

Investigation of Ontologies in Software-Engineering-(Meta-)Research

Dmitrii Seletkov

Institute for Program Structures and Data Organization (IPD)

Advisor: Dipl.-Inform. Angelika Kaplan

Abstract[Deutsch]. Wissensgewinnung ist eine essentielle Eigenschaft des menschlichen Wesens, die uns von den anderen Kreaturen unterscheidet. Ein Instrument für Einordnung und Übertragung des Wissens, das nicht nur von Menschen, sondern auch von Computern verstanden werden kann, sind Ontologien. In dieser Proseminar-Arbeit werden die Ontologien in Software-Engineering-Meta-Forschung untersucht und analysiert, um die fundamentalen Kenntnisse in diesem wissenschaftlichen Bereich zu erwerben. Darüber hinaus haben Ontologien den Bezug auf semantische Suche, die in fast allen modernen Suchmaschinen wie Google, Bing, Yandex etc. verwendet wird. Das Ziel dieser Proseminar-Arbeit ist jedoch die Untersuchung von wissenschaftlichen Suchmaschinen mit dem Fokus auf die Anwendung der Ontologien in ihnen und wie sie zur Software-Engineering-Meta-Forschung beitragen könnten.

Abstract. Retrieving Knowledge is an essential part of human being that distinguishes us from other creatures. One of the instrument for organizing and transferring knowledge that is understandable not only to humans, but to computers as well, is an ontology. This seminar work aims at investigating ontologies in Software-Engineering-Meta-Research and trying to analyze the retrieved information, in order to grasp an understanding and awareness of fundamentals in this scientific field. Moreover, ontologies are closely related to the Semantic search that is used by every contemporary search engine such as Google, Bing, Yandex etc. The purpose of this seminar was, however, about the investigation of scientific search engines with focus on applying ontologies in them and how they could contribute to the Software-Engineering-Meta-Research.

1 Introduction

The term ontology originates from philosophy. Its etymology gives us the direct definition i.e. a science about being. More broadly, ontology is «a branch of metaphysics concerned with the nature and relations of being» [21]. But how does it relate to Computer Science? As far as we know, Information and Computer science are very precise and cannot admit a vague descriptions, as it could be met in philosophy. Thus, ontology in Computer Science is «a list of concepts and categories in a subject area that shows the relationships between them» [20]. This means simply the supplement of a common language, in order to facilitate general understanding of knowledge in machine way.

The term Meta-research is «the use of scientific methodology to study science itself» [18]. This means that researchers are trying to understand the research itself i.e. how other researches should be conducted, what practices in the research are the most effective and in what research fields they should be used.

These two terms and especially their combination in Software Engineering, namely how ontologies can contribute to the Meta-research and how Meta-research changes the ontology-based approaches in Software Engineering are part of the rather young scientific field that arouses a strong interest in it. Moreover, it is not a secret that ontology-based systems are actively used in the contemporary search mechanism. Therefore, it attracts an attention, how exactly ontologies are used not only in the traditional search, but also in the scientific one and how these mechanisms can contribute to the Software-Engineering-Meta-Research.

The remainder of this seminar work is structured as follows. Section 2 sets the necessary foundations to the terms ontologies and Meta-research; Section 3 describes the main use cases of utilizing Ontologies in Software-Engineering-Meta-Research; Section 4 is about ontology-based systems in scientific search. Finally, in section 5 are made the conclusions regarding to the whole seminar work.

2 Foundations

Before we go to the main sections of this seminar work, there should be introduced the main terms and concepts that will be used during the whole work and are necessary for understanding it.

2.1 Ontology in Computer Science

Knowledge is a fundamental concept of our world. There are some requirements that Knowledge should fulfill in order to be represented in Computer Science i.e. Knowledge must be adequate, representative in machine form and, finally, be able to elaborate new Knowledge. Therefore, scientists have been trying to create a method of representation of Knowledge, since the inventing of the first computers. Here it is very important to establish a difference between information and knowledge. The contemporary computers can represent informa-

tion for a very long time, but we still need people to interpret it. Thus, we should be able not only represent the information that we already have, but also give this information the semantics, namely to give the terms the necessary meaning. That is why we need ontologies. An ontology in Computer Science is «an explicit specification of a conceptualization» [9], where a conceptualization is an abstract model of some knowledge domain and an explicit specification that represents classes/concepts/terms, their properties and attributes, exemplars of classes. In sum, an ontology sets the general understanding of a knowledge domain and, moreover, the common language for representing of this knowledge.

2.1.1 Ontology languages

An ontology language must: have a clear syntax (computer readable), have formal semantic (unique meaning), be effectively computer analyzable and have sufficient descriptive power.¹

Here it should be noticed that there already exist some approaches that are known to all informatics students. They are ER-models and UML-diagrams. Indeed, they can be used for representative explanation of domain knowledge. Nevertheless, they are still complex and badly readable for computers. The first ontology languages that we look at will be the Resource Description Framework (RDF) and RDF with schema vocabulary (RDFS), which was recommended by W3C [29].

The idea of RDF is representing of everything though triples *<subject, predicate, object>*, where subject, predicate and object are resources; every resource is identified by a unique reference (URI) and every RDF-statement consists of resources. The syntax that is used for machine interpretation is based on XML. In order to make it clearer, let us represent the actual seminar work and its relations using RDF. The representing in triples, graph and XML-notation see in Figure 1.

Although RDS allows to describe the predicates (properties) of resources, it is still not permitted to express between predicate and classes of resources. This problem is solved in RDFS. Hence, the expressions like *rdfs:subClassOf* (subset relation), *rdfs:range*, *rdf:type* etc. become possible. For example, *<isWrittenBy, type, Property>* means to understand that *isWrittenBy* is a property or *<isWrittenBy, range, Person>* - those, who writes is a person.

Despite the opportunities that RDFS gives us, it still has the following problem: it is not expressive enough i.e. there is no local restrictions on a knowledge domain e.g. it is not possible to make a predicate *hasChild* differ whether it is applied on a class *Person* or a class *Animal* (no semantic difference); there is no transitive, symmetric or reverse predicate e.g. it is not possible to express with a predicate *touches* if a thing *A* touches another thing *B*, then *B* also touches *A*; and many other restrictions.

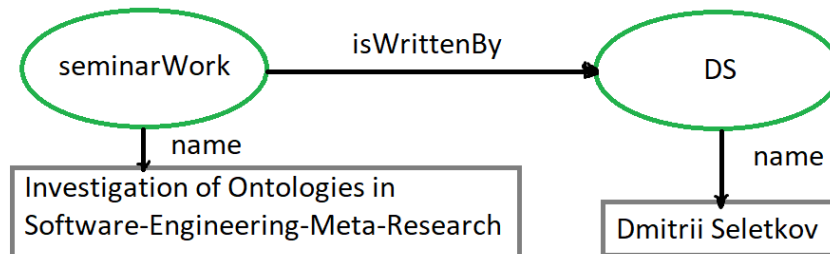
All these restrictions made scientist to elaborate a new family of knowledge representation languages that is called Description Logic (DL). It consists of different dialects and differs from other approaches with formal semantics (more expressive and usually solvable) and instruments of logical analysis. As an example of one of the simplest DL dialects we will look

¹The following subsection is fully based on [15], unless it is stated otherwise The literal citations will be quoted and cited.

```

<seminarWork, name, Investigation of Ontologies in Software-Engineering-Meta-Research>
<seminarWork, isWrittenBy, DS>
<DS, name, Dmitrii Seletkov>

```



```

<rdf:Description rdf:about="#seminarWork">
  <csclub:isWrittenBy rdf:resource="#DS"/>
  <csclub:name>Investigation of Ontologies in Software-Engineering-Meta-Research</csclub:name>
</rdf:Description>
<rdf:Description rdf:about="#DS">
  <csclub:name>Dmitrii Seletkov</csclub:name>
</rdf:Description>

```

Figure 1: RDF example

at EL DL that can be solved in polynomial time and could help us to understand the main concepts and terms of contemporary ontology language standards such as Web Ontology Language.

The main difference between EL DL and RDFS is a clear separation between Terminology (to define the concepts and relations in an ontology schema) and Assertions (define data or concrete individuals that pertain to the concepts in the schema). The architecture of a EL DL can be observed in Figure 2. The main syntax can be described through:

- Names of *Concepts* (also classes) such as *Person*, *Female*, ...
- Names of *roles* (also relations) such as *isWrittenBy*, *hasChild*, ...

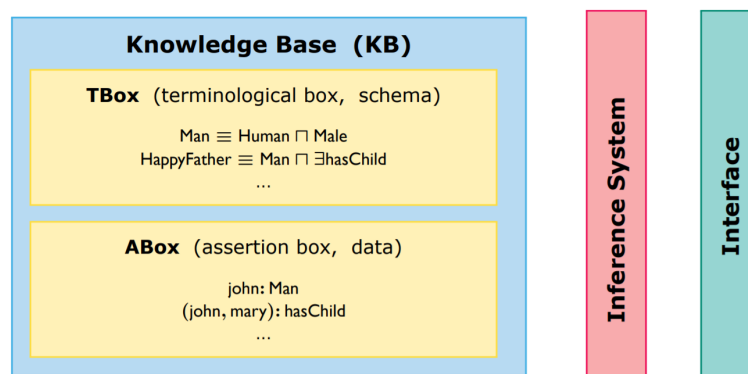


Figure 2: Architecture of Description Logic [15, lecture 8, slide 1]

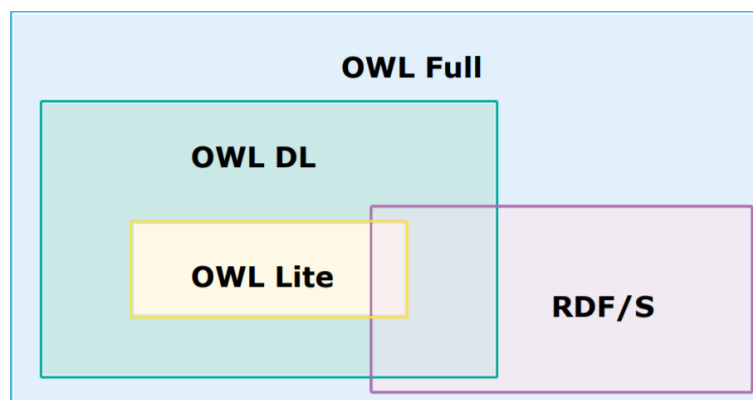


Figure 3: OWL dialects [15, lecture 8, slide 40]

- Concept *Thing* \top , Everything (every concept) is a *Thing*
- Conjunction (logical and) \sqcap ; Existence quantor \exists
- Definition or equivalence \equiv ; Primitive definition or subset \sqsubseteq

Let us illustrate the above-named semantics with trivial examples. For example, $Person \sqcap Female$ is an intersection of classes *Person* and *Female* i.e. *woman*; $Person \sqcap \exists hasChild.\top$ is a *Person*, who has a child i.e. a *parent*; $Person \sqcap \exists hasChild.\top \sqcap \exists hasChild.\top$ is simply a *grandparent*. The last but not least, let us introduce examples with definition and subsets: we can define a *father* as $Father \equiv Person \sqcap Male \sqcap \exists hasChild.\top$ the statement $Father \sqsubseteq Person$ would be also valid.

It is noteworthy, that in comparison with a boolean logic there is not negation (\neg), disjunction (\vee) and all quantor (\forall). Nevertheless, it allows us to express many ideas though writing concepts and connecting them via roles and everything with a rather simple syntax e.g. without inserting variables x, y, z , etc., as it is required in the first-order predicate logic.

Now, it is time to introduce the latest standard in ontology languages for Semantic Web (more in subsection 4.1), namely the Ontology Web Language (OWL) [4]. By the establishing of a new standard there were set two main purposes. Firstly, the integrating of DL concepts and, secondly, the retainment of the already existing ontologies i.e. full compatibility with RDFS, otherwise the existing ontologies had to be discarded. As a family of DL languages OWL has also diverse dialects e.g. OWL Full, OWL DL, OWL, etc. (see Figure 3) Lite, that serve as appropriate tools for different types of tasks. In sum, the syntax of OWL is based on RDF, XML (complement RDF and does not follow all rules of RDF) and DL theory (elaborated logical foundations), has an abstract syntax and graphical representation in UML.

In this seminar work there is no intention to observe all dialects and all features in OWL, thus, we will consider only some features of OWL EL (a dialect of OWL based on DL EL). Hence, the main aspects of OWL can be summed up in the following itemization (instead of XML notation the abstract syntax is used):

- Classes are concepts and also help to express the definitions (\sqsubseteq and \equiv) with corresponding reserved words *Class(name1 partial name2)* and *Class(name1 complete name2)*
- There are existence (\exists) and **all** (\forall) quantors that are expressed by *restriction(prop someValuesFrom(class))* and *restriction(prop allValuesFrom(class))*
- It is possible to **disjoint** classes (*DisjointClasses(class1...classN)*)
- Both conjunction and **disjunction** are permitted (*unionOf(...)* and *intersectionOf(...)*)
- **Negation** of classes is in (*complementOf(class)*)

As we can see, the OWL is more powerful in comparison with DL EL or RDFS. Therefore, one statement can be expressed with different variants. For example, $A \sqsubseteq B$ can be expressed by three absolutely equivalent variants: *Class(A partial B)*, *SubClassOf(A B)*, or *DisjointClasses(A complementOf(B))*. It can be a good so-called syntactic sugar, but also cause many problems as well.

2.1.2 Expediency and reasons for using of Ontologies

After the consideration of power of DL languages and some standards such as RDFS and OWL, we can now understand, what opportunities and expressiveness we have, namely we are able to: determine concepts(classes) of knowledge domain, organize them in hierarchy (subclass, superclass), determine attributes, relations and restrictions of classes and, finally, determine instances of classes and assign the values to the corresponding attributes and relations. This all allows us to reach a common understanding of terms that can be used for interchanging of pure information independent on languages, definitions and other syntactic barriers. Moreover, ontology languages make possible to reuse the obtained knowledge and to standardize them.

Then, should we develop an ontology? Of course, if it is desired to fulfill all aspects which were mentioned above, then it is strongly recommended and maybe even urgent. For instance, a structured and standardized collection of medical terms SNOMED CT ² that includes definitions, descriptions, synonyms etc. for medical diseases is so huge that there does not exist any person who could keep in mind even a small part of knowledge of this ontology. Once the ontology is established, it is possible simply to insert new entries (e.g. new diseases) without knowing any technical details of ontologies and the ontology automatically facilitates the classification of the new entries by integrating them properly in existing hierarchy based on determined concepts and relations. Besides, it allows any doctor to find the medical that he/she is looking for and see not only the whole information about this term, but also its place in the hierarchy. Sometimes, it is superfluous to develop an ontology e.g. if it needed to organize less than 100 terms or the similar tasks. But sometimes it is not possible to express

²<http://www.snomed.org/>

everything that is desired e.g. «Despite consisting of a powerful family of semantically well-founded formalisms’, DLs are not endowed with the basic mathematical resources necessary for representing neural networks, for instance the arithmetical operations.» [7].

Therefore, despite the power of ontologies, it is always very important to consider all advantages and drawbacks, namely the expediency, that the using of ontology gives us.

2.2 Meta-research

2.2.1 Motivation

A statement «Science is one of the main drivers of human progress» is indisputable and proved by the growing number of published papers and publishing authors [11]. Naturally people face the problems such as data sharing, replications of experiments, its ownership and many others. Moreover, the research practices suffer from lack of systematization and inefficiency. For example, according to [17] 85% of resources in biomedical research are wasted because of above-called reasons. Therefore, there exists the urgent need of Meta-research, which aims to include the evaluation of diverse researches with succeeding suggestions of improvement for research practices.

2.2.2 Areas of Meta-research

Discussing the existing problems about contemporary research a logical question arises, whether it is possible to provide a map of ongoing efforts in the field Meta-research and connect the multiple already made but still fragmented attempts across science. These goals were set by J. Ioannidis et al.’ work «Meta-research: Evaluation and Improvement of Research Methods and practices» [12].

His suggestions can be summed up in the following categorization. There are five major areas of interests in meta-research: methods, reporting, reproducibility, evaluation and incentives. Each of them was not only explicitly defined and illustrated with examples, but also for each area were found the existing initiatives. These mentioned features of the work can be summarized as following categorizations:

1. **Methods:** practices for performing research (e.g. study design, methods, statistics). With **specific interests** in biases and questionable practices in conducting research, methods to reduce such biases, meta-analysis. Existing **Initiatives** such as Cochrane Collaboration for systematic reviews of health care or Campbell Collaboration for the same ones but in social science.
2. **Reporting:** publications of standards and study registrations (e.g. study registration, information to patients, public and policy-makers). With **specific interests** in biases and questionable practices in reporting, explaining, disseminating and popularizing research. Existing **Initiatives** such as ClinicalTrials.gov for clinical trials registrations or EQUATOR network for reporting standards for research.

3. **Reproducibility:** methods for verifying research (e.g. sharing data and methods, replicability). With **specific interests** in overcoming of obstacles to sharing data, methods and replications. Existing **Initiatives** such as YODA for sharing data in clinical research or BITSS for transparency in social science.
4. **Evaluation:** improvements for scientific quality (e.g. pre- and post-publication peer reviews, research funding criteria). With **specific interests** in effectiveness, costs, and benefits of old and new approaches to peer review. Existing **Initiatives** such as Peer Review Congress for evidence on peer review or ArXiv for preprinting articles.
5. **Incentives:** rewards and penalties for research (e.g. promotion criteria, penalties in research evaluation). With **specific interests** in accuracy, effectiveness and benefits of old and new approaches to ranking and evaluating the performance. Existing **Initiatives** such as REWARD for reducing waste and rewarding diligence in research or AAAS for science policy.

Certainly, it has to be said that this is an «nonexhaustive list». Also, it is worth of remarking that neither in this paper nor in the referenced and citing papers the initiatives in SE-Meta-Research are considered. Therefore, strictly based on definitions and comparing of initiatives in other fields, the mentioned in this work approaches in Software Engineering will be categorized, if it is possible and if not, the limitations of such classification will be discussed.

3 Ontologies in Software-Engineering-Meta-Research

This section aims to show different types of ontologies, which are used by scientists for Meta-research in Software Engineering, and also to compare them.

Firstly, in order to show all this diversity, the Biolchini et al. [1] classification of empirical studies in SE has been chosen. It shows «the concepts of Primary and Secondary Studies on Software Engineering at a high level» [8]. According to this classification any experimental study comprises two types of investigations: Primary and Secondary [1]. Primary studies are used for evaluation of the researcher's hypothesis and represented above all by Controlled Experiments. In contrast, Secondary studies serve for comparisons between individual investigations to generalize the results and are represented by systematic reviews.

Secondly, in order to make the introduced ontologies comparable, the template «Problem, Objectives, Suggested method and Future works» that was used by me during the reading of papers, will be applied. Although this segmentation cannot transfer the whole information contained in papers, it should be sufficient for the set intentions.

3.1 Ontology to support systematic reviews in Software Engineering

Before we go on to systematic reviews, let me introduce Evidence-based Software Engineering (EBSE), whose main instrument they are. EBSE was evolved by Kitchenham [14]. The author

supposes that SE might benefit from the evidence-based methods, as it was done in medicine with appearance of Evidence-based Medicine (EBM). The goal of EBSE is «to provide the means by which current best evidence from research can be integrated with practical experience and human values in the decision making process regarding the development and maintenance of software» [14]. In other words, what SE practice works, when, where and which tools and standards are needed for that. This all can and should be proven by experiments using Systematic reviews (SRs), where SRs are a «form of secondary study that uses a well-defined methodology to identify, analyze and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable» [13] or simplified a tool to obtain accurate knowledge by analyzing the primary studies to eliminate possible biases.

Having so detailed introduction of origin SRs, Biolchini has set a **Problem**: produce knowledge that can be based on scientific methodology. Besides, according to his work this problem implies the following **Objectives**:

- Discussing the significance of experimental studies, particularly SRs and their use in supporting software processes
- Present a template designed to support systematic reviews in SE
- Introduce development of ontologies to describe knowledge regarding such experimental studies

So as to fulfill the determined objectives, the author introduces, firstly, a template for systematic review protocol (Figure 4) that was created based on corresponding Guidelines [13] and his own experience. It is worth to notice the later relevant parts of them, such as Problem (SR target, describing the research context), Intervention (observation target in SR), control (initial data already possessed by researcher), outcome measure (metrics to measure effects), experimental design (which statistical analysis method will be applied on the collected data).

Secondly, the author **suggests** «Scientific research ontology» that is organized in level-structure (entities of levels can be seen as concepts) and the levels possess taxonomic and meronymic hierarchies (can be seen as roles), namely *is_a* and *has* relations. The following paragraph summarizes and analyzes the main components and features of the suggested ontology level by level with pertaining relations between them.

Level 0: Different knowledge of domains that are involved in the conduction of SRs in SE, represented by *Experimental Method*, *Primary Research* and *Research Synthesis*

For the further discussion the domain *Primary Research* is selected, but the analog statements and conclusions can be derived and provided by the remaining ones.

Level 1: The conceptual entity, represented by *Primary Study Element* is the highest level of hypernym of the *Primary Research* and subsumes the concepts in the lower levels of hierarchy.

1. Question Formularization 1.1. Question Focus 1.2. Question Quality and Amplitude - Problem - Question. - Keywords and Synonyms - Intervention - Control - Effect - Outcome Measure - Population. - Application - Experimental Design 2. Sources Selection 2.1. Sources Selection Criteria Definition 2.2. Studies Languages 2.3. Sources Identification - Sources Search Methods - Search String - Sources List 2.4. Sources Selection after Evaluation 2.5. References Checking 3. Studies Selection 3.1. Studies Definition - Studies Inclusion and Exclusion Criteria Definition - Studies Types Definition 3.2. Procedures for Studies Selection	3.3. Selection Execution - Initial Studies Selection - Studies Quality Evaluation - Selection Review 4. Information Extraction 4.1. Information Inclusion and Exclusion Criteria Definition 4.2. Data Extraction Forms 4.3. Extraction Execution - Objective Results Extraction i) Study Identification ii) Study Methodology iii) Study Results iv) Study Problems - Subjective Results Extraction i) Information through Authors ii) General Impressions and Abstractions 4.4. Resolution of divergences among reviewers 5. Results Summarization 5.1. Results Statistical Calculus 5.2. Results Presentation in Tables 5.3. Sensitivity Analysis 5.4. Plotting 5.5. Final Comments - Number of Studies - Search, Selection and Extraction Bias - Publication Bias - Inter-Reviewers Variation. - Results Application - Recommendations
---	---

Figure 4: Systematic review protocol template [1]

Level 2: The main concepts, represented by *Structure of Study* and *Quality of Study*

Level 3: The subcategories of one of the main concept Structure of study, represented by *Problem*, *Hypothesis*, *Intervention*, *Control*, *Measurement*, *Outcome* and *Unit of Study*

Level 4: The entities of the subcategory *Outcome* that is demonstrating the ontological hybridism, having not only taxonomic relations, represented by *Target Outcome* and *Surrogate Outcome*, but also meronymic relations, represented by *Endpoint*, *Incidence*, *Prevalence*, *Effect Modification* and *Effect Modifier*

As we can observe, the above described part of the ontology **results** in an object that is directly linked with Systematic review protocol template. The full ontology conceptualizes on roles we can read in the paper, but the given example in depth of levels depicts the power comprehensive cover of SR needs.

Though the presented ontology at the moment of publication was only in development, it can be esteemed as robust for the entire diversity of SRs in different types of Studies. Nevertheless, the author points out on possible **future works**, which are mainly related to merging of the presented Scientific Research Ontology with Software Engineering Ontology and to the successive integrating them into eSEE (experimental Software Engineering Environment). Moreover, this could lead towards a wider Experimental Software Engineering Ontology that theoretically will combine all received evidence-based knowledge in Software Engineering.

3.2 Ontologies for Controlled Experiments on Software Engineering

In contrast to the previous subsection, this one is about a part of Primary Studies, whereas one of the main subjects is Controlled Experiments. They serve Experimental Software Engineering as an instrument to build a body of knowledge for diverse and exceeding software practices to support making successful decision in SE.

Obviously, there exist many problems in this field and one of them is trying to be solved by R. E. Garcia [8]. This **Problem** is the sharing of knowledge among research groups. It requires replication of Controlled Experiments. The generated Knowledge during these experiments is registered in the so-called Lab Packages (procedures, the results and conclusions). However, researchers face difficulties reviewing the lab packages and suffer from the lack of standardization that leads to obstacles in sharing knowledge among research groups.

According to the problem the following **Objectives** were set:

- Explore ontologies to support knowledge transfer, helping to elucidate the associated concepts of controlled experiments and their relationships.
- Present an Ontology to experimental studies, named EXPEROntology - tool for knowledge transfer, assisting researchers, reviewers, and meta-analysts in designing, conducting and evaluating controlled experiments.
- Validate the ontology, whilst instantiating it to a controlled experiment.

Before we come to the suggested ontology, it is noteworthy to talk about the main object of the current research, in particular five Controlled Experiments phases and its components [31]:

1. Definition: hypotheses and experiment goals
2. Planning: execution plan and environment; subjects and their profiles; dependent and independent variables; validity
3. Operation: preparation, execution, data validation
4. Analysis: analysis of collected data
5. Packaging : artifacts, procedures, results into Lab Packages (is recommended executing parallel with each phase)

Some above named components show up in the **suggested** ontology, which can be seen on Figure 5 and has the following ordered workflow, consisting as usual of concepts and roles (relations) between them:

1. *Lab Package* from *Original Experiment* (created by *Designer*) is used for *Replication* (by *Replicator*) and generation of a new *Lab Package*.

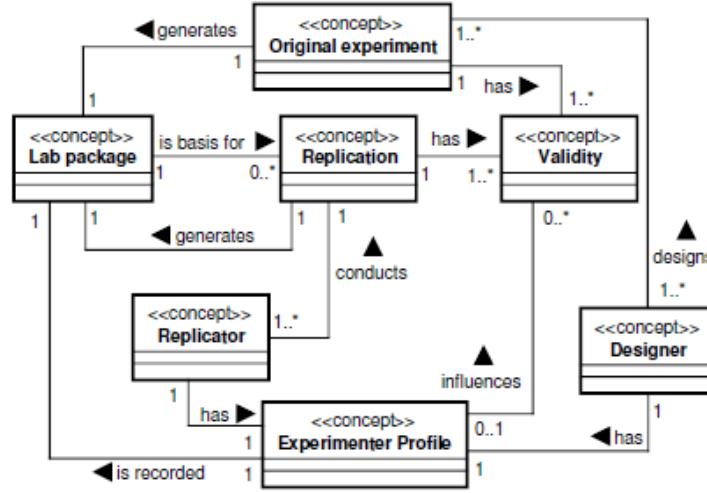


Figure 5: Ontology for Controlled Experiment [8]

2. *Designer* and *Replicator* have *Experimenter Profile*: negative influence attests a lack of experience, positive - high experience
3. *Original Experiment* and *Replication* evaluated regarding to *Validity* with four types: conclusion (relationship between the treatment and outcome), internal (relationship between the factors and the outcome), construct (relation theory and observation) and external (generalization).

But they are so superficial designed that, apparently, they cannot satisfy all requirements of researchers. Therefore, Garcia et al. goes deeper and refines the above described ontology, presenting the EXPERontology for Lab Package that can be observed in Figure 6. Besides, four years later the author publishes the guidelines for this ontology [26], extending the earlier work and providing more circumstantial description of that. Let us focus below on the main structures and features of the presented ontology, corresponding to its workflow and how it refers to earlier set up phases of the Controlled Experiments:

1. Definition: establishing of *Initial hypothesis*, composed by *Object of study*, *Purpose*, *Quality focus* in a specific *Context*.
2. Planning: generating *Hypothesis formalized* from Definition phase. Experimenter defines *Dependent*, *Independent Variables* and *Experimental Object* that contains *Technology* and *Artifacts* to be used in controlled experiment. Furthermore, in this phase the *Experimental design* is created and refined by *Subject* and its *Profile*.
3. Operation: elaborating *Execution plan*, which is obtained by *Tasks*

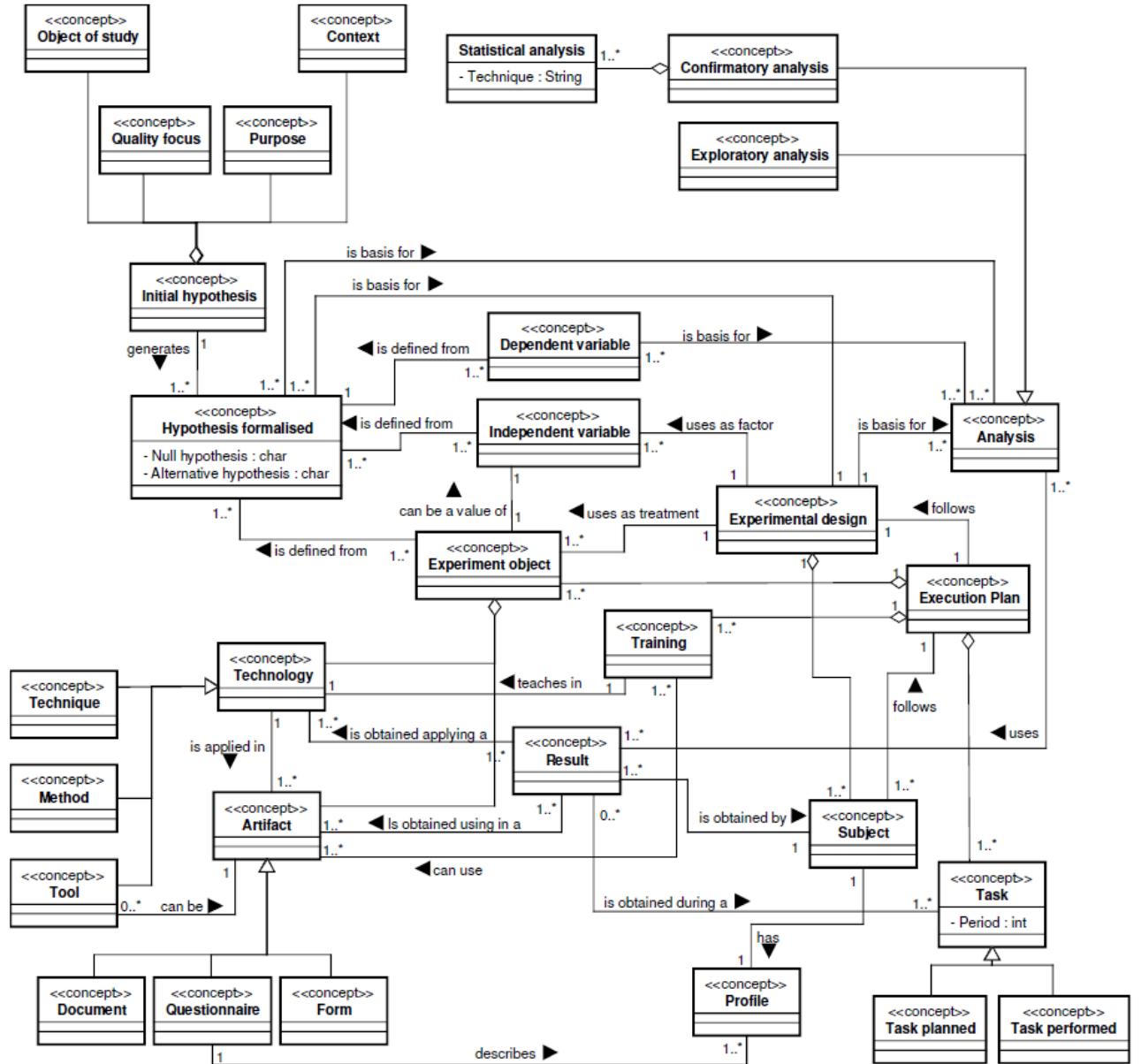


Figure 6: Ontology for Lab Packages [8]

4. Analysis: gathering *Results* from Operation phase are to be used in *Analysis* of different types, such as *Confirmatory*(testing *Hypothesis formalized*) or *Exploratory*(investigating new relationships)

Since it is desired to make all these concepts working, we are missing the connection axioms for the entire ontology. It is enough to define four predicates:

1.

$$Design(subject, SetOfTreatment)$$

2.

$$\begin{aligned} & Factor(f_1, \dots, f_n) \\ & \forall f \in Factor, \exists Treatment(f) = v_1, \dots, v_n, n \geq 2 \\ & dom(Treatment) = Artifact \cup Technology \\ & SetOfTreatment = (vf_1, \dots, vf_n) | \forall f, vf_n \in Treatment(f_n) \end{aligned}$$

3.

$$\begin{aligned} & \forall subject, SetOfTreatment \\ & Execution(subject, SetOfTreatment) \rightarrow Task(ta_1) \wedge \dots \wedge Task(ta_n) \end{aligned}$$

4.

$$Task(ta_n) \rightarrow Training(subject, t_r, a, p) \vee Applying(subject, t_e, a, p)$$

The last but not least is the validating of presented ontology by obtaining the **Results** on existing data. Thus, in order to illustrate the power of the suggested ontology, an experiment that was originally conducted by Basili and Selby [2] has been chosen. The objective of the study was «compare three state-of-the-practice software testing techniques: a) code reading by stepwise abstraction, b) functional testing using equivalence partitioning and boundary value analysis, and c) structural testing using 100 percent statement coverage criteria.» [2]. So, the Experimental Design of the study: 32 Subjects were divided in 3 groups (advanced, intermediate and junior) and each of them applied 3 software testing techniques(code reading, functional testing and structural testing) on 3 different pieces of software(text processor, numeric abstract data type and database maintainer). According to this description the EXPEROntology was instantiated as the following (for instance only for one subject S1).

1.

$$\begin{aligned} & Design(S1, Advanced, Code Reading, P3) \\ & Design(S1, Advanced, Functional Testing, P2) \\ & Design(S1, Advanced, Structural Testing, P1) \end{aligned}$$

2.

$Factor = (Expertise, Technique, Program)$
 $for\ Expertise, Treatment = \{Advanced, Intermediate, Junior\}$
 $for\ Technique, Treatment = \{Code\ Reading, Functional\ Testing, Structural\ Testing\}$
 $for\ Program, Treatment = \{P1, P2, P3\}$
 $SetOfTreatment = (Advanced, Code\ Reading, P3)$

3. and 4.

$Execution() \rightarrow$

$Training(S1, \{Code\ Reading \vee Functional\ Testing \vee Structural\ Testing\}, --, --) \wedge \dots$
 $Applying(S1, \{Code\ Reading \vee Functional\ Testing \vee Structural\ Testing\}, P3, --) \wedge \dots$

After the instantiation of the ontology, we can observe that all pieces of information given in the conducted study can be encapsulated and saved in EXPEROntology without losing any information.

Besides, we can observe the missing values on the predicate: Training - artifact and period of training, Applying - period for the application. After looking in the study we can see that it does not bring them indeed. It allows us to make one more important conclusion «the ontology can also be used as a mechanism to improve the obtained data set from the Lab Package» [8].

Also, the authors intend in **future works** to combine the presented EXPEROntology with OntoTest. By merging these two ontology it will be possible to create a whole architecture that «supports the development of environments/tools to automate software testing activities and related experimental studies» [8].

We have just looked at the ontology for the controlled experiments. However, there exists another work from alternative perspective, which was suggested by H. Siy and Y. Wu [27]. Although they had the similar **Objectives**:

- Present an ontology for analyzing empirical studies of SE, in particular the design of software engineering experiment
- Encapsulate the experience experts by means of an ontology for experimental designs using Protege OWL

they are aware of the **Problem** wider in the enterprise way, namely a badly designed experiment can increase the cost and risk of invalid results. And the solving of this problem includes not only the fundamental knowledge of the mechanisms, methods and tools, thus improving our understanding of which one works best under what situation, but also the way software engineers work, think and interact with each other.

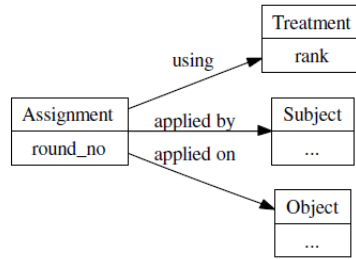


Figure 7: Ontology fragment depicting the concepts involved in the design of experiments [27]

So as to solve this problem, the author **suggest** the following ontology concept. First of all, the main concepts and their roles (Figure 7 are presented:

- *Treatment* Software Engineering Technique/Method/Process being studied
- *Subject* Person, Developer/Student participated on experiment
- *Object* Entity, Program/Model
- *Assignment* Relation between all of them; in an assignmentInstance: a subjectInstance is assigned to apply a treatmentInstance in an objectInstance.

Secondly, in order to bring all of these concepts into a working order, the connection axioms are set. In sum, we get four necessary constraints and the purposes they fulfill:

1. No *Subject* can be assigned a *Treatment* that is less sophisticated than the other ones he was already assigned to. What for: subject assigned to one treatment may use the knowledge and experience gained from that treatment.
2. No *Treatment* was applied by only one *Subject*. What for: experiment subjects have varying backgrounds and abilities that implies different results. If only one subject is related, then it is not scientifically meaningful.
3. *Subject* is assigned to several *Treatments*. What for: assess the variability introduced by that subject.
4. An *Object* is treated by several *Subjects*. What for: provide a way to untangle subject performance from object complexity.

Finally, the created ontology concept was evaluated on experiments on software inspections [3]. Aligning to presented concepts the concrete values i.e. *Treatment* - reading technique, *Subject* - reviewer, *Object* - software requirements document. The **Results** were «encouraging» [27]. The author states that it was not found any validation after the applying all

constraints. Moreover, if the ontology was fed by invalid assignments, it was possible to observe inconsistencies as expected.

It remains to be said that the authors consider the **Future** of their presented ontology and the preliminary results that it has shown as a step towards organizing and accommodating such an ontology, which can also support other Software Engineering knowledge domains (not only the systematic review process).

3.3 Comparative analysis

To sum up, we could observe in this section four ontologies that were introduced by three different groups of researchers. Although the scientists had set different objectives and problems that they tried to solve, here can be discussed similarities of suggested in this sections ontologies. Firstly, it is remarkable that all researchers point on the advantages and opportunities which the adoption of ontologies give them in realization of their intentions. In the researcher's opinion, an ontology is the best tool to accumulate any experience or knowledge (especially, from experiments), formalize them for later representing, sharing and transferring. Nevertheless, ontology is not a silver bullet, because of the apparent drawbacks. It cannot depict precisely all objects of the real world and relations between them. An ontology is still restricted by the first-order logic. But sometimes it can be enough to get desired results. Secondly, even though only Biolchini et al. [1] intended to work directly on the supporting of systematic reviews, the other presented ontology also pertain to them e.g. Garcia et.al. [8] evaluated their ontology on experiments with V&V techniques [2] and H. Siy and Y. Wu [27] on experiments with software inspections [3]. This congeniality shows us, how important systematic reviews are and thereby EBSE in modern Software Engineering.

To the very end, let us discuss the differences between the presented ontologies. First of all, according to the categorization in subsection 2.2.2, the Biolchini et al. for supporting systematic reviews [1] must definitely belong to the area Methods as by analogy provided initiative Cochrane Collaboration for Systematic reviews of health care; Garcia et al. two ontologies [8][26] exactly match the category Reproducibility, since their main objective to support replication of the SE studies; the last one by H. Siy and Y. Wu [27] might belong to as Reproducibility as Methods because of similarities with three ontologies presented before. Finally, we could observe that the modeling of each presented ontology was done by using of different ontology languages or at least different kinds of the same language what can also lead to the barriers by applying the presented ontologies and making them the standards in the research field.

4 Ontology-based systems in scientific search

During the searching and reading of the suggested papers and the papers found by reference-based search, the two facts have appeared in sight: Ontologies are used in the contemporary

search engines and the scientific world is using ontologies successfully for a large number of tasks. Therefore, the papers about Ontology-based systems in the scientific search were found and will be analyzed further in this section. Moreover, the presented mechanisms and how they can be applied in Software-Engineering-Meta-Research

4.1 Semantic Web

Every discussion about ontologies is incomplete, if the concept of Semantic web is omitted. Therefore, in this subsection the Semantic Web is presented and compared with a traditional Web (WWW).

The first definition of Semantic Web was coined by T. Berners-Lee: «The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation» [5]. But what are the purposes that we are unable to use the traditional Web and urgently need an implementation of the Semantic one?

Firstly, we are all aware of the constant growing of our Web, namely the information in it. Thus, indexing and retrieving information becomes an unbearable task if we do it manually or semi-automatically as it happens in the current Web. Secondly, the current Web that we have is a Web of HTML-documents (or other formats), which means only humans can really understand this information and refer to these relations in a document. The Semantic Web should become the Web of meaning and thereby understood not only by humans, but also by machines. Thirdly, every language has an uncountable number of synonyms with the same meaning, but different word representation. Besides, the current Web is getting multi-lingual. As in the example: «English shopping website would use the word price to refer to an items price, while a Dutch website would use the word prijs, a French website would use the word prix...» [23] the same information differ in other languages and thus different for the machines. And finally, to summarize the current paragraphs, the main purposes of the Semantic Web were introduced by W3C that states «The Semantic Web is about two things. It is about common formats for interchange of data, where on the original Web we only had interchange of documents. Also it is about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing» [29].

4.2 Semantic Search Engine

The main requirement for the Semantic Web is a search engine for the proper interpretations of the user's search intentions, ability to detect the relations between found results and pretension of their with respect to relevance.

Before we go to the semantic search engine, the basic one should be surveyed. Compressed information about the traditional search engine architecture is retold further based on the description in [10]. Before the real search query begins, the crawler indexes the already

available information in the background, by storing, marking and organizing the fetched data from WWW. It allows the search engine to collect and output data with respect to the highest page rank, when the real search query is given.

Knowing the main steps of the basic search engine we are now able to see the main feature of the Semantic search engine. The thing that distinguishes the semantic search engine and brings it to the next level is using ontologies. In the following paragraph one variation of the Semantic search engine based on ontology is presented based on description of Fang. et al. [30], which makes easier the discussing features and opportunities of using ontologies in semantic search.

The basic search engine suffers from a **Problem** that it searches the web documents based on keywords, when instead should be based on their content. Therefore, there were set the **Objectives**: use ontologies as the knowledge representation domain, use OWL DL to represent them and finally, employ them in search engine; create a preliminary implementation of a framework and test it.

Getting rid of particular details and summing up the algorithm, the **suggested** method has the following steps (after the input of a query):

1. Parse the keywords in the query, which relate to concepts in ontology and forward it to the Reasoner.
2. Reasoner returns RDF triples as input to the next step.
3. Retrieve the referring to these concepts documents based on extended Term-Document Matrix (TDM), where the extended TDM is constructed before, with the three steps: Calculating the basic TDM (like in a basic search engine), defining the relevance between ontologies and documents and calculating the extended TDM based on the previous step. This allows for the extended TDM depicting the connections between concepts in the domain ontologies.
4. Sort the retrieved documents by relevance and output them to the user.

This constriction of the semantic search engine **results** in the improving of the main parameter for the search query, namely precision and recall that can be observed on Figure 8. To be remarked, the experiment is evaluated on one topic(with corresponding ontology) in 3600 manually classified documents for three topics. Of course, it is one of the first approaches and authors' **future** works are going to optimize the relevance levels. To conclude the main differences between basic search engine and semantic search engine based on ontologies [10, p.653]:

- Query in basic search engine takes less time, but the Result of a query in semantic search engine is much more relevant.
- Basic search engine crawler uses all domains based on keywords. The semantic one considers the logical meaning instead.

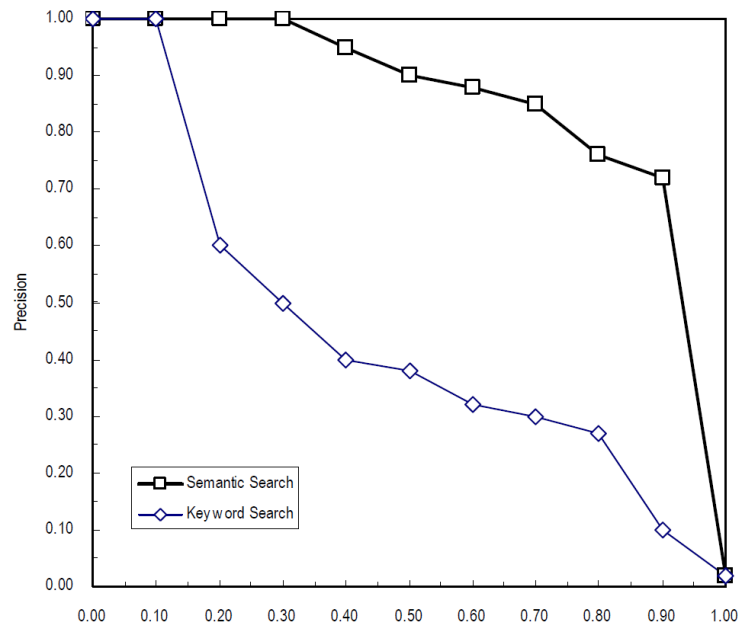


Figure 8: Precision-recall graph comparison of semantic and keyword search [30]

- Basic search engine construction costs in implementation and time are less, whereas the semantic one reduces costs of searching for a user.

4.3 Examples in scientific world

As it was said in the previous subsection the main feature of the semantic search engine is using the ontologies. But which of them should be used adopted and which left out. In the next section there will be an of introduced all scientific search ontologies «palettes» and in the sections after that it will be discussed what requirements they should fulfill and how they can interact and can be used together.

4.3.1 Scientific ontologies

With constant growing number of publishing articles in different scientific fields the already announced problems have arisen. One of the solutions is the so-called Semantic publishing that replaces the direct publishing with the smarter one, providing not only the original text, but also the information for understanding this text. Scientist, who are interested that their studies and publications will be read and potentially spread better can spend some time that allows ontologies to receive information in proper way and thus, semantic search engines work better. However, there are many initiatives and types of ontologies which differ from each other drastically. This fact arises a **Problem**: the selecting of suitable ontologies that sophisticate the initial requirements and purposes. Making it as the main **Objective** solving

this problem A. Ruiz-Iniesta and O. Corcho present a review of such ontologies [24]. For the better understanding the authors **suggest** a plain categorization which contains only three main classes of ontologies. The information about these classes and the examples relating to them is displayed in the following enumerated paragraph:

1. «ontologies for describing document structure (sections, paragraphs, etc.)». The ontologies of this class are primarily concentrating on structure of a scientific publication. Some of the concepts of such ontology could be Author, Title, Volume, etc. As example, Document ontology. For the semantic search input it used a simple extension of HTML-file that an author of a publication should enter manually.
2. «ontologies for describing the rhetorical elements (introduction, results etc.)». The ontologies of this class are primarily concentrating on rhetorical structure of a publication that does not depend on any research field. Moreover, they mainly consist of three parts: header, body, tail. Some of concepts could be for header: Creator, Title etc.; for body: Introduction, Method etc.; for tail: Acknowledgment, Reference etc. As example, Ontology of Rhetorical Box (ORB). For the semantic search input the RDF format is used.
3. «ontologies for describing bibliographies and cites structure». The ontologies of this class are primarily concentrating on citing and referencing of a scientific publication. Because of importance of them, as an example, it is worth of noting the two main ontologies in this class, namely FaBiO and CiTo. « FaBiO, the FRBR-aligned Bibliographic Ontology, an ontology for recording and publishing bibliographic records of scholarly endeavours on the Semantic Web, and CiTO, the Citation Typing Ontology, an ontology for the characterization of bibliographic citations, both factually and rhetorically» [22]. For the Semantic search input is used OWL 2 DL.

It is to be noted that this classification can be non-exhaustive. For example, Brack et al. point out to the broader list of ontology species. Below are some of them [6] : argumentative with concepts like claims, constructive and comparative statements about other work, activity based with concepts like sequence of research activities, aspects and elements of research articles. Two separate classes that is not in the [24], but given in [6], in my opinion should be also noticed. One of them is a class of domain-specific ontologies. For mathematics it would be with concepts like proofs, assertions, etc. or machine learning with concepts like dataset, neural network, etc. Another one is a class of the taxonomy building domain-specific research ontologies. A good and well-known example for this category is Computer Science Ontology(CSO). CSO is a «a large-scale, automatically generated ontology of research areas, which includes about 14K topics and 162K semantic relationships»[25], whose main purpose is creating comprehensive taxonomy for the Computer Science. CSO has two features over other attempts to make a classification for all terms of the science, namely «i) it includes a very large number of topics that do not appear in other classifications, and ii) it can be updated automatically»[25].

4.3.2 Requirements for scientific ontologies

As we could see there is huge diversity in selecting of suitable ontologies. But it is very important to create a list of requirements which should be fulfilled by the ontologies, in order to be considered at all. But before we go on to them it is also necessary to speak about how these ontologies are integrated in the current scientific search engines. In this seminar work it was actively used the scientific search engines such as Google Scholar and Microsoft Academic. The main skeleton of them are Knowledge Graphs(KGs) (Microsoft Academic Knowledge Graph or SciGraph). This type of graphs is created with a purpose to interlink all available research articles through citations, authors, journals and so on. According to [6] there two main parts that a KG contain: «(1) an ontology describing a conceptual model, and (2) the corresponding instance data following the constraints posed by the ontology» [6]. It implies a necessity for designing of an appropriate ontology and populating of trough instances. Further we are focusing only on the first part of implication which refers to ontology design.

Based on systematic literature review [13] that was discussed above in subsection 3.1 and interviewing experts in KGs Brack et al. [6] elaborated the requirement analysis and, consequently, possible approaches for contracting Open Research Knowledge Graphs (ORKG) and , in particular, the ontology design requirements. The authors suppose that the ORKG have primarily seven use cases that can be improved by semantic approaches. They are: Getting research field overview, finding related work, assessing relevance, extracting relevant information, getting recommended articles, obtaining deep understanding and reproducing results. These use cases have functional and non-functional requirements such as (based on [6]): domain-specification (high like in CSO or low in ORB) and granularity (how circumstantial it is in relation to the number of concepts e.g. high in FaBiO low in Document Ontology) of the ontology, coverage (how many scientific fields are represented) and quality (how reliable the research is) of the instance data. According to these non-functional needs we can combine the use cases in four main groups, those which require:

1. High domain-specification, granularity, quality and low coverage, namely extract relevant information and get research field overview with functional requirements: maintaining of the structure, evolution and relevance of contained information.
2. High coverage, low domain-specification, granularity, quality, namely find related works and get recommended articles with functional requirements: supporting searching for related work (also in natural language) and focusing on particular parts of work.
3. High quality, medium domain-specification and granularity, low coverage, namely obtain deep understanding and reproduce results with functional. requirements: linking to the related artifacts such as source code or datasets, linking or representing of semantic description such as guidelines or standards
4. Medium domain-specification, coverage, quality and low granularity, namely assess relevance with functional requirements: relevance for the matching of search interests and highlighting of most relevant zones in an article.

We can definitely observe that the first two classes are the poles of classification and, thus, approximated as the last two classes that can be referred to the first ones. Therefore Brack et al. [6] suggest two approaches: manual (for the first class) and (Semi-)automatic (for the second class). The manual approach can be carried out only with the help of community and with supporting of meta-modeling i.e. special templates for inputting the information. The (semi-)automatic approach means the usage of methods such as automatic construction of ontologies based on Natural language processing (NLP) or automatic information extraction from text in sentence or phrasal level that analyzes text and looks for language templates and extracts information from it.

4.3.3 Other ways

The list of given examples of existing approaches for creating ontology-based semantic search would be incomplete, if it is only said about diverse scientific ontologies and KGs. Also there exist some independent separate approaches. All of them cannot be listed here, but some of them that came across during my search of relevant papers could and I found them interesting.

There are many approaches in the medical field. One of them is Ontology based Semantic Search Engine for Cancer [23] that was made for analysis of cancer, its categories, types, causes, symptoms, etc.

Another approach [19] should support retrieving information from scientific abstracts that was also evaluated on medical papers, but Milward et al. suppose that other research fields could also benefit from it.

Finally, there is an approach [28], whose terminal purpose is creating Semantic Web Expert System (SWES) that is able to give answers like an expert for different types of queries, in particular scientific ones. SWES should use Ontology Web Search Engine that will look for the already existing and available ontologies in the Web, index, merge them and allow using of all knowledge that is accumulated in them.

4.4 Using of Knowledge Graphs in Software-Engineering-Meta-Research

After the obtaining of necessary knowledge in the field of KGs and the fact that a skeleton of every KG are ontologies, we can now discuss how the main disadvantage of the suggested in the section 3 Software-Engineering-Meta-Research ontologies can be solved, namely the different approaches and incompatibility to each other (at least the multiple using of them is neither provided nor discussed). Despite all authors posit that their approaches are the best and their developed ontology can be easily completed by needs of an user, the user could need, for example, both ontologies for packaging of a controlled experiment and systematic reviews for this experiment. Or another possible use case could be: many different researchers could have used the different presented ontologies and might have obtained the great results, but there is still no opportunity to query for the results through all ontologies, unless one ontology is integrated into another, that could be a very labor-intensive task. Since the KGs can combine diverse ontologies, enable the semantic search and, consequently, the effective

analysis of obtained knowledge, they can be used in Software-Engineering-Meta-Research. This makes possible not only using of multiple ontologies, but also making connected the knowledge that was obtained by them.

Unfortunately, there are not many works in this direction, but, for instance, Ze-Qi Lin et al. [16] suggest the using of Software Knowledge Graph for software intelligent development. In this paper the Software Knowledge Graph consists of nodes and directed edges, where «graph, nodes represent software knowledge entities (e.g., classes, issue reports and business concepts), and directed edges represent various relationships between these entities (e.g., method invocation and traceability link)» [16]. That depicts the structure of any ontology. Moreover, the Software Knowledge Graph is a connecting tissue between different documents, such as source code files, PDFs or XML entities, and Intelligent Development Environment, which includes the assistance to software construction and testing and verification process.

Based on the facts of the successful using ontologies in Software-Engineering-Meta-Research and the integrating them into KGs approaches, in order to have different ontologies inside one system, there is no foreseeable reasons, why the suggested in section 3 ontologies cannot be combined and used together within one Software-Engineering-Meta-Research project.

5 Conclusion

In this seminar work, based on extensive references that were obtained by using both reference-based and data-based searches, the knowledge about different concepts has been received, summed up, presented and analyzed. Among them were ontology in computer science, different ontology languages, their expediency and limitations. Then, the term Meta-research and the corresponding categorization, that is used during the whole work, have been presented. The combination of these two terms in Software Engineering, namely the usage and the application of ontologies in Software-Engineering-Meta-Research have been investigated and a part of the current state-of-the-art in this field has been reported. Besides, the presented ontologies have been compared and analyzed by extracting their advantages and disadvantages. Moreover, during the search processes the affinity of the above-named terms with Semantic Web and Semantic search engine have been found out and, thus, it has been decided to investigate the ontology-based systems in the scientific search. For this purpose the necessary foundations in this field have been briefly set up and, then, the diversity of existing scientific ontologies have been investigated, presented and attempted to be classified. To the very end, the requirements for the present scientific ontologies integrated in the Open Research Knowledge Graph (ORKG) have been discussed and also, the other approaches that were covered on ORKG have been introduced.

Furthermore, the aim of this seminar work was not only to find the necessary papers and references, to summarize and report them, but also to categorize, compare, and analyze them and, finally, to determine the possible perspectives and insights for the works performed in the future.

It has been shown, how using ontologies has changed a large number of the scientific fields such as Mathematics, Medicine and, especially Computer Science. In my pinion, the using of ontologies in Software-Engineering-Meta-Research has already become rather significant by playing a major role in the research field. Moreover, I suppose its influence will become more prominent overall regarding the workings in this particular scientific field and it will manage to change Software Engineering towards a more advantageous and exceedingly remarkable future.

References

- [1] Jorge Calmon de Almeida Biolchini et al. “Scientific research ontology to support systematic review in software engineering”. In: *Adv. Eng. Informatics* 21.2 (2007), pp. 133–151.
- [2] V. R. Basili and R. W. Selby. “Comparing the Effectiveness of Software Testing Strategies”. In: *SE-13* (1987), pp. 1278–1296. ISSN: 0098-5589. DOI: 10.1109/tse.1987.232881.
- [3] V. R. Basili, F. Shull, and F. Lanubile. “Building knowledge through families of experiments”. In: 25 (1999), pp. 456–473. ISSN: 0098-5589. DOI: 10.1109/32.799939.
- [4] Sean Bechhofer et al. *OWL Web Ontology Language Reference*. 2004. URL: <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>.
- [5] Tim Berners-Lee, James Hendler, and Ora Lassila. “The Semantic Web”. In: 284 (2001), pp. 34–43. ISSN: 0036-8733. DOI: 10.1038/scientificamerican0501-34.
- [6] Arthur Brack et al. “Requirements Analysis for an Open Research Knowledge Graph”. In: (May 20, 2020). arXiv: 2005.10334v1 [cs.DL].
- [7] Fred Freitas and Fernando Lins. “The Limitations of Description Logic for Mathematical Ontologies: An Example on Neural Networks”. In: *ONTOBRAS-MOST*. Ed. by Andreia Malucelli and Marcello Peixoto Bax. Vol. 938. CEUR Workshop Proceedings. CEUR-WS.org, 2012, pp. 84–95. URL: <http://ceur-ws.org/Vol-938>.
- [8] Rogério Eduardo Garcia et al. “An Ontology for Controlled Experiments on Software Engineering”. In: *Proceedings of the Twentieth International Conference on Software Engineering & Knowledge Engineering (SEKE’2008), San Francisco, CA, USA, July 1-3, 2008*. Knowledge Systems Institute Graduate School, 2008, pp. 685–690.
- [9] Thomas R. Gruber. “A Translation Approach to Portable Ontology Specifications”. In: *Knowledge Acquisition* 5.2 (1993), pp. 199–220.
- [10] Paras Nath Gupta et al. “A novel architecture of ontology based semantic search engine”. In: *Int. J. Sci. Technol* 1.12 (2012), pp. 650–654.
- [11] John P. A. Ioannidis, Kevin W. Boyack, and Richard Klavans. “Estimates of the Continuously Publishing Core in the Scientific Workforce”. In: 9 (), e101698. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0101698.

-
- [12] John P. A. Ioannidis et al. “Meta-research: Evaluation and Improvement of Research Methods and Practices”. In: 13 (), e1002264. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1002264.
- [13] Barbara Kitchenham and Stuart M. Charters. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Tech. rep. 2007. URL: https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering.
- [14] Barbara A. Kitchenham, Tore Dyba, and Magne Jorgensen. “Evidence-Based Software Engineering”. In: *26th International Conference on Software Engineering (ICSE '04)*. Edinburgh, Scotland, May 2004, pp. 273–281.
- [15] Boris Konev. *Lecture notes of course Ontology and knowledge representation*. 2010. URL: <https://www.lektorium.tv/speaker/2680>.
- [16] Zeqi Lin et al. “Intelligent Development Environment and Software Knowledge Graph”. In: *J. Comput. Sci. Technol* 32.2 (2017), pp. 242–249.
- [17] Malcolm R. Macleod et al. “Biomedical research: increasing value, reducing waste”. In: 383 (2014), pp. 101–104. ISSN: 0140-6736. DOI: 10.1016/s0140-6736(13)62329-6.
- [18] *Meta-Research, Wikipedia*. Last accessed on 2020-06-05. URL: <https://en.wikipedia.org/wiki/Metascience>.
- [19] David Milward et al. “Ontology-Based Interactive Information Extraction From Scientific Abstracts”. In: 6 (2005), pp. 67–71. ISSN: 1531-6912. DOI: 10.1002/cfg.456.
- [20] *Ontology (computing), Oxford Lerner’s online dictionary*. Last accessed on 2020-06-05. URL: <https://www.oxfordlearnersdictionaries.com/definition/american-english/ontology>.
- [21] *Ontology, Merriam-Webster online dictionary*. Last accessed on 2020-06-05. URL: <https://www.merriam-webster.com/dictionary/ontology>.
- [22] Silvio Peroni and David M. Shotton. “FaBiO and CiTO: Ontologies for describing bibliographic resources and citations”. In: *J. Web Semant.* 17 (2012), pp. 33–43. DOI: 10.1016/j.websem.2012.08.001.
- [23] Syam Raj and S Sarumathi. “Ontology based Semantic Search Engine for Cancer”. In: *International Journal of Computer Applications* 95.5 (2014), pp. 39–43. ISSN: 0975-8887. DOI: 10.5120/16594-6308.
- [24] Almudena Ruiz-Iniesta and Oscar Corcho. “A review of ontologies for describing scholarly and scientific documents”. In: *Proceedings of the 4th Workshop on Semantic Publishing co-located with the 11th Extended Semantic Web Conference (ESWC 2014), Anissaras, Greece, May 25th, 2014*. Ed. by Alexander Garcia Castro et al. Vol. 1155. CEUR Workshop Proceedings. CEUR-WS.org, 2014. URL: <http://ceur-ws.org/Vol-1155/paper-07.pdf>.

-
- [25] Angelo Salatino et al. “The Computer Science Ontology: A Comprehensive Automatically-Generated Taxonomy of Research Areas”. In: (2019). URL: <http://oro.open.ac.uk/66268/1/DataIntel-CS0.camera.ready.0R0.pdf>.
 - [26] Lilian Passos Scatalon, Rogério Eduardo Garcia, and Ronaldo Celso Messias Correia. “Packaging Controlled Experiments Using an Evolutionary Approach Based on Ontology(S)”. In: *Proceedings of the 23rd International Conference on Software Engineering & Knowledge Engineering (SEKE'2011), Eden Roc Renaissance, Miami Beach, USA, July 7-9, 2011*. Knowledge Systems Institute Graduate School, 2011, pp. 408–413.
 - [27] Harvey Siy and Yan Wu. *An Ontology to Support Empirical Studies in Software Engineering*. en. Mar. 2012. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.216.8899;%20http://csalpha.ist.unomaha.edu/~hsiy/research/icc09.pdf>.
 - [28] Olegs Verhodubs. “Towards the Ontology Web Search Engine”. In: (May 4, 2015). arXiv: 1505.00755v1 [cs.IR].
 - [29] W3C. *Semantic web Activity*. 2001. URL: <http://www.w3.org/2001/>.
 - [30] Wei-Dong Fang et al. “Toward a semantic search engine based on ontologies”. In: *2005 International Conference on Machine Learning and Cybernetics*. Vol. 3. 2005, 1913–1918 Vol. 3.
 - [31] Claes Wohlin et al. *Experimentation in Software Engineering - An Introduction*. Vol. 6. The Kluwer International Series in Software Engineering. Kluwer, 2000. ISBN: 978-1-4613-7091-8. DOI: 10.1007/978-1-4615-4625-2.