Serdica J. Computing **7** (2013), No 3, 281–316

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

MAPPING AND MERGING OF ANATOMICAL ONTOLOGIES

Peter Petrov

Abstract. The problem of mapping and merging ontologies in general is an important one in the area of ontology engineering. The same problem considered within the narrower area of anatomical ontologies (AOs) is important in bioinformatics because solving it could enable the transfer of data and the application of knowledge obtained from various model organisms to other model and non-model organisms, and even to research areas such as those of human health and medicine.

This paper presents a detailed summary of the author's PhD research done in the period 2007–2013. The paper's main topic is the problem of mapping and merging of multiple species-specific AOs and the related approaches, methods, and procedures that can be used for solving it.

ACM Computing Classification System (1998): J.3, E.1, G.2.2, G.2.3, I.2.1, I.2.4.

Key words: ontology, anatomical ontology, ontology mapping, anatomical ontology mapping, ontology merging, anatomical ontology merging, external knowledge source, algorithm, graph, directed acyclic graph.

*This article presents the principal results of the Ph.D. thesis *Intelligent systems in bioinformatics: mapping and merging anatomical ontologies* by Peter Petrov, successfully defended at the St. Kliment Ohridski University of Sofia, Faculty of Mathematics and Informatics, Department of Information Technologies, on 26 April 2013.

In this paper the current state of the AO merging and mapping problem is first reviewed. Then a formalization of the problem is suggested. Based on this formalization, an algorithmic procedure for mapping AOs is proposed, which utilizes both syntactic and semantic techniques, including the usage of several existing external knowledge sources (EKSs) containing anatomical information. After that a necessary and sufficient condition is outlined pertaining to the process of merging two given AOs.

Next, the computer program AnatOM developed as part of this study is described. An analysis is done of the results obtained through the use of AnatOM while mapping and merging three particular couples of species-specific AOs. A discussion is presented about the main problems encountered while doing this research. At the end, some perspectives for future development of the work are suggested, and the author's view of this study's contributions is presented.

1. Introduction: importance of the problem and motivation.

The problem of ***ontology mapping*** and ***ontology merging*** is of key importance in the ontology research field in general [3]. Instead of these concepts, often the general concepts ***ontology mediation*** and ***ontology integration*** are used, which indicate both ***mapping*** and ***merging*** of ontologies.

The importance of ontology mediation comes from the fact that ontologies are usually designed and developed by multiple unrelated parties (scientific groups and organizations, software companies, others). This usually leads to the emergence of multiple heterogeneous ontologies which model similar or even identical domains of study. For various reasons (mostly economical and financial), it is virtually impossible for an agreement to be reached between these multiple parties for using a common ontology that would include all the knowledge contained in the many existing heterogeneous ontologies. This in turn makes the exchange of knowledge and information between these distinct parties and between their software systems also difficult, if not practically impossible. The goal of ontology mediation is the creation of an environment which allows free exchange of information and knowledge between the different parties, based on a common ontology which is the result of the mediation/integration of the many distinct heterogeneous ontologies [2].

In this study ***anatomical ontologies (AOs)*** are used. The goal is the ***mediation*** of these ontologies. Two main stages are considered while mediating AOs: 1) mapping the ontologies (discovering the correspondences between them); 2) merging the ontologies.

The motivation for undertaking this study can be found in the context of three main challenges.

Often in biology, it happens that the experimental data obtained for a given organism (e.g., a model organism) may turn out to be more general and thus applicable to other organisms. The current state of knowledge and information in the abovementioned distinct species-specific AOs makes it difficult or even impossible to perform intelligent cross-species searches or mining in the structured information which they contain.

The individual species-specific AOs are useful for extracting or querying data from databases containing information on a particular organism. But performing integrative searches which query multiple heterogeneous anatomical databases is still a difficult task. The reason for this is that each distinct anatomical database uses its own underlying ontology, and distinct ontologies are usually designed and developed based on different purposes, principles, and goals. There is still a lack of inter-ontology connections between the distinct AOs and an almost complete lack of connections from the anatomy of a given organism to other biological domains of study of that organism as e.g. its genotype or phenotype [40].

As of today there is also a lack of reliable mechanisms for cross-querying anatomical data from humans and the same data from various model and non-model organisms. This is due to the significant differences in their terminologies [40] which naturally hinder the attempts to extrapolate or transfer all the accumulated knowledge for various model and non-model organisms to areas such as those of human health and of medicine.

This paper presents a detailed summary of its author's PhD research done in the period 2007–2013. The main subject of this work is the problem of mapping and merging of species-specific AOs and the approaches, methods, and procedures which can be used for solving it.

The paper is organized in 10 parts. Part 1 is this introduction in which the importance of the AO mapping and merging problem is argued for. The motivation for undertaking this study is also presented here. Part 2 presents an informal description of the problem of mapping and merging of AOs. Part 3 describes the objectives of this study. Part 4 is an overview of the current state of the problem. Part 5 is an attempt to formalize the problem at hand. Parts 6 and 7 contain the essence of this research work as they describe the proposed methods for mapping and merging of AOs. In part 8 the computer program AnatOM is described, which is built as part of this study, and which implements the methods proposed and presented in the previous two parts. The AnatOM program semi-automates the processes of mapping and merging of AOs. Part 9 presents the

experiments performed with AnatOM on mapping and merging of several actual AOs of particular categories of organisms, as well as the results obtained from these experiments. It also contains a discussion of various problems encountered during the experiments. In part 10 the perspectives for future development of this research work are outlined, as well as the conclusions that can be drawn from it. That last part also presents the author's view about the main contributions of this study, both scientific and applied.

2. Informal description of the AO mapping and merging problem. The problem of mapping and merging AOs can be described as taking several species-specific AOs as input, detecting various relations (links, connections) between their terms, and generating one general, common, ***species-neutral*** anatomical ontology (called ***super-ontology***) as output. That common AO should include all the knowledge contained in the input AOs. The particular species-specific AOs used in this study describe the anatomies of several very well-studied ***categories of organisms*** usually called ***model organisms*** in biology: mouse (Mus musculus), frog (Xenopus), and zebrafish (Danio rerio).

The concepts of ***ontology mapping*** and ***ontology merging*** seem similar but are quite different in their meanings. Mapping is the process of finding semantic links, or relations, or correspondences between several given input ontologies while merging is the process of creating a new output ontology which is to be used as a union of the given ontologies.

Ideally, the mapping and merging have to be done in an adequate way (adequate in the sense of anatomy, biology, evolutionary science). This means that a given term describing an anatomical part (organ, tissue, cell, etc.) from one organism (e.g., mouse), has to be mapped to a term describing some anatomical part from another organism (e.g., zebrafish), only if the two parts they denote are anatomically similar, or if one of them is considered to have evolved into the other over the course of the evolution of these organisms.

In some cases producing this mapping seems to be simple and even trivial. For example it is obvious that the following anatomical concepts should be mapped onto one another: *brain (mouse) = brain (zebrafish) = brain (frog).*

The difficulties come from the fact that usually (for most terms) generating this terminological mapping is far from being so obvious, as the following examples show.

capillary (mouse) = microcirculatory vessel (zebrafish)
ear (mouse) = auditory apparatus (frog)
myeloid leukocyte (zebrafish) = myeloid cell (frog)

This can be viewed as ***problem #1***—how do we find the mappings between the different terms (concepts) from the given input ontologies, and how do we merge these terms into generalized terms belonging to the output super-ontology?

Other difficulties arise after the mappings between the input ontology terms have already been established. For example, after figuring out that these two terms can be mapped onto each other:

capillary (mouse) = microcirculatory vessel (zebrafish),

a natural question comes up about what should be done with their parent terms and their child terms from the input species-specific AOs and how those parent and child terms should be mapped onto each other.

This can be viewed as ***problem #2***—how can we do the mapping and merging of the relations from the two input ontologies in order to arrive at a complete mapping and merging of the two input AOs as a whole?

3. Objectives of this study. The first main objective of this study was to develop a method and an algorithm for mapping and merging of AOs.

The second main objective was the implementation of this method in the form of an intelligent computer program to be used by biologists and anatomists. This program would semi-automate the processes of mapping and merging of two or more AOs by using the expert knowledge available in several ***external knowledge sources*** (EKSs).

To achieve these objectives the following concrete tasks needed to be solved:

1. defining a formal statement of the problem;

2. representing in a formal form the input species-specific AOs and the available external knowledge sources—UMLS [7, 8], FMA [9, 10], WordNet [11, 12, 13];

3. creating several complementary analytical formal models which describe the processes of mapping and merging of AOs;

4. developing an algorithm based on these models which performs mapping and merging of the input AOs;

5. designing and developing an integrated computer program implementing the mapping and merging algorithms developed as part of this study;

6. making sure that this program is compatible with the current standards for declarative representation of ontologies and AOs in particular.

4. Current state of the problem.

4.1. Ontology—definitions, components, types. Ontology languages. Probably the most popular definition of ontology in the sense of informatics is the one provided by Tom Gruber in [1] which states that an ontology is a ***"specification of a conceptualization"***. This definition is similar but not identical to the initial meaning of the ontology concept known from philosophy[1].

The definition of ontology (in the sense of computer science and informatics) found in Wikipedia is more verbose than Gruber's terse definition. It states that: ***"an ontology formally represents knowledge as a set of concepts within a domain, using a shared vocabulary to denote the types, properties and interrelationships of those concepts; an ontology can be used for generally describing the domain of study as well as for inferring new knowledge about the objects from the domain"***.

The components[2] or building blocks comprising an ontology can be listed as follows: classes, relations, attributes, individual terms, functional terms, restrictions, rules, axioms, events. Not all of them are necessarily present in each and every ontology. The most important among these components are ***classes*** and ***relations***.

Ontologies can be classified based on various properties which they possess. Based on their ***purpose*** ontologies can be classified into: 1.1) ***application*** ontologies and 1.2) ***reference*** ontologies. Based on their ***specificity***, they are usually divided into three groups: 2.1) ***generic*** (upper-level, top-level) ontologies; 2.2) ***core*** ontologies; 2.3) ***domain*** ontologies. Based on their ***expressiveness*** ontologies are grouped into 3.1) ***lightweight*** ontologies and 3.2) ***heavyweight*** ontologies [20, 21].

Ontologies, used currently as formal models for ***knowledge representation*** and ***knowledge inference (reasoning)***, emerged from some earlier, less formal models which served similar purposes such as ***semantic networks*** [27, 28, 24, 29] and ***frame languages*** [22, 23].

Nowadays many formal languages exist for representing ontologies. It could be argued that the most significant ones are **RDF**[3]/**RDFS**[4] [25], **OWL**[5] [24, 26], **OBO**[6] [6]. This study has touching points mostly with the OBO lan-

[1] `http://en.wikipedia.org/wiki/Ontology`

[2] `http://en.wikipedia.org/wiki/Ontology_components`

[3] `http://www.w3.org/RDF/`

[4] `http://www.w3.org/TR/rdf-schema/`

[5] `http://www.w3.org/TR/owl-features/`

[6] `http://www.geneontology.org/GO.format.obo-1_2.shtml`

guage, as this is the language through which most publicly available AOs are currently represented.

4.2. Applications of ontologies in life sciences. Ontologies are very widely used these days in life sciences (biology, biomedicine, medicine, anatomy, genetics, proteomics, comparative genomics, etc.) as models for knowledge representation and knowledge inference. Such ontologies are denoted as ***bioontologies*** and the respective research projects in which they are used are called bioontology projects. As part of this study a detailed review has been done of some of the most significant bioontology systems and research projects: the **Gene Ontology (GO)**[7] [32], **GALEN**[8] [33, 34, 35, 36], **UMLS**[9] [7, 8], **FMA**[10] [9, 10], **OBO** and **OBOFoundry**[11] [5, 6, 39].

4.3. Mediating ontologies—mapping and merging ontologies in general. The very rapid development and popularization of the ontologies as models for knowledge representation and knowledge reasoning in the last decade has been strongly influenced by the idea for the creation of a global semantic network[12] usually denoted as the ***Semantic Web***.

The Semantic Web is a popular idea for further development of the ***World Wide Web***. The goal is the enrichment of the World Wide Web with semantic information, which would enable the processing of the information available within it by automated agents and systems, thus turning it practically from a global Web network to a global Semantic Web network. It is generally assumed that within the Semantic Web all the data would be annotated through the use of ontologies. Therefore the idea of a global Semantic Web is directly related to the problem of mediating or integrating ontologies which are different by their origins but close by their domains of study [3].

One of the problems for implementing the Semantic Web following its original conceptual form is that there is no mechanism through which it can be imposed on distinct individuals, groups, scientific and business organizations, to stick to a single commonly adopted standard set of ontologies [3] for the Semantic Web annotations. It cannot be expected that these distinct parties will ever agree on using one common terminology or one common set of standard widely-accepted ontologies [42] to describe all the various domains of study that are

[7] `http://www.geneontology.org/`

[8] `http://www.opengalen.org/`, `http://www.openclinical.org/prj_galen.html`

[9] `http://www.nlm.nih.gov/research/umls/`

[10] `http://sig.biostr.washington.edu/projects/fm/`

[11] `http://obofoundry.org/`

[12] `http://en.wikipedia.org/wiki/Semantic_Web`

subject of human knowledge and research in general. Therefore other solutions need to be sought instead of imposing such a common terminology. The alternative approach is to enable data, information, and knowledge exchange between the distinct parties and their systems by overcoming (i.e., mediating) the distinctions between the heterogeneous ontologies which they are based on.

The goal of the ontology mediation process is to enable the reuse of ontologies and the sharing of the knowledge annotated through them, thus bridging the gap between various distinct software systems and enabling their interaction [3]. This is why the mediation of ontologies which model similar or even common domains of study is so important [15, 16]. Then in its own turn the problem of mediating the AOs of multiple distinct categories of organisms (species, genera, families, etc.) happens to be a particular case of the general ontology mediation problem.

The terminology related to the process of ***ontology mediation*** which is used in this study has been adopted from [3]. The two terms ***ontology mediation*** and ***ontology integration*** are used as synonyms; they are used as general terms denoting any of the following processes: (i) ontology mapping; (ii) ontology alignment, ontology matching; (iii) ontology merging.

With ***ontology mapping***, the correspondence links between the ontologies are stored separately from the ontologies which are being mapped. These links can be used e.g. for querying heterogeneous knowledge sources through some common interface or for transforming information between multiple distinct representations [3]. The process of automated or semi-automated detection (or discovery) of such links is known as ***ontology alignment*** [3]. With ***ontology merging***, a new output ontology is created which includes all the knowledge from the input ontologies. The main challenge when merging ontologies is to make sure that all the similarities, as well as all the distinctions, between the ontologies which are being merged are reflected in the new, output ontology [3].

In this study it is assumed that the semantic differences between the concepts of ***ontology mapping*** and of ***ontology alignment*** are relatively small, and so only the first term is usually used. Still, while doing so the union of both meanings is meant.

As part of this work, several well-known, general purpose methods, algorithms, software systems, and tools have been reviewed for mapping and merging of ontologies, such as **MAFRA**[13] [43], **RDFT** [44], **PROMPT** [45, 46], **Anchor-PROMPT** [47], **QOM** [48, 49], **OntoMerge**[14] [30, 38]. These are considered

[13] `http://mafra-toolkit.sourceforge.net/`

[14] `http://cs-www.cs.yale.edu/homes/dvm/daml/ontology-translation.html`

general purpose ones as they can be used for mediating ontologies which model arbitrary domains of study.

4.4. Mapping and merging AOs—the Uberon project. With respect to mapping and merging AOs, the Uberon[15] project is probably the most significant and the largest in scale to date. The main goal of Uberon is the unification (i.e., the merging, the integration) of the many existing species-specific AOs in one (or in a set of just a few) common, general species-neutral AOs.

The project started in 2008–2009 [31]. One of its main goals is to fill in the gap formed by the lack of a common species-neutral ontology which would describe the anatomies of a wide range of species. This lack turns out to be the main hurdle for transferring the available research data from various model organism to areas such as those of human health and of medicine. Another goal of the project is to bridge the gap between the reference **CARO**[16] ontology (an upper-level and rather abstract ontology) and the various species-specific AOs (either existing ones or still to be designed and implemented in the future). Within the Uberon project, multiple publicly available ontologies are examined and studied. Some of these ontologies are purely anatomical ones while others indirectly (i.e., implicitly) contain nested AOs.

In the beginning of 2012, the authors of Uberon announced the first mature version of the Uberon ontology [40]. Apart from the many internal terms and relations which it contains, the Uberon ontology also contains many external links (cross links) to a variety of existing species-specific AOs, which contributes to Uberon's tight integration with these AOs.

5. Formalization of the problem. Here the input data and the EKSs used in this study are formalized. While doing this the author's perception of ontologies mainly as ***directed acyclic graphs (DAGs)*** has been used.

Three models are presented here: **model #1**—the model of the input ontologies presented as DAGs; **model #2**—the model of the input ontologies after they have been mapped onto each other; **model #3**—the model of the output ontology which is called ***super-ontology*** throughout this text.

Formalizing the two input ontologies is done by representing them as follows:

$$\boldsymbol{O_1 : G_1 = DAG_1 = (V_1, E_1);\ F_1\ :\ E_1 \to C = \{c_1, c_2, \ldots, c_n\}}$$

$$\boldsymbol{O_2 : G_2 = DAG_2 = (V_2, E_2);\ F_2\ :\ E_2 \to C = \{c_1, c_2, \ldots, c_n\}}$$

[15] `http://uberon.org/`

[16] `http://www.bioontology.org/wiki/index.php/CARO:Main_Page`

Here O_1 and O_2 are the two input ontologies. Each of them consists of a DAG denoted as G_k or DAG_k and a function F_k which colors the edges of this graph. The vertices of these graphs represent the terms of the two input ontologies, while the edges represent the relations between the terms. $C = \{c_1, c_2, \ldots, c_n\}$ is a set of the colors used by the coloring function. Each of these colors represents one of the several existing types of ***inner-ontology (IO) relations***. Each of these existing relations is a ***subsumption relation*** of some kind.

The types of relations which are most commonly used in AOs are the following: ***is_ a*** (generalization/specialization) and ***part_ of*** (aggregation/membership). Sometimes other relation types are used too but for the purposes of this study, it is assumed that only these two relation types are present, i.e., it is assumed that n=2, c_1=***is_ a***, c_2=***part_ of***.

Here V_1 is the set of terms (also called concepts) from the AO of one particular organism, and V_2 is the set of terms from the AO of another organism. So

$$V_1 = \{v_{11}, v_{12}, \ldots, v_{1n_1}\}, \quad |V_1| = n_1;$$
$$V_2 = \{v_{21}, v_{22}, \ldots, v_{2n_2}\}, \quad |V_2| = n_2$$

where n_1 and n_2 are the counts of the terms in the two input ontologies.

The relations in these two given DAGs are called inner-ontology (IO) relations. These relations are of the kind parent–child. As mentioned, these are mostly ***is_ a*** and ***part_ of*** relations but in practice others are present too. The inner-ontology relations are always asymmetric (i.e., it does matter which term is the parent and which one is the child). So in this study the different types of IO relations from the input ontologies are modeled through coloring their respective edges (in the given DAGs) in different colors.

Here a few notes are made which aim to clarify a few important aspects of this study's terminology.

1) In this study the concept of ***parent–child relation*** means any asymmetric relation. As both ***is_ a*** and ***part_ of*** are asymmetric, they are both viewed as parent–child relations.

2) The AOs used here sometimes contain not just information about the adult organism, but also about phases (prenatal or postnatal) of the organism's development. So in the input ontologies, terms can be found which are related to the processes and the phases of this development. Such relations are for example ***develops_ from***, ***start_ stage***, ***end_ stage***, ***preceded_ by***. These relations are not directly considered in this study as the study is only concerned with the anatomies of the adult organisms. In theoretical aspect though, they are no

different than the ***is_ a*** and ***part_ of*** relations. So the methods and algorithms described in this text can handle them without any special modifications.

3) This study practically does not deal with inner-ontology relations other than ***is_ a*** and ***part_ of***. Actually many of the existing AOs do not contain other types of relations. In the AOs which do contain other relations, there are no uniform meanings of these relations which can be considered valid beyond the borders of the particular ontology they are found in. In a theoretical aspect though, the algorithmic approach for mapping and merging AOs, suggested in this study can be applied to other asymmetric inner-ontology relation types and not just to ***is_ a*** and ***part_ of***.

4) Throughout this study, when ontologies or AOs in particular are discussed, the two terms ***terms*** and ***concepts*** are always used as synonyms. The graph theory concepts ***edge*** and ***arc*** are also used as such even though according to certain authors the ***arc*** concept should be used for directed graphs only, while the ***edge*** concept should be used for undirected graphs only.

The available EKSs $\boldsymbol{T_s}$ and the information contained in them are represented formally as a family of two sets: ***the set of the terms*** and ***the set of the relations***, defined by each of the available EKSs.

- ***Set of terms***

$$\boldsymbol{Ms = \{t_{s1}, t_{s2}, \ldots, t_{sm_s}\}}$$

Here $\boldsymbol{t_{sk}}$ *=* ***(id$_{sk}$; name$_{sk}$)*** denotes a term***,*** $\boldsymbol{id_{sk}}$ is a string identifier of the term $\boldsymbol{t_{sk}}$, $\boldsymbol{name_{sk}}$ is the name of the term $\boldsymbol{t_{sk}}$ (and is also a string), $\boldsymbol{m_s}$ is the count of terms in the EKS $\boldsymbol{T_s}$.

- ***Set of subsumption relations***

$$\boldsymbol{R'_{T_s} = R^{is\_a}_{T_s} \subseteq M_s \times M_s; \quad R_{T_s}{}'' = R^{part\_of}_{T_s} \subseteq M_s \times M_s}$$

These are the relations which each EKS defines over the set of its terms i.e. these are the relations ***is_ a*** and ***part_ of***, as defined by the respective EKS $\boldsymbol{T_s}$ ($\boldsymbol{s = 1, 2, 3}$). As already noted, the EKSs usually define other relations too (other than ***is_ a*** and ***part_ of***) but this study is limited to considering ***is_ a*** and ***part_ of*** relations only.

Based on the formal symbols and definitions (about the input AOs and about the EKSs) introduced here, in the text which follows a formal statement of the mapping and merging problem is suggested, as well as an algorithmic method for solving that problem.

It is important to note that, even though this study is about mapping and merging AOs, the method and the algorithmic procedures suggested here are

general enough and could be naturally applied to other domains of study (other than anatomy). The only prerequisite for that would be the existence of EKSs which are semantically close enough to that other domain of study. In fact one of the EKSs used here (WordNet) could be applied to almost any domain because it is a general-purpose knowledge source, i.e., one that is not tied to any particular domain of study.

6. Proposed method for mapping AOs.

6.1. Formalization of the problem statement. This study's main objective is to find the semantic links between two given AOs $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$—e.g., between the mouse AO and the zebrafish AO. This is done mainly through the use of the available EKSs and the sets of ***is_a*** and ***part_of*** relations which they define between their terms. In this study three EKSs are used—$\boldsymbol{T_1}$, $\boldsymbol{T_2}$, $\boldsymbol{T_3}$ (***UMLS***, ***FMA***, ***WordNet*** respectively). One of the goals here is to detect, or predict, or discover a set of reliable, i.e., adequate semantic links between the terms of the two input ontologies. These links should make sense (be adequate) from both a biological and an anatomical point of view, and should be of one of the following relation types:

$\boldsymbol{R_1} = \boldsymbol{R_{syn}}$ (synonymy),
$\boldsymbol{R_2} = \boldsymbol{R_{hyper}}$ (hypernymy), $\boldsymbol{R_3} = \boldsymbol{R_{hypo}}$ (hyponymy),
$\boldsymbol{R_4} = \boldsymbol{R_{holo}}$ (holonymy), $\boldsymbol{R_5} = \boldsymbol{R_{mero}}$ (meronymy).

So the goal is the detection of semantic links from the just mentioned types $\boldsymbol{R_k} \subseteq (\boldsymbol{V_1} \times \boldsymbol{V_2}) \cup (\boldsymbol{V_2} \times \boldsymbol{V_1})$, such that these links are anatomically adequate (i.e., make sense from an anatomical point of view). Of greatest interest is the detection of synonymy links as they allow one to directly come up with a mapping of the two input ontologies $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$ onto each other. This then leads to their merging in one general output ontology denoted as $\boldsymbol{O_{super}}$ and called ***super-ontology*** in this study.

6.2. Algorithmic solution for mapping AOs.

Phase 1—Generating thesauri. This is the first preparatory phase of the algorithm. During this phase from the two input AOs $\boldsymbol{O_k}$ $\boldsymbol{(k = 1, 2)}$ their respective thesauri $\boldsymbol{Th_k}$ $\boldsymbol{(k = 1, 2)}$ are built. The thesauri $\boldsymbol{Th_k}$ are tabular structures similar to hash tables. The table $\boldsymbol{Th_k[t.id]}$ maps the identifiers of all terms $\boldsymbol{t} \in \boldsymbol{V_k}$ to a list of their respective names. These names can be either primary names or alternative names (the alternative names are the synonyms of the term $\boldsymbol{t}$ as defined by the input ontology). Thus for any $\boldsymbol{id}$ from the input ontology $\boldsymbol{O_k}$, the list $\boldsymbol{Th_k[id]}$ contains strings which define the names of the term $\boldsymbol{t} \in \boldsymbol{V_k}$ having $\boldsymbol{id}$ as its identifier.

Phase 2—Mapping the two input ontologies to the available EKSs. This is the second preparatory phase of the algorithm. Here each of the two input ontologies is mapped to all the EKSs that are available. As already noted, the particular EKSs used in this study are T_1=***UMLS***, T_2=***FMA***, T_3=***WordNet***, but the count of the EKSs is not significant. Actually, during this phase not the ontologies themselves but their thesauri Th_1 and Th_2 which were generated from them, are mapped to the EKSs. This mapping is performed as described below. The process below is described in terms of T_1 but the same process is then repeated using T_2 and T_3.

Procedure 2.1. For each identifier k of a term $t \in V_1$ **:** the list $L = Th_1[k]$ is retrieved from the thesauri Th_1 of O_1.

Procedure 2.2. For each term name $s \in L$ **:** all distinct identifiers (identifiers defined by the EKS T_1) are retrieved which correspond to the name s.

$$RS_1 = \{(t^I.id) \mid t^I \in T_1 \text{ and } t^I.name = s\}$$

Step 2.2.1. For each identifier id from RS_1 **:** the set

$$RS_2 = \{(t^{II}.id,\ t^{II}.name) \mid t^{II} \in T_1 \text{ and } t^{II}.id = t^I.id\}$$

is retrieved from the EKS T_1.

After this step, the synonyms of s (as defined by the EKS T_1) are now known. These are called the T_1− ***synonyms of*** s. Strictly speaking this is the set

$$RS_2^* = \{t^{II}.id \mid t^{II} \in T_1 \text{ and } t^{II}.id = t^I.id\}$$

which consists of the first components of the ordered couples, contained in RS_2.

Step 2.2.2. For each identifier id from RS_1 **:** the set

$$RS_3 = \{(t^{III}.id) \mid t^{III} \in T_1 \text{ and } ((t^{III}, t^I) \in R_{T_1}^{is\_a} \text{ or } (t^{III}, t^I) \in R_{T_1}^{part\_of})\}$$

is retrieved from the EKS T_1.

The result of performing this step is that the following two sets are now known:

• The meronyms of s defined by T_1 and called T_1-***meronyms of*** s; strictly speaking this is the set:

$$RS_{3,1} = \{t^{III}.id \mid t^{III} \in T_1 \text{ and } (t^{III}, t^I) \in R_{T_1}^{part\_of}\}.$$

• The hyponyms of s defined by T_1 and called T_1-*hyponyms of* s; strictly speaking this is the set:

$$RS_{3,2} = \{t^{III}.id | t^{III} \in T_1 \text{ and } (t^{III}, t^{I}) \in R_{T_1}^{is\_a}\}.$$

It should be noted that at this point the following two conditions are usually met:

$$RS_{3,1} \cup RS_{3,2} = RS_3 \text{ and } RS_{3,1} \cap RS_{3,2} = \emptyset.$$

Step 2.2.3. For each identifier id from RS_1 **:** the set

$$RS_4 = \{(t^{IV}.id) | t^{IV} \in T_1 \text{ and } ((t^{I}, t^{IV}) \in R_{T_1}^{is\_a} \text{ or } (t^{I}, t^{IV}) \in R_{T_1}^{part\_of})\}.$$

is retrieved from the EKS T_1.

The result from performing this step is that the following two sets are now known:

• The holonyms of s defined by T_1 and called T_1-*holonyms of* s; strictly speaking this is the set:

$$RS_{4,1} = \{t^{IV}.id | t^{IV} \in T_1 \text{ and } (t^{I}, t^{IV}) \in R_{T_1}^{part\_of}\}.$$

• The hypernyms of s defined by T_1 and called T_1-*hypernyms of* s; strictly speaking this is the set:

$$RS_{4,2} = \{t^{IV}.id | t^{IV} \in T_1 \text{ and } (t^{I}, t^{IV}) \in R_{T_1}^{is\_a}\}.$$

It should be noted that at this point the following two conditions are usually met:

$$RS_{4,1} \cup RS_{4,2} = RS_4 \text{ and } RS_{4,1} \cap RS_{4,2} = \emptyset$$

Applying the above procedures and steps for all the EKSs T_s completes the process of mapping the input ontologies to the available EKSs.

Phase 3—Discovering cross-ontology synonymy links and cross-ontology parent–child links (is_ a and part_ of). During this phase, three separate algorithmic procedures are applied, which are denoted as ***DM (direct matching)***, ***SMP (source matching predictions)*** and ***CMP (child matching predictions).*** After applying each of these algorithmic procedures the two DAGs get linked through new sets of ***cross-ontology semantic links*** called

DM, SMP, and CMP links respectively. In this way another graph $\boldsymbol{G = (V, E)}$ emerges, called an ***intermediate result graph***. So each of the three procedures described below adds new cross-ontology links to the intermediate result graph $\boldsymbol{G}$, thus modifying the graph $\boldsymbol{G}$.

Procedure 3.1. During the execution of this procedure (called **DM**) direct (textual, syntactic) matches are discovered, and based on them predictions are generated for cross-ontology synonym terms from the two ontologies. Among the terms from the two ontologies simple textual matching of their names is sought. So this procedure is practically trivial—it iterates over all terms $\boldsymbol{t_1 \in V_1}$ and $\boldsymbol{t_2 \in V_2}$ and checks if the following holds true: $\boldsymbol{t_1.name = t_2.name}$. When such a match is found, $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ are marked as synonyms, and it is also recorded that this synonymy prediction originates from direct matching (***DM***). These predictions are called ***DM–predictions***.

Procedure 3.2. During the execution of this procedure (called **SMP**) more predictions are generated. These are synonymy cross-ontology links and parent-child cross-ontology links (***is_ a*** and ***part_ of***). At this point the two input ontologies are already mapped to the available EKSs. Based on that, several simple logical rules are applied from a predefined set of rules. Applying these rules generates predictions which are called ***SMP–predictions***. Here are the logical rules that are applied by SMP.

Rule (A) If two terms $\boldsymbol{t_M \in O_1}$ and $\boldsymbol{t_Z \in O_2}$ have been found to be synonyms of the same term $\boldsymbol{t \in T_k}$, $\boldsymbol{t_M}$ and $\boldsymbol{t_Z}$ are marked as predicted (through SMP) cross-ontology synonyms of each other.

Rule (B) If the term $\boldsymbol{t_j \in O_j}$ has been found to be a synonym of the term $\boldsymbol{t \in T_k}$ and if the term $\boldsymbol{t_{3-j} \in O_{3-j}}$ has been found to be a (***is_ a/part_ of***) child/parent of $\boldsymbol{t}$, then $\boldsymbol{t_j}$ is marked as a predicted (through SMP) cross-ontology (***is_ a/part_ of***) parent/child of $\boldsymbol{t_{3-j}}$ (here $\boldsymbol{j = 1}$ ***or*** $\boldsymbol{2}$ and respectively $\boldsymbol{3 - j = 2}$ ***or*** $\boldsymbol{1}$).

Through the application of the above rules the SMP procedure discovers a set of cross-ontology links (both synonymy and parent-child links) between the vertices of the graphs $\boldsymbol{DAG_1}$ and $\boldsymbol{DAG_2}$ (i.e., between the terms from $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$). The evidences (i.e., the arguments) about their reasonableness and adequacy originate in the information contained in the available EKSs. For these cross-ontology links, the SMP procedure records their types (***synonymy, is_ a, part_ of***) and the fact that they originate from SMP.

Procedure 3.3. This is the **CMP** (child matching predictions) procedure. This procedure generates yet more cross-ontology links (i.e., predictions) in addition to those generated by DM and SMP. They are called ***CMP-predictions.***

Here, in parallel to describing the ***CMP*** procedure, several definitions are introduced, which associate a ***score*** to each and every link (either inner-ontology or cross-ontology link) from the ***intermediate result graph G*** (viewed in the state it had before the execution of the CMP procedure). These definitions are given following a hierarchical approach, moving from something simpler onto something more complex, so actually they build upon each other. The definitions let us arrive at one final number called the ***final (aggregated) CMP score***, which is assigned to what is called the ***final (aggregated) CMP link***, which is introduced by CMP between terms $t_1 \in V_1$ and $t_2 \in V_2$.

The CMP procedure tries to find new links from the types R_1, R_2, R_3, R_4 and R_5 between two terms $t_1 \in V_1$ and $t_2 \in V_2$. The CMP considers several ***patterns of connectivity*** which include the vertices $t_1 \in V_1$ (parent 1) and $t_2 \in V_2$ (parent 2), as well as the children of t_1 and t_2 from the two input ontologies. These patterns are sought in the graph G, and more specifically in the state that graph had after the DM and SMP procedures finished their executions, but before the CMP procedure is executed. The following three patterns of connectivity are considered by the CMP procedure:

(1) $t_1 \in V_1 \leftarrow t_{ch1} \in V_1 \leftrightarrow t_{ch2} \in V_2 \rightarrow t_2 \in V_2$ (called **U-pattern**);

(2) $t_1 \in V_1 \leftarrow t_{ch2} \in V_2 \leftrightarrow t_{ch1} \in V_1 \rightarrow t_2 \in V_2$ (called **X-pattern**);

(3) $t_1 \in V_1 \leftarrow t_{ch1} \in V_1 \rightarrow t_2 \in V_2$ or $t_1 \in V_1 \leftarrow t_{ch2} \in V_2 \rightarrow t_2 \in V_2$ (called **V-pattern**).

In this notation, the unidirectional arrows $\rightarrow$ and $\leftarrow$ denote sets of parent–child links which either come from SMP, or are inner-ontology (IO) links. These links are asymmetric and their arrows always point from the child to the parent term. Contrariwise, the bidirectional arrows $\leftrightarrow$ denote sets of synonymy links which originate from DM or SMP. These are symmetric links.

Each occurrence of a pattern (of one of these three types) between t_1 and t_2 (the two parent terms) is called a ***pattern instance***. It is important to note that all asymmetric links within a given pattern instance either denote ***is_a*** or denote ***part_of*** links, i.e., it is not allowed to mix these two types of parent–child links within a single pattern instance.

Based on the patterns of connectivity found, the CMP procedure introduces new cross-ontology links between the ontology terms t_1 and t_2, called ***individual CMP links***. To assign scores to them, the concepts ***score of a set of non-CMP links*** and ***score of a pattern instance*** are introduced first. The score of the pattern instance also becomes the ***score of the individual CMP link***. At the end of the day, the scores of all individual CMP links between t_1 and t_2 are aggregated through the use of an ***aggregation function***. Here are

the main definitions related to the CMP procedure.

Definition 1 (Conj). The $\boldsymbol{Conj}$ function takes $\boldsymbol{N}$ arguments from the interval [0, 1] and returns a result in [0, 1]. It is defined recursively as follows:

1.1. $\boldsymbol{Conj(A_1, A_2) = A_1 \bullet A_2}$;

1.2. $\boldsymbol{Conj(A_1, A_2, ..., A_N) = Conj(Conj(A_1, A_2, ..., A_{N-1}), A_N)}$, for $\boldsymbol{N \geq 3}$.

Definition 2 (Disj). The $\boldsymbol{Disj}$ function takes $\boldsymbol{N}$ arguments from the interval [0, 1] and returns a result in [0, 1]. It is defined recursively as follows:

2.1. $\boldsymbol{Disj(A_1, A_2) = A_1 + A_2 - A_1 \bullet A_2}$;

2.2. $\boldsymbol{Disj(A_1, A_2, ..., A_N) = Disj(Disj(A_1, A_2, ..., A_{N-1}), A_N)}$, for $\boldsymbol{N \geq 3}$.

Definition 3 (score of a non-CMP link). The score of a link that does not originate from CMP is defined as follows:

$$\boldsymbol{score}(s_{ij}) = \begin{cases} \boldsymbol{I}, & \textit{if } s_{ij} \textit{ is an IO link} \\ \boldsymbol{D}, & \textit{if } s_{ij} \textit{ is a DM link} \\ \boldsymbol{f(T)}, & \textit{if } s_{ij} \textit{ is a SMP link originating from the EKS } \boldsymbol{T} \end{cases}$$

In this formula ***IO link*** denotes an inner-ontology link; ***DM link*** denotes a link which originates from the DM procedure; ***SMP link*** denotes a link which originates from the SMP procedure; $\boldsymbol{s_{ij}}$ is a link from one of the kinds IO, DM, SMP; $\boldsymbol{I}$ and $\boldsymbol{D}$ are constants from [0, 1] (usually $\boldsymbol{I = 1}$ and $\boldsymbol{D = 1}$); $\boldsymbol{f(T)}$ is a constant assigned up front to the EKS $\boldsymbol{T}$.

Definition 4 (score of a set of non-CMP links). The score of a set of non-CMP links is defined as follows: $\boldsymbol{score(\bar{S}_i) = Disj_{j=1}^{m_i}(score(s_{ij}))}$. In this formula $\boldsymbol{Disj}$ is the function from definition 2, $\boldsymbol{s_{ij}}$ are either IO or DM or SMP links, and the $\boldsymbol{Disj}$ function is applied over all links from $\boldsymbol{\bar{S}_i}$.

Definition 5 (score of a pattern instance, i.e., of an individual CMP link). The score of an individual CMP link $\boldsymbol{e}$ (the score of the pattern instance which has given rise to this link) is defined as: $\boldsymbol{score(e) = p.Conj_{i=1}^{n}(score(\overline{S}_i))}$. Here the number $\boldsymbol{p} \in (0, 1)$ is a constant (called ***CMP penalty constant***), $\boldsymbol{Conj}$ is the function from definition 1, the $\boldsymbol{Conj}$ function is applied over all sets of links taking part in the pattern instance which has given rise to the CMP link $\boldsymbol{e}$.

Definition 6. Let $\boldsymbol{t_1 = Par1 \in V_1}$ and $\boldsymbol{t_2 = Par2 \in V_2}$ denote two terms from the two input AOs. Let $\boldsymbol{G}$ denote the ***intermediate result graph*** obtained from $\boldsymbol{DAG_1}$ and $\boldsymbol{DAG_2}$ after all the **DM links** and all the **SMP links** have been introduced by procedures 3.1 (DM) and 3.2 (SMP). Let also:

(6.1) $\boldsymbol{u = \{u_1,\ u_2,\ \ldots,\ u_{N_u}\}}$ be the set of all concrete occurrences of ***U-patterns***, in which the terms $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ take part as parent terms ($\boldsymbol{N_u \geq 1}$);

(6.2) $\boldsymbol{x = \{x_1,\ x_2,\ \ldots,\ x_{N_x}\}}$ be the set of all concrete occurrences of ***X-patterns***, in which the terms $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ take part as parent terms ($\boldsymbol{N_x \geq 1}$);

(6.3) $\boldsymbol{w = \{w_1,\ w_2,\ \ldots,\ w_{N_w}\}}$ be the set of all concrete occurrences of ***V-patterns***, in which the terms $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ take part as parent terms ($\boldsymbol{N_w \geq 1}$);

(6.4) $\boldsymbol{N_u + N_x + N_w > 0}$;

(6.5) $\boldsymbol{PIS(t_1, t_2) = u \cup x \cup w}$ be the set of all concrete occurrences of any patterns in which the terms $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ take part as parents (***PIS*** is an abbreviation from ***pattern instance set***);

(6.6) $\boldsymbol{|PIS(t_1, t_2)| > 0}$.

Then the final result of the execution of the ***CMP procedure*** (as far as $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ are concerned) is that one ***generalized (final, aggregated) CMP link*** denoted as $\boldsymbol{e_{CMP}(t_1,\ t_2)}$ is introduced between the terms $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$. Its score is defined as follows:

$$\boldsymbol{score_{CMP}(t_1, t_2) \ = \ \underset{\forall p \in PIS(t_1,\, t_2)}{MAX} (score(p)).}$$

Here $\boldsymbol{p}$ denotes one particular pattern instance from ***PIS***. So the ***MAX*** function is taken over all pattern instances in which $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ are involved as parent terms.

The ***MAX*** function is one particular ***aggregation function*** that is used here. When implementing the CMP procedure other aggregation functions may be used instead.

In the above given description of the ***CMP*** procedure, it was shown how ***individual CMP links*** can be introduced between any two terms $\boldsymbol{t_1 \in V_1}$ and $\boldsymbol{t_2 \in V_2}$ as long as the condition (6.4) (or its equivalent condition (6.6)) is met. Numeric scores were then defined for the individual CMP links. At the end, it was shown how the set of ***individual CMP links*** between any two terms $\boldsymbol{t_1 \in V_1}$ and $\boldsymbol{t_2 \in V_2}$ can be aggregated into a ***generalized (final, aggregated) CMP link*** $\boldsymbol{e_{CMP}(t_1, t_2)}$ between the terms $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$. These ***aggregated CMP links*** between terms from the two input ontologies, together with their scores, form the final result of the execution of the CMP procedure.

7. Proposed method for merging AOs. In this part several key definitions are given and two important statements are presented (their proofs omitted in this paper) which solve theoretically the questions about transforming the ***intermediate result graph G*** into a special ***generalized graph*** denoted as

G^*, and about generating (provided that certain conditions are met) an output ***super-ontology*** from the generalized graph G^*.

Definition 7. Let RZ be a relation over the set of vertices of the intermediate result graph G ($RZ \subseteq V \times V$), defined as follows: $(v_1, v_2) \in RZ$, if the vertices $v_1$and v_2 are connected by at least one synonymy link in the graph G. By default it is also defined that $(v, v) \in RZ$ for each $v \in V$.

Clearly the relation RZ defined in this way is symmetric, i.e., if $(v_1, v_2) \in RZ$, then $(v_2, v_1) \in RZ$ too. This is so because the synonymy links in G are (by their nature) bidirectional (i.e., undirected) links. Also, by definition the relation RZ is reflexive.

Definition 8. Let RZ^* denote the transitive closure of the relation RZ.

Apparently, RZ^* is an equivalence relation defined over the set of vertices $V$of the graph G.

Definition 9. The generalized graph $G^* = (V^*, E^*)$ is defined in the following way: let $V^* = \{C_1^*, C_2^*, ..., C_n^*\}$ be the set of the equivalence classes (in V), which are produced by the partitioning of V imposed by the relation RZ^*; let also an edge $e = (C_k^*, C_l^*)$ colored in $c \in \{is\_a, part\_of\}$ belong to E^* if and only if $\exists u \in C_k^*$ and $\exists v \in C_l^*$ such that $u, v \in V$ and the vertices u and v are connected by an edge $(u, v) \in E$ having the same color c.

Definition 10. A cycle from the graph G is called ***acceptable*** if all edges which are part of it represent ***synonymy links***, i.e., if it contains no edge of any of the types ***is_a*** and ***part_of***. A cycle from the graph G is called ***unacceptable*** if it is not acceptable, i.e., if it contains at least one edge of the types ***is_a*** or ***part_of***.

Statement 1. If the graph G contains only acceptable cycles, the graph G^* is acyclic.

Statement 2. If the graph G^* is acyclic, the graph G contains only acceptable cycles.

These two statements provide a ***necessary and sufficient condition*** for the graph G^* to be acyclic. Apparently only then is it possible to generate an output ***super-ontology*** from it. As already noted, the proofs of these two statements are omitted here. They are quite simple and can be found in the complete text of the author's dissertation. These two statements give the theoretical basis for answering the questions when it is possible to generate an output super-ontology from the intermediate result graph and when it is not.

8. AnatOM—a computer program for mapping and merging AOs. Here the computer program AnatOM is described. Its name is an

abbreviation from ***Anatomical Ontologies Merger***. The AnatOM program implements the theoretical models and algorithmic procedures presented in the preceding two parts.

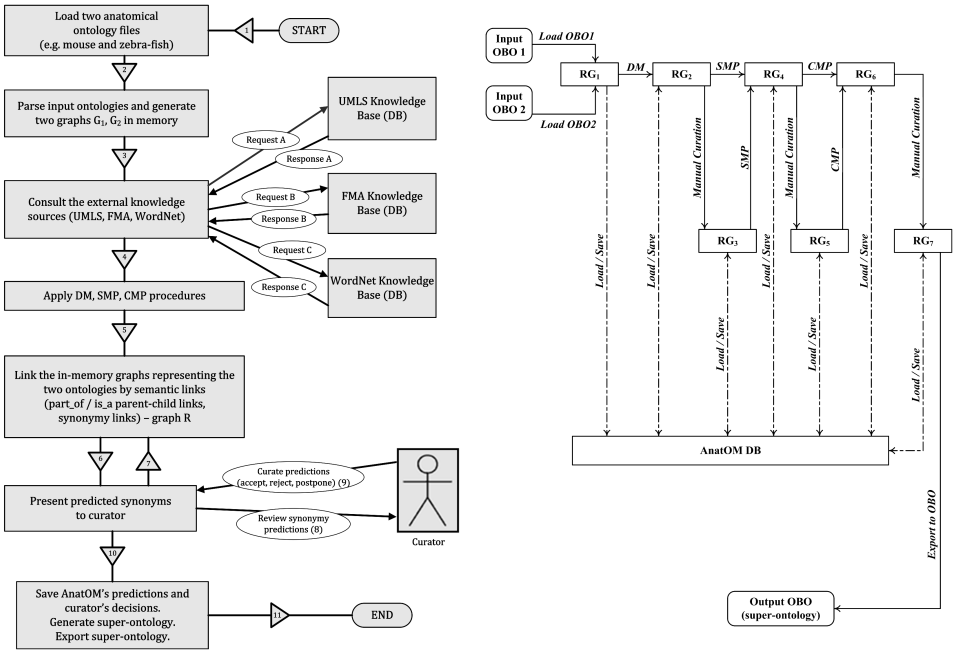

Fig. 1. The main execution stages in a typical run of the AnatOM program

AnatOM has been developed as part of this study with a particular practical goal in mind—the semi-automation of the processes of mapping and merging AOs which are typically performed manually by anatomists. For this purpose, the three algorithmic procedures are integrated in AnatOM to arrive at one complete, integrated application solving the mapping and merging tasks. Thus the program is designed to aid anatomists in the processes of mapping and merging of AOs. The anatomists using the program are supposed to review and manually edit (i.e., curate) the predictions auto-generated by the program, thus refining the mappings, and to then merge the input species-specific AOs, thus generating a species-neutral super-ontology as an output.

The AnatOM program is a typical ***standalone desktop application*** with a ***graphical user interface (GUI)*** implemented on the ***Microsoft.NET***[17] platform. The programming language used in its development is ***C#*** – probably

[17] `http://www.microsoft.com/net`

the most popular and widely-used language for the Microsoft.NET framework.

The input ontologies are entered into AnatOM as OBO files (files describing ontologies in a declarative manner through the use of the OBO ontology language[18]). The available EKSs (UMLS, FMA, WordNet) are represented as ***relational databases*** managed by a MySQL RDBMS [14]. Apart from these 3 databases, AnatOM uses an additional 4th MySQL database for storing various temporary intermediate data from the operations it does and the calculations it performs.

In Figure 1 two diagrams are presented which aim to illustrate the main stages and transitions in a typical run of the AnatOM program. In the diagram on the right $\boldsymbol{RG_k}$ denotes the ***intermediate result graph*** and the various states it goes through during the execution of the program. When the appropriate command is triggered (through the GUI menus of the program), the program reads and loads the two OBO files of the two input AOs. This is done by the ***OBOParser.NET*** module. Then a special factory object converts the objects that resulted from loading the OBO files and transforms them into Graph, Node, Edge objects, i.e., into objects exposed by the ***Graph.NET*** module. Again from the GUI menus of AnatOM separate threads are started which execute the three algorithmic procedures—DM, SMP, CMP. These threads implement the main logic of the program so they form the ***logical module*** of AnatOM. The threads process the two Graph objects $\boldsymbol{G_1}$ and $\boldsymbol{G_2}$ representing the two input AOs by adding new edges to them. These added edges have one of their ends in $\boldsymbol{G_1}$, and the other one in $\boldsymbol{G_2}$. These edges are actually the ***auto-predicted cross-ontology links*** described earlier in the text. The DM and CMP procedures have no need to communicate with the three relational databases representing the three EKSs (UMLS, FMA, WordNet). SMP is the procedure which communicates with these databases. This database communication is done through the ***DataAccess*** module. As noted already, the main logic of AnatOM is implemented in three classes defining three separate threads of execution which correspond to the three algorithmic procedures DM, SMP, CMP. These threads work on the ***intermediate result graph*** and modify it. The modified graph is passed on to a ***visualization module*** which presents it in graphical form to the user. At the end, the ***export module*** allows the user to export the ***result graph*** into a ***super-ontology*** represented as an OBO file. The modules comprising the AnatOM program are described below in brief.

8.1. OBOParser.NET. The OBOParser.NET module is a specialized library developed for reading and loading OBO files in memory. It follows an

[18] `http://www.geneontology.org/GO.format.obo-1_2.shtml`

event-based design. This module defines structures describing the main elements from the OBO syntax [4]—***stanzas, terms, term identifiers, term names, term synonyms, relations between terms***. These structures are returned to the calling module after OBOParser.NET has read and parsed an OBO file given to it. The OBOParser.NET module implements a typical ***recursive descent parser***. This module gets called when AnatOM is instructed to load the input AOs which is done from the File menu in the AnatOM GUI.

8.2. Graph.NET. The Graph.NET module is a general purpose library for handling graphs. It defines classes such as Node, Edge, and Graph, and allows storing arbitrary objects as properties of each Node object and each Edge object from the Graph.

To its calling modules Graph.NET provides methods for: adding nodes/vertices, adding and removing edges, getting all edges of the graph, getting all edges between two given nodes, getting all parents or children of a given node, getting the start and end nodes of a given edge, counting all nodes or edges in the graph, applying various useful graph algorithms as e.g. the topological sort algorithm, Tarjan's algorithm [19], Johnson's algorithm [18].

The library can be easily extended and enhanced with other useful algorithms for working with graphs or for processing graphs. It is general and open enough and can be easily integrated with programs other than AnatOM.

8.3. DataAccess. DataAccess is a standard module, allowing AnatOM to execute various SQL statements (select, insert, update, delete), as well as SQL stored procedures and SQL functions, which are part of the 4 relational databases mentioned already in this text.

DataAccess contains general classes used for communication with the 4 relational databases which are used by AnatOM (***umls***, ***fma***, ***wordnet***, and ***anatom***). The first 3 databases represent the available EKSs. The 4th database contains: 1) the KNOWLEDGE table containing a set of very important ***species-neutral anatomical statements*** obtained through manual edits (i.e., curations) done by an anatomist (as of now these statements are valid mainly in the context of vertebrate animals); 2) all intermediate data from the computations performed by AnatOM.

8.4. Graphical user interface and logical module. The logical module manages the intermediate result graph and implements the three algorithmic procedures (DM, SMP, CMP). The graphical user interface (GUI) of the program allows controlling the execution of the procedures and presenting their results in both tabular and graphical forms to the user. The algorithmic procedures are

executed in AnatOM in separate threads. At any moment while working with the program, the user has access to the intermediate result graph and can view and edit it. The 3 threads comprising the logical module can be started from the Action menu in the AnatOM GUI.

8.5. Visualization module—GraphVisualizer.NET. GraphVisualizer.NET is the module used for visualizing the graphs AnatOM works with. It implements the graph visualization approach based on physical forces [17] due to its clarity and relative simplicity. In this approach the graph is modeled as a physical system: its vertices are viewed as ***material points*** possessing certain ***electric charges***, and its edges are viewed as ***springs*** connecting the respective material points. The algorithm is iterative and works until the physical system reaches its ***equilibrium***—the state in which the system possesses the least (zero) amount of kinetic energy.

The visualization module is called when the user selects in the AnatOM GUI one particular cross-ontology link predicted by AnatOM. When that happens, the two nodes (connected through the selected link), together with all the edges connected to these two nodes, are visualized in a graphical form by GraphVisualizer.NET.

8.6. Export module. The export module acts on the intermediate result graph and if a particular condition is met, it generates an output super-ontology in the form of an OBO file. The necessary and sufficient condition, which needs to be met in order for it to be possible to generate an output super-ontology, is that the ***intermediate result graph should contain no unacceptable cycles***. The AnatOM program provides two main ways of tracking the available cycles in the intermediate result graph: 1) counting the cycles that are present; 2) exporting the cycles into an external text file in a certain format.

The export module is called from the Result menu in the AnatOM GUI. For its execution to succeed, the abovementioned condition needs to be met. If that is not the case, some human intervention is needed from an anatomist, who needs to perform certain amount of manual editing, i.e., manual curation work on the set of predictions generated by AnatOM, until the condition is met.

9. Experiments and results. Discussion. Here the results from the pairwise mappings and mergings of the AOs of three particular categories of species are analysed (mouse /Mus musculus/, zebrafish /Danio rerio/, frog /Xenopus/). The mappings were obtained through the use of AnatOM, and then manually curated and assessed by an expert in anatomy. These expert assessments

of the ***automatic predictions generated by AnatOM*** were represented by the numbers 1, 2, 3, having the following meanings:

1—a fully accurate prediction: the two terms (from the two input ontologies) are indeed related through the semantic link predicted by the program; the type of the predicted relation (***is_ a***, ***part_ of, synonymy)*** is also accurate;

2—a partially accurate prediction: the two terms are indeed related as the program predicted but not through that type of relation which was predicted by the program; for example the program predicted a ***synonymy*** link but in reality the semantic relation which exists between the two terms is ***is_ a*** or ***part_ of***;

3—a fully inaccurate prediction: the semantic link predicted by the program is inaccurate and makes no sense whatsoever from an anatomical point of view; no other (close or distant to the predicted one) type of relation exists between the two terms either.

These three types of expert assessments are used here for analysing the results obtained through AnatOM when doing pairwise mappings and mergings of the abovementioned 3 species-specific AOs.

9.1. DM procedure results analysis. The DM procedure is syntactic in nature because it does not consider any term semantics; it just looks for pure textual, i.e., syntactic matches between the term names. Still, the links it generates are semantic links (synonymy links in particular) as are the links generated by the other two algorithmic procedures described in this text.

The DM procedure generates only synonymy link predictions. As noted, it is based solely on textual matching between term names from the two input ontologies. The manual expert assessment of the relations predicted by DM shows a very high percentage of the fully accurate predictions—more than 95% in all three couples of ontology mappings and mergings (mouse–zebrafish, mouse–frog, zebrafish–frog). This result is expectedly high given the nature of the DM procedure.

9.2. SMP procedure results analysis. The SMP procedure is semantic in nature; it consults the EKSs and based on the information contained in them generates predictions for cross-ontology links. Two of the external knowledge sources (UMLS, and FMA) are highly specialized and contain very high-quality information. The third EKS (WordNet) is a general-purpose one: it is a large lexical database of the English language; naturally the information contained in it is not that adequate from an anatomical point of view.

The SMP procedure uses heavily the information from the available EKSs and on its base it generates predictions about parent–child links (***is_ a*** and

part_ of) as well as about ***synonymy*** links. The percentage of the fully accurate predictions is very high—over 86% for both types of predictions (parent–child and synonymy) for all 3 couples of ontology mappings and mergings. This too is an expectedly high result given the nature of the SMP procedure and the big amount of adequate anatomical knowledge available in the EKSs used by AnatOM, and more specifically in UMLS and FMA.

9.3. CMP procedure results analysis. The CMP procedure is purely structural in nature (i.e., it is based solely on the structure of the intermediate result graph). But as it uses the output data, generated by DM and SMP as its input data, it could be said that indirectly CMP is both syntactic and semantic in nature.

The CMP procedure generates only synonymy links between terms of the two input ontologies. It does not use directly any of the available EKSs; as noted, it only uses them indirectly (because the output from the DM and SMP procedures is the input for the CMP procedure, and because the SMP procedure does use the EKSs).

The results from the CMP procedure are summarized in Table 1. In the first column of the table the ontology couple (on which CMP was run) is presented. In the second column the origin of the predicted links is shown. In the next 4 columns the total count of the generated predictions is shown and this count is then broken down by the values of the manually obtained expert assessments.

For the origin values (second column) of the generated predictions the following notations are used: (i) ***CMP Only*** means that these links were obtained solely through the application of the CMP procedure; they were not predicted by either DM or by SMP; (ii) ***CMP + Other*** means that these links were predicted by CMP and by one or both of the other two procedures (DM and SMP); (iii) ***CMP Any*** means that these links were predicted by CMP regardless of whether they were also predicted by some of the other procedures or not. So the counts on rows 3.n+4 from the table (in bold) are always equal to the sums of the counts from rows 3.n+3 and 3.n+2 (n=0, 1, 2) from the table.

Apparently for ***CMP + Other*** the percentage of the fully accurate predictions (score=1) is very high (which is expected)—it is higher than 93%. This means that the predictions generated by CMP have the greatest chance of being accurate when they are confirmed by at least one of the other two procedures (DM and SMP).

It can be seen that for ***CMP Any*** the percentage of the fully accurate predictions (score=1) is still satisfactory—about 15%–22%. In fact, for ***CMP Any*** the majority of the predictions are partially accurate (score=2) which is mostly

Table 1. CMP procedure results summary

Couple	Origin	Total count	Score 1	Score 2	Score 3
Mus-Danio	CMP Only	693	21 (3.03%)	517 (74.60%)	155 (22.37%)
Mus-Danio	CMP + Other	109	104 (95.41%)	4 (3.67%)	1 (0.92%)
Mus-Danio	**CMP Any**	**802**	**125 (15.59%)**	**521 (64.96%)**	**156 (19.45%)**
Mus-Xenopus	CMP Only	595	26 (4.37%)	503 (84.54%)	66 (11.09%)
Mus-Xenopus	CMP + Other	125	120 (96.00%)	4 (3.20%)	1 (0.80%)
Mus-Xenopus	**CMP Any**	**720**	**146 (20.28%)**	**507 (70.42%)**	**67 (9.30%)**
Danio-Xenopus	CMP Only	566	23 (4.06%)	427 (75.44%)	116 (20.50%)
Danio-Xenopus	CMP + Other	146	137 (93.84%)	7 (4.79%)	2 (1.37%)
Danio-Xenopus	**CMP Any**	**712**	**160 (22.47%)**	**434 (60.96%)**	**118 (16.57%)**

due to the fact that the input ontologies used differ in their ***depths*** and ***granularities***. A good result is observed in the fully inaccurate predictions (score=3). Their percentage is not high (it is less than 20%).

For ***CMP Only*** there is a certain amount of fully accurate predictions (score=1)—about 3%–4%, which can still be taken as a good result. This result means that the CMP procedure which is purely algorithmic and does not rely directly on the EKSs is still able to discover fully accurate semantic links which were missed by both DM and SMP. The percentage of the fully inaccurate predictions (score=3) is again low enough. It does not exceed 22%. Here it is again the case that the majority of the predictions are partially accurate (score=2) which, as was noted already, is due to the differences in the ***depths*** and ***granularities*** of the input ontologies used.

The ***granularity*** of an ontology denotes the average size (in semantic sense) of the step between a given parent term and a given child term. Suppose we have a sample ontology A which declares

finger *part_ of* forelimb (1)

and another sample ontology B which declares

finger *part_ of* hand (2)

hand *part_ of* forelimb (3)

It can be seen that the statement (1), which ontology A defines in a single step, is defined in ontology B in two steps—(2) and (3). That is why it is said that ontology A is ***coarser grained*** than B, and ontology B is ***finer grained*** than A.

The ***depth*** (or the ***height***) of a given ontology is defined by the difference of the level numbers of its leaf terms and its root terms. The most general terms (roots) in an ontology are those terms which do not possess any parents; the most specialized terms (leaves) in an ontology are those which do not possess

any children. This depth concept is analogical to the depth concept from graph theory where it is typically used for trees. Here it is used for DAGs (for the DAGs representing the ontologies in question).

9.4. Analysis of the merging process and the generated super-ontologies. A few facts and observations are presented here related to the mapping and merging of the AOs of the mouse and the zebrafish while working on them with the AnatOM program, and more precisely while running the DM and the SMP procedures on them.

The mouse AO contains information only about ***the anatomy of the adult organism***; it contains no terms describing the prenatal or postnatal development of the organism. The zebrafish AO contains both terms describing ***the anatomy of the adult organism*** and terms describing ***the development of the organism***. The names of the terms from the adult mouse ontology start with the string "MA". The names of the terms from the zebrafish ontology start with the string "ZFA" (when they describe anatomical terms of the adult ogranism) or with the string "ZFS" (when they describe developmental phases). In the two ontologies there are 2982 MA terms, 2712 ZFA terms, and 46 ZFS terms in total.

In this study, the count of the ***original terms*** (terms from the input ontologies) from which a given ***generalized term*** (term from the super-ontology) originates is called ***degree of the generalized term***. It is obvious that the ZFS terms don't have corresponding terms in the mouse AO. So while merging the two ontologies they generate generalized terms of degree 1. Of greatest interest are the generalized terms of degree $\geq \mathbf{2}$. It turns out that their count is 255, which means that about 10% of the mouse and zebrafish terms have corresponding terms in the other ontology. A more detailed analysis reveals that the super-ontology contains 5470 generalized terms in total: 1 term has a degree of 5, 12 terms have a degree of 3, 242 terms have a degree of 2, and 5215 terms have a degree of 1.

These results are mentioned here for illustrative purposes mostly. They were obtained after applying the DM and SMP procedures only, and they were obtained only from the automatic processing which AnatOM does. If the CMP procedure is run too, then some more work is needed from an anatomist's side, for manually editing AnatOM's auto-predicted links, before one can proceed with the generation of an output super-ontology.

All in all, the main observation which can be made while mapping and merging the AOs of mouse and zebrafish through the use of AnatOM, is that about 10% of the two input ontologies can actually be mapped onto each other through the use of synonymy links. The 10% overlap (match) between the terms from the two AOs leads to the generation of generalized terms with a degree of 2

or higher.

9.5. Discussion of certain problems encountered while doing this study. In this study most of the experiments have been performed with the AOs of mouse and zebrafish. Also, some experiments have been performed with the AO of frog looking for its links to the other two AOs.

From a biological point of view, some topics need to be discussed in an attempt to explain the results from the merging of AOs by using AnatOM.

First, it has to be stressed that the input AOs are not homogenous with respect to their contents. Some include developmental stages and some are only focused on the anatomy of the adult organism. Here, certain conflicts can be expected simply because many embryonic terms cannot be matched directly to terms pertaining to an adult organism. An example is the relation of the term "coelom" to the term "pericardium". In the vertebrates analyzed in this study the two terms can be matched with the relation *"pericardium" is_a "coelom"*, but also with the relation *"pericardium" part_of "coelom"*.

Second, an important contradiction should be noted which originates from the conflict between different descriptions of potentially identical structures in Latin and English. In this study there has been some struggle with the establishing of adequate relations between loosely formulated terms such as *"cardiac muscle tissue"* and strictly formulated terms such as *"myocardium"*. One can use both terms as synonyms, but strictly speaking they may be related in other ways too (e.g., through a *part_of* relation). This mainly depends on the definition meant by the authors of the ontology and the interpretation of the analyzer using it. This problem can be described as the problem of the ***semantic load of terms***. Particularly difficult to handle are those terms that are ***semantically over-loaded*** or ***semantically under-loaded***. Such are, e.g., the terms "portion of organism substance", "portion of tissue", "Xenopus anatomical entity", "acellular anatomical structure", "anatomical set", "multi-tissue structure", "anatomical space", "surface structure".

Third, there are also some problems with the definitions of certain terms within the input ontologies themselves. A good example is the statement *"bulbus arteriosus" part_of "heart"*. Actually, the formulation of the term "bulbus arteriosus" leads to the conclusion that this structure cannot be part of the heart but has to be part of the arterial system (and therefore possess smooth rather then cardiac musculature). Whenever such problematic definitions have not led to formations of cycles in the ***intermediate result graph***, they have been accepted in this study in an attempt to keep the respective input AOs as unaltered as possible.

Finally, the following problem has to be noted too: in some cases ***different relations can exist between the same terms depending on the anatomical context***. The mouse AO defines the statement *"maxilla" part_ of "upper jaw"*. On the other hand, AnatOM discovers that the relation *"maxilla" synonym "upper jaw"* also makes sense. The program finds that based on the information available in WordNet. But these two statements lead to the existence of some cycles in the intermediate result graph while trying to merge the AOs of mouse and zebrafish. This problem comes from the fact that the 1st statement is valid for all *mammals* while the 2nd one is valid for all other *jawed vertebrates*. This problem also reveals certain limitations of the previously mentioned KNOWLEDGE table which was designed and populated as part of doing this study, and which is heavily used by AnatOM. In future enhancements or extensions of this study, it would make sense for this table to become more specialized, and even to turn into a set of tables, each of which pertains to a particular specialized anatomical context, i.e., to a particular category of species (mammals, fish, amphibia, reptiles, etc.).

10. Conclusions. Here the main conclusions are presented about the contributions of this study as seen by its author. These contributions are broken down into scientific and applied ones, though the boundary between the two types is sometimes not fully clear if not vague. At the end several directions for potential future work, related to this study, are presented.

10.1. Scientific contributions.

1) In this study, a detailed review was done of the current state of the problem domain. Several existing general-purpose systems for mapping and merging of ontologies were described. Making a direct comparison between these systems and the AnatOM program developed as part of this study is practically impossible for two reasons: (i) most of the popular existing systems do not support the OBO language—the main de-facto standard language for describing biomedical ontologies and AOs in particular; (ii) there is a lack of clear and objective criteria which would allow direct comparison of the results obtained from the AnatOM program on one side, and the results that can be obtained from existing systems on the other, as well as a lack of ways for automating such a comparison; this is because the goal there would be to perform a ***semantic comparison*** between these results so the only possible approach would be non-automatic and would involve manual work to be done by an expert in anatomy.

2) The 3 algorithmic procedures DM, SMP, and CMP were formalized and presented in terms of a consistent theoretical framework. Each of these 3 procedures complements the previous ones in the process of discovering semantic

cross-ontology links between two given AOs. The ***DM procedure*** is widely known and is used almost always when the goal is to integrate several distinct ontologies into a common ontology. It gives the basis or starting point for finding cross-ontology mappings. Some general ideas about the ***SMP procedure*** have been adopted in this study from certain scientific publications [37, 41], but in this study these ideas were supplemented, formalized, and adapted to the context of mapping AOs through the use of EKSs. The ***CMP procedure*** is an original one and has been developed as part of this study. It allows the discovery of reasonable (from anatomical point of view) cross-ontology links, which are missed both by DM and by SMP.

3) An important ***necessary and sufficient condition*** was proved which makes it possible to get from model #2 to model #3 (these models were mentioned at the beginning of this text). This necessary and sufficient condition solves theoretically the question about when it is possible to generate a valid, common, output, species-neutral anatomical ontology (super-ontology) from two given, input, species-specific anatomical ontologies. The proof (omitted in this paper) of this condition gives an explicit procedure showing how the generation of the super-ontology can be done, once the mappings between the two input ontologies have already been established (by the DM, SMP, and CMP procedures).

4) Even though the concrete subject of this study is the mapping and merging of AOs in particular, the method and the algorithm suggested here are ***general enough*** and so they could be applied to other domains, and not just to the anatomical domain. The only condition for this to be possible is to have a set of EKSs which are semantically close enough to that other domain of study, to which the method presented here would be applied.

10.2. Applied contributions.

1) The program AnatOM was developed as a complete, integrated solution for semi-automatic mediation, i.e., integration of AOs.

2) As part of AnatOM several modules were developed which are valuable from a practical point of view even when considered outside of the context of AnatOM. Such are for example: the graph module, the graph visualization module, the OBO parser module, the super-ontology export module.

3) The algorithmic procedures DM and SMP, formally described here, were implemented as part of AnatOM. The original CMP procedure suggested in this study was also implemented and built into the AnatOM program. As noted already, these three procedures comprise the logical module of the AnatOM program.

4) Some popular and some not so popular algorithms working on graphs

were implemented such as Tarjan's algorithm [19], Johnson's algorithm [18], as well as the algorithm for visualizing graphs based on balancing of physical forces [17].

10.3. Directions for future development. Many directions for future development of this study could be pointed out. Here the most significant of them are listed.

1) Future work can be done in improving the ***sensitivity*** and the ***specificity*** of the CMP procedure. The difficulty here comes from the fact that the different input AOs have different depths and granularities. Achieving an even smaller improvement in that direction would lead to serious improvement in the results obtained through the CMP procedure (with respect to their adequacy from anatomical point of view).

2) Some alternative, improved schemes (either probabilistic ones or not) could be sought for scoring the automatically predicted cross-ontology links. Naturally it can be assumed that one such scheme is better than another, if the scores given by the former are closer than the scores given by the latter, to the actual scores which an expert in anatomy would manually assign to these automatically predicted links.

3) In the future, a procedure which builds upon CMP or is an alternative to CMP could be developed. Such a procedure could consider different types of patterns of connectivity in the intermediate result graph, obtained from the execution of the DM and SMP procedures. This is a rather difficult task as it requires quite some creativity and ingenuity to come up with a procedure which would generate links that are more adequate (from an anatomical standpoint) than the links generated by the CMP procedure as it stands now.

4) More serious tests with wider coverage could be performed while merging three or more AOs. So far serious tests have been performed with three couples of organisms and their respective AOs (as mentioned already). As to merging 3 or more AOs into a single super-ontology only some basic tests have been performed. There is enough room for future work in that direction with respect to performing additional tests, with respect to manual curation of the results obtained from them, and with respect to improving the existing algorithmic procedures (mostly the CMP procedure) based on these results.

5) Some work can be done on developing a procedure for automatic elimination of unacceptable cycles present in the intermediate result graph.

6) The graph visualization algorithm integrated in AnatOM can be improved to a certain extent. Currently in some cases it presents a small problem where some edges are visualized too close to each other (i.e., the angles between the

segments representing these edges are too small). That would better be avoided because it makes the visual perception of these edges difficult. So some correcting procedure could be developed here which would run after the main visualization procedure and which would try to increase those angles which are too small.

7) The AnatOM program could be modified to support ontology languages other than OBO (e.g., OWL or RDFS). This modification would be relatively small and easy. Currently though, practically all publicly available AOs are represented through the use of the OBO language which is the *de facto* standard for representing biological and biomedical ontologies.

8) By using AnatOM and the methods suggested here, some more concrete, more practical, and much larger-in-scale problems could be attacked in the future. An attempt could be made to merge more AOs into a much larger-in-scale common super-ontology, which would integrate the anatomical knowledge for a much wider set of organisms. This super-ontology could then be used as an underlying central model for developing algorithms and tools for cross-species text searching and cross-species text mining in anatomical texts available in various electronic libraries or on the Internet in the general.

REFERENCES

[1] Gruber T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, **5** (1993), No 2, 199–220.

[2] Gruber T. R. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, **43 (**1995), No 4–5, 907–928.

[3] De Bruijn J., M. Ehrig, C. Feier, F. Martín-Recuerda, F. Scharffe, M. Weiten. Ontology mediation, merging, and aligning. Semantic Web Technologies (Eds J. Davies, R. Studer, P. Warren), John Wiley and Sons, 2006, 95–113.

[4] Grenon P., B. Smith, L. Goldberg. Biodynamic ontology: applying BFO in the biomedical domain. Ontologies in Medicine (Ed. P. M. Pisanelli), *Studies in Health Technology and Informatics*, **102** (2004), Amsterdam, 20–38.

[5] Smith B., M. Ashburner, C. Rosse, J. Bard, W. Bug et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology,* **25** (2007), 1251–1255.

[6] Day-Richter J. The OBO Flat File Format Specification. Version 1.2, 2006. http://www.geneontology.org/GO.format.obo-1_2.shtml, 12 February 2014

[7] Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, **32** (2004). doi: 10.1093/nar/gkh061

[8] Official web site of UMLS by the U.S. National Library of Medicine (NLM). http://www.nlm.nih.gov/research/umls/ 12 February 2014.

[9] Rosse C., J. L. Mejino Jr. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of Biomedical Informatics,* **36** (2003), 478–500.

[10] Official web site of the FMA by the University of Washington. http://sig.biostr.washington.edu/projects/fm/AboutFM.html, 12 February 2014

[11] Fellbaum C. WordNet: An electronic lexical database. MIT Press, Cambridge, MA, 1998.

[12] Miller G. A. WordNet: a lexical database for English. *Communications of the ACM*, **38** (1995), No. 11, 39–41.

[13] Official web site of the WordNet project by the Princeton University. http://wordnet.princeton.edu/, 12 February 2014

[14] Official web site of the MySQL RDBMS. http://www.mysql.com/, 12 February 2014

[15] Zlatareva N., M. Nisheva. Alignment of heterogeneous ontologies: a practical approach to testing for similarities and discrepancies. In: Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference (Eds D. Wilson, H. Chad Lane), Coconut Grove, Florida, US, May 15–17, 2008, ISBN 978-1-57735-365-2, AAAI Press, Menlo Park, California, 2008, 365–370.

[16] Nisheva–Pavlova M. Mapping and merging domain ontologies in digital library systems. In: Proceedings of the 5th International Conference on Information Systems and Grid Technologies, Sofia, May 27-28, 2011, ISSN 1314-4855, St. Kliment Ohridski University Press, Sofia, 2011, 107–113.

[17] Tollis I. G., G. Di Battista, P. Eades, R. Tamassia. Graph drawing: algorithms for the visualization of graphs. Prentice Hall, 1999.

[18] Johnson D. B. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, **4** (1975), No 1, 77–84.

[19] Tarjan R. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, **1** (1972), No 2, 146–160.

[20] Oberle D. Semantic management of middleware. Springer, New York, 2006.

[21] Happel H.-J., S. Seedorf. Applications of ontologies in software engineering. In: Proceedings of the 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006), 5th International Semantic Web Conference (ISWC 2006), Athens, GA, USA, November 2006.

[22] Minsky M. L. A framework for representing knowledge. MIT-AI Laboratory Memo 306, June, 1974. Reprinted in The Psychology of Computer Vision (Ed. P. Winston), McGraw-Hill, 1975. Shorter versions in Mind Design (Ed. J. Haugeland), MIT Press, 1981 and in Cognitive Science (Eds A. Collins, E. E. Smith), Morgan-Kaufmann, 1992.

[23] Minsky M. L. Frame-system theory. Theoretical issues in natural language processing (Eds R. C. Schank, B. L. Nash-Webber), Preprints of a conference at Massachusetts Institute of Technology, Cambridge, MA, 1975.

[24] The description logic handbook (Eds F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider), Cambridge University Press, 2002.

[25] McBride B. The Resource Description Framework (RDF) and its vocabulary description language RDFS. The Handbook on Ontologies in Information Systems (Eds S. Staab, R. Studer), Springer Verlag, 2003.

[26] Antoniou G., F. van Harmelen.Web Ontology Language: OWL, Handbook on ontologies (Eds S. Staab, R. Studer), Springer, Berlin, 2004, 67–92.

[27] Sowa J. F. Semantic networks. Encyclopedia of Artificial Intelligence (Ed. S. C. Shapiro), Wiley, New York, 1987, Revised and extended for the 2nd edition, 1992.

[28] Richens R. H. Preprogramming for mechanical translation. *Mechanical Translation*, **3** (1956), No 1, 20–25.

[29] Quillian M. R. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, **12** (1967), No 5, 410–430.

[30] Dou D., D. McDermott, P. Qi. Ontology translation by ontology merging and automated reasoning. In: Proceedings of EKAW2002 Workshop on Ontologies for Multi-Agent Systems, Siguenza, Spain, 2002, 3–18.

[31] Uberon: towards a comprehensive multi-species anatomy ontology. (Eds M. Haendel, G. V. Gkoutos, S. Lewis, C. Mungall), Nature Proceedings, Buffalo, NY, 2009, doi:10.1038/npre.2009.3592.1

[32] Ashburner M. et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, **25** (2000), 25–29.

[33] Rector A. L., W. A. Nowlan, S. Kay. Unifying medical information using an architecture based on descriptions. In: Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care R. (Ed. A. Miller), IEEE Computer Society Press, Los Alamitos, California, 1990, 190–194.

[34] Rector A. L., W. A. Nowlan, S. Kay. Foundations for an electronic medical record. *Methods Inf Med*, **30** (1991), No 3, 179–86.

[35] Nowlan W. A., A. L. Rector, S. Kay, B. Horan, A. Wilson. A patient care workstation based on a user centred design and a formal theory of medical terminology: PEN&PAD and the SMK formalism. In: Proceedings of the 15th Annual Symposiumon Computer Applications in Medical Care (Ed. P. D. Clayton), McGraw- Hill Inc., Washington DC, 1991, 855–857.

[36] Rector A. L., J. E. Rogers, P. E. Zanstra, E. van der Haring. OpenGALEN: open source medical terminology and tools. In: AMIA Annual Symposium Proceedings, National Center for Biotechnology Information, U. S. National Library of Medicine, Bethesda MD, USA, 2003, 982–983.

[37] Aleksovski Z., W. ten Kate, F. van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In: Proceedings of the International Workshop on Ontology Matching, 5 November, 2006, Athens, Georgia, USA, 2006, 13–24.

[38] Dou D., D. McDermott, P. Qi. Ontology translation by ontology merging and automated reasoning. In: Proceedings of the EKAW2002, Workshop on Ontologies for Multi-Agent Systems, Spain, 2002, 3–18.

[39] Ashburner M., C. J. Mungall, S. E. Lewis. Ontologies for biologists: a community model for the annotation of genomic data. *Cold Spring Harbor Symposia on Quantitative Biology*, **68 (**2003), 227–236.

[40] Mungall C. J., C. Torniai, G. V. Gkoutos, S. E. Lewis, M. A. Haendel. Uberon, an integrative multi-species anatomy ontology. *Genome Biology,* **13** (2012), R5. `http://genomebiology.com/2012/13/1/R5`, 12 February 2014

[41] Van Ophuizen E. A. A., J. Leunissen. An evaluation of the performance of three semantic background knowledge sources in comparative anatomy. *Journal of Integrative Bioinformatics*, **7** (2010), 124–130.

[42] Visser P. R. S., Z. Cui. On accepting heterogeneous ontologies in distributed architectures. In: Proceedings of the ECAI98 Workshop on Applications of Ontologies and Problem-Solving Methods, Brighton, UK, 1998.

[43] Maedche A., B. Motik, N. Silva, R. Volz. MAFRA—A mapping framework for distributed ontologies. In: Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW-2002, Madrid, Spain, 2002.

[44] Omelayenko B. RDFT: A mapping meta-ontology for business integration. In: Proceedings of the Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002) at the 15-th European Conference on Artificial Intelligence, Lyon, France, 2002, 76–83.

[45] Noy N. F., M. A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In: Proceedings of 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, Texas, USA, 2000, 450–455.

[46] Noy N. F., M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, **59 (**2003), No 6, 983–1024.

[47] Noy N. F., M. A. Musen. Anchor-PROMPT: Using non-local context for semantic matching. In: Proceedings of the Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA, USA, 2001.

[48] Ehrig M., S. Staab. QOM—quick ontology mapping. In: Proceedings of the 3rd International Semantic Web Conference (ISWC2004) (Eds F. van Harmelen, S. McIlraith, D. Plexousakis), LNCS, Springer, Hiroshima, Japan, 2004, 683–696.

[49] Ehrig M., Y. Sure. Ontology mapping–an integrated approach. In: Proceedings of the 1st European Semantic Web Symposium ESWS'2004**,** Heraklion, Greece, Lecture Notes in Computer Science, Vol. **3053,** Springer Verlag, 2004, 76–91.

Peter Petrov
Faculty of Mathematics and Informatics
"St. Kl. Ohridski" University of Sofia
5, J. Bourchier Blvd, P.O. Box 48
1164 Sofia, Bulgaria
e-mail: p.a.petrov@gmail.com

Received November 11, 2013
Final Accepted December 12, 2013