protein family entries in InterPro, a database of protein families and domains (see page 92), additional protein family names were extracted from InterPro. For example, for the UniProtKB entry ‘KT81L_HUMAN’

[2] `http://enzyme.expasy.org/` – access date 2012-11-07.

Source	Entries	Distinct labels
UniProtKB human (cleansed)	19,052	90,920
MeSH protein	5,097	47,210
MeSH SCR	183,030	462,673

Table 9.2: Statistics on entries and associated labels considered for the alignment of (parts of) UniProtKB with MeSH.

the family name "Type II keratin" was extracted from the associated InterPro entry 'IPR003054'. In total, 90,920 distinct labels of UniProtKB entries were considered.

For each MeSH entry, the main heading and all associated entry terms were extracted as labels, and for each MeSH SCR entry the so-called name of substance and all given synonyms. In total, 47,210 distinct labels of MeSH entries and 462,673 distinct labels of MeSH SCR entries were considered. The number of entries and associated labels of UniProtKB, MeSH and MeSH SCR considered as input for the PROTEIN alignment project are summarized in table 9.2.

Preprocessing of Labels

To cope with morphological variation, the inflection file "LRAGR" of the UMLS Specialist Lexicon (see page 56) was used. No stemmer was used, because the stemmers tested in a preliminary study turned out to truncate too many domain-specific terms, the names of specific gene and proteins in particular. Each UniProtKB, MeSH and MeSH SCR label was looked up in the inflection file. If a label was found to denote a plural form of a noun, the associated singular form was extracted and added to the label set of the underlying entry. Furthermore, punctuation marks in labels were replaced by spaces, the task-specific stop words "gene", "protein", "family", "member", "domain", and "subunit" were removed, and the labels were lower-cased and tokenized, taking spaces as token boundaries.

A special preprocessing step was applied to labels of MeSH and MeSH SCR entries that contain species specifications, such as the label "IL2 protein, human" of the MeSH SCR entry 'C508594'. To make them compatible with labels of UniProtKB entries, for which species membership is usually specified in terms of an identifier (TaxID) of the NCBI taxonomy (see page 41), provided as value of the field "Taxonomic Identifier", the species specifications had to be removed. To reach this goal, organism names from the NCBI taxonomy were compiled and matched against MeSH and MeSH SCR labels. If a MeSH label was found to contain an organism name as substring, the substring was removed. The corresponding TaxID was kept for the subsequent comparison to TaxIDs associated to UniProtKB entries.

Matching Approach

A two-step matching approach was applied. In the first step, for each UniProtKB entry all extracted protein and gene names were matched against all MeSH and MeSH SCR labels. Each pair of UniProtKB and MeSH labels was tested whether it consisted of the same tokens ignoring the order of tokens. In addition, if a TaxID was associated to the MeSH label it was checked whether it matched the TaxID of the entry to which the UniProtKB label belonged (for the original Protein alignment, restricted to human proteins, this was always TaxID "9606", denoting *Homo sapiens*). If both conditions were fulfilled, the mapping of the underlying pair of entries was accepted. In the second step, for each UniProtKB entry that was not yet successfully mapped, the previously compiled enzyme and protein family names were matched against all MeSH and MeSH SCR labels. Again, if a pair of labels fulfilled the above mentioned conditions, the mapping of the underlying pair of entries was accepted. UniProtKB entries that were still not mapped were marked as "not mapped".

To compile the final alignment, the accepted mappings were further filtered and processed. For UniProtKB entries involved in more than one mapping, the most suitable one was determined and the remaining ones were dropped. For this purpose, a Lucene-based ranking procedure was used that basically relied on a fine-tuned TF-IDF weight [Gospodnetić and Hatcher, 2005, chapter 3.3]. For the ranking, the types of labels that caused the mapping were considered, amongst others. For example, given a particular UniProtKB entry, a mapping that involved a MeSH entry of which a label matched the recommended name of the UniProtKB entry was ranked higher than a mapping that involved a MeSH entry of which a label matched an alternative name of the UniProtKB entry only. Mappings that involved a MeSH SCR entry were replaced by mappings involving a MeSH entry instead, exploiting the existing links between MeSH SCR and MeSH (see page 140). If more than one MeSH entry was linked to a MeSH SCR entry, multiple mappings were created as replacement (these were the only cases, in which multiple mappings per UniProtKB entry were allowed).

For the comparison also a baseline approach was applied. For each UniProtKB entry all protein and gene names were matched to all MeSH labels, based on a Lucene index. From all matching label pairs (if any), the one ranked highest by Lucene was selected and the underlying pair of UniProtKB and MeSH entries was accepted as mapping. To cope with term variants, prior to the matching procedure all protein and gene names and all MeSH labels were Porter-stemmed [Porter, 1980], lower-cased, and punctuation marks were removed.

Matching Results

The first matching step resulted in mappings for 67% of the human protein entries considered, the second step in mappings for additional 11%. Applying the baseline approach resulted in mappings for 70% of the human protein entries considered. Hence, the matching procedure was able to detect related MeSH entries for 78% of the protein entries considered from UniProtKB, while the baseline approach could detect related MeSH entries for 70% only. The results are summarized in table 9.3.

9.3 Evaluation

To evaluate the proposed matching approach, a subset of the automatically generated alignment was compared to a manually created gold standard using evaluation measures adapted to the task of alignment evaluation.

Gold Standard

The gold standard used consists of a sample of 550 UniProtKB entries. They had randomly been selected from the total set of 19,052 entries used as input for the Protein alignment and a domain expert had manually mapped them to at least one most closely related—in the best case equivalent—MeSH entry. Since MeSH is a multi-hierarchy, the expert had mapped some UniProtKB entries to several most closely related MeSH entries, residing in different branches of MeSH. Furthermore, 58 UniProtKB entries (10.5%) had been mapped to the general MeSH entry 'D011506' ("Proteins"). In the evaluation, these entries have then been considered as "not mapped".

Evaluation Measures

For the evaluation relaxed precision and recall measures were used (see formula 4.2 on page 62) and precision-accentuated $F_{0.5}$-scores were computed by setting β to 0.5 in formula 4.4 (page 63). In contrast to classical precision and recall the relaxed variants thereof pay tribute to the fact that mappings to entries of a hierarchical knowledge resource (MeSH) are compared. The relaxed measures are based on overlap proximity (see formula 4.1 on page 62). As proximity function σ a function was chosen that lets pairs of mappings score according to the reciprocal length of the shortest path between their mapping targets in the MeSH hierarchy. Let u_a and u_g denote UniProtKB entries, m_a and m_g MeSH entries, $a = (u_a, m_a)$ a mapping contained in the Protein alignment

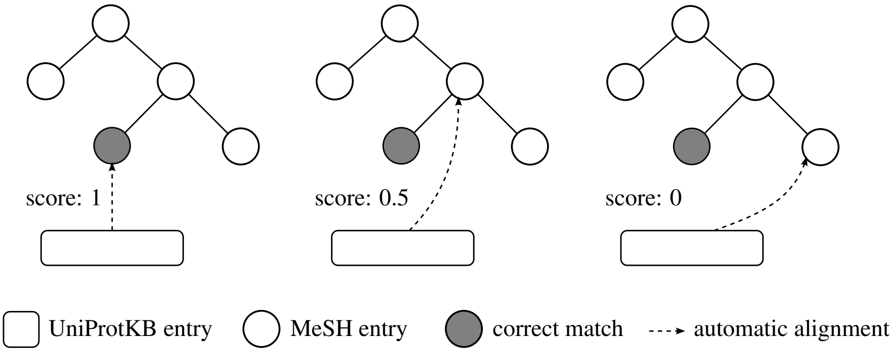


Figure 9.1: Scores achieved by three exemplary automatically generated mappings in a gold standard-based evaluation using the proximity function specified in formula 9.1 (page 146).

and $g = (u_g, m_g)$ a mapping contained in the gold standard alignment. Then the chosen proximity function σ may be specified as

$$\sigma(a,g) := \begin{cases} \frac{1}{p(m_a,m_g)+1} & \text{if } eq(u_a,u_g) \wedge (eq(m_a,m_g) \vee h(m_a,m_g) \vee h(m_g,m_a)) \\ 0 & \text{otherwise,} \end{cases} \tag{9.1}$$

where p denotes the length of the shortest path between two entries in the MeSH hierarchy, eq the ‘equivalent-to’ relation and h the hierarchy-forming relation of MeSH. According to this proximity function a mapping involving exactly the correct MeSH entry scores “1” (as in the case of standard precision and recall), a mapping that involves a MeSH entry that is not the correct one, but is linked to the correct one by a direct or indirect hierarchical relationship in the MeSH hierarchy, scores the reciprocal length of the shortest path between the correct and the identified MeSH entry in the MeSH hierarchy. All remaining mappings score zero.

Figure 9.1 illustrates the scoring logic for three exemplary mappings. On the left, a fully correct mapping is shown (i.e., the automatically identified and the correct mapping targets are identical). It scores “1”. In the middle, a mapping involving a too general mapping target is shown. Since the automatically identified mapping target is directly subordinated to the correct one in the MeSH hierarchy, the mapping scores “0.5”. On the right, a mapping is shown whose identified mapping target is neither identical with the correct one, nor is it linked by a hierarchical relationship to the correct one in the MeSH hierarchy. Accordingly, it scores “0”.

Procedure	Matches (%)		Precision	Recall	$F_{0.5}$-score
Step 1	12,691	(67%)	93,0	70,7	87,5
Step 2	2,102	(11%)	72,7	8,0	27,7
Step 1 + 2	14,793	(78%)	**90,2**	**78,5**	**87,6**
Baseline	13,321	(70%)	85,3	66,6	80,8

Table 9.3: Results of the evaluation of the PROTEIN alignment.

Evaluation Results

The evaluation of the PROTEIN alignment against the manually created gold standard, applying the relaxed precision and recall measures introduced above, resulted in 90% precision, 79% recall and an $F_{0.5}$-score of 88%. The analogous evaluation of the alignment generated by the baseline approach resulted in 85% precision, 67% recall and an $F_{0.5}$-score of 81%. Considering the steps of the matching procedure separately, in analogous evaluations for the alignment generated by the first step only 93% precision, 71% recall and 88% $F_{0.5}$-score were achieved, and for the second step 73% precision, 8% recall, and 28% $F_{0.5}$-score. The evaluation results are summarized in table 9.3.

9.4 Documentation, Release and Maintenance

The PROTEIN alignment has been documented in terms of a proceedings paper [Beisswanger et al., 2010]. A public release embedded in the Semedico system and the initiation of the maintenance of the PROTEIN alignment are pending.

9.5 Use Case

The PROTEIN alignment has been designed to enhance the “Genes and Proteins” facet of the semantic search engine Semedico. It allows to extend the current facet, based on an almost flat list of protein entries derived from the protein database UniProtKB with hierarchical superstructures from MeSH. In contrast to the current facet a hierarchically structured variant would enable taxonomic search strategies.

The following example is intended to illustrate the benefit of using a hierarchically structured “Genes and Proteins” facet (the numbers were compiled on July 5, 2012): Conducting a document search with the PubMed search engine (see page 92) using

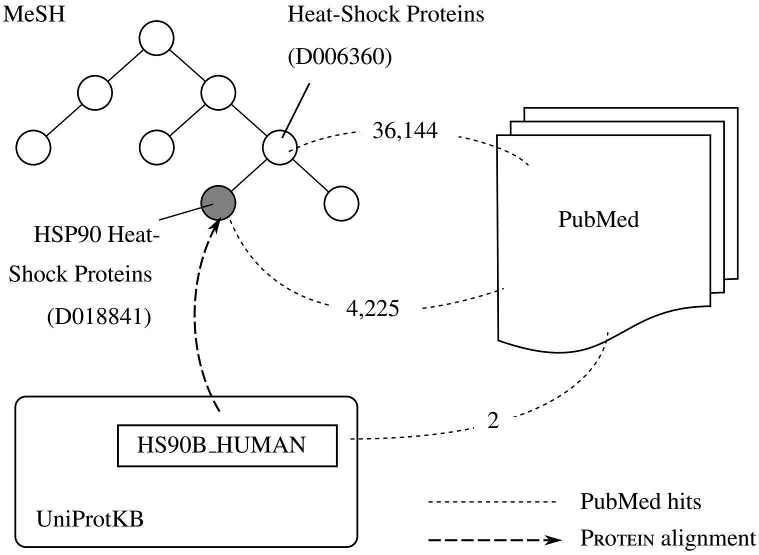


Figure 9.2: Example search utilizing the PROTEIN alignment. The increasing number of documents found (2—4,225—36,144) reflects the increasing conceptual generality of the search terms involved ("Heat shock protein HSP 90-beta" from UniProtKB entry HS90B_HUMAN—"HSP90 Heat-Shock Proteins"—"Heat-Shock Proteins"). The numbers are from July 5, 2012.

the exact string "Heat shock protein HSP 90-beta" (derived from UniProtKB entry 'HS90B_HUMAN') as query term results in only two documents as hits. In order to generalize the search to get additional hits, the searcher would have to come up with appropriate new query terms. In many cases, this is no trivial task, and PubMed currently fails to provide any support to solve it. If the search would have been carried out using a faceted search engine instead, provided with a hierarchically structured facet based on the PROTEIN alignment, the searcher would have been able to easily generalize the document search by entering the facet's term hierarchy with the search term and ascending in it using the GUI of the search engine. For example, the broader term "HSP90 Heat-Shock Proteins" (derived from MeSH entry 'D018841') could be selected as new search term, or the even more general term "Heat-Shock Proteins" (derived from the MeSH entry 'D006360'), without requiring that the searcher would know these terms in advance or search for them in external knowledge resources. For the exact string "HSP90 Heat-Shock Proteins" a PubMed search would result in 4,225 documents, and for "Heat-Shock Proteins" in 36,144. The example is outlined in figure 9.2.

Chapter 10

Consolidation of Ontology Reference Alignments

This chapter deals with the evaluation of the validity and reusability of three manually created or curated ontology alignment datasets, referred to as the Anatomy, Lod and Bridge dataset. Each of the datasets has been (or is still) used as reference for the evaluation of automatic ontology matching systems. The Anatomy dataset comprises a standard reference alignment from the biomedical domain. It deals with anatomy as application domain and consists of correspondences based on the 'equivalent-to' relation. The Lod and the Bridge alignments are two more recent datasets that excel in a broad domain coverage and the provision of 'equivalent-to' *and* 'is-a'-based correspondences. The evaluation has first been described in Beisswanger and Hahn [2012].

10.1 Reference Alignment Datasets

The Anatomy dataset comprises the reference alignment used in the OAEI 2010 Anatomy track (see page 37) with input ontologies. The alignment links pairs of equivalent classes from the anatomy branch of the NCI Thesaurus (NCI) [Sioutos et al., 2007], describing human anatomy, and the mouse adult gross anatomy ontology, which itself is based on the Anatomical Dictionary for the Adult Mouse [Hayamizu et al., 2005]. The alignment was created in a combined manual and automatic effort. First, an automatic matching approach was applied, utilizing lexical and structural techniques. Sec-

Dataset	Domain	Alignments	Ontologies	≡	⊑	Total
Anatomy	Anatomy	1	2	1,520	0	1,520
Lod	Various	7	8	85	2,260	2,339
Bridge	Various	10	17	?	?	4,876

Table 10.1: Statistics on the Anatomy, Lod and Bridge alignments. The symbols "≡" and "⊑" denote 'equivalent-to' and 'is-a'-based correspondences.

ond, an extensive manual curation step was carried out, consolidating the automatically achieved results [Bodenreider et al., 2005]. In the OAEI Anatomy track, a slightly revised version of the original alignment is used that contains 1,520 'equivalent-to'-based correspondences.

The Lod dataset, published by Jain et al. [2010], consists of seven manually created reference alignments of pairs of eight different schemata of Linked Open Data (LOD) sets. The schemata cover general information as well as particular domains ranging from geography over scientific publications to entertainment and social networks. The reference alignments comprise a total of 2,339 distinct correspondences, the vast majority of which (over 96%) relating to 'is-a' relationships, the remaining ones to 'equivalent-to' relationships. The original purpose of the reference alignments has been the evaluation of a matching system tuned to cross-link LOD schemata [Jain et al., 2010].

The Bridge dataset denotes another multi-domain dataset. It has been created by Mascardi et al. [2009] for the evaluation of a structural matching approach utilizing top level ontologies as semantic bridge in the matching process. The dataset consists of ten manually created reference alignments of selected pairs of 17 input ontologies, covering various domains, ranging from anatomy, biology and geography to gastronomy, travel and entertainment. The reference alignments contain a total of 4,876 distinct correspondences based on the relation types 'equivalent-to', 'is-a', and a generic 'related' relation. However, the relation types are not distinguished in the alignment files (for this reason, table 10.1 lacks coverage statistics for 'equivalent-to' and 'is-a'-based correspondences).

An overview of the sample datasets is given in table 10.1.

10.2 Evaluation

The Anatomy, the Lod and the Bridge datasets have been evaluated by applying ten basic quality checks for ontology alignments (section 4.5.5, figure 4.3).

Check 1 ("Is the alignment provided together with the input ontologies it is based on in the appropriate release versions, including imported ontologies, if applicable?") revealed that the Anatomy dataset, as it has been used in the OAEI evaluation campaigns for several years now, includes as input ontologies the anatomy branch of the NCI Thesaurus in the version from 2006-02-13, and the mouse adult gross anatomy ontology (MA) from 2007-01-18, both in OWL format. However, the anatomy alignment itself was created based on the NCI Thesaurus version 04.09a from 2004-09-10, and MA from 2004-11-22 [Bodenreider et al., 2005]. Obviously, different release versions of the input ontologies have been mixed up for the creation and the use of the reference alignment. Furthermore, the check revealed that the Lod and Bridge datasets do not include any input ontologies at all. Instead, URLs for download are provided in the respective publications [Jain et al., 2010; Mascardi et al., 2009]. The attempt to download the ontologies from the specified URLs revealed two major problems. First, it turned out that many ontology URLs do not point to a distinct ontology version. Instead, they either specify a webspace at which always the most recent version of the respective ontology is provided, or they refer to a general website of the ontology that provides download links for several versions of the ontology (usually the most recent and selected older ones). While, in the first case, the user cannot choose among alternative (in particular, older) ontology versions anymore at the specified URL, in the second case the user misses the information which version is the required one. In the latter case the decision was made to download always the most recent ontology version. The second problem arose from the fact that the Web is dynamic. Web contents are moved or deleted, which results in broken URLs. A total of eight out of 17 URLs specified for the input ontologies of the Bridge alignments, and additional URLs of ontologies imported by them, turned out to be already broken. Six of the concerned ontologies could be retrieved from the cache of the Semantic Web search engine Swoogle (see page 3.8) and two could be found by searching the Web. So, at least it was possible to proceed and run the remaining quality checks.

Since, according to the outcome of check 1, the quality checks were run using different input ontology versions (Anatomy) or at least potentially different ones (Lod and Bridge) than those underlying the respective alignments, for all three datasets check 2a and 2b were compulsory. Check 2a ("Check whether all classes persist to which correspondences in the alignment refer.") confirmed that for the Anatomy dataset all classes involved in the alignment (i.e., classes participating in at least one correspondence) are still contained in the available versions of the input ontologies. Hence, for the Anatomy alignment class consistency is preserved. In contrast, for the Lod alignments the check revealed that a total of 143 classes were missing in the downloaded input ontologies, affecting 413 correspondences, across the alignments, and for the Bridge alignments 12 classes were missing, affecting 18 correspondences. However, analyzing the affected correspondences in the Lod and the Bridge datasets revealed that in fact much fewer classes were missing in the ontologies than indicated by check 2a. The major reason for *seemingly* missing classes turned out to be simple errors in the alignment files, such as ordinary character mistakes or omissions, or mixed up name spaces, precluding the recovery of classes in the input ontologies. For example, the removal of erroneous whitespace from local names of classes in the LOD alignment mapping tables, the number of classes referred to in the alignments but not available in the corresponding input ontologies dropped from 143 to the much smaller number of 47, and the number of correspondences concerned decreased from 413 to 182. Because of the strong impact of whitespace removal, the revised versions of the LOD alignments were taken as basis for subsequent checks. Another reason for seemingly missing classes turned out to be the fact that a few correspondences in the BRIDGE alignments refer to ontology elements that are not explicitly specified to be classes in the input ontologies (i.e., they are not typed as ‘owl:Class’ or ‘rdfs:Class’) and thus were ignored by the implementation of the quality checks that was used in the context of this thesis.

In the anatomy alignment as it is contained in the ANATOMY dataset classes involved in correspondences are referenced by URIs. However, a curator of the alignment kindly provided the original mapping table on which the alignment is based. The mapping table lists both, pairs of URIs and of class labels. Running check 2b (“If classes are referred to by URI-label pairs in the alignment, check whether the URI-label pairs still persist.”), testing whether given URI-class combinations remained across different versions of the alignment input ontologies, 85 NCI classes and 34 MA classes were found for which the labels had changed. A manual inspection revealed that in most cases the labels have been made more precise in the new ontology versions (e.g., the label of class NCI_C12443 was changed from “Cortex” to “Cerebral Cortex”), were replaced by synonyms (e.g., the label of class NCI_C33178 was changed from “Nostril” to “External Nare”), or minor spelling or syntax modifications were made (e.g., the label of class MA_0000475 was changed from “aortic arch” to “arch of aorta”), while the meaning of the classes remained stable and the correspondences were still valid. However, the check also helped to reveal six mistakes in the alignment that were probably caused by shifts in the mapping table. For example, the class NCI_C49334 “brain white matter” was found to be mistakenly mapped to MA_0000810 “brain grey matter” and NCI_C49333 “brain gray matter” to MA_0000820 “brain white matter”. For the LOD and the BRIDGE alignments no URI-label pair data was available. Hence, this check could not be run.

Check 3 (“Check whether the alignment is made available in a machine readable format.”) revealed that the reference alignments included in the ANATOMY and the BRIDGE datasets are distributed in the Alignment API format [Euzenat, 2004] and thus can easily be accessed and used via the corresponding JAVA-based Alignment API [Euzenat, 2004]. In contrast, the alignments from the LOD dataset are not represented in a standard format but come in a comma delimited, three column format instead. They contain typical mistakes of manually created documents, such as additional whitespace or (few) missing values, which, however, hinder automatic processing. Check 4 (“Check whether the ontology classes in the alignment are referred to in terms of unique identifiers.”) confirmed that classes in the ANATOMY and the BRIDGE alignments are referenced via their URIs. However, for the LOD dataset the check revealed that local names are used as references to classes instead. A major problem with the local name-based approach is ambiguity. Although local names are unique in their original name space, across name spaces they can be ambiguous. Accordingly, correspondences in an alignment of ontologies that mix up classes from different name spaces, or that import whole ontologies with classes from a different name space, are potentially affected by ambiguities. Across the LOD alignments 30 cases of ambiguous local names were found.

Check 5 ("Check whether for all correspondences in the alignment the type of the assumed relationship is explicitly specified.") approved that across the examined alignments for each correspondence a relation type had been specified, such as 'equivalent-to' for the correspondence linking the classes 'NCI_C12717' ("Femur") and 'MA_000 1359' ("femur") in the ANATOMY alignment. However, for the BRIDGE dataset it turned out that all correspondences are based on the 'equivalent-to' relation, although the journal paper describing the alignments reports on a substantial number of correspondences that are based on the 'is-a' and 'related' relation [Mascardi et al., 2009]. Asked about this circumstance, a curator of the dataset explained that the relation types specified in the alignment files were mere technical artifacts that should be disregarded. In fact, the relation type information was suppressed already at creation time of the alignments, because it was of no importance for the evaluations carried out at that time. To cope with this situation, all relation types in the BRIDGE dataset were considered as unknown. Unfortunately this also meant that the remaining checks (requiring information on relation types) could not be run on this dataset.

Check 6 ("Check whether there are cases in the alignment in which a class from ontology O_1 is linked by 'equivalent-to' relationships to several classes in ontology O_2, while the latter are not linked by 'equivalent-to' relationships themselves, or vice versa.") identified 39 cases in the ANATOMY dataset and 10 cases in the LOD dataset in which a class from one input ontology of an alignment was linked by an 'equivalent-to' relationship to more than one class in the other input ontology. In none of the cases, the multiple equivalent classes were linked by 'equivalent-to' relationships themselves in the respective ontologies. All cases of multiple mapping targets were manually screened by forming class pairs and inspecting them individually. The analysis revealed that for 20 class pairs from the ANATOMY dataset in fact an 'equivalent-to' relationship holds, though not being explicitly specified in the respective ontology. Cross-checking with the most recent versions the respective ontologies revealed that 12 class pairs from the NCI Thesaurus have meanwhile been merged, while for another three class pairs from the NCI Thesaurus a merger was proposed to the NCI team in the context of this thesis. An example is the class pair NCI_C33708 ("suprarenal artery") and NCI_C52844 ("adrenal artery"). Meanwhile the mergers have been accepted and included in the NCI Thesaurus. Furthermore, 18 class pairs were identified in the ANATOMY alignment and five in the LOD alignments that were linked by other than 'equivalent-to' relationships in the respective ontologies. In 12 cases, a 'part-of' relationship was concerned (ANATOMY), in eight cases an 'is-a' relationship (four cases from ANATOMY and four from LOD), in two cases a sibling relationship (ANATOMY), and in one case a ‘disjoint-with’ relationship (Lod). An inspection of these relationships revealed that the majority of them is correct. The conclusion was drawn that for each of these class pairs only one ‘equivalent-to’-based correspondence in the alignment can be correct, while the other one must be wrong and has to be removed from the alignment. The same was found to apply to another five cases of multiple mapping targets in the Lod alignments, although in these cases no relationships between the class pairs concerned were present in the respective ontologies.

Check 7 (“Check whether there are cases in the alignment in which a pair of classes c_1 from O_1 and c_2 from O_2 are linked by an ‘equivalent-to’ relationship, while not every subclass of c_1 is linked to c_2 and all superclasses of c_2, and c_1 to every superclass of c_2 by a ‘is-a’ relationship, and vice versa.”) inferred 10,415 ‘is-a’-based correspondences for the Anatomy dataset and 772 for the Lod dataset exploiting ‘equivalent-to’-based correspondences from the manual alignments in combination with the taxonomic structure of the corresponding input ontologies. While in the case of the Anatomy dataset all detected correspondences were new (because of its innate focus on ‘equivalent-to’-based correspondences), for the Lod dataset still 540 (70%) of them were new, i.e., not yet contained in the alignments.

Check 8 (“Check whether there are pairs of classes from O_1 and O_2 with identical labels (or local names) that are not linked by an ‘equivalent-to’ relationship in the alignment.”) detected 13 class pairs with identical class labels or local names in the input ontologies of the Anatomy alignment, and 37 in the input ontologies of the Lod alignments, for which no ‘equivalent-to’-based correspondence existed in the respective alignment. The check was run after applying a simple string normalization procedure to all class labels and local names that comprised, amongst others, the splitting of CamelCase expressions, lowercasing, and the removal of underscores. A manual inspection of the class pairs concerned revealed that two class pairs from the Anatomy dataset in fact referred to slightly differently defined concepts. (For example, the classes MA_0000323 and NCI_C12378 share the label “gastrointestinal system”. However, while the MA class fits the usual understanding of “gastrointestinal system” comprising the stomach, intestine and the structures from mouth to anus, the NCI class does not, but includes, in addition, accessory organs of digestion, such as the pancreas and the liver. Instead, the NCI anatomy branch comes with another class, NCI_C22510 “gastrointestinal tract”, which corresponds to MA_0000323 “gastrointestinal system”.) Furthermore, it turned out that in the Lod dataset in three cases an ‘is-a’ relationship instead of the ‘equivalent-to’ relationship, as proposed by the check, holds between the pair of classes sharing a name. For example, one class in a class pair sharing the name “Genre” referred to the general concept of genre, and the other one to the more specific concept of *music* genre. For the remaining eleven class pairs in the Anatomy dataset and 34 in the Lod dataset an ‘equivalent-to’-based correspondence turned out to be effectively missing in the respective alignments. An example is the class pair (NCI_C33460, MA_0002730) from the Anatomy dataset sharing the label “renal papilla”. In the Lod dataset some input ontology pairs import classes from the same third-party ontologies. Thus, in nearly half of the analyzed cases it turned out that the classes did not only have identical names, but, in fact, denoted the same classes.

Check 9 ("Check whether there are pairs of classes from O_1 and O_2 with labels (or local names) with identical syntactic heads that fulfill the condition that one label (or local name) includes the other, but the class pair is not linked by an 'is-a' relationship in the alignment.") identified 3,127 class pairs in the input ontologies of the ANATOMY alignment and 57 in input ontologies of the LOD alignments for which the label (or local name) of one class included the label (or local name) of the other one and the two shared the syntactic head, and for which no 'is-a'-based correspondence existed yet in the respective manual alignment. The check was run after normalizing the class labels and local names as described above (see check 8). A manual analysis of the class pairs from the LOD dataset revealed that 52 'is-a'-based correspondences were in fact missing in the respective alignments, of which 24 had already been detected by check 7. Furthermore, five proposed 'is-a' relationships were judged to be imprecise or wrong. For example, for the classes named "Label" and "RecordLabel" an 'equivalent-to'-based correspondence should be added to the respective alignment instead of an 'is-a'-based one, and for the classes named "Book" and "InBook", and "Conference" and "Attending-A-Conference" no correspondences should be added at all.

Check 10 ("Determine how many non-trivial correspondences occur in the alignment.") revealed that in the ANATOMY dataset 916 correspondences (60%) and in the LOD dataset 158 correspondences (7%) are trivial ones (see page 4.5.5). In the BRIDGE dataset the number of trivial correspondences could not be computed because of the missing type specification for correspondences.

The results of applying the basic quality checks to the ANATOMY, LOD and BRIDGE alignments are summarized in table 10.2.

Check	Topic	ANATOMY	LOD	BRIDGE
Check 1	Input ontologies	wrong versions	-	-
Check 2a	Missing classes	-	143 (47)	12
Check 2b	URI-label changes	121	-	-
Check 3	Standard format	yes	no	yes
Check 4	URIs as references	yes	no	yes
Check 5	Explicit relation types	yes	yes	no
Check 6	Multiple equivalences	39	10	-
Check 7	Inferable correspondences	10,415	540	-
Check 8	Label identity, but no $\equiv$	13	37	-
Check 9	Label inclusion, but no $\sqsubseteq$	3,127	57	-
Check 10	Trivial correspondences	916 (60%)	158 (7%)	-

Table 10.2: Results of applying basic quality checks to the ANATOMY, LOD and BRIDGE alignments. The symbols "$\equiv$" and "$\sqsubseteq$" denote 'equivalent-to' and 'is-a'-based correspondences.

Chapter 11

Discussion of the Case Studies

In this chapter, GRO, MaHCO, BioTop and the Protein alignment are discussed as the resources that resulted from the case studies presented in chapters 6–9. To assess the expressiveness of GRO, MaHCO and BioTop, a comparison against existing biomedical domain and top domain ontologies is carried out (section 11.4). In addition, the suggestions for improvement are revisited that have been achieved for the Anatomy, Lod and Bridge alignments in the case study presented in chapter 10. Finally, the practical realization of the previously proposed approach to building ontological background knowledge for biomedicine across the five case studies is discussed and factors of successful ontology development are derived.

11.1 The Gene Regulation Ontology

GRO has been developed as an ontology about the regulation of gene expression (see chapter 6). While existing knowledge resources on this topic either suffer from a fragmented coverage or a limited computational accessibility (see section 6.1), this is not the case with GRO. It describes basic processes of gene regulation together with their participants and relationships between them in the formal ontology language OWL DL. Generally it is the first publicly available ontology that is entirely dedicated to the field of gene regulation.

The fact that GRO covers key concepts of gene regulation and relationships between them has been utilized in different corpus annotation projects. Subsets of GRO classes and relationships between them have been used as vocabulary for the annotation of various types of named entities and relations in domain-specific documents [Buyko et al., 2010; Kim and Rebholz-Schuhmann, 2011; Thompson et al., 2009]. The annotated documents in turn served as training data for machine learning-based NER tools and a powerful relation extraction system [Buyko et al., 2011]. Although the formal rigor of an ontology has no direct impact on the annotation result, there are good reasons for using an ontology-backed annotation vocabulary. First, ontologies can help define the annotation scope and the right coverage and granularity of the annotation vocabulary. Second, ontologies can support the complex task of relation annotation by encoding classes and relationships between them. Third, there is first evidence that using an ontology-based annotation vocabulary (at this point assuming a formal and expressive ontology that adheres to basic ontological distinctions) leads to more correct and unambiguous annotations than using an ad-hoc compiled vocabulary [Kawazoe et al., 2006]. The growing number of semantically annotated corpora that rely on ontologies as annotation vocabulary [see, e.g., Bada et al., 2010; Kim et al., 2008] indicates the increasing acknowledgment of these reasons.

The formal representation of gene regulation processes, their participants and explicit relationships between them by GRO has been utilized for the manual compilation of domain-specific inferencing rules [Kim and Rebholz-Schuhmann, 2011]. The authors of the rules report that incorporating the latter in their domain-specific event extraction system substantially improved the performance of the system in three different evaluation settings [Kim and Rebholz-Schuhmann, 2011]. Using GRO-based rules, their system even outperformed the above mentioned machine learning-based system on a "real-world" reference dataset that had been compiled from a manually curated database on gene regulation [Hahn et al., 2009]. The performance advantage can be assumed to result from the fact that the manually created rules have been tailored to the extraction of gene regulation events, while the machine learning-based system takes a more general approach to biomedical event extraction, making it applicable to a broader spectrum of tasks. Whether a broad scope or a particularly high performance (at the expense of a narrower scope) is more advantageous strongly depends on the application context.

The formal representation of explicit relationships between gene regulation processes and their participants has further been exploited for the construction of formal definitions for classes of GO [Kim et al., 2010]. Based on GRO, Kim et al. [2010] achieved full formal definitions for 75% of the subset of GO classes that represent gene regulatory processes. They report that an alternative approach, which was run on the entire GO, achieved definitions for only 15% of the classes in the mentioned subset. The relevance of the work by Kim et al. [2010] lies in the fact that on the one hand GO is by far the most important biomedical ontology on which various real-world applications depend and on the other hand, it still lacks formal rigor. This precludes the detection of modeling mistakes by logical consistency checking, as well as the use of GO in reasoning-based applications, especially in the field of biomedical NLP.

Hence, GRO contributes to the automatic extraction of facts from the biomedical literature (which under the term "automatic biocuration" increasingly supports human database curators who struggle with the fast publishing rate of biomedical articles [Baumgartner et al., 2007]) in at least three different ways. As annotation vocabulary for the creation of training data it supports a machine learning-based event extraction systems. As conceptual basis for inferencing rules it backs a rule-based event extraction systems. Furthermore, it helped to formalize GO, which itself is used in various ways for information extraction tasks.

The evaluation of GRO, presented in section 6.3, resulted in valuable hints how GRO could further be improved. Most importantly, it revealed that in order to preserve (or reestablish) the value of GRO, an update with external knowledge resources would be required from which GRO reuses contents. Furthermore, the evaluation revealed that some GRO classes, 'is-a' relationships and formal class definitions should be checked for their correctness. Some classes should be checked for missing subclasses, formal class definitions and 'disjoint-with' relationships to other classes. In addition, missing natural language annotations should be added, incorrectly formatted ones corrected and empty ones removed, amongst others. The evaluation also revealed that in order to further strengthen the compliance of GRO with quality standards stated by the OBO Foundry, maintenance would have to be resumed, cross-product links to existing OBO Foundry ontologies provided, the compliance with RO and the OBO naming conventions increased, efforts made to attract new users and a collaboration with developers of existing OBO Foundry ontologies started.

To sum up, GRO takes a novel approach by representing common knowledge on gene regulation both formally and comprehensively. Backing ML-based and rule-based automatic event extraction systems and facilitating the formalization of parts of GO, it supports the increasingly important task of automatic biocuration in various ways. Worth mentioning in this context is the fact that GRO will play an important role in the 2013 edition of the BioNLP Shared Task[1] (see also page 7). The latter will cover six event extraction tasks, of which the "GRO Task" will concern GRO-based corpus annotation and automatic fact extraction. The involvement in the Shared Task can be expected to strongly increase the visibility of GRO in the biomedical NLP community and promote its further development. GRO could additionally profit from the suggestions for improvement that resulted from the evaluations of GRO, carried out in the context of this thesis. Correcting individual mistakes in GRO and making GRO guideline compliant could increase its value and strengthen analysis results based on it. In addition, further tying it up with the OBO Foundry ontologies could increase its visibility in the bio-ontology community.

[1] `http://2013.bionlp-st.org/` – access date 2013-01-13.

11.2 The Major Histocompatibility Complex Ontology

MaHCO has been developed as an ontology on the MHC of multiple species (see chapter 7). It represents MHC genes, alleles, chains and molecules as classes in a common framework. The classes are hierarchically organized and linked by additional non-taxonomic relationships. This is a completely novel approach. Existing knowledge resources on the MHC on the one hand comprise flat files and database tables, which fail to hierarchically organize MHC data, often represent genes and alleles separately from chains and proteins, and data on different species separately from one another. On the other hand, they comprise broad-coverage thesauri and biomedical ontologies that cover the MHC only fragmentarily as one among many other topics (see section 7.1). As a computational representation of aspects of the MHC, MaHCO complements the IMGT ontology [Duroux et al., 2008] and ONTIE [Sathiamurthy et al., 2005] as two existing ontologies on parts of immunology. MaHCO is the first publicly available ontology about the MHC of multiple species that is implemented in a formal ontology language, *viz.*, OWL DL.

The coverage of key concepts of the MHC by MaHCO was a prerequisite for using it in support of a domain-specific corpus annotation project [Hahn et al., 2008]. A subset of MaHCO core classes was used as annotation vocabulary for the annotation of a corpus of immunology abstracts, which in turn was used as training data for a machine learning-based NER tool. In a gold standard-based evaluation, the tool achieved an F-score of 82.8%, with 83.1% precision and 82.5% recall, denoting a state-of-the-art performance of biomedical NER tools at that time [Hirschman et al., 2007]. Given the complexity of the MHC, a major benefit of using an ontology as source for the annotation vocabulary was that it helped define the annotation scope and the right coverage and granularity of the annotation vocabulary.

The elaborate hierarchical representation of HLA alleles and chains in MaHCO has qualified it as conceptual backbone for domain-specific browsing and search applications. The class hierarchy of MaHCO HLA has been used in support of different computational immunology Web-tools, developed at the Hannover Medical School. An example is the “HLA Module Explorer” of the PeptideCheck tool, which enables the intuitive browsing of HLA alleles, the selection of individual alleles, and their comparison to other HLA alleles on the level of peptide modules. This procedure is important for finding the best donor/recipient match for hematopoietic stem cell transplantation [DeLuca et al., 2009]. Although the HLA Module Explorer solely uses the class hierarchy of MaHCO HLA and leaves further features of it unexploited, it clearly profits from the strict classification of HLA alleles provided by MaHCO as ontology. Instead of exactly mirroring the HLA nomenclature, MaHCO classifies HLA alleles strictly according to their characteristics, avoiding the auxiliary “rollover” groups contained in the HLA nomenclature at that time (see page 110). This novel approach enabled the intuitive and convenient browsing of HLA alleles for the first time.

The evaluation of MaHCO, presented in section 7.3, provided important hints on how the ontology could further be improved. Most importantly, it revealed that in order to preserve (or restore) the value of MaHCO, an update regarding the external knowledge resources, on which it depends, would be required. The evaluation further revealed that the expressiveness of MaHCO could be increased by adding 'disjoint-with' relationships between most pairs of sibling classes in the ontology[2] and adding closure axioms to existing 'encodes' relationships between HLA allele and corresponding HLA chain classes. For example, in MaHCO the fact that HLA-A alleles encode HLA-A chains is currently stated as

$$\text{`HLA-A'} \sqsubseteq \exists\text{`encodes'}.\text{`HLA-A\_Chain'},$$

while in fact HLA-A alleles encode HLA-A chains and only HLA-A chains as gene products. This fact could explicitly be stated by adding a closure axiom:

$$\text{`HLA-A'} \sqsubseteq \exists\text{`encodes'}.\text{`HLA-A\_Chain'} \sqcap \forall\text{`encodes'}.\text{`HLA-A\_Chain'}.$$

The evaluation further helped to detect that currently many HLA allele and HLA chain classes share a class label, due to the lack of distinct names for HLA chains in the HLA nomenclature. To prevent duplicate labels in future releases of the ontology, the suffix "chain" should be added to labels of HLA chain classes (e.g., the label "B*4402" of class 'B_4402_Chain' would become "B*4402 chain"). However, the short labels for chain classes should be kept as alternative labels, because they are widely-used in external knowledge resources and the domain-specific literature. In addition, missing natural language annotations should be added to MaHCO, wrongly formatted ones revised and empty ones removed. Unused annotation properties should be considered for removal.

The evaluation also revealed that the maintenance efforts for MaHCO could be reduced by reorganizing MaHCO according to the modularization approach by Rector [2003]. For this purpose, the multi-hierarchical (asserted) class hierarchy of MaHCO would have to be transformed into a tree structure, and classes would have to be provided with formal definitions that enable reasoners to compute the previously hard-coded alternative classifications.

Finally, the evaluation indicated that in order to increase the value and visibility of MaHCO for the bio-ontology community, its adherence to the OBO Foundry principles would further have to be strengthened. Similarly as for GRO, this would require to make MaHCO subject to continuous maintenance, provide cross-product links to existing OBO Foundry ontologies, improve the compliance with RO and with the OBO naming conventions, add missing verbal class definitions, make efforts to attract new

[2] Exceptions arise from classes that represent different HLA classifications schemes. For example, the class 'A23' referring to serological typing and the class 'A_23' referring to sequencing-based typing are sibling classes that are *not* disjoint.

users and start a collaboration with developers of existing OBO Foundry ontologies.

To sum up, MaHCO excels in the formal representation of the MHC of multiple species. Due to its coverage of basic MHC classes and the strict ontological organization of HLA alleles and HLA chains it has successfully been used in specialized NLP and immunoinformatics applications. The further development of MaHCO could profit from realizing suggestions for improvement that resulted from the evaluation of MaHCO. Realizing these suggestions could make MaHCO an even more reliable tool for researchers and clinicians who profit from a formal representation of the MHC.

11.3 BioTop

BioTop has been developed as a formal top level ontology for molecular biology and biomedicine that is implemented in OWL DL (see chapter 8). A distinguishing feature of BioTop is that it represents fundamental biomedical classes and relations in a formal ontology language *and* explicitly specifies their meanings in terms of axioms. In contrast, the currently widely-used top level approaches for biomedicine (*viz.*, the UMLS Semantic Network and the GENIA ontology) either suffer from an informal representation, such as the UMLS Semantic Network, or from a limited coverage and expressiveness, such as the GENIA ontology (see section 8.1).

BioTop remedies several major shortcomings of the GENIA ontology, from which it originates [Schulz et al., 2006a]. Being integrated with the top level ontology BFO, in contrast to the GENIA ontology it is provided with an ontological grounding. Representing additional classes, including dependent continuants (e.g., functions and qualities) and occurrents (processes), it closes conceptual gaps of the GENIA ontology. Furthermore, due to the formal class definitions, conceptual relations and 'disjoint-with' relationships between classes that it contains it is much more expressive than the GENIA ontology. Although meanwhile a second GENIA ontology has been released, covering biological processes and molecular functions [Kim et al., 2008], BioTop by far exceeds the (meanwhile two) GENIA ontologies in terms of ontological grounding, formal rigor and expressiveness.

BioTop has been employed in various applications (see section 8.6). For example, it was used as conceptual basis for the development of an ontology on biological taxa [Schulz et al., 2008] and an application oriented ontology on antibiotics resistance patterns [Schober et al., 2010]. In both cases, it provided the necessary ontological grounding and basic domain-specific distinctions, which otherwise would have been missed or had to be reinvented.

Furthermore, an alignment of BioTop and the UMLS Semantic Network has been created [Schulz et al., 2009a]. By means of this alignment, a substantial proportion of mistakes in the semantic classification of concepts in the UMLS Metathesaurus has been detected (see page 136) and the mistakes were subsequently proposed for revision. The relevance of this finding lies in the fact that the UMLS Metathesaurus is the world-wide largest publicly available terminology system for biomedicine, on which many practical applications depend. The reason why this result could be achieved is that the alignment between BioTop and the UMLS Semantic Network allowed to pass the formal rigor of BioTop on to the UMLS Semantic Network, which has long been used for the semantic categorization of Metathesaurus concepts but lacks the necessary formalization to detect incompatible combinations of semantic type assignments itself.

BioTop was further used for the semantic integration of biomedical domain ontologies in a pilot study on a subset of the OBO ontologies [Beisswanger et al., 2008c]. At the time when BioTop was created the OBO ontologies lacked any deeper form of conceptual integration. Indeed, integration efforts have been made based on the automatic detection of implicit relationships between classes of different OBO ontologies. For example, the composition of labels of GO was investigated [Bada and Hunter, 2008; Burgun and Bodenreider, 2005; Myhre et al., 2006; Ogren et al., 2004] and was exploited to derive formal class definitions across ontologies [Mungall, 2004; Mungall et al., 2011a; Wroe et al., 2003]. However, these efforts missed the grounding on a formally rigid and abstract ontological framework, which is critically required [Rosse et al., 2005]. The integration of the OBO ontologies based on BioTop as semantic bridge is of the advantage that BioTop itself constitutes such a framework. An additional benefit is that in contrast to comparably expressive top domain ontologies for biology and biomedicine, such as GFO-Bio [Hoehndorf et al., 2008], BioTop relies on the top level ontology BFO, which is also the recommended top level for the OBO ontologies [OBO Foundry, 2012, principle FP14 (under review)]. This circumstance can be expected to strongly simplify the integration process.

The evaluation of BioTop (presented in section 8.3) resulted in suggestions how BioTop could further be improved. Amongst others, it helped to identify potentially missing subclasses, formal class definitions and 'disjoint-with' relationships between classes, missing class labels and verbal definitions, classes that should be considered for a split, empty annotations that should be removed and unused conceptual relations and annotation properties that should be considered for removal. It also indicated that the quality of natural language annotations could be improved by introducing explicit annotation guidelines.[3] To further improve the adherence to the OBO Foundry principles, cross-product links to OBO Foundry ontologies would have to be provided and the collaboration with developers of OBO Foundry ontologies strengthened. In addition, verbal class definitions would have to be added, if applicable, and additional users would have to be recruited.

[3] The evaluation concerns BioTop version 2008-02-19. Meanwhile annotation guidelines have been established for BioTop, such that the current BioTop version 2012-01-29 supports the unambiguous automatic access of natural language annotations.

To sum up, BioTop is a new top domain ontology for molecular biology and biomedicine that excels in the coverage of fundamental biomedical classes and relations, formal rigor and expressiveness. It has shown to provide the necessary ontological grounding and domain-specific distinctions for the successful development of new biomedical domain ontologies, enough formal rigor to detect mistakes in applications of widely-used though informal existing top level resources for biomedicine, and the appropriate coverage to integrate various domain ontologies from the OBO library. Although the integration of the whole OBO library based on BioTop is pending, it promises a coherent and expressive, large-coverage conceptual resource for biomedicine with BioTop as top level that would allow for cross-ontology consistency checking and other value-adding inferencing services, as they are required by sophisticated biomedical NLP applications, amongst others. The further development of BioTop could profit from implementing the suggestions for improvement that resulted from the evaluation of it (though they would first have to be cross-checked with the current version of BioTop).

11.4 Expressiveness of GRO, MaHCO and BioTop

The goal of the case studies on GRO, MaHCO and BioTop was the construction of formal and (apart from the MaHCO HLA extension) also expressive ontologies. To assess whether the intended expressiveness has been achieved, the ontologies were compared to external domain and top domain ontologies. As external domain ontologies GO, SO, CL, PRO, ChEBI and the FMA library were selected (see section 2.2.3) and as external top domain ontologies GFO-Bio and SBUO (see section 8.1). The domain ontologies were chosen for three reasons. First, they cover domains related to those covered by GRO and MaHCO. Second, they have a central position in the OBO library [Smith et al., 2007]. Third, they are actively used and maintained. The top domain ontologies were chosen for their thematic scope and intended formal rigor.

The expressiveness of ontologies was assessed based on three criteria: the proportion of formally defined classes, the 'disjoint-with' relationship/class ratio ("DR/C"), and the conceptual relationship/class ratio ("CR/C"), where conceptual relationships were counted as proposed on page 26. The proportion of defined classes was considered because it is an evident figure for the expressiveness of an ontology. However, in addition DR/C and CR/C values were considered, because the proportion of defined classes does neither reflect the provision of 'disjoint-with' relationships between classes that are crucial for effective logical consistency checking [see, e.g., Meilicke et al., 2008; Rector et al., 2004], nor the number of conceptual relationships expressed in terms of *partial* formal class definitions in ontologies. To abstract from the size of ontologies, relationship/class ratios were compared instead of direct relationship counts.

For each of the selected OBO ontologies two OWL versions were downloaded from the

Ontology	Origin	Classes	%Defined	CR/C	DR/C
Domain ontologies					
GRO 0.5	internal	506	13%	0.64	0.18
MaHCO 1.0.1	internal	7,928	-	0.42	0.02
MaHCO core 1.0.1	internal	105	13%	0.26	**1.21**
GO 2008-05-26	external	26,136	-	0.27	-
SO 2008-05-26	external	1,498	13%	0.32	0.02
CL 2008-05-26	external	857	-	0.24	-
PR 2008-05-26	external	667	-	0.15	-
ChEBI 2008-05-26	external	19,109	-	0.43	-
FMA 2008-05-26	external	75,145	-	0.59	-
GO 2012-04-24	external	36,554	21%	0.39	-
SO 2012-04-24	external	4,090	11%	0.28	-
CL 2012-04-24	external	2,021	**28%**	**1.03**	0.04
PRO 2012-04-24	external	28,755	21%	0.27	0.02
ChEBI 2012-04-24	external	31,470	-	0.76	-
FMA 2012-04-24	external	80,469	-	0.57	-
Top domain ontologies					
BioTop 2008-02-19	internal	175	**26%**	**0.56**	0.58
GFO-Bio	external	164	21%	0.40	0.18
SBUO	external	144	17%	0.16	**0.74**

Table 11.1: Statistics on internal and external biomedical domain and top domain ontologies. “CR/C” and“DR/C” stand for the relationship/class ratio for conceptual and ‘disjoint-with’ relationships, respectively. The numbers refer to the asserted class hierarchy of the respective ontology. Percentages and ratio values were rounded.

OBO website (`http://obofoundry.org/`), the first one from May 26, 2008 (a time at which GRO, MaHCO and BioTop had reached a rather mature state) and the second one from April 24, 2012 (a time at which the funding for GRO and MaHCO had already expired for three years). The top domain ontology GFO-Bio was downloaded from the GFO-Bio website (`http://www.onto-med.de/ontologies/gfo-bio/index.jsp`) and the (factored version of) SBUO from the SBUO website (`http://www.cs.man.ac.uk/~rector/ontologies/simple-top-bio/`) on May 8, 2012. Because both top domain ontologies have not changed since 2006, only a single version of them was considered.

For all internal and external domain and top domain ontologies class counts, proportions of formally defined classes and CR/C and DR/C values were calculated (table 11.1). Overall, the numbers confirm the expectation that regarding class counts top domain ontologies are much smaller than domain ontologies. Furthermore, they reveal that between 2008 and 2012 many new classes have been added to the external domain ontologies, and new formal class definitions have been added to GO, SO, CL and PR. For GO, CL and PR even the percentage of defined classes has increased.

A comparison of the figures for GRO and MaHCO with those for the external domain ontologies from the OBO library revealed that GRO and MaHCO *core* excel in high DR/C values. Furthermore, with respect to the percentage of formal class definitions they exceeded or were at least strongly competitive with the selected OBO ontologies in 2008, while in 2012 they had been overtaken by three of them. Similarly, with respect to CR/C values, GRO exceeded all and MaHCO all but two of the selected OBO ontologies in 2008, while in 2012 GRO had been overtaken by two and MaHCO by three of them. These results directly reflect the fact that the funding for GRO and MaHCO has expired in 2009, and hence their maintenance was discontinued, while the OBO ontologies are still maintained, partially by professional curator teams who coordinate requests from a growing user community and special interest groups [see, e.g., Leonelli et al., 2011]. Furthermore, between 2008 and 2012 in the bio-ontologies community efforts were made to introduce explicit cross links between some of the OBO ontologies [Meehan et al., 2011; Mungall et al., 2011a,b], leading to the creation of new formal class definitions for some of the OBO ontologies.

A comparison of the figures for BioTop with those for the external top domain ontologies GFO-Bio and SBUO revealed that BioTop particularly excels in the provision of full formal class definitions and a high CR/C value. In both respects, it outperforms GFO-Bio and SBUO. Regarding DR/C values, it outperforms GFO-Bio and *is* outperformed by SBUO.[4]

The figures in table 11.1 also led to the unexpected finding that in some respects the analyzed top domain ontologies are less expressive than (some of) the analyzed domain ontologies. For example, half of the external domain ontologies in their 2012 versions excel in an equal or higher proportion of defined classes than GFO-Bio and SBUO. As another example, the CR/C figures for GRO and MaHCO are higher than those for GFO-Bio and SBUO, and the DR/C figure for MaHCO core is higher than those of all analyzed top domain ontologies. A possible explanation for these findings is that the increasing use of the analyzed domain ontologies in real-world applications may have pushed their development on the level of formal semantics, while the analyzed top domain ontologies largely miss this trigger through real world applications.

When interpreting the results of this comparison study, it must be considered that the CR/C figures depend on how conceptual relationships were counted (see page 26).

[4] The comparison is based on BioTop version 2008-02-19. The current BioTop version 2012-01-29 holds the same proportion of fully defined classes, the DR/C value increased to 0.78, and the CR/C dropped to 0.31. Accordingly, the current BioTop version outperforms GFO-Bio and SBUO with respect to DR/C values and *is* outperformed by GFO-Bio with respect to CR/C values.

11.5 The Protein Alignment

The PROTEIN alignment has been developed as bridge between parts of the protein database UniProtKB and the MeSH thesaurus (see chapter 9). It links specific protein entries of UniProtKB with the hierarchically organized protein group, family and complex entries of MeSH. Being expressed in a machine processable format, it facilitates the seamless browsing of protein entries across granularity levels. At the time of the release of the alignment it was the only resource providing this facility. Meanwhile, PRO is developing towards a second such resource [Natale et al., 2011]. However, with regard to human proteins it still lacks far behind the coverage of the merger of the PROTEIN alignment with its input resources (see section 9.1).

The PROTEIN alignment is intended to support the semantic document retrieval system Semedico (see page 139). It facilitates the creation of a hierarchically structured protein facet. Being the first of its kind, this facet for the first time enables protein-centered faceted search. In comparison, existing document retrieval systems support only more restricted variants of protein-centered search. PubMed, as the standard document retrieval system for the life sciences, allows to search for documents about a certain protein by entering a free text query or a query composed of entries of the hierarchically structured MeSH. It neither supports non-free text queries beyond the coverage of MeSH nor the extension or restriction of the query results achieved. This is different with the related document retrieval system GoPubMed[5] [Doms and Schroeder, 2005]. It allows to filter query results based on the GO and the MeSH hierarchy. Obviously, the coverage of these two resources restricts its filtering facilities. While MeSH covers proteins on a rather coarse-grained level only (it represents protein families, groups and complexes, but no individual protein classes), GO represents cell components, molecular functions and biological processes *associated* with proteins and other gene products, but no proteins themselves. A detailed protein-centered search is enabled by protein-centered search systems, such as iHOP[6] [Hoffmann and Valencia, 2004]. Similar to Semedico, iHOP incorporates NER tools and normalizers that in a preparatory stage enrich documents with semantic metadata, including synonyms or links to databases entries. If subsequently a protein name is entered as search term, not only a set of documents matching this term is retrieved, but also links to corresponding database entries, comprising additional factual information about the protein under scrutiny. However, iHOP is incapable of taxonomic searching.

To evaluate the matching approach used to generate the PROTEIN alignment, parts of the alignment were compared to a manually created gold standard, applying relaxed precision and recall measures and computing $F_{0.5}$-scores (i.e., F-scores with an emphasis on precision, see section 9.3). The choice of the evaluation measures reflects requirements

[5] `http://www.gopubmed.com` – access date 2013-01-24.

[6] `http://www.ihop-net.org/UniPub/iHOP/` – access date 2013-01-24.

from the information retrieval scenario for which the alignment was designed (see section 9.5). The claim is that even if the automatic procedure cannot detect the fully correct MeSH entry for a UniProtKB entry, the detection of a slightly more specific or more general MeSH entry still enables the user of the document retrieval system utilizing the alignment to pass from UniProtKB-based protein terms to MeSH-based ones and hence to correctly generalize or specialize the original search utilizing the MeSH hierarchy. The emphasis on precision reflects the assumption that false mappings that would lead to the retrieval of irrelevant documents are worse than missing mappings due to the negligence of taxonomic relation that particular proteins share.

The evaluation of the Protein alignment resulted in 90% precision, 79% recall and an $F_{0.5}$-score of 88%, compared to 85% precision, 67% recall and an $F_{0.5}$-score of 80% that were obtained for a baseline approach (see table 9.3). The numbers show that regarding precision the performance of the proposed matching procedure is acceptable. An analysis of the false predictions revealed that three-fourth of all incorrect mappings were due to the unresolved ambiguity of gene symbols. Regarding recall, the numbers indicate that more than 20% of the considered UniProtKB entries were *not* mapped to MeSH entries by the proposed approach, although based on the gold standard data only about 10% of not-mapped entries would have been expected (see page 145). The inspection of missed mappings revealed that many of the involved UniProtKB entries come with rather technical names that cannot easily be matched to MeSH labels. An example is the entry ‘YI020_HUMAN’ named “FAM75-like protein FLJ43859”.

In an attempt to increase recall the matching procedure was extended by a third step that resembles the first one but allows for partial token matches and contradicting TaxIDs (see section 9.2). Furthermore, additional UniProtKB name types were considered for the mapping (e.g., short forms of gene and protein names and so-called “international non-proprietary names” [see UniProt Consortium, 2012]). By means of this third step, 1,267 additional UniProtKB entries could be mapped, an increase of 7%. However, the $F_{0.5}$-score dropped from 88% to 85% due to a decrease in precision (86%). Accordingly, the overall effect of the additional step was negative, wherefore it was not included in the matching procedure.

A post-processing analysis of the Protein alignment revealed that the UniProtKB entries which the matching approach was able to map to a MeSH entry occur on average in ten times as many Medline abstracts (*viz.*, 111.3 abstracts) as those it was unable to map (*viz.*, 12.6 abstracts).[7] The numbers suggest that the negative effect of missing mappings on search results, as it would be expected based on the moderate recall figures stated above, is probably less serious in practice.

The terminological heterogeneity of protein and gene names might raise concerns about

[7] The analysis is based on four million Medline abstracts that were published between 1990 and 2008. They were automatically annotated with genes and proteins using the gene name normalizer Geno [Wermter et al., 2009].

the size of the gold standard that has been used for the evaluation. It contains mappings for 550 of the 19,052 UniProtKB entries considered for the matching task. To assess the plausibility of the evaluation performed on this gold standard, assume that the random sample of 550 UniProtKB entries would have been drawn from an infinite set of entries, and the precision and recall estimates, resulting from the comparison of the automatically detected mappings with the gold standard, would be 0.5 (50%). Then the standard deviation of the estimates would be ± 2.1, which is acceptable. Since the sample was in fact drawn from the finite set of 19,052 UniProtKB entries and the determined precision and recall values are considerably higher than 0.5 the real standard deviation is even lower than the stated upper limit for it.

The Protein alignment is currently restricted to UniProtKB entries representing *human* proteins. However, the results of a preliminary study on UniProtKB entries for 29 important model organisms indicate that the proposed matching approach is able to detect mappings to MeSH headings for a similar proportion of protein entries as for human, evidencing that the alignment could easily be extended. Finally, it is important to notice that the alignment is based on a thesaurus and a domain-specific database as input. Accordingly, it is neither an ontology alignment nor ontological background knowledge in the strict sense. The implications of this fact (including the fact that no strict ‘is-a’ relationships are provided between entries of the input resources) must be considered when using the alignment in applications. However, the alignment has been designed for information retrieval purposes, for which thesaurus-style background knowledge is commonly accepted as being sufficient.

To sum up, by bridging parts of the protein database UniProtKB and the MeSH thesaurus, the Protein alignment facilitates the creation of a hierarchically structured protein facet for semantic document retrieval. This facet enables a novel combination of advanced strategies for protein-centered search. The alignment itself has been created by a language-based automatic matching approach that showed a decent performance on a manually created gold standard, in particular with respect to precision. An evaluation that measures the effect of the Protein alignment on searching in real retrieval settings is pending. However, on the basis of exemplary searches it has been demonstrated which gains can be expected for retrieval results (see section 9.5).

11.6 Consolidation of Ontology Reference Alignments

The Anatomy, Lod and Bridge alignments were evaluated regarding aspects of their validity and reusability (see chapter 10). The evaluations were carried out by running basic quality checks on the alignments, as proposed in section 4.5.5. Running these checks on the mentioned alignments was advisable because the latter have themselves been used as reference standard in evaluations and hence their quality is of topmost importance (the Anatomy alignment is still used as reference alignment in annual evaluation campaigns, see page 37).

Check 1 revealed that the alignments have been published without their precise input ontologies. This seriously hampers their reusability. For our evaluation it meant that the Web had to be searched for missing ontologies and recursive ontology import problems had to be solved. Furthermore, class and label persistence had to be checked in terms of running check 2a, before the remaining checks could be run. Furthermore, it made the results of all remaining checks subject to the caveat that they were run on input ontologies that are possibly not identical with the original ones.

Check 2–4 revealed obstacles and irregularities in the alignment files, such as ambiguous class references, name space confusions, typing errors and formatting mistakes. They hindered the automatic lookup of referenced classes in the respective input ontologies and required time-consuming preparatory work before the remaining checks could be run. Interestingly, even in the Bridge and the Anatomy alignments, which have been published in a machine-processable standard format, spelling mistakes and class label confusions were detected, though considerably fewer than in the Lod alignments, which have been published in manually formatted files. A possible explanation for this is that manually typed lists of class labels, local names or URIs might have been used as input for the automatic creation of the final alignment files. To avoid such mistakes in the future, automatic forms of sanity-checking and data cleansing should be applied to manually created documents, for which proper spelling, case-sensitivity, and the use of special delimiters is crucial.

Check 5 led to the particularly disappointing discovery that the very promising Bridge alignments, which according to their authors contain correspondences based on the relations ‘equivalent-to’, ‘is-a’, and ‘related’, were found to come without relation type encodings, lowering their usefulness for various evaluation scenarios. According to the authors of the alignments, relation type information was not kept when creating the alignments because at that time it was not needed. As a consequence, check 6–10 could not be run on the Bridge alignments because these checks require relation type encodings.

By far the most interesting evaluation results were achieved for the Anatomy alignment. Based on check 2b, 6 and 8, a total of 30 erroneous correspondences were detected that should be removed from the alignment (this accounts for 2% of the complete alignment and 5% of its non-trivial subset) and 14 missing correspondences that should be added. The proposed changes were communicated to the organizers of the OAEI Anatomy track, who reviewed and accepted them, changed the alignment accordingly and use the new version of the alignment in the OAEI evaluation campaigns since 2011 [Euzenat et al., 2011a], a circumstance that clearly shows the relevance and effectiveness of these basic quality checks in practice.

Furthermore, check 10 revealed that two-thirds of the correspondences in the Anatomy alignment are trivial, i.e., they can be detected by a simple string matching tool. Since the alignment is quite large with respect to the number of correspondences contained, the remaining third still constitutes a valuable reference for evaluation. However, the predominance of trivial correspondences should be considered when interpreting the results that matching systems achieve on this reference alignment or when comparing the results to the ones achieved by the same systems on different reference alignments. In fact, the OAEI Anatomy track organizers are aware of this limitation and compute, in addition to standard recall and precision, a measure which they call “recall+” [Aguirre et al., 2012]. It refers to the non-trivial correspondences that a system is able to detect.

For the Lod alignments, based on check 6 ten erroneous ‘equivalent-to’-based correspondences were detected that should be removed. Furthermore, based on checks 8 and 9, a total of 35 missing ‘equivalent-to’-based correspondences and based on check 9 a total of 52 ‘is-a’-based correspondences were detected that should be added. If in addition the full results from check 7 would be considered, the number of newly proposed ‘is-a’-based correspondences would be even higher. Whether ‘is-a’-based correspondences should be included in an alignment is a design choice, as well as whether automatically inferable ‘is-a’-based correspondences should be included, as identified by check 7. However, a clear decision should be made and either all or none of the automatically inferable ‘is-a’-based correspondences included in the alignment, but not only some of them, as in the Lod alignments. As mentioned above, check 6–10 could not be run on the Bridge alignments due to missing relation type encodings.

To sum up, running the ten basic quality checks on the Anatomy, Lod and Bridge alignments revealed suggestions for improvement for all three datasets. Despite the fact that the alignments have already been used as reference alignments in evaluations, shortcomings of them were detected that seriously hamper their validity and reusability. The checks revealed issues that concern all three datasets, such as the failure to provide the original input ontologies of the alignments, and it made individual strengths and weaknesses of the three datasets apparent. For the Lod alignments suggestions for improvement could be derived from all ten checks. For the Lod and the Bridge alignments the format checks proved to be particularly helpful to substantially improve their usability. For the Anatomy alignment, which has been used in public evaluation campaigns for several years, the checks at the content level rendered very positive effects. Particularly encouraging is the fact that based on the suggestions made in the context of this thesis, the Anatomy alignment has been revised and the new version is now used in the Anatomy track of the OAEI evaluation campaigns, increasing the strength of evaluation results achieved in this track.

11.7 Realization of Life Cycle Model

The approach to building ontological background knowledge for biomedicine, proposed in the context of this thesis, consisting of a five-staged life cycle model for biomedical ontologies and ontology alignments, has been implemented in five different case studies (see chapters 6–10). Since novel and useful resources resulted from the first four case studies (*viz.*, GRO, MaHCO, BioTop and the Protein alignment) and existing resources have been enhanced in the fifth one (*viz.*, the Anatomy, Lod and Bridge alignments) the implementations can be considered successful. The benefit arising from the implementation of the proposed evaluation procedures attracts particular attention. It helped to increase the quality of concrete ontologies and ontology alignments by revealing shortcomings with regard to multiple levels and aspects. Besides individual mistakes, the evaluations carried out also revealed some systematic shortcomings that need to be solved, described below. They constitute valuable suggestions for improvement of the concerned resources.

Evaluating GRO, MaHCO and BioTop revealed that across the three ontologies only few classes have been provided with alternative class labels. Alternative class labels are optional annotations. However, the field of biomedicine is well known for a rich inventory of synonyms [Spasic et al., 2005]. Hence, a low number of alternative class labels in a biomedical ontology can be taken as a hint for a weak lexical coverage. The lexical coverage of an ontology, in turn, has an impact on various applications of ontologies. Examples include automatic concept recognition (i.e., the automatic recognition of verbal mentions of ontology classes in text—an important first step in most fact extraction approaches) [see, e.g., Beisswanger et al., 2008b; Jonquet et al., 2009] and ontology matching [see, e.g., Cruz et al., 2009; Jain et al., 2010; Jiménez-Ruiz and Grau, 2011; Mascardi et al., 2009]. To improve the results of these applications, additional efforts should be made to provide alternative class labels for biomedical ontologies. Semi-automatic tools that support the discovery of class labels from natural language texts may be used for assistance [see, e.g., Wächter and Schroeder, 2010].

The evaluation of GRO, MaHCO and BioTop further revealed that in none of the ontologies the possibilities of OWL DL have been exploited consistently. For example, missing explicit ‘disjoint-with’ relationships between classes and lacking closure axioms have been detected. For OWL ontologies it is particularly important to explicitly specify any intended meaning before applying a reasoner to rule out side effects of the underlying open world assumption [Rector et al., 2004]. Hence, additional efforts should be made to make the three ontologies more expressive by adding missing facts and specifications, additional explicit ‘disjoint-with’ relationships and closure axioms in particular.

The evaluation of GRO and MaHCO revealed that both ontologies neglect to cite the origin of some reused classes and relations. The policy for knowledge reuse in both ontologies is to recreate classes and relations from external ontologies, provide them with new URIs and reference their origin in terms of 'gro:reference' and 'mhc:reference' annotations, respectively. However, for many reused classes and relations such annotations were found to be either missing or suffer from formatting mistakes. The latter hampers the automatic processing of concerned annotations. To allow the benefits of knowledge reuse to take effect, additional efforts should be made to state the origin of *all* reused classes and relations in GRO and MaHCO explicitly in a consistently formatted way that makes them automatically processable.

Overall the proposed approach to building ontological background knowledge for biomedicine has successfully been implemented in the above-mentioned case studies. However, there is one major shortcoming with the implementations, *viz.*, the incomplete or missing implementation of some of the stages of the underlying life cycle model. In the case studies on GRO, MaHCO and BioTop the evaluation stage has been omitted at development time (the comprehensive evaluations of the guideline compliance of GRO, MaHCO and BioTop, presented in sections 6.3, 7.3 and 8.3, have been carried out retrospectively). In the case study on the Protein alignment the maintenance stage has been omitted. Finally, in the case studies on GRO and MaHCO maintenance has been discontinued. Below, the incomplete implementation of the evaluation and maintenance stage in different case studies is discussed.

Evaluation Issues

Skipping the evaluation stage of ontology development and instead evaluate ontologies retrospectively seems to be a widespread pattern, beyond the work presented in this thesis. For many important biomedical ontologies in the OBO library the corresponding primary publications do not mention a proper ontology evaluation. For example, Ashburner et al. [2000] introduce GO, Eilbeck et al. [2005] SO, Bard et al. [2005] CL, Natale et al. [2011] PRO and Degtyarenko et al. [2008] ChEBI. At most, they mention certain aspects of evaluation, such as the usefulness of an ontology for a particular purpose [see, e.g., Ashburner et al., 2000; Eilbeck et al., 2005] and its conceptual coverage [see, e.g., Bard et al., 2005]. An exception is the work by Rosse and Mejino [2003]. They introduce the FMA and announce a comprehensive evaluation of it. However, even in this case, large parts of the evaluation were pending at publication time. Analogously to the retrospective evaluations carried out in the context of this thesis, for some of the mentioned ontologies subsequent evaluations have been run. For example, Smith et al. [2003] carried out a critical analysis of the structure of GO, Ogren et al. [2004] studied the compositional structure of GO terms, and Zhang and Bodenreider [2006] evaluated the adherence of the FMA to a set of ontological modeling principles.

In the literature, a broad spectrum of ontology evaluation approaches has been proposed (see section 3.6). However, the missing evaluation of ontologies at development time—both in internal and external ontology projects—indicates that evaluation has not yet become an integral part of practical ontology development. To improve the situation, obviously, the transfer between the theoretical and the practical level of ontology evaluation must be improved. Two important aspects to be considered in this context are *education* and *tooling*. Regarding education, teaching ontology developers the importance of ontology evaluation (e.g., by addressing ontology evaluation more prominently in ontology development guides, which so far mainly focus on the design and implementation of ontologies) could strengthen practical ontology development. Interestingly, even the OBO Foundry principles currently do not claim the systematic evaluation of ontologies, although in other respects they exceed classical ontology development guides by far. There is a preliminary OBO Foundry principle that stipulates that "The ontology must be a faithful representation of the domain and fit for the stated purpose." [OBO Foundry, 2012, principle FP19 (under review)]. To further promote quality assurance for ontologies, this implicit claim should be tightened towards an explicit claim for a thorough intrinsic and extrinsic evaluation of ontologies.

With regard to new tools for ontology development first steps have been taken in the context of this thesis. A three-step evaluation procedure for checking the guideline compliance of ontologies has been proposed, of which parts have been implemented in a configurable way. To further improve the usability of the procedure, its three steps could be merged and implemented, as far as possible, in terms of a test suite. The latter could automatically be run on ontologies and provide suggestions for improvement, similar to the existing pitfall scanner Oops! (see page 60), which is currently incorporated in the three-step procedure, though on a much broader scale. To increase the visibility of the test suite, it should be designed as plug-in for widely-used ontology editors, such as Protégé. The development of new evaluation procedures and planning of new tools for ontology evaluation in the context of this thesis responds to previously stated demands by Rubin et al. [2008] ("Another future direction for ontology research is in developing metrics for ontology quality and in creating tools to enable the user community to evaluate ontology quality." [Rubin et al., 2008, page 87]) and Obrst et al. [2007], who mention the definition of "defined units of measure" and "well-defined engineering practices" as important next steps in ontology evaluation research [Obrst et al., 2007, page 152].

Maintenance Issues

The reason for the discontinuation of the maintenance of GRO, MaHCO and the PROTEIN alignment is that they have been developed in research projects that provided for the development, but not for the maintenance of the resources. In retrospect, to consider the development of resources decoupled from maintenance was a short-sighted view. Each of the mentioned resources is the result of considerable investments. Since they are not maintained anymore, they are increasingly becoming obsolete, putting the investment made to create them at risk. To preserve the value of the mentioned resources, the maintenance of the resources would have to be resumed.

For large-scale ontology and thesaurus projects, on which public services depend (e.g., GO, the MeSH and the NCI Thesaurus), maintenance procedures have been established. Furthermore, tools have been provided for the maintenance and collaborative development of ontologies [Noy et al., 2010, 2006; Tudorache et al., 2008]. However, neither procedures nor tools are sufficient, if in middle- or small-scale projects there is no staff available who could carry out the maintenance work. Hence, in contrast to ontology evaluation, for which education and tooling have been identified as important aspects to be considered, to achieve the goal of continuous maintenance *planning* and *funding* seem to be important factors. For future ontology and ontology alignment projects, additional efforts should be made in the planning phase. When in this early phase of projects funds and person months would be assigned to the maintenance stage of the particular project, too, the long-term preservation of ontologies could be ensured.

Three further observations on maintenance were made, in the context of this thesis. The first observation is that although the reuse of knowledge is generally accepted as good modeling practice, the fact that it faces ontology developers with additional maintenance work, obviously, is mostly neglected. In fact, ontologies which reuse knowledge from external knowledge resources do not only have to be adapted to changing domain knowledge and new application requirements, but also to changes in the resources from which they reuse knowledge. The same applies to ontology alignments and their input ontologies. In case of GRO, MaHCO and the Protein alignment, already a few years of missing maintenance led to the obsolescence of reused classes, relations and annotations. The classes of MaHCO are concerned in particular, because most of them were derived from domain-specific databases, which themselves are frequently updated with new entries. In spring 2010 the whole HLA nomenclature was changed [Marsh et al., 2010], obsoleting the labels of thousands of MaHCO classes all at once. Neither in the case of GRO, nor in the case of MaHCO community-based maintenance activities took place, although both ontologies had been made freely available to the bio-ontology community. Hence, in order to preserve (or reestablish) the value of the mentioned resources for intended applications, funds for maintenance would have to be allocated that allow to employ staff to continuously update them. Alternatively, for MaHCO HLA, which is comparatively lightweight, it would be worth to test the use of automatic update routines. They could regularly check the databases on which the ontology depends for changes and transfer these changes automatically to the ontology.

The second observation is that obviously it is challenging for ontology developers to keep quality standards up (e.g., to preserve the proportion of annotated or fully defined classes) when the ontology grows. GRO 0.5, for example, contains 15% more classes compared to its precursor GRO 0.4, but 13% fewer classes with a full formal definition, 2% fewer classes with a verbal definition, a 12% lower relationship/class ratio for conceptual relationships and a nearly 13% lower relationship/class ratio for 'disjoint-with' relationships (the numbers were derived from table 6.6). Similar observations can be made regarding external biomedical ontologies. For example, SO from 2012-05-08 contains 173% more classes than SO from 2008-05-26, but a 15% lower proportion of formally defined classes, a 47% lower relationship/class ratio for conceptual relationships and no 'disjoint-with' relationships at all anymore. Furthermore, PRO from 2012-05-08 contains 4,211% more classes than PRO from 2008-05-26 (which represents a very early developmental state of PRO), but an 80% lower relationship/class ratio for conceptual relationships (the numbers on SO and PRO were derived from table 11.1). Based on the assumption that the decreased proportions of formally and verbally defined classes and the decreased relationship/class ratios for conceptual and 'disjoint-with' relationships are not the result of design decisions, additional attention should be payed to the validation step in the maintenance cycle of ontologies to keep quality standards up. Running the same simple tools as they were proposed for the evaluation stage of the ontology life cycle could help to recognize missing pieces of information, such as formal or verbal class definitions. In addition, tools that check the preservation of ontology contents across ontology releases would be beneficial (see section 4.7).

The third observation is that obviously the use of ontologies and ontology alignments in concrete applications has a strong positive effect on the respective resources by triggering maintenance activities. The concrete use of resources does not only expose possible errors that should be fixed, but it also poses new demands on resources that require their improvement, refinement or extension. For example, GRO has been used by Kim and Rebholz-Schuhmann [2011] for information extraction tasks and by Kim et al. [2010] for the construction of formal definitions for GO classes. In the context of these works, 70 new classes and eight new relation types were requested, of which all classes (some with slight revisions) and six relations were added to GRO. These changes led to the current version GRO 0.5. As another example, the alignment of BioTop and the UMLS Semantic Network revealed shortcomings of BioTop, such as missing classes, faulty 'disjoint-with' relationships between classes, unrecognized ambiguities and granularity mismatches Schulz et al. [2009a], which should be resolved. Similar experiences are reported by Meehan et al. [2011]. They mention that the process of interlinking the Cell Ontology with other ontologies by introducing appropriate formal class definitions exposed conceptual gaps in the ontology that required the addition of new classes. Accordingly, to accelerate the development process and to trigger maintenance of ontologies and ontology alignments, they should extensively be tested in practice by using them in different applications.

To generally prevent that in future ontology and ontology alignment projects life cycle stages are incompletely implemented or even skipped, ontology and ontology alignment management tools should be used. They could assist developers in project planning, scheduling and execution. In particular, they could assist in selecting an appropriate life cycle model, schedule the processes and activities involved in ontology development, and implement the selected model, the latter by informing the ontology developer about guidelines and tools associated to upcoming processes and activities. First examples for such tools are already available [Noy et al., 2010; Suárez-Figueroa et al., 2010]. Promising is the work by Noy et al. [2010]. They initiated the integration of established tools (basically WebProtégé and the BioPortal) to support ontology development across different life cycle stages, ranging from their implementation to their public release and maintenance.

11.8 Summary

By means of five different case studies, dealing with the development of GRO, MaHCO, BioTop and the Protein alignment, as well as the evaluation of the Anatomy, Lod and Bridge alignments, the effectiveness of the approach to building ontological background knowledge for biomedicine, which has been proposed in the context of this thesis, has been demonstrated. In particular, the evaluation procedures that the approach incorporates have shown to be effective in practice. With GRO, MaHCO, BioTop and the Protein alignment, four novel resources have been developed. In addition, important suggestions for improvement of existing ontology reference alignments have been achieved.

The case studies demonstrate that very diverse applications benefit from ontological background knowledge for biomedicine, utilizing different facets of it. While some applications profit from the proper organization of domain knowledge, others utilize the structured controlled vocabulary, the explicit semantics, or the possibility of automatic classification and logical consistency checking provided by ontologies. In order to satisfy the requirements of intended applications, it is important to consider the respective requirements profile already early in the development of ontologies and ontology alignments.

The case studies on GRO, MaHCO, BioTop and the Protein alignment illustrate that given the need for ontological background knowledge for biomedicine, in some cases the development of new ontologies is appropriate, while in others the creation of an alignment of existing knowledge resources is more beneficial. Furthermore, the case studies on GRO, MaHCO and BioTop demonstrate that in case that a new ontology is developed, there are different possibilities of involving and reusing existing knowledge resources. While BioTop directly revises and extends an existing ontology and imports two additional ones, all three ontologies adopt individual classes or relations from external ontologies. MaHCO even contains classes that were derived from domain-specific databases. However, the case studies also delivered evidence that the benefits of knowledge reuse can only take effect if it is carried out in an explicit and transparent way and the adopted knowledge is regularly maintained.

All five case studies demonstrate the benefit of a thorough evaluation and the continuous maintenance of biomedical ontologies or ontology alignments, respectively. While the evaluation helps to ensure that the respective resources are valid, reusable and adhere to important design and implementation guidelines, continuous maintenance is important to regularly update them and keep them effective for intended applications. The evaluation of ontology reference alignments is of particular importance because their quality defines the strength of the evaluation results that depend on them.

The case studies further revealed that to guide ontology developers in the implementation of the proposed approach to building ontological background knowledge for biomedicine, as well as the evaluation procedures that it incorporates, the provision of appropriate ontology and ontology alignment management and evaluation tools would be beneficial.

Part IV

Conclusions

Chapter 12

Conclusions

Today, biomedicine is one of the few areas in which ontologies are used on a large scale in practice. They are mainly utilized as background knowledge in applications to cope with the large volumes of biomedical data. The continuous generation of new biomedical data and development of new tools to manage it result in a constant demand for new ontological background knowledge. From an ontology engineering perspective this makes biomedicine a particularly relevant field of application. However, it is also a field that poses specific challenges on ontologies and their creation, which cannot reliably be met by standard approaches to ontology development. In order to cope with these challenges, in the context of this thesis an approach to building ontological background knowledge was developed that is tailored to biomedicine as field of application. The effectiveness of this approach in practice has been shown in several case studies. They resulted in three new biomedical ontologies, one new biomedical alignment and the improvement of existing ontology reference alignments.

12.1 Contributions

In the context of this thesis an approach to building ontological background for biomedicine has been developed that is based on a five-staged life cycle model for biomedical ontologies and ontology alignments. It stands out from existing approaches in several respects.

First, the proposed approach combines the conventional strategy of ontology development with the complementary strategy of ontology matching, which is still often neglected in practice. This combination of strategies is novel and increases the flexibility of our approach. There are cases, in which ontological background knowledge is required that already exists, but is scattered over different ontologies. While the development of new ontologies would lead to redundancy and further knowledge fragmentation, the creation of ontology alignments would help to bridge ontologies and hence foster knowledge integration. The proposed approach is flexible enough to respond to such cases. Its effectiveness in practice was demonstrated in different case studies, carried out in the context of this thesis. In three of them ontologies and in one an alignment were developed. Given that the number of cases, in which ontology matching as strategy to building ontological background knowledge would be advantageous, will grow with the number of available biomedical ontologies. Then the chosen two-track strategy will become increasingly important.

Second, the life cycle model for ontologies and ontology alignments, on which the proposed approach to building ontological background for biomedicine relies, is specifically tailored to biomedicine as field of application. Each stage of the model responds to specific characteristics of this field. For example, the "requirements analysis" stage helps to ensure that useful ontological background knowledge is developed, despite the very heterogeneous requirements that different biomedical applications pose on it. The "design and implementation" stage encourages to express domain knowledge in form of axioms to rule out unintended interpretations that often occur in interdisciplinary fields like biomedicine. The "documentation and release" stage promotes sharing and reuse of ontological background knowledge, which is an important prerequisite for achieving ontology-mediated interoperability of biomedical databases and knowledge stores. In addition, the "maintenance" stage allows to regularly update ontological background knowledge and adapt it to the frequent changes that occur in a highly dynamic field like biomedicine. Overall, the domain adaption of the life cycle model facilitates the development of valid, useful and reusable ontological background knowledge even for the challenging field of biomedicine.

Third, the proposed life cycle model excels in providing an in-depth description of the individual life cycle stages. For each stage that it comprises associated subtasks and activities are specified at a granularity as it is known from practical guides for ontology development. The major advantage of this detailed description is that it facilitates the implementation of the model, and hence promotes its use and benefit in practice.

Fourth, the proposed approach to building ontological background knowledge for biomedicine incorporates elaborate evaluation approaches for ontologies and ontology alignments. They include a three-step procedure for checking the guideline compliance of ontologies, as well as a set of quality checks for testing basic aspects of the validity and reusability of ontology alignments. The proposed procedures are highly explicit. They assess the quality of ontologies and ontology alignments with regard to a broad range of aspects, including some hitherto neglected, technical ones. While from an ontologist's perspective the latter might seem secondary, in practice they have apparently a strong influence on the (re)usability of ontologies and ontology alignments.

In several case studies, carried out in the context of this thesis, the three-step procedure for checking the guideline compliance of ontologies has shown to be an effective means to improve the quality of ontologies on various levels. In addition, the basic quality checks for ontology alignments have proven to effectively increase the validity, accessibility and (re)usability of the latter and, as a positive side effect, the quality of their input ontologies. The relevance of these basic checks in practice is clearly stressed by the fact that based on results of them the reference alignment of the Anatomy track of the annual OAEI evaluation campaigns and the anatomy branch of the NCI Thesaurus, as one of its input resources, have both been improved and enhanced. Meanwhile updated versions of both artifacts are used in practice, providing a strong basis for upcoming evaluations and applications.

In addition to this new approach to building ontological background knowledge for biomedicine, four concrete knowledge resources have been developed in the context of this thesis. They comprise the biomedical domain ontologies GRO and MaHCO, a basic version of the top domain ontology for molecular biology and biomedicine BioTop and the Protein alignment, linking parts of the protein database UniProtKB and the broad-coverage thesaurus MeSH. Each of these resources contributes substantially to existing ontological background knowledge for biomedicine. They structure and formalize previously unstructured domain knowledge, and hence make it computationally accessible and utilizable for various applications. In addition, they enhance already formalized knowledge by integrating and linking it, which so far has been scattered over various mostly disconnected knowledge resources. From the perspective of current biomedical and clinical research, GRO, MaHCO and the Protein alignment deal with particularly important subdomains of biomedicine. The regulation of gene expression, modeled by GRO, plays a crucial role in various biomolecular contexts. They range from developmental and metabolic regulation to dysregulation causing disease. Furthermore, differences in the major histocompatibility complex, represented by MaHCO, determine the compatibility of donors and recipients for organ and hematopoietic stem cell transplantations. Finally, a plurality of proteins, as they are hierarchically organized by the Protein alignment, are involved in nearly every cellular process. For example, they catalyze biochemical reactions and perform structural and mechanical functions.

In the context of this thesis, it was shown in various applications that the newly developed resources do not only cover relevant contents, but are also useful in practice. The use cases range from biomedical NLP tasks and applications, such as the domain-specific semantic annotation of text corpora, fact extraction and document retrieval based on faceted search, to applications from the field of computational immunology, dealing with browsing and comparing domain-specific factual data. The mentioned applications utilize background knowledge very differently. However, each of them in some respect profits from the availability of *ontological*, as compared to simpler forms of background knowledge. They profit either from the strict ontological classification and hierarchical organization of domain-specific entities, from the computational accessibility of domain knowledge that allows for automatic verification, inferencing and search, or from the interlinkage of resources, which allows to browse, inference on and search knowledge across resources. The importance of the mentioned applications lies in their potential to assist biomedical and clinical researchers in coping with the increasing amounts of available data and domain knowledge. While applications such as document and fact retrieval can assist researches in planning new "wet lab" studies, applications such as corpus annotation and automatic fact extraction from text foster automatic biocuration and facilitate studies "in silico". The potential of the domain ontology GRO, developed in the context of this thesis, for automatic biocuration will be studied on a larger scale in the upcoming BioNLP Shared Task 2013, a widely noticed international competition on biomedical information extraction.

In the course of this thesis, several factors were identified that obviously have a strong impact on the successful creation and curation of ontological background knowledge for biomedicine. The choice of an appropriate life cycle model and the adherence to design and implementation guidelines during ontology development were identified as dominant factors for the development of clearly structured, expressive, well annotated, properly documented, useful and reusable ontologies. Evaluation (during development) and validation (during maintenance) were identified as factors of crucial importance for assuring the quality of ontological background knowledge before it is used in practice. Maintenance was found to be vital for the constant improvement and enhancement of ontological background knowledge and its adaption to new requirements from the underlying field of application. In the context of maintenance, versioning was found to be an important factor in order to prevent confusions between different versions of ontologies or ontology alignments. Furthermore, the joint release of alignments with their original input ontologies was found to be crucial for the applicability of alignments as reference standards in evaluations. Practical use cases were found to be a factor with a strong influence on maintenance activities and hence the progress of ontological background knowledge. Knowledge reuse (ranging from the reuse of ontology elements to the creation of ontology alignments) was identified as a crucial factor of preventing knowledge fragmentation and redundancy and promoting instead knowledge integration and interoperability between knowledge-based systems. However, knowledge reuse was found to be only effective if it is carried out in a transparent way and if the reused knowledge is maintained.

As additional factors of the successful creation of ontological background knowledge for biomedicine and beyond, the availability of appropriate tools, the education of developers and the availability of funding were identified. These factors will be addressed in the perspectives below. The mentioned factors of the successful creation of ontological background knowledge are summarized in appendix A, in the form of suggestions for biomedical ontology developers.

12.2 Perspectives

In the context of this thesis, a large gap between theory and practice in biomedical ontology development was noticed. Important issues of ontology development, for which approaches and solutions have been provided on the theoretical level, were found to be still unsolved in practice. For example, the comprehensive literature on ontology evaluation contrasts with a weak evaluation culture in practice. To achieve further progress in the field of biomedical ontology development, the mentioned gap must be bridged. To achieve this goal, in this thesis guidelines and quality standards for the process of developing ontologies and ontology alignments were promoted. In addition, procedures were provided to test whether the guidelines and quality standards are adhered to in practice. However, a lot of effort is still required. It is expectable that the key to success will be strongly associated with the above-mentioned factors—tools, education, planning and funding—and their mutual impact.

Tools are a proven means to transfer methodical advances in ontology research into practice. For example, ontology editors like Protégé-OWL [Knublauch et al., 2004] facilitate the implementation of ontologies in the state-of-the-art ontology language OWL, without requiring programing skills. As another example, reasoning tools like HermiT [Motik et al., 2009] enable the efficient consistency checking of OWL ontologies, without requiring the understanding of complex reasoning algorithms. However, in the context of this thesis three gaps in current tool support were identified that would be worthwhile to be filled. First, ontology management tools should be created that guide developers through the complete life cycle of ontologies or ontology alignments. Such tools could ensure that all important life cycle stages and associated tasks are considered and adequately implemented in practice. First steps towards such tools have already been taken [Noy et al., 2010; Suárez-Figueroa et al., 2010], whereas the proposal to integrate and extend already established tools (*viz.*, WebProtégé and the BioPortal) by Noy et al. [2010] is particularly promising. Second, evaluation tools should be provided that assist ontology and ontology alignment developers by checking if important development guidelines and quality criteria were adhered to. The evaluation procedures that have been proposed in the context of this thesis could be translated into such tools. They would then complement and extend simple pitfall scanning tools, such as Oops! [Poveda-Villalon et al., 2012]. Third, tools should be provided to check the persistence of contents of ontologies and ontology alignments across different versions. They could ensure that no content was lost during the update step of a maintenance cycle.

The specialized education and training of ontology developers could equip them with core competencies and skills required to successfully create and curate ontological background knowledge. For example, a simple way of teaching ontology developers the importance of ontology evaluation and maintenance would be extending the focus of practical ontology development guides from design and implementation of ontologies to the whole ontology life cycle. A more expensive way would be the selective training of ontology developers. Courses could be offered to students and researchers in areas such as biomedicine and bioinformatics, providing them with skills required to create and curate ontological background knowledge. First steps in this direction have been taken by Boeker et al. [2012], who compiled and tested a curriculum that trains students with a background in biomedicine and computer science in the practical development of biomedical ontologies using Protégé-OWL. Neuhaus et al. [2011] go even further and propose a complete course of studies intended to create “the ontologists of the future”. They recommend a body of knowledge that should be taught to future ontologists and skills they should develop. Each of the mentioned approaches could raise the awareness of critical issues of ontology and ontology alignment development, such as the evaluation and maintenance of newly developed resources.

Finally, sufficient planning and appropriate funding models could contribute to the successful creation and curation of ontological background knowledge. In the context of this thesis, both factors were observed to be instrumental for whether enough time and resources are available to properly execute ontology and ontology alignment projects. Classical research projects with a limited run time have turned out to be a suboptimal setting for ontology and ontology alignment development. Once they terminate, funding expires and the responsible persons leave the project, leaving the developed resources unattended. Missing maintenance, in turn, rapidly decreases the value of knowledge resources, especially when they concern a dynamic field of application, such as biomedicine. Which funding models in the academic context (outside large national institutions) could supply long-term maintenance is currently an open issue. Until it is answered, strategies should be investigated how to save maintenance costs. A currently popular strategy is to share ontological background knowledge with the community and trust on the corrective power of collaborative effort. However, at least in case of the two domain ontologies developed in the context of this thesis (*viz.*, GRO and MaHCO) this strategy did not prevent that the ontologies become obsolete.[1] Work on tool support that could further increase the effectiveness of the community-based strategy, embedded in the NCBO BioPortal, is under way [Noy et al., 2009, 2010]. An alternative strategy to reduce maintenance costs would be to establish automatic update mechanisms for ontologies and ontology alignments. Given, for example, an ontology that imports portions of another one, then the second ontology could automatically be checked for changes and possible changes could be transfered to the first ontology. However, a drawback in this case would be that a manual post-processing step would be required to repair the ontology in case that logical inconsistencies were introduced through the update and to check the empirical plausibility of the updated ontology.

[1] In fact, there were requests for further improvements and extensions of GRO, but none of the interested parties updated the ontology and put it back to the OBO library or NCBO BioPortal.

12.3 Final Remarks

This work responds to several future directions that have been specified by leading scientists in the area of biomedical ontologies. Three future directions specified by Bodenreider and Stevens [2006] in their survey article titled “Bio-ontologies: current trends and future directions” are the establishment of scientific techniques for building ontologies, the further development of ontology validation and certification, and the precise capture of biological knowledge in a computational form through formal and expressive biomedical ontologies [Bodenreider and Stevens, 2006, page 269]. In this thesis, the three directions are tackled in the form of a comprehensive approach to building ontological background knowledge for biomedicine, evaluation approaches for biomedical ontologies and their application in practice, and a guideline-based approach to ontology design and implementation, which promotes the creation of formal and expressive ontologies.

In their survey article on practical uses of biomedical ontologies, Rubin et al. [2008] specify the increasing importance of ontology matching—as a means to relate existing ontologies to one another—and the establishment of tools that enable the evaluation of the quality of ontologies as two additional future directions of biomedical ontologies [Rubin et al., 2008, pages 86 and 87]. In this thesis, the first direction is responded to by an approach to building ontological background knowledge for biomedicine that incorporates ontology matching as a complementary strategy to ontology development. The second direction is addressed by elaborate evaluation procedures for assessing the quality of biomedical ontologies and ontology alignments, which can be translated into the requested tools. The mentioned evaluation procedures, in combination with standard evaluation approaches for ontologies and ontology alignments—such as checking their logical consistency—should be run on all biomedical ontologies and ontology alignments before they are publicly released and used in practice. Although the procedures have been compiled for biomedical ontological background knowledge, to large parts they are general enough to be applied to other forms of ontological background knowledge.

Part V

Appendix

Appendix A

Suggestions for Biomedical Ontology Developers

Before starting a new ontology or ontology alignment project:

1. Ensure that funding for development *and* maintenance is secured.
2. Ensure that persons are in charge of the development task who have the necessary knowledge and skills.
3. Consider the revision, extension or matching of existing ontologies as alternative to developing ontologies from scratch.
4. Choose an appropriate life cycle model for the new ontology or ontology alignment.

Running a new ontology or ontology alignment project:

1. Carry out a thorough requirements analysis.
2. Use a standard ontology language (e.g., OWL) and exploit available tools.
3. Adhere to approved design and implementation guidelines.
4. Reuse knowledge and explicitly cite its origin.

5. Evaluate your ontology or ontology alignment, also intrinsically.
   (a) Check the logical consistency.
   (b) Check the correctness and coverage of contents.
   (c) Check the compliance with design and implementation guidelines.
6. Test your ontology or ontology alignment in practical use cases.
7. Document your ontology or ontology alignment.
   (a) Publish a research paper or technical report.
   (b) Provide meta data annotations in the corresponding ontology or ontology alignment document.
8. Make your ontology or ontology alignment publicly available.
   (a) Include a version number.
   (b) For ontology alignments, include the original input ontologies.
9. Maintain your ontology or ontology alignment to preserve and enhance its value.

Appendix B

Example Ontology

Below, the classes and conceptual relationships are listed that are contained in the example ontology introduced in chapter 4. Classes and relationships below the horizontal line are only contained in the extended version of the example ontology (see page 68).

Conceptual Relations

1. ‘has-agent’
2. ‘has-patient’

Classes

1. ‘Entity’
2. ‘Continuant’
    - Continuant $\sqsubseteq$ Entity
3. ‘Occurrent’
    - Occurrent $\sqsubseteq$ Entity
4. ‘TranscriptionRegulator’

   - TranscriptionRegulator $\sqsubseteq$ Continuant

5. ‘RegulationOfTranscription’

   - RegulationOfTranscription $\sqsubseteq$ Occurrent
   - RegulationOfTranscription $\sqsubseteq \exists$ has-patient.Transcription
   - RegulationOfTranscription $\sqsubseteq \exists$ has-agent.TranscriptionRegulator

6. ‘Transcription’

   - Transcription $\sqsubseteq$ Occurrent

7. ‘TranslationRegulator’

   - TranslationRegulator $\sqsubseteq$ Continuant

8. ‘RegulationOfTranslation’

   - RegulationOfTranslation $\sqsubseteq$ Occurrent
   - RegulationOfTranslation $\sqsubseteq \exists$ has-patient.Translation
   - RegulationOfTranslation $\sqsubseteq \exists$ has-agent.TranslationRegulator

9. ‘Translation’

   - Translation $\sqsubseteq$ Occurrent

APPENDIX C

Constructors and Axioms

C.1 Constructors

In OWL DL, the following class and property constructors are available:

$$C_i \sqcap C_j \quad \text{(intersection)} \tag{C.1}$$

$$C_i \sqcup C_j \quad \text{(union)} \tag{C.2}$$

$$\neg C \quad \text{(complement)} \tag{C.3}$$

$$\{o_1, \ldots, o_k\} \quad \text{(enumeration)} \tag{C.4}$$

$$\exists R.C \quad \text{(existential restriction)} \tag{C.5}$$

$$\forall R.C \quad \text{(universal restriction)} \tag{C.6}$$

$$R.o \quad \text{(has-value restriction)} \tag{C.7}$$

$$\geqslant_n R \quad \text{(unqualified at-least restriction)} \tag{C.8}$$

$$\leqslant_n R \quad \text{(unqualified at-most restriction)} \tag{C.9}$$

$$\exists U.D \quad \text{(existential data restriction)} \tag{C.10}$$

$$\forall U.D \quad \text{(universal data restriction)} \tag{C.11}$$

$$U.v \quad \text{(has-value data restriction)} \tag{C.12}$$

$$\geqslant_n U \quad \text{(unqualified at-least data restriction)} \tag{C.13}$$

$$\leqslant_n U \quad \text{(unqualified at-most data restriction)} \tag{C.14}$$

$$R^{-} \quad \text{(inverse),} \tag{C.15}$$

where C, C_i and C_j denote classes, D a data range, R an object property, U a datatype property, o and $o_1, \ldots, o_k$ individuals, v a data value and k and n natural numbers.

C.2 Axioms

Axioms are subdivided into terminological and assertional axioms. Terminological axioms describe the relations between classes and between properties. OWL DL supports the following terminological axioms:

$$C_i \sqsubseteq C_j \quad \text{(class subsumption)} \tag{C.16}$$
$$C_i \equiv C_j \quad \text{(class equivalence)} \tag{C.17}$$
$$C_i \sqcap C_j \sqsubseteq \bot \quad \text{(class disjointness)} \tag{C.18}$$
$$R_i \sqsubseteq R_j \quad \text{(object property subsumption)} \tag{C.19}$$
$$R_i \equiv R_j \quad \text{(object property equivalence)} \tag{C.20}$$
$$U_i \sqsubseteq U_j \quad \text{(data property subsumption)} \tag{C.21}$$
$$U_i \equiv U_j \quad \text{(data property equivalence)} \tag{C.22}$$
$$Tr(R) \quad \text{(transitive object property),} \tag{C.23}$$

where C_i and C_j denote classes, R_i and R_j object properties and U_i and U_j datatype properties.

Assertional axioms (also called “facts” or “assertions”) describe the nature of individuals. OWL DL supports the following assertional axioms:

$$o_i = o_j \quad \text{(individual equality)} \tag{C.24}$$
$$o_i \neq o_j, i \neq j \quad \text{(individual inequality)} \tag{C.25}$$
$$C(o) \quad \text{(class assertion)} \tag{C.26}$$
$$R(o_i, o_j) \quad \text{(object property assertion)} \tag{C.27}$$
$$U(o, v) \quad \text{(datatype property assertion),} \tag{C.28}$$

where C denotes a class, R an object property, U a datatype property, o, o_i and o_j individuals and v a data value.

Using the above mentioned constructors and axioms, domain and range restrictions and further characteristics of object and datatype properties may be expressed as follows

$$\geqslant_1 R \sqsubseteq C \quad \text{(domain of object property)} \tag{C.29}$$
$$\top \sqsubseteq \forall R.C \quad \text{(range of object property)} \tag{C.30}$$
$$\geqslant_1 U \sqsubseteq C \quad \text{(domain of datatype property)} \tag{C.31}$$

$$\top \sqsubseteq \forall U.D \quad \text{(range of datatype property)} \tag{C.32}$$

$$R \equiv R^{-} \quad \text{(symmetric object property)} \tag{C.33}$$

$$\top \sqsubseteq \leqslant_1 R \quad \text{(functional object property)} \tag{C.34}$$

$$\top \sqsubseteq \leqslant_1 R^{-} \quad \text{(inverse functional object property)} \tag{C.35}$$

$$\top \sqsubseteq \leqslant_1 U \quad \text{(functional datatype property),} \tag{C.36}$$

where R and U denote an object and a datatype property, respectively, C a class and D a data range.

Appendix D

Supplemental Tables

Prefix	Namespace
gro	`http://www.bootstrep.eu/ontology/GRO#`
mhc	`http://purl.org/stemnet/MHC#`
dc	`http://purl.org/dc/elements/1.1/`
rdfs	`http://www.w3.org/2000/01/rdf-schema#`
owl	`http://www.w3.org/2002/07/owl#`
skos	`http://www.w3.org/2004/02/skos/core#`

Table D.1: Prefix-namespace mappings used in this thesis.

ID	Pitfall	GRO	MaHCO	BioTop
P04	creating unconnected ontology elements	2	-	1
P07	merging different concepts in the same class	3	-	2
P08	missing annotations	536	7,935	48
P11	missing domain or range in properties	28	5	32
P13	missing inverse relationships	8	4	32
P20	swapping label and comment	-	1	2
P22	using different naming criteria in the ontology	1	1	-
P24	using recursive definition	1	-	1

Table D.2: Modeling pitfalls detected by Oops! in GRO, MaHCO and BioTop. The evaluation results of GRO are presented in detail on page 101, of MaHCO on page 118 and of BioTop on page 131.

ID	Foundry Principle	GRO	MaHCO	BioTop
FP01	Open	1	1	1
FP02	Format	1	1	1
FP03	URIs	1	1	1
FP04	Versioning	1	1	1
FP05	Delineated content	0	0	0
FP06	Textual definitions	1	0	0
FP07	Relations	0	0	1
FP08	Documented	1	1	1
FP09	Users	-1	-1	-1
FP10	Collaboration	-1	0	0
FP11	Locus of authority	1	1	1
FP12	Naming conventions	0	0	1
FP16	Maintenance	-1	-1	1

Table D.3: Adherence of GRO, MaHCO and BioTop to “accepted” OBO Foundry principles, where “1” denotes full, “0” partial, and “-1” missing adherence. The evaluation results of GRO are presented in detail on page 101, of MaHCO on page 118 and of BioTop on page 131.

Appendix E

Acronyms

API	Application Programming Interface
BFO	Basic Formal Ontology
CCO	Cell Cycle Ontology
ChEBI	Ontology of Chemical Entities of Biological Interest
CL	Cell Ontology
CV	controlled vocabulary
DAG	directed acyclic graph
DL	description logic
DLA	dog leukocyte antigen
DNA	deoxyribonucleic acid
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
FMA	Foundational Model of Anatomy
GO	Gene Ontology
GRO	Gene Regulation Ontology
GUI	graphical user interface
GvHD	graft versus host disease
GvL	graft versus leukemia
HLA	human leukocyte antigen

HSCT	hematopoietic stem cell transplantation
IEDB	Immune Epitope Database
IMGT	ImMunoGeneTics
IMR	INOH Molecule Role Ontology
LOD	Linked Open Data
MaHCO	Major Histocompatibility Complex Ontology
MeSH	Medical Subject Headings
MHC	major histocompatibility complex
mRNA	messenger RNA
NER	named entity recognition
NLP	natural language processing
OAEI	Ontology Alignment Evaluation Initiative
OBO	Open Biomedical Ontologies
OBR	Ontology of Biomedical Reality
ODP	ontology design pattern
OL	ontology learning
ONTIE	Ontology of Immune Epitopes
OWA	open world assumption
OWL	Web Ontology Language
PRO	Protein Ontology
RDF	Resource Description Framework
RNA	ribonucleic acid
RO	OBO Relation Ontology
rRNA	ribosomal RNA
SBUO	Simple Bio Upper Ontology
SCR	Supplementary Concept Records
SO	Sequence Ontology
SUMO	Suggested Upper Merged Ontology
TF	transcription factor
tRNA	transfer RNA
SN	UMLS Semantic Network
UNA	unique name assumption
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium

Bibliography

[Aguirre et al. 2012] Aguirre, José L.; Eckert, Kai; Euzenat, Jérôme; Alfio, Ferrara; van Hage, Willem R.; Hollink, Laura; Meilicke, Christian; Nikolov, Andriy; Ritze, Dominique; Scharffe, François; Shvaiko, Pavel; Zamazal, Ondřej Šváb; Trojahn, Cássia; Jiménez-Ruiz, Ernesto; Grau, Bernardo C.; Zapilko, Benjamin: Results of the Ontology Alignment Evaluation Initiative 2012. In: *OM 2012 – Proceedings of the 7th International Workshop on Ontology Matching*. CEUR Workshop Proceedings, 2012.

[Alberts et al. 2002] Alberts, Bruce; Johnson, Alexander; Lewis, Julian; Raff, Martin; Roberts, Keith; Walter, Peter: *Molecular Biology of the Cell*, 4th Edn. Garland Science, 2002.

[Antezana et al. 2009] Antezana, Erick; Egana, Mikel; Blonde, Ward; Illarramendi, Aitzol; Bilbao, Inaki; De Baets, Bernard; Stevens, Robert; Mironov, Vladimir; Kuiper, Martin: The Cell Cycle Ontology: an application ontology for the representation and integrated analysis of the cell cycle process. In: *Genome Biology*, Vol. 10, No. 5, pp. R58, 2009.

[Apweiler et al. 2004] Apweiler, Rolf; Bairoch, Amos; Wu, Cathy H.; Barker, Winona C.; Boeckmann, Brigitte; Ferro, Serenella; Gasteiger, Elisabeth; Huang, Hongzhan; Lopez, Rodrigo; Magrane, Michele; Martin, Maria J.; Natale, Darren A.; O'Donovan, Claire; Redaschi1, Nicole; Yeh, Lai Su L.: UniProt: the universal protein knowledgebase. In: *Nucleic Acids Research*, Vol. 32, No. Database issue, pp. D115–D119, 2004.

[Aronson 2001] Aronson, Alan R.: Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: *AMIA 2001 – Proceedings of the*

25th Annual Symposium of the American Medical Informatics Association, pp. 17–21. Hanley & Belfus, 2001.

[Ashburner et al. 2000] Ashburner, Michael; Ball, Catherine A.; Blake, Judith A.; Botstein, David; Butler, Heather; Cherry, J. M.; Davis, Allan P.; Dolinski, Kara; Dwight, Selina S.; Eppig, Janan T.; Harris, Midori A.; Hill, David P.; Issel-Tarver, Laurie; Kasarskis, Andrew; Lewis, Suzanna E.; Matese, John C.; Richardson, Joel E.; Ringwald, Martin; Rubin, Gerald M.; Sherlock, Gavin: Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. In: *Nature Genetics*, Vol. 25, No. 1, pp. 25–29, 2000.

[Baader et al. 2003] Baader, Franz; Calvanese, Diego; McGuinness, Deborah L.; Nardi, Daniele; Patel-Schneider, Peter F. (Eds.): *The Description Logic Handbook. Theory, Implementation, and Applications*, Cambridge University Press, 2003.

[Baader et al. 2007] Baader, Franz; Horrocks, Ian; Sattler, Ulrike: Description Logics. In: van Harmelen, Frank; Lifschitz, Vladimir; Porter, Bruce (Eds.): *Handbook of Knowledge Representation*, Chap. 3, pp. 135–180. Elsevier, 2007.

[Bada and Hunter 2008] Bada, Michael; Hunter, Lawrence: Identification of OBO nonalignments and its implications for OBO enrichment. In: *Bioinformatics*, Vol. 24, No. 12, pp. 1448–1455, 2008.

[Bada et al. 2010] Bada, Michael; Hunter, Lawrence E.; Eckert, Miriam; Palmer, Martha: An overview of the CRAFT concept annotation guidelines. In: *LAW 2010 – Proceedings of the 4th Linguistic Annotation Workshop*, pp. 207–211. Association for Computational Linguistics, 2010.

[Bard et al. 2005] Bard, Jonathan; Rhee, Seung; Ashburner, Michael: An ontology for cell types. In: *Genome Biology*, Vol. 6, No. 2, pp. R21, 2005.

[Baumgartner et al. 2007] Baumgartner, William A.; Cohen, Kevin B.; Fox, Lynne M.; Acquaah-Mensah, George; Hunter, Lawrence: Manual curation is not sufficient for annotation of genomic databases. In: *Bioinformatics*, Vol. 23, No. 13, pp. i41–48, 2007.

[Baxevanis 2000] Baxevanis, Andreas D.: The molecular biology database collection: an online compilation of relevant database resources. In: *Nucleic Acids Research*, Vol. 28, No. 1, pp. 1–7, 2000.

[Bechhofer et al. 2004] Bechhofer, Sean; Harmelen, Frank van; Hendler, Jim; Horrocks, Ian; McGuinness, Deborah L.; Patel-Schneider, Peter F.; Stein, Lynn A.: *OWL Web Ontology Language Reference*. 2004. URL: `http://www.w3.org/TR/owl-ref/` – access date: 2012-12-14.

[Beisswanger et al. 2007] Beisswanger, Elena; DeLuca, David S.; Blasczyk, Rainer; Hahn, Udo: An ontology for major histocompatibility complex (MHC) alleles and molecules. In: *AMIA 2007 – Proceedings of the 31st Annual Symposium of the American Medical Informatics Association*, pp. 41–45, 2007.

[Beisswanger and Hahn 2012] Beisswanger, Elena; Hahn, Udo: Towards valid and reusable reference alignments – ten basic quality checks for ontology alignments and their application to three different reference data sets. In: *Journal of Biomedical Semantics*, Vol. 3, No. Suppl 1, pp. S4, 2012.

[Beisswanger et al. 2008a] Beisswanger, Elena; Lee, Vivian; Kim, Jung J.; Rebholz-Schuhmann, Dietrich; Splendiani, Andrea; Dameron, Olivier; Schulz, Stefan; Hahn, Udo: Gene Regulation Ontology (GRO): design principles and use cases. In: *MIE 2008 – Proceedings of the 21st International Congress of the European Federation for Medical Informatics*, pp. 9–14. IOS Press, 2008.

[Beisswanger et al. 2008b] Beisswanger, Elena; Poprat, Michael; Hahn, Udo: Lexical properties of OBO ontology class names and synonyms. In: *SMBM 2008 – Proceedings of the 3rd International Symposium on Semantic Mining in Biomedicine*, pp. 13–20. Turku Centre for Computer Science, 2008.

[Beisswanger et al. 2008c] Beisswanger, Elena; Schulz, Stefan; Stenzhorn, Holger; Hahn, Udo: BioTop: an upper domain ontology for the life sciences. A description of its current structure, contents and interfaces to OBO ontologies. In: *Applied Ontology*, Vol. 3, No. 4, pp. 205–212, 2008.

[Beisswanger et al. 2010] Beisswanger, Elena; Wermter, Joachim; Hahn, Udo: Aligning UniProt and MeSH – a case study on human protein terms. In: *MEDINFO 2010 – Proceedings of the 13th World Congress on Medical and Health Informatics*, pp. 1030–1034. IOS Press, 2010.

[Berners-Lee et al. 2001] Berners-Lee, Tim; Hendler, James; Lassila, Ora: The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. In: *Scientific American Magazine*, Vol. 284, No. 5, pp. 34–43, 2001.

[Blomqvist 2009] Blomqvist, Eva: *Semi-Automatic Ontology Construction Based on Patterns*. Department of Computer and Information Science, Linköpings universitet. Dissertation, 2009.

[Blomqvist 2010] Blomqvist, Eva: Ontology patterns – typology and experiences from design pattern development. In: *SAIS 2010 – Proceedings of the 26th Workshop of the Swedish Artificial Intelligence Society*, pp. 55–64. Linköping University Electronic Press, 2010.

[Bodenreider et al. 2005] Bodenreider, Olivier; Hayamizu, Terry F.; Ringwald, Martin; Coronado, Sherri de; Zhang, Songmao: Of mice and men: aligning mouse and human anatomies. In: *AMIA 2005 – Proceedings of the Annual Symposium of the American Medical Informatics Association*, pp. 61–65, 2005.

[Bodenreider and Stevens 2006] Bodenreider, Olivier; Stevens, Robert: Bio-ontologies: current trends and future directions. In: *Briefings in Bioinformatics*, Vol. 7, No. 3, pp. 256–274, 2006.

[Boehm 1986] Boehm, Barry W.: A spiral model of software development and enhancement. In: *ACM SIGSOFT Software Engineering Notes*, Vol. 11, No. 4, pp. 14–24, 1986.

[Boeker et al. 2012] Boeker, Martin; Schober, Daniel; Raufie, Djamila; Grewe, Niels; Röhl, Johannes; Jansen, Ludger; Schulz, Stefan: Teaching good biomedical ontology design. In: *ICBO 2012 – Proceedings of the 3rd International Conference on Biomedical Ontology*. CEUR Workshop Proceedings, 2012.

[Brank et al. 2005] Brank, Janez; Grobelnik, Marko; Mladenic, Dunja: A survey of ontology evaluation techniques. In: *SiKDD 2005 – Proceedings of the Conference on Data Mining and Data Warehouses*, pp. 166–169, 2005.

[Brickley and Guha 2004] Brickley, Dan; Guha, Ramanathan V.: *RDF Vocabulary Description Language 1.0: RDF Schema*. 2004. URL: `http://www.w3.org/TR/rdf-schema/` – access date: 2012-08-22.

[Burgun and Bodenreider 2005] Burgun, Anita; Bodenreider, Olivier: An ontology of chemical entities helps identify dependence relations among Gene Ontology terms. In: *SMBM 2005 – Proceedings of the 1st International Symposium on Semantic Mining in Biomedicine*. CEUR Workshop Proceedings, 2005.

[Buyko et al. 2010] Buyko, Ekaterina; Beisswanger, Elena; Hahn, Udo: The GeneReg corpus for gene expression regulation events - an overview of the corpus and its in-domain and out-of-domain interoperability. In: *LREC 2010 – Proceedings of the 7th International Conference on Language Resources and Evaluation*, pp. 2662–2666. European Language Resources Association, 2010.

[Buyko et al. 2011] Buyko, Ekaterina; Faessler, Erik; Wermter, Joachim; Hahn, Udo: Syntactic simplification and semantic enrichment – trimming dependency graphs for event extraction. In: *Computational Intelligence*, Vol. 27, No. 4, pp. 610–644, 2011.

[Ceusters 2006] Ceusters, Werner: Towards a realism-based metric for quality assurance in ontology matching. In: *FOIS 2006 – Proceedings of the 4th International Conference on Formal Ontology in Information Systems*, pp. 321–332. IOS Press, 2006.

[Ciaramita et al. 2005] Ciaramita, Massimiliano; Gangemi, Aldo; Ratsch, Esther; Saric, Jasmin; Rojas, Isabel: Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In: *IJCAI 2005 – Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 659–664. Morgan Kaufmann Publishers Inc., 2005.

[Cimiano 2006] Cimiano, Philipp: *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*, Springer, 2006.

[Cruz et al. 2009] Cruz, Isabel F.; Antonelli, Flavio P.; Stroe, Cosmin: Efficient selection of mappings and automatic quality-driven combination of matching methods. In: *OM 2009 – Proceedings of the 4th International Workshop on Ontology Matching*. CEUR Workshop Proceedings, 2009.

[Davis et al. 1988] Davis, Alan M.; Bersoff, Edward H.; Comer, Edward R.: A strategy for comparing alternative software development life cycle models. In: *IEEE Transactions on Software Engineering*, Vol. 14, No. 10, pp. 1453–1461, 1988.

[Day-Richter 2006] Day-Richter, John: *The OBO Flat File Format Specification, version 1.2*. 2006. URL: `http://www.geneontology.org/GO.format.obo-1_2.shtml` – access date: 2012-04-18.

[de Coronado et al. 2009] de Coronado, Sherri; Wright, Lawrence W.; Fragoso, Gilberto; Haber, Margaret W.; Hahn-Dantona, Elizabeth A.; Hartel, Francis W.; Quan, Sharon L.; Safran, Tracy; Thomas, Nicole; Whiteman, Lori: The NCI Thesaurus quality assurance life cycle. In: *Journal of Biomedical Informatics*, Vol. 42, No. 3, pp. 530–539, 2009.

[Degtyarenko et al. 2008] Degtyarenko, Kirill; Matos, Paula de; Ennis, Marcus; Hastings, Janna; Zbinden, Martin; McNaught, Alan; Alcántara, Rafael; Darsow, Michael; Guedj, Mickaël; Ashburner, Michael: ChEBI: a database and ontology for chemical entities of biological interest. In: *Nucleic Acids Research*, Vol. 36, No. Database issue, pp. 344–350, 2008.

[DeLuca et al. 2009] DeLuca, David S.; Beisswanger, Elena; Wermter, Joachim; Horn, Peter A.; Hahn, Udo; Blasczyk, Rainer: MaHCO: an ontology of the major histocompatibility complex for immunoinformatic applications and text mining. In: *Bioinformatics*, Vol. 25, No. 16, pp. 2064–2070, 2009.

[Ding et al. 2004] Ding, Li; Finin, Tim; Joshi, Anupam; Pan, Rong; Cost, R. S.; Peng, Yun; Reddivari, Pavan; Doshi, Vishal; Sachs, Joel: Swoogle: a search and metadata engine for the Semantic Web. In: *CIKM 2004 – Proceedings of the 13th International Conference on Information and Knowledge Management*, pp. 652–659. ACM Press, 2004.

[Doms and Schroeder 2005] Doms, Andreas; Schroeder, Michael: GoPubMed: exploring PubMed with the Gene Ontology. In: *Nucleic Acids Research*, Vol. 33, No. Web Server issue, pp. W783–W786, 2005.

[Dowell et al. 2009] Dowell, Karen G.; McAndrews-Hill, Monica S.; Hill, David P.; Drabkin, Harold J.; Blake, Judith A.: Integrating text mining into the MGI biocuration workflow. In: *Database*, Vol. 2009, pp. bap019, 2009.

[Duroux et al. 2008] Duroux, Patrice; Kaas, Quentin; Brochet, Xavier; Lane, Jérôme; Ginestoux, Chantal; Lefranc, Marie-Paule; Giudicelli, Véronique: IMGT-Kaleidoscope, the formal IMGT-ONTOLOGY paradigm. In: *Biochimie*, Vol. 90, No. 4, pp. 570–583, 2008.

[Ehrig and Euzenat 2005] Ehrig, Marc; Euzenat, Jérôme: Relaxed precision and recall for ontology matching. In: *Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies*, pp. 25–32. CEUR Workshop Proceedings, 2005.

[Eilbeck et al. 2005] Eilbeck, Karen; Lewis, Suzanna E.; Mungall, Christopher J.; Yandell, Mark; Stein, Lincoln; Durbin, Richard; Ashburner, Michael: The Sequence Ontology: a tool for the unification of genome annotations. In: *Genome Biology*, Vol. 6, No. 5, pp. R44, 2005.

[Euzenat 2004] Euzenat, Jérôme: An API for ontology alignment. In: *ISWC 2004 – Proceedings of the 3rd International Semantic Web Conference*, pp. 698–712. Springer, 2004.

[Euzenat 2007] Euzenat, Jérôme: Semantic precision and recall for ontology alignment evaluation. In: *IJCAI 2007 – Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pp. 348–353. Morgan Kaufmann Publishers Inc., 2007.

[Euzenat et al. 2011a] Euzenat, Jérôme; Ferrara, Alfio; van Hage, Willem R.; Hollink, Laura; Meilicke, Christian; Nikolov, Andriy; Ritze, Dominique; Scharffe, François; Shvaiko, Pavel; Stuckenschmidt, Heiner; Sváb-Zamazal, Ondrej; Trojahn dos Santos, Cássia: Results of the Ontology Alignment Evaluation Initiative 2011. In: *OM 2011 – Proceedings of the 6th International Workshop on Ontology Matching*. CEUR Workshop Proceedings, 2011.

[Euzenat et al. 2011b] Euzenat, Jérôme; Meilicke, Christian; Stuckenschmidt, Heiner; Shvaiko, Pavel; Trojahn dos Santos, Cássia: Ontology Alignment Evaluation Initiative: six years of experience. In: *Journal on Data Semantics XV*, pp. 158–192. Springer, 2011.

[Euzenat et al. 2008] Euzenat, Jérôme; Mocan, Adian; Scharffe, François: Ontology Alignments. In: Hepp, Martin; Leenheer, Pieter; Moor, Aldo; Sure, York (Eds.): *Ontology Management: Semantic Web, Semantic Web Services and Business Applications*, Chap. 6, pp. 177–206. Springer, 2008.

[Euzenat and Shvaiko 2007] Euzenat, Jérôme; Shvaiko, Pavel: *Ontology Matching*, Springer, 2007.

[Fernandez et al. 1997] Fernandez, M.; Gómez-Pérez, Asunción; Juristo, N.: Methontology: from ontological art towards ontological engineering. In: *AAAI Technical Report SS-97-06, Papers of the AAAI Spring Symposium on Ontological Engineering*, pp. 33–40. AAAI Press, 1997.

[Fortuna et al. 2007] Fortuna, Blaž; Grobelnik, Marko; Mladenić, Dunja: OntoGen: semi-automatic ontology editor. In: *Proceedings of the Symposium on Human Interface 2007 – Part II*, pp. 309–318. Springer, 2007.

[Funk and Reid 1983] Funk, Mark E.; Reid, Carolyn A.: Indexing consistency in MEDLINE. In: *Bulletin of the Medical Library Association*, Vol. 71, No. 2, pp. 176–183, 1983.

[Gangemi 2005] Gangemi, Aldo: Ontology design patterns for Semantic Web content. In: *ISWC 2005 – Proceedings of the 4th International Semantic Web Conference*, pp. 262–276. Springer, 2005.

[Gangemi et al. 2005a] Gangemi, Aldo; Catenacci, Carola; Ciaramita, Massimiliano; Lehmann, Jens: Ontology evaluation and validation: an integrated formal model for the quality diagnostic task / Laboratory for Applied Ontologies, ISTC-CNR, Roma/Trento, Italy. Technical Report, 2005.

[Gangemi et al. 2005b] Gangemi, Aldo; Catenacci, Carola; Ciaramita, Massimiliano; Lehmann, Jos: A theoretical framework for ontology evaluation and validation. In: *SWAP 2005 – Proceedings of the 2nd Italian Semantic Web Workshop*. CEUR Workshop Proceedings, 2005.

[Giudicelli and Lefranc 1999] Giudicelli, Véonique; Lefranc, Marie-Paule: Ontology for immunogenetics: The IMGT-ONTOLOGY. In: *Bioinformatics*, Vol. 15, No. 12, pp. 1047–1054, 1999.

[Glimm et al. 2010] Glimm, Birte; Horrocks, Ian; Motik, Boris: Optimized description logic reasoning via core blocking. In: *IJCAR 2010 – Proceedings of the 5th International Joint Conference on Automated Reasoning*, pp. 457–471. Springer, 2010.

[Golbreich et al. 2007] Golbreich, Christine; Horridge, Matthew; Horrocks, Ian; Motik, Boris; Shearer, Rob: OBO and OWL: leveraging Semantic Web technologies for the life sciences. In: *ISWC 2007 and ASWC 2007 – Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference*, pp. 169–182. Springer, 2007.

[Gospodnetić and Hatcher 2005] Gospodnetić, Otis; Hatcher, Erik: *Lucene in Action*, 1st Edn. Manning Publication Co., 2005.

[Greenbaum et al. 2010] Greenbaum, Jason A.; Vita, Randi; Zarebski, Laura.M.; Sette, Alessandro; Peters, Bjoern: Ontology development for the Immune Epitope Database. In: Flower, Darren R.; Davies, Matthew N.; Ranganathan, Shoba (Eds.): *Bioinformatics for Immunomics*, pp. 47–56. Springer, 2010.

[Grenon et al. 2004] Grenon, Pierre; Smith, Barry; Goldberg, Louis: Biodynamic ontology: applying BFO in the biomedical domain. In: *Studies in Health Technology and Informatics*, Vol. 102, pp. 20–38, 2004.

[Gruber 1993] Gruber, Thomas R.: A translation approach to portable ontology specifications. In: *Knowledge Acquisition*, Vol. 5, No. 2, pp. 199–220, 1993.

[Grüninger and Fox 1995] Grüninger, Michael; Fox, Mark S.: Methodology for the design and evaluation of ontologies. In: *Proceedings of the IJCAI 1995 Workshop on Basic Ontological Issues in Knowledge Sharing*, pp. 6.1–6.10. AAAI Press, 1995.

[Guarino 1998] Guarino, Nicola: Formal ontology and information systems. In: *FOIS 1998 – Proceedings of the 1st International Conference on Formal Ontology in Information Systems*, pp. 3–15. IOS Press, 1998.

[Guarino and Welty 2002] Guarino, Nicola; Welty, Christopher: Evaluating ontological decisions with OntoClean. In: *Communications of the ACM*, Vol. 45, No. 2, pp. 61–65, 2002.

[Haase and Völker 2008] Haase, Peter; Völker, Johanna: Ontology learning and reasoning – dealing with uncertainty and inconsistency. In: Costa, Paulo C. G.; d'Amato, Claudia; Fanizzi, Nicola; Laskey, Kathryn B.; Laskey, Kenneth J.; Lukasiewicz, Thomas; Nickles, Matthias; Pool, Michael (Eds.): *Uncertainty Reasoning for the Semantic Web I*, pp. 366–384. Springer, 2008.

[Hahn et al. 2008] Hahn, Udo; Beisswanger, Elena; Buyko, Ekaterina; Poprat, Michael; Tomanek, Katrin; Wermter, Joachim: Semantic annotations for biology: a corpus development initiative at the Jena University Language & Information Engineering (JULIE) Lab. In: *LREC 2008 – Proceedings of the 6th International Conference on Language Resources and Evaluation*, pp. 2257–2261. European Language Resources Association, 2008.

[Hahn et al. 2009] Hahn, Udo; Tomanek, Katrin; Buyko, Ekaterina; Kim, Jung J.; Rebholz-Schuhmann, Dietrich: How feasible and robust is the automatic extraction of gene regulation events? A cross-method evaluation under lab and real-life conditions. In: *BioNLP 2009 – Proceedings of the 8th Workshop on Biomedical Natural Language Processing*, pp. 37–45. Association for Computational Linguistics, 2009.

[Hammar and Sandkuhl 2010] Hammar, Karl; Sandkuhl, Kurt: The state of ontology pattern research: a systematic review of ISWC, ESWC and ASWC 2005–2009. In: *WOP 2010 – Proceedings of the 2nd Workshop on Ontology Patterns, collocated with the 9th International Semantic Web Conference*. CEUR Workshop Proceedings, 2010.

[Hanisch et al. 2005] Hanisch, Daniel; Fundel, Katrin; Mevissen, Heinz-Theodor; Zimmer, Ralf; Fluck, Juliane: ProMiner: rule-based protein and gene entity recognition. In: *BMC Bioinformatics*, Vol. 6, No. Suppl 1, pp. S14, 2005.

[Hartmann et al. 2004] Hartmann, Jens; Sure, York; Giboin, Alain; Maynard, Diana; Suárez-Figueroa, María del C.; Cuel, Roberta: Methods for Ontology Evaluation / KnowledgeWeb project. Deliverable D1.2.3, 2004.

[Hayamizu et al. 2005] Hayamizu, Terry; Mangan, Mary; Corradi, John; Kadin, James; Ringwald, Martin: The Adult Mouse Anatomical Dictionary: a tool for annotating and integrating data. In: *Genome Biology*, Vol. 6, No. 3, pp. R29, 2005.

[Hearst 2006] Hearst, Marti: Design recommendations for hierarchical faceted search interfaces. In: *Proceedings of the International ACM SIGIR 2006 Workshop on Faceted Search*, pp. 26–30, 2006.

[Hirschman et al. 2007] Hirschman, Lynette; Krallinger, Martin; Valencia, Alfonso (Eds.): *Proceedings of the Second BioCreative Challenge Evaluation Workshop*. CNIO Centro Nacional de Investigaciones Oncológicas, Madrid, 2007.

[Hirschman et al. 2005] Hirschman, Lynette; Yeh, Alexander S.; Blaschke, Christian; Valencia, Alfonso: Overview of BioCreAtIvE: critical assessment of information extraction for biology. In: *BMC Bioinformatics*, Vol. 6, No. Suppl 1, pp. S1, 2005.

[Hoehndorf et al. 2008] Hoehndorf, Robert; Loebe, Frank; Poli, Roberto; Herre, Heinrich; Kelso, Janet: GFO-Bio: A biological core ontology. In: *Applied Ontology*, Vol. 3, No. 4, pp. 219–227, 2008.

[Hoffmann and Valencia 2004] Hoffmann, Robert; Valencia, Alfonso: A gene network for navigating the literature. In: *Nature Genetics*, Vol. 36, No. 7, pp. 664, 2004.

[Holdsworth et al. 2009] Holdsworth, Rhonda; Hurley, Carolyn K.; Marsh, Steven G.; Lau, Marie; Noreen, Harriet J.; Kempenich, Jane H.; Setterholm, Michelle; Maiers, Martin: The HLA dictionary 2008: a summary of HLA-A, -B, -C, -DRB1/3/4/5, and -DQB1 alleles and their association with serologically defined HLA-A, -B, -C, -DR, and -DQ antigens. In: *Tissue Antigens*, Vol. 73, pp. 95–170, 2009.

[Horridge 2011] Horridge, Matthew: *A practical guide to building OWL ontologies using Protégé 4 and CO-ODE tools*, v1.3. The University Of Manchester, Manchester, UK. Manual, 2011.

[Horridge and Bechhofer 2011] Horridge, Matthew; Bechhofer, Sean: The OWL API: a Java API for OWL ontologies. In: *Semantic Web*, Vol. 2, No. 1, pp. 11–21, 2011.

[Horrocks and Patel-Schneider 2004] Horrocks, Ian; Patel-Schneider, Peter: Reducing OWL entailment to description logic satisfiability. In: *Journal of Web Semantics*, Vol. 1, No. 4, pp. 345–357, 2004.

[Horrocks et al. 2007] Horrocks, Ian; Patel-Schneider, Peter F.; McGuinness, Deborah L.; Welty, Christopher A.: OWL: a description logic based ontology language for the Semantic Web. In: Baader, Franz; Calvanese, Diego; McGuinness, Deborah; Nardi, Daniele; Patel-Schneider, Peter F. (Eds.): *The Description Logic Handbook:*

Theory, Implementation, and Applications, 2nd Edn., Chap. 14. Cambridge University Press, 2007.

[Howe et al. 2008] Howe, Doug; Costanzo, Maria; Fey, Petra; Gojobori, Takashi; Hannick, Linda; Hide, Winston; Hill, David P.; Kania, Renate; Schaeffer, Mary; St Pierre, Susan; Twigger, Simon; White, Owen; Rhee, Seung Y.: Big data: the future of biocuration. In: *Nature*, Vol. 455, No. 7209, pp. 47–50, 2008.

[Jain et al. 2010] Jain, Prateek; Hitzler, Pascal; Sheth, Amit P.; Verma, Kunal; Yeh, Peter Z.: Ontology alignment for Linked Open Data. In: *ISWC 2010 – Proceedings of the 9th International Semantic Web Conference*, pp. 402–417. Springer, 2010.

[Jiménez-Ruiz and Grau 2011] Jiménez-Ruiz, Ernesto; Grau, Bernardo C.: LogMap: logic-based and scalable ontology matching. In: *ISWC 2011 – Proceedings of the 10th International Semantic Web Conference, Part I*, pp. 273–288. Springer, 2011.

[Jonquet et al. 2009] Jonquet, Clement; Shah, Nigam H.; Musen, Mark A.: The Open Biomedical Annotator. In: *AMIA-TBI 2009 – American Medical Informatics Association Symposium on Translational BioInformatics*, pp. 56–60, 2009.

[Joslyn et al. 2009] Joslyn, Cliff; Paulson, Patrick; White, Amanda M.: Measuring the structural preservation of semantic hierarchy alignment. In: *OM 2009 – Proceedings of the 4th International Workshop on Ontology Matching*. CEUR Workshop Proceedings, 2009.

[Kawazoe et al. 2006] Kawazoe, Ai; Jin, Lihua; Shigematsu, Mika; Barerro, Roberto; Taniguchi, Kiyosu; Collier, Nigel: The development of a schema for the annotation of terms in the BioCaster disease detecting/tracking system. In: *KR-MED 2006 – Proceedings of the 2nd International Workshop on Formal Biomedical Knowledge Representation*, pp. 77–85. CEUR Workshop Proceedings, 2006.

[Kim et al. 2009] Kim, Jin-Dong; Ohta, Tomoko; Pyysalo, Sampo; Kano, Yoshinobu; Tsujii, Jun'ichi: Overview of BioNLP'09 shared task on event extraction. In: *BioNLP 2009 – Proceedings of the 8th Workshop on Biomedical Natural Language Processing: Shared Task*, pp. 1–9. Association for Computational Linguistics, 2009.

[Kim et al. 2003] Kim, Jin-Dong; Ohta, Tomoko; Tateisi, Yuka; Tsujii, Jun'ichi: GENIA corpus – semantically annotated corpus for bio-textmining. In: *Bioinformatics*, Vol. 19, No. Suppl 1, pp. i180–182, 2003.

[Kim et al. 2008] Kim, Jin-Dong; Ohta, Tomoko; Tsujii, Jun'ichi: Corpus annotation for mining biomedical events from literature. In: *BMC Bioinformatics*, Vol. 9, pp. 10 (1:25), 2008.

[Kim et al. 2010] Kim, Jung J.; Lee, Vivian; Rebholz-Schuhmann, Dietrich: Semantic representation of Gene Ontology terms by using Gene Regulation Ontology. In:

KR-MED 2012 – Proceedings of the 4th International Workshop on Formal Biomedical Knowledge Representation, hosted by Bio-Ontologies 2010. CEUR Workshop Proceedings, 2010.

[Kim and Rebholz-Schuhmann 2011] Kim, Jung J.; Rebholz-Schuhmann, Dietrich: Improving the extraction of complex regulatory events from scientific text by using ontology-based inference. In: *Journal of Biomedical Semantics*, Vol. 2, No. Suppl 5, pp. S3, 2011.

[Klyne and Carroll 2004] Klyne, Graham; Carroll, Jeremy J.: *Resource Description Framework (RDF): Concepts and Abstract Syntax*. 2004. URL: `http://www.w3.org/TR/rdf-concepts/` – access date: 2012-12-21.

[Knublauch et al. 2004] Knublauch, Holger; Fergerson, Ray W.; Noy, Natalya F.; Musen, Mark A.: The Protégé-OWL plugin: an open development environment for Semantic Web applications. In: *ISWC 2004 – Proceedings of the 3rd International Semantic Web Conference*, pp. 229–243. Springer, 2004.

[Köhler et al. 2006] Köhler, Jacob; Munn, Katherine; Rüegg, Alexander; Skusa, Andre; Smith, Barry: Quality control for terms and definitions in ontologies and taxonomies. In: *BMC Bioinformatics*, Vol. 7, pp. 212 (1:12), 2006.

[Larman and Basili 2003] Larman, Craig; Basili, Victor R.: Iterative and incremental developments. A brief history. In: *Computer*, Vol. 36, No. 6, pp. 47–56, 2003.

[Leonelli et al. 2011] Leonelli, Sabina; Diehl, Alexander; Christie, Karen; Harris, Midori; Lomax, Jane: How the Gene Ontology evolves. In: *BMC Bioinformatics*, Vol. 12, No. 1, pp. 325 (1:7), 2011.

[Little 2007] Little, Ann-Margaret: An overview of HLA typing for hematopoietic stem cell transplantation. In: *Methods in Molecular Medicine*, Vol. 134, pp. 35–49, 2007.

[Marsh 2003] Marsh, Steven G.: HLA nomenclature and the IMGT/HLA sequence database. In: *Novartis Foundation Symposium*, Vol. 254, pp. 165–173, 2003.

[Marsh et al. 2010] Marsh, Steven G.; Albert, Ekkehard D.; Bodmer, Walter F.; Bontrop, Ronald E.; Dupont, Bo; Erlich, Henry A.; Fernández-Viña, Marcelo; Geraghty, Daniel E.; Holdsworth, Rhonda; Hurley, Carolyn K.; Lau, Marie; Lee, Kyung W.; Mach, Bernard; Maiers, Martin; Mayr, Wolfgang R.; Müller, Carlheinz R.; Parham, Peter; Petersdorf, Effie W.; Sasazuki, Takehiko; L., Strominger J.; Svejgaard, Arne; Terasaki, Paul I.; Tiercy, Jean-Marie; Trowsdale, John: An update to HLA nomenclature, 2010. In: *Bone Marrow Transplantation*, Vol. 45, No. 5, pp. 846–848, 2010.

[Marsh et al. 2002] Marsh, Steven G.; Albert, Ekkehard D.; Bodmer, Walter F.; Bontrop, Ronald E.; Dupont, Bo; Erlich, Henry A.; Geraghty, Daniel E.; Hansen, John A.; Mach, Bernard; Mayr, Wolfgang R.; Parham, Peter; Petersdorf, Effie W.;

Sasazuki, Takehiko; Schreuder, Geziena M.; Strominger, Jack L.; Svejgaard, Arne; Terasaki, Paul I.: Nomenclature for factors of the HLA system, 2002. In: *European Journal of Immunogenetics*, Vol. 29, No. 6, pp. 463–515, 2002.

[Mascardi et al. 2009] Mascardi, Viviana; Locoro, Angela; Rosso, Paolo: Automatic ontology matching via upper ontologies: a systematic evaluation. In: *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 5, pp. 609–623, 2009.

[Masolo et al. 2003] Masolo, Claudio; Borgo, Stefano; Gangemi, Aldo; Guarino, Nicola; Oltramari, Alessandro: Ontology Library / WonderWeb project. Deliverable D18, 2003.

[McCray 2003] McCray, Alexa T.: An upper level ontology for the biomedical domain. In: *Comparative and Functional Genomics*, Vol. 4, No. 1, pp. 80–84, 2003.

[Meehan et al. 2011] Meehan, Terrence F.; Masci, Anna M.; Abdulla, Amina; Cowell, Lindsay G.; Blake, Judith A.; Mungall, Christopher J.; Diehl, Alexander D.: Logical development of the Cell Ontology. In: *BMC Bioinformatics*, Vol. 12, pp. 6 (1:12), 2011.

[Meilicke and Stuckenschmidt 2009] Meilicke, Christian; Stuckenschmidt, Heiner: An efficient method for computing alignment diagnoses. In: *RR 2009 – Proceedings of the 3rd International Conference on Web Reasoning and Rule Systems*, pp. 182–196. Springer, 2009.

[Meilicke et al. 2009] Meilicke, Christian; Stuckenschmidt, Heiner; Zamazal, Ondřej Šváb: A reasoning-based support tool for ontology mapping evaluation. In: *ESWC 2009 – Proceedings of the 6th European Semantic Web Conference*, pp. 878–882. Springer, 2009.

[Meilicke et al. 2008] Meilicke, Christian; Völker, Johanna; Stuckenschmidt, Heiner: Learning disjointness for debugging mappings between lightweight ontologies. In: *EKAW 2008 – Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management*, pp. 93–108. Springer, 2008.

[Miles and Bechhofer 2009] Miles, Alistair; Bechhofer, Sean: *SKOS Simple Knowledge Organization System Reference*. 2009. URL: `http://www.w3.org/TR/skos-reference/` – access date: 2012-08-17.

[Mizoguchi 2003] Mizoguchi, Riichiro: Tutorial on ontological engineering. Part 2: Ontology development, tools and languages. In: *New Generation Computing*, Vol. 22, No. 1, pp. 61–96, 2003.

[Mortensen et al. 2012] Mortensen, Jonathan M.; Horridge, Matthew; Musen, Mark A.; Noy, Natalya F.: Modest use of ontology design patterns in a repository of biomedical ontologies. In: *WOP 2012 – Proceedings of the 3rd Workshop on Ontology Patterns*. CEUR Workshop Proceedings, 2012.

[Motik et al. 2009] Motik, Boris; Horrocks, Ian; Shearer, Rob: Hypertableau reasoning for description logics. In: *Journal of Artificial Intelligence Research*, Vol. 36, pp. 165–228, 2009.

[Mottaz et al. 2008] Mottaz, Anais; Yip, Yum; Ruch, Patrick; Veuthey, Anne-Lise: Mapping proteins to disease terminologies: from UniProt to MeSH. In: *BMC Bioinformatics*, Vol. 9, No. Suppl 5, pp. S3, 2008.

[Mungall 2004] Mungall, Christopher J.: Obol: integrating language and meaning in bio-ontologies. In: *Comparative and Functional Genomics*, Vol. 5, No. 6-7, pp. 509–520, 2004.

[Mungall et al. 2011a] Mungall, Christopher J.; Bada, Michael; Berardini, Tanya Z.; Deegan, Jennifer; Ireland, Amelia; Harris, Midori A.; Hill, David P.; Lomax, Jane: Cross-product extensions of the Gene Ontology. In: *Journal of Biomedical Informatics*, Vol. 44, No. 1, pp. 80–86, 2011.

[Mungall et al. 2011b] Mungall, Christopher J.; Batchelor, Colin R.; Eilbeck, Karen: Evolution of the Sequence Ontology terms and relationships. In: *Journal of Biomedical Informatics*, Vol. 44, No. 1, pp. 87–93, 2011.

[Myhre et al. 2006] Myhre, Simen; Tveit, Henrik; Mollestad, Torulf; Laegreid, Astrid: Additional Gene Ontology structure for improved biological reasoning. In: *Bioinformatics*, Vol. 22, No. 16, pp. 2020–2027, 2006.

[Natale et al. 2011] Natale, Darren A.; Arighi, Cecilia N.; Barker, Winona C.; Blake, Judith A.; Bult, Carol J.; Caudy, Michael; Drabkin, Harold J.; D'Eustachio, Peter; Evsikov, Alexei V.; Huang, Hongzhan; Nchoutmboube, Jules; Roberts, Natalia V.; Smith, Barry; Zhang, Jian; Wu, Cathy H.: The Protein Ontology: a structured representation of protein forms and complexes. In: *Nucleic Acids Research*, Vol. 39, No. Database issue, pp. 539–545, 2011.

[Netzer et al. 2009] Netzer, Yael; Gabay, David; Adler, Meni; Goldberg, Yoav; Elhadad, Michael: Ontology evaluation through text classification. In: *Advances in Web and Network Technologies, and Information Management*, pp. 210–221. Springer, 2009.

[Neuhaus et al. 2011] Neuhaus, Fabian; Florescu, Elizabeth; Galton, Antony; Grüninger, Michael; Guarino, Nicola; Obrst, Leo; Sanchez, Arturo; Vizedom, Amanda; Yim, Peter; Smith, Barry: Creating the ontologists of the future. In: *Applied Ontology*, Vol. 6, No. 1, pp. 91–98, 2011.

[Noy and Musen 2003] Noy, Natalya F.; Musen, Mark A.: The PROMPT suite: interactive tools for ontology merging and mapping. In: *International Journal of Human-Computer Studies*, Vol. 59, No. 6, pp. 983 – 1024, 2003.

[Noy et al. 2009] Noy, Natalya F.; Shah, Nigam H.; Whetzel, Patricia L.; Dai, Benjamin; Dorf, Michael; Griffith, Nicholas; Jonquet, Clement; Rubin, Daniel L.; Storey, Margaret A.; Chute, Christopher G.; Musen, Mark A.: BioPortal: ontologies and integrated data resources at the click of a mouse. In: *Nucleic Acids Research*, Vol. 37, No. Web Server issue, pp. W170–173, 2009.

[Noy et al. 2010] Noy, Natalya F.; Tudorache, Tania; Nyulas, Csongor; Musen, Mark: The ontology life cycle: integrated tools for editing, publishing, peer review, and evolution of ontologies. In: *AMIA 2010 – Proceedings of the 34th Annual Symposium of the American Medical Informatics Association*, pp. 552–556, 2010.

[Noy et al. 2006] Noy, Natalya F.; Chugh, Abhita; Liu, William; Musen, Mark A.: A framework for ontology evolution in collaborative environments. In: *ISWC 2006 – Proceedings of the 5th International Semantic Web Conference*, pp. 544–558. Springer, 2006.

[Noy and McGuinness 2001] Noy, Natalya F.; McGuinness, Deborah L.: Ontology development 101: a guide to creating your first ontology / Knowledge Systems, AI Laboratory, Stanford University. Technical Report KSL-01-05, 2001.

[OBO Foundry 2012] OBO Foundry: *The OBO Foundry Principles*. 2012. URL: `http://obofoundry.org/crit.shtml` – access date: 2012-02-11.

[Obrst et al. 2007] Obrst, Leo; Ceusters, Werner; Mani, Inderjeet; Ray, Steve; Smith, Barry: The evaluation of ontologies. In: Baker, Christopher J. O.; Cheung, Hei-Hoi (Eds.): *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, pp. 139–158. Springer, 2007.

[Ogren et al. 2004] Ogren, Philip V.; Cohen, Kevin B.; Acquaah-Mensah, George K.; Eberlein, Jens; Hunter, Lawrence: The compositional structure of Gene Ontology terms. In: *PSB 2004 – Proceedings of the 9th Pacific Symposium on Biocomputing*, pp. 214–225. World Scientific Publishing, 2004.

[Oxford Journals 2012] Oxford Journals: *Nucleic Acids Research*. 2012. URL: `http://nar.oxfordjournals.org/` – access date: 2012-08-22.

[Paul 2003] Paul, William E.: *Fundamental Immunology*, 5th Edn. Lippincott Williams & Wilkins, 2003.

[Pease 2011] Pease, Adam: *The Suggested Upper Merged Ontology (SUMO)*. 2011. URL: `http://www.ontologyportal.org/` – access date: 2012-02-29.

[Pilato 2004] Pilato, Michael: *Version Control with Subversion*, O'Reilly & Associates, Inc., 2004.

[Porter 1980] Porter, Martin F.: An algorithm for suffix stripping. In: *Program*, Vol. 14, No. 3, pp. 130–137, 1980.

[Poveda-Villalon et al. 2012] Poveda-Villalon, María; Suárez-Figueroa, María del C.; Gómez-Pérez, Asunción: Validating ontologies with OOPS! In: *EKAW 2012 – Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management*, pp. 267–281. Springer, 2012.

[Pyysalo et al. 2012] Pyysalo, Sampo; Ohta, Tomoko; Rak, Rafal; Sullivan, Dan; Mao, Chunhong; Wang, Chunxia; Sobral, Bruno; Tsujii, Jun'ichi; Ananiadou, Sophia: Overview of the ID, EPI and REL tasks of BioNLP Shared Task 2011. In: *BMC Bioinformatics*, Vol. 13, No. Suppl 11, pp. S2, 2012.

[Rector 2003] Rector, Alan L.: Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In: *K-CAP 2003 – Proceedings of the 2nd International Conference on Knowledge Capture*, pp. 121–128. ACM Press, 2003.

[Rector et al. 2004] Rector, Alan L.; Drummond, Nick; Horridge, Matthew; Rogers, Jeremy; Knublauch, Holger; Stevens, Robert; Wang, Hai; Wroe, Chris: OWL pizzas: practical experience of teaching OWL DL. Common errors & common patterns. In: *EKAW 2004 – Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management*, pp. 63–81. Springer, 2004.

[Rector et al. 2006a] Rector, Alan L.; Rogers, Jeremy E.; Bittner, Thomas: Granularity, scale and collectivity: when size does and does not matter. In: *Journal of Biomedical Informatics*, Vol. 39, No. 3, pp. 333–349, 2006.

[Rector et al. 2006b] Rector, Alan L.; Stevens, Robert; Rogers, Jeremy: *Simple Bio Upper Ontology*. 2006. URL: `http://www.cs.man.ac.uk/~rector/ontologies/simple-top-bio/` – access date: 2012-05-08.

[Rosse et al. 2005] Rosse, Cornelius; Kumar, Anand; Mejino, José L. V.; Cook, Daniel L.; Detwiler, Landon T.; Smith, Barry: A strategy for improving and integrating biomedical ontologies. In: *AMIA 2005 – Proceedings of the Annual Symposium of the American Medical Informatics Association*, pp. 639–643, 2005.

[Rosse and Mejino 2003] Rosse, Cornelius; Mejino, José L. V.: A reference ontology for biomedical informatics: the Foundational Model of Anatomy. In: *Journal of Biomedical Informatics*, Vol. 36, No. 6, pp. 478–500, 2003.

[Royce 1970] Royce, Walker W.: Managing the development of large software systems: concepts and techniques. In: *WesCon 1970 – Technical papers presented at the of Western Electronic Show and Convention*, pp. 1–9, 1970.

[Rubin et al. 2008] Rubin, Daniel L.; Shah, Nigam; Noy, Natalya F.: Biomedical ontologies: a functional perspective. In: *Briefings in Bioinformatics*, Vol. 9, No. 1, pp. 75–90, 2008.

[Sasaki et al. 2008] Sasaki, Yutaka; Thompson, Paul; Cotter, Philip; McNaught, John; Ananiadou, Sophia: Event frame extraction based on a gene regulation corpus. In: *COLING 2008 – Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 761–768, 2008.

[Sathiamurthy et al. 2005] Sathiamurthy, Muthuraman; Peters, Bjoern; Bui, Huynh-Hoa; Sidney, John; Mokili, John; Wilson, Stephen S.; Fleri, Ward; McGuinness, Deborah L.; Bourne, Philip E.; Sette, Alessandro: An ontology for immune epitopes: application to the design of a broad scope database of immune reactivities. In: *Immunome Research*, Vol. 1, pp. 2 (1:10), 2005.

[Schober et al. 2010] Schober, Daniel; Boeker, Martin; Bullenkamp, Jessica; Huszka, Csaba; Depraetere, Kristof; Teodoro, Douglas; Nadah, Nadia; Choquet, Remy; Daniel, Christel; Schulz, Stefan: The DebugIT core ontology: semantic integration of antibiotics resistance patterns. In: *Studies in Health Technology and Informatics*, Vol. 160, No. Pt 2, pp. 1060–1064, 2010.

[Schober et al. 2009] Schober, Daniel; Smith, Barry; Lewis, Suzanna E.; Kusnierczyk, Waclaw; Lomax, Jane; Mungall, Christopher J.; Taylor, Chris F.; Rocca-Serra, Philippe; Sansone, Susanna-Assunta: Survey-based naming conventions for use in OBO Foundry ontology development. In: *BMC Bioinformatics*, Vol. 10, pp. 125 (1:9), 2009.

[Schulz et al. 2006a] Schulz, Stefan; Beisswanger, Elena; Hahn, Udo; Wermter, Joachim; Kumar, Amand; Stenzhorn, Holger: From GENIA to BioTop: towards a top-level ontology for biology. In: *FOIS 2006 – Proceedings of the 4th International Conference on Formal Ontology in Information Systems*, pp. 103–114. IOS Press, 2006.

[Schulz et al. 2009a] Schulz, Stefan; Beisswanger, Elena; van den Hoek, László; Bodenreider, Olivier; van Mulligen, Erik M.: Alignment of the UMLS semantic network with BioTop: methodology and assessment. In: *Bioinformatics*, Vol. 25, No. 12, pp. i69–76, 2009.

[Schulz et al. 2006b] Schulz, Stefan; Beisswanger, Elena; Wermter, Joachim; Hahn, Udo: Towards an upper-level ontology for molecular biology. In: *AMIA 2006 – Proceedings of the Annual Symposium of the American Medical Informatics Association*, pp. 694–698, 2006.

[Schulz et al. 2009b] Schulz, Stefan; Boeker, Martin; Stenzhorn, Holger; Niggemann, Jörg M.: Granularity issues in the alignment of upper ontologies. In: *Methods of Information in Medicine*, Vol. 48, No. 2, pp. 184–189, 2009.

[Schulz et al. 2008] Schulz, Stefan; Stenzhorn, Holger; Boeker, Martin: The ontology of biological taxa. In: *Bioinformatics*, Vol. 24, No. 13, pp. i313–321, 2008.

[Schulze-Kremer et al. 2004] Schulze-Kremer, Steffen; Smith, Barry; Kumar, Amand: *Revising the UMLS Semantic Network.* 2004. URL: `http://ontology.buffalo.edu/medo/UMLS_SN.pdf` – access date: 2013-01-24.

[Sioutos et al. 2007] Sioutos, Nicholas; Coronado, Sherri d.; Haber, Margaret W.; Hartel, Frank W.; Shaiu, Wen-Ling; Wright, Lawrence W.: NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information. In: *Journal of Biomedical Informatics*, Vol. 40, No. 1, pp. 30–43, 2007.

[Sirin et al. 2007] Sirin, Evren; Parsia, Bijan; Grau, Bernardo C.; Kalyanpur, Aditya; Katz, Yarden: Pellet: a practical OWL-DL reasoner. In: *Journal of Web Semantics*, Vol. 5, No. 2, pp. 51–53, 2007.

[Smith 2006] Smith, Barry: Against idiosyncrasy in ontology development. In: *FOIS 2006 – Proceedings of the 4th International Conference on Formal Ontology in Information Systems*, pp. 15–26. IOS Press, 2006.

[Smith et al. 2007] Smith, Barry; Ashburner, Michael; Rosse, Cornelius; Bard, Jonathan; Bug, William; Ceusters, Werner; Goldberg, Louis; Eilbeck, Karen; Ireland, Amelia; Mungall, Christopher J.; Leontis, Neocles; Rocca-Serra, Philippe; Ruttenberg, Alan; Sansone, Susanna-Assunta; Scheuermann, Richard H.; Shah, Nigam; Whetzel, Patricia L.; Lewis, Suzanna E.: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. In: *Nature Biotechnology*, Vol. 25, No. 11, pp. 1251–1255, 2007.

[Smith et al. 2005] Smith, Barry; Ceusters, Werner; Klagges, Bert; Köhler, Jacob; Kumar, Amand; Lomax, Jane; Mungall, Christopher J.; Neuhaus, Fabian; Rector, Alan L.; Rosse, Cornelius: Relations in biomedical ontologies. In: *Genome Biology*, Vol. 6, No. 5, pp. R46 (1:15), 2005.

[Smith et al. 2003] Smith, Barry; Williams, Jennifer; Schulze-Kremer, Steffen: The ontology of the Gene Ontology. In: *AMIA 2003 – Proceedings of the 27the Annual Symposium of the American Medical Informatics Association*, pp. 609–613. Hanley & Belfus, 2003.

[Spasic et al. 2005] Spasic, Irena; Ananiadou, Sophia; McNaught, John; Kumar, Anand: Text mining and ontologies in biomedicine: Making sense of raw text. In: *Briefings in Bioinformatics*, Vol. 6, No. 3, pp. 239–251, 2005.

[Stenzhorn et al. 2008] Stenzhorn, Holger; Schulz, Stefan; Beisswanger, Elena; Hahn, Udo; van den Hoek, László; van Mulligen, Erik M.: BioTop and ChemTop – top-domain ontologies for biology and chemistry. In: *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference*. CEUR Workshop Proceedings, 2008.

[Stevens et al. 2007] Stevens, Robert; Aranguren, Mikel E.; Wolstencroft, Katy; Sattler, Ulrike; Drummond, Nick; Horridge, Matthew; Rector, Alan L.: Using OWL to model biological knowledge. In: *International Journal of Man-Machine Studies*, Vol. 65, No. 7, pp. 583–594, 2007.

[Stevens et al. 2000] Stevens, Robert; Goble, Carole A.; Bechhofer, Sean: Ontology-based knowledge representation for bioinformatics. In: *Briefings in Bioinformatics*, Vol. 1, No. 4, pp. 398–414, 2000.

[Stojanovic 2004] Stojanovic, Ljiljana: *Methods and Tools for Ontology Evolution*. Universität Karlsruhe (TH), Fakultät für Wirtschaftswissenschaften. Dissertation, 2004.

[Strömbäck et al. 2007] Strömbäck, Lena; Hall, David; Lambrix, Patrick: A review of standards for data exchange within systems biology. In: *Proteomics*, Vol. 7, No. 6, pp. 857–867, 2007.

[Suárez-Figueroa and Gómez-Pérez 2008] Suárez-Figueroa, María del C.; Gómez-Pérez, Asunción: Building ontology networks: how to obtain a particular ontology network life cycle? In: *I-SEMANTICS 2008 – Proceedings of the 3rd International Conference on Semantic Systems*, pp. 142–149. The Journal of Universal Computer Science, 2008.

[Suárez-Figueroa et al. 2010] Suárez-Figueroa, María del C.; Gómez-Pérez, Asunción; Muñoz-García, Óscar; Vigo, Martín: gOntt, a tool for scheduling and executing ontology development projects. In: *SEKE 2010 – Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering*, pp. 614–619. Knowledge Systems Institute Graduate School, 2010.

[Suchanek et al. 2009] Suchanek, Fabian M.; Sozio, Mauro; Weikum, Gerhard: SOFIE: a Self-Organizing Framework for Information Extraction. In: *WWW 2009 – Proceedings of the 18th International World Wide Web Conference*, pp. 631–640. ACM, 2009.

[Thompson et al. 2009] Thompson, Paul; Iqbal, Syed; McNaught, John; Ananiadou, Sophia: Construction of an annotated corpus to support biomedical information extraction. In: *BMC Bioinformatics*, Vol. 10, No. 1, pp. 349, 2009.

[Tomanek et al. 2007] Tomanek, Katrin; Wermter, Joachim; Hahn, Udo: A reappraisal of sentence and token splitting for life sciences documents. In: *MEDINFO 2007 – Proceedings of the 12th World Congress on Medical Informatics*, pp. 524–528. IOS Press, 2007.

[Tsarkov and Horrocks 2006] Tsarkov, Dmitry; Horrocks, Ian: FaCT++ description logic reasoner: system description. In: *IJCAR 2006 – Proceedings of the International Joint Conference on Automated Reasoning*, pp. 292–297. Springer, 2006.

[Tudorache et al. 2008] Tudorache, Tania; Noy, Natalya F.; Tu, Samson; Musen, Mark A.: Supporting collaborative ontology development in Protégé. In: *ISWC*

2008 – Proceedings of the 7th International Semantic Web Conference, pp. 17–32. Springer, 2008.

[UniProt Consortium 2012] UniProt Consortium: *User Manual of the UniProt Knowledgease, Swiss-Prot Protein Knowledgebase and TrEMBL Protein Database*. 2012. URL: `http://web.expasy.org/docs/userman.html` – access date: 2012-07-09.

[U.S. National Library of Medicine 2012] U.S. National Library of Medicine: *Detailed Indexing Statistics: 1965-2011*. 2012. URL: `http://www.nlm.nih.gov/bsd/index_stats_comp.html` – access date: 2012-08-22.

[Uschold and Grüninger 1996] Uschold, Mike; Grüninger, Michael: Ontologies: principles, methods and applications. In: *Knowledge Engineering Review*, Vol. 11, No. 2, pp. 93–155, 1996.

[Uschold and King 1995] Uschold, Mike; King, Martin: Towards a methodology for building ontologies. In: *Proceedings of the IJCAI 1995 Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.

[van Hage et al. 2007] van Hage, Willem R.; Isaac, Antoine; Aleksovski, Zharko: Sample evaluation of ontology-matching systems. In: *EON 2007 – Proceedings of the 5th International Workshop on Evaluation of Ontologies and Ontology-Based Tools*, pp. 41–50. CEUR Workshop Proceedings, 2007.

[van Hage et al. 2005] van Hage, Willem R.; Katrenko, Sophia; Schreiber, Guus: A method to combine linguistic ontology-mapping techniques. In: *ISWC 2005 – Proceedings of the 4th International Semantic Web Conference*, pp. 732–744. Springer, 2005.

[Völker 2009] Völker, Johanna: *Learning Expressive Ontologies*, AKA Verlag / IOS Press, 2009.

[Völker et al. 2008] Völker, Johanna; Haase, Peter; Hitzler, Pascal: Learning expressive ontologies. In: *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pp. 45–69. IOS Press, 2008.

[Völker et al. 2005] Völker, Johanna; Vrandečić, Denny; Sure, York: Automatic evaluation of ontologies (AEON). In: *ISWC 2005 – Proceedings of the 4th International Semantic Web Conference*, pp. 716–731. Springer, 2005.

[Vrandečić 2010] Vrandečić, Denny: *Ontology Evaluation*. Fakultät für Wirtschaftswissenschaften des Karlsruher Instituts für Technologie (KIT). Dissertation, 2010.

[W3C OWL Working Group 2009] W3C OWL Working Group: *OWL 2 Web Ontology Language Document Overview*. 2009. URL: `http://www.w3.org/TR/owl2-overview/` – access date: 2012-08-14.

[Wächter and Schroeder 2010] Wächter, Thomas; Schroeder, Michael: Semi-automated ontology generation within OBO-Edit. In: *Bioinformatics*, Vol. 26, No. 12, pp. i88–96, 2010.

[Wermter et al. 2009] Wermter, Joachim; Tomanek, Katrin; Hahn, Udo: High-performance gene name normalization with GeNo. In: *Bioinformatics*, Vol. 25, No. 6, pp. 815–821, 2009.

[Wikimedia Commons 2013] Wikimedia Commons, user “ArneLH”: *A diagram showing at which stages in the DNA—mRNA—protein pathway expression can be controlled.* 2013. URL: http://en.wikipedia.org/wiki/File:Gene_expression_control.png – access date: 2013-01-20.

[Wroe et al. 2003] Wroe, Chris J.; Stevens, Robert; Goble, Carole A.; Ashburner, Michael: A methodology to migrate the Gene Ontology to a description logic environment using DAML+OIL. In: *PSB 2003 – Proceedings of the 8th Pacific Symposium on Biocomputing 2003*, pp. 624–635. World Scientific Publishing, 2003.

[Yamamoto et al. 2004] Yamamoto, Satoko; Asanuma, Takao; Takagi, Toshihisa; Fukuda, Ken I.: The molecule role ontology: an ontology for annotation of signal transduction pathway molecules in the scientific literature. In: *Comparative and Functional Genomics*, Vol. 5, No. 6-7, pp. 528–536, 2004.

[Zhang and Bodenreider 2006] Zhang, Songmao; Bodenreider, Olivier: Law and order: assessing and enforcing compliance with ontological modeling principles in the Foundational Model of Anatomy. In: *Computers in Biology and Medicine*, Vol. 36, No. 7–8, pp. 674 – 693, 2006.