



Evaluation of Automatic Ontology Matching for Materials Sciences and Engineering

Master's Thesis in Computer Science at the
Albert-Ludwigs-Universität Freiburg, Faculty of
Engineering

Report A 02/20

Prepared by:

M. Sc. Engy Nasr

Matriculation Number:

4559022

First Supervisor (Uni):

Prof. Dr. Hannah Bast

Second Supervisor (EMI/Uni):

Prof. Dr.-Ing. habil. Stefan Hiermaier

Supervisor at Fraunhofer (EMI):

M. Sc. Martin Huschka

Februar 2020
Freiburg, Germany

Evaluation of Automatic Ontology Matching for Materials Sciences and Engineering

Master's Thesis in Computer Science at the Albert-Ludwigs-Universität Freiburg, Faculty of Engineering

Report A 02/20

Prepared by:

E. Nasr

Master Student – Digital Engineering

Supervisor:

M. Sc. M. Huschka

Research Fellow – Digital Engineering

Dr.-Ing. J. Fritsch

Head of Department – Material Dynamics

p. p.

Prof. Dr.-Ing. habil. S. Hiermaier

Director of Ernst-Mach-Institut

Table of contents

Preface	5
Acknowledgements	6
Declaration of academic integrity	7
Zusammenfassung	8
Abstract	10
1 Introduction	12
1.1 Thesis Approach	12
1.2 Thesis Structure	13
2 Related Work	15
3 Theoretical Background	17
3.1 Materials Modelling	17
3.1.1 The Four RoMM Materials Models	19
3.1.2 Solvers and Multi Scaling	20
3.2 Ontologies and Interoperability	22
3.2.1 How to Create an Interoperable Ontology	23
3.2.2 The Semantics Spectrum and its Computer Languages	26
3.2.3 Upper Level Ontologies and the BFO	28
3.2.4 Popular Ontologies and Ongoing Ontologies' Projects	30
3.2.5 Materials Sciences Ontologies	31
3.3 Ontology Matching	32
3.3.1 Ontology Matchers	35
3.4 Performance Evaluation of Ontology Matchers	38
4 Methodology	42
4.1 Ontology Matchers Preparation	42
4.1.1 Development Platform	44
4.1.2 Agreement Maker Light (AML)	45
4.1.3 LogMap	50
4.2 Benchmark Development	51
4.2.1 Preparations of Materials Sciences Ontologies	52
4.2.2 Test Cases Design	53
4.2.3 Manual Reference Alignments	55
4.2.4 Background Knowledge Ontologies	56

4.3	Performance Evaluation Schema	58
4.4	Performance Evaluation Workflow for Matching of Materials Sciences Ontologies	58
5	Results and Discussion	61
5.1	Number of Alignments Using the Materials Sciences Benchmark	61
5.2	Evaluation of Ontology Matchers using the Materials Sciences Benchmark	64
5.2.1	1 st Test Case	65
5.2.2	2 nd Test Case	68
5.2.3	3 rd Test Case	71
6	Conclusion and Future Work	74
6.1	Conclusion	74
6.2	Future Work	75
7	List of Tables	78
8	List of Figures	79
9	List of Abbreviations	82
10	References	84
	Appendix	88

Preface

This report was written by Engy Nasr within the framework of the examination regulations for the Masters of Science in Computer Science at the Faculty of Engineering.

The work was carried out in cooperation between the University of Freiburg and the Fraunhofer Institute for High-Speed Dynamics, Ernst-Mach-Institut, EMI, Freiburg, Germany and was prepared at Fraunhofer EMI.

Prof. Dr. Hannah Bast was in charge of the supervision at the University of Freiburg.

The University of Freiburg (Albert-Ludwigs-Universität Freiburg) is a public research university based in Freiburg im Breisgau, Baden-Württemberg, Germany. In 1457, the Habsburger have founded the university to be the second university in Austrian-Habsburg territory after the University of Vienna. Nowadays, the Albert-Ludwigs-Universität Freiburg is the fifth-oldest university in Germany. The university consists of eleven faculties, including the faculty of Engineering that has three departments, Microsystems Engineering, Sustainable Systems Engineering and my department, Computer Science. My field of study in Computer Science is Cognitive Technical Systems, which is one among two other field of studies in the department; Cyber-Physical Systems and Information Systems. Examples of my field of study are Robotics and autonomous intelligent systems, artificial intelligence and machine learning, image processing and computer graphics.

Martin Huschka (Digital Engineering) was in charge of the supervision at EMI. At Fraunhofer EMI, experimental, computer-aided and analytical methods are used in the five business units Defense, Security, Automotive, Space and Aviation to develop solutions aimed at improving the safety and reliability of components and structures under dynamic loading.

Acknowledgements

First and foremost, praises and thanks are to Allah, the Almighty, for His showers of blessings throughout my whole life and ever more during the tenure of my thesis.

I am highly indebted to my thesis examiner Prof. Dr. Hannah Bast for her kind vision, sincerity and motivation that have deeply inspired me.

I would like to express my deepest and sincere gratitude to my thesis supervisor M. Sc. Martin Huschka for his constructive and valuable suggestions during the planning, development and writing of the thesis. His willingness to generously give his time to the thesis has been very much appreciated. I would also like to thank Dr.-Ing. Michael Dlugosch for his advice, assistance, enthusiastic encouragement and guidance throughout my whole working period at Fraunhofer EMI. Thank you for believing in me and giving me such wonderful opportunities to learn and create.

My grateful thanks to M. Sc. Thomas Soot, who took his time, knowledge and effort to help creating the Materials Sciences manual reference alignments for the thesis.

My special gratitude and thanks to Ms. Johanna Holz for her constructive comments and feedback that made the finalization of the thesis possible.

To my amazing parents, thank you for encouraging me in all of my pursuits, thanks for the prayers, care and sacrifices you have made for my life.

Declaration of academic integrity

Hereby I declare that I wrote the present work independently and that I did not use any literature sources and aids other than quoted. Passages of the thesis obtained verbatim or in the meaning from other sources are marked and their origin is indicated. This also applies to drawings, sketches and illustrations, as well as to sources from the Internet. If it should be proven that single parts of the thesis/seminar paper have been copied from other works without indication of source, I am also aware that this can have legal consequences.

Freiburg, .02.2020

Engy Nasr

Zusammenfassung

Aufgrund der exponentiellen Zunahme unterschiedlicher Informations- und Kommunikationstechnologien stellt die Verwaltung heterogener Datenmengen und -arten zukünftig eine große Herausforderung dar. Strategien für den effizienten Umgang mit diesen Datenmengen befinden sich noch in der Entwicklung und bedienen sich Konzepten der Logik, der Semantik und der Computerwissenschaften, um so Schlüsse aus heterogen strukturierten Daten zu ziehen. Ontologien stellen dabei eine vielversprechende Lösung als sehr aussagekräftiges Wissensrepräsentationssystem mit starker Semantik dar. Dementsprechend werden Ontologien von Institutionen gemäß ihrer Bedürfnisse entwickelt und etabliert, was zu zahlreichen Ontologieansätzen für spezifische Domänen von unterschiedlichsten Akteuren führt. Um die Interoperabilität zwischen Akteuren und Systemen zu ermöglichen, ist das Mapping von Ontologien untereinander von entscheidender Bedeutung. Eine Möglichkeit, ein solches Mapping zwischen verschiedenen Ontologien herzustellen, ist Ontologie-Matching, bei dem tatsächliche Beziehungen und Zuordnungen zwischen Entitäten unterschiedlicher Ontologien gefunden werden.

Nach Huschka und Dlugosch hält die Digitalisierung „zunehmend Einzug in das klassische Ingenieurwesen und gerade im Bereich der Werkstoffwissenschaften zeichnen sich bahnbrechende Innovationen durch moderne Methoden der Datenverarbeitung ab.“ [1] Viele automatische Ontologie-Matching-Tools (Matcher) wurden durch die „Ontology Alignment Evaluation Initiative (OAEI)“ für die Domänen der Bioinformatik und Biomedizin eingeführt, evaluiert und befürwortet. Bisher wurden jedoch noch keine Evaluierungen für das automatisierte Ontologie-Matching (Evaluierungsworkflows) für die Materialwissenschaft und Werkstofftechnik (MatWerk) vorgenommen. Es gibt folglich keine verfügbaren Benchmarks mit materialwissenschaftlichen oder werkstofftechnischen Ontologien, um die bereits vorhandenen Ontologie-Matcher im Hinblick auf das MatWerk-Feld zu bewerten. Die Benchmarks anderer Domänen können nicht für die Materialwissenschaft und Werkstofftechnik verwendet werden, da sich die in den Ontologien repräsentierten Konzepte mit der Domäne unterscheiden. Daher ist der in dieser Arbeit erstellte MatWerk-Benchmark, inklusive Test-Cases, Referenz- und unterstützender Ontologien, sowie die Durchführung von Evaluierungen der vorhandenen Ontologie-Matcher von entscheidender Bedeutung für die Anwendung von automatisiertem Ontologie-Matching in der Domäne der Materialwissenschaft und Werkstofftechnik.

Die Evaluierungsergebnisse haben ergeben, dass die Verwendung des Ontologie-Matching-Tools „LogMap“ für Anwendungen empfohlen wird, die hohe Präzision der Zuordnung erfordern. „Agreement Maker Light (AML)“, ein weiteres Ontologie-Matching-Tool, beinhaltet einen Background-Knowledge-Ontologie Matcher, der unterstützende Background-Knowledge-Ontologien verwendet und so die Anzahl der korrekten Zuordnungen durch das Tool drastisch erhöht. Folglich ist eine ausgereifte Background-Knowledge-Ontologie (semantische Brücke) der Materialwissenschaften und Werkstofftechnik essentiell, um den Anteil korrekter und vollständiger Matching-Ergebnisse zu maximieren. Die Verwendung des Property-Matchers wird nicht empfohlen, bis dieser angepasst wurde, um logisch korrekte Zuordnungen auszugeben. Letzten Endes wird vollautomatisches Ontologie-Matching nicht erreicht, da alle gefundenen Zuordnungen nur „Äquivalenz“ repräsentieren und die Zuordnung zu unter- oder übergeordneten Klassen durch die Matcher nicht vorgenommen wird. Folglich sind noch Eingriffe des Menschen erforderlich, da die Ergebnisse nicht zu 100 % korrekt und vollständig sind.

Abstract

A challenge of managing various amounts and types of data has raised due to the exponential increase in information and communication technologies. Creating strategies for handling those exponentially increased amounts of data using logic, semantics and computers, allowing reasoning between various data entities, is still under progress. Ontologies present a promising solution as a high-level knowledge representation system with strong semantics. Accordingly, institutions develop and establish ontologies according to their needs, resulting in various ontologies created for a specific field by different actors. To enable interoperability between actors and systems, mapping between ontologies is crucial. One way to establish a mapping between different ontologies is ontology matching, where actual relations and alignments between entities of different ontologies are found.

As Huschka and Dlugosch have stated, "Digitization is increasingly entering classical engineering disciplines. Especially in the field of materials science, pioneering innovations are characterized by modern methods of data processing". [1] Many automatic ontology matching tools (matchers) have been introduced, evaluated and approved by the "Ontology Alignment Evaluation Initiative (OAEI)", recently in the field of Bioinformatics and Biomedical domains. However, until now, there have not yet been evaluations performed for automatic ontology matching (evaluation workflows) for Materials Sciences. Consequently, there is no available benchmark for Materials Sciences ontologies to evaluate the already existing ontology matchers in terms of the Materials Sciences domain. Other domains' benchmarks cannot be used as Materials Sciences benchmark due to the different concepts represented in different domains' ontologies. Therefore, in this thesis, creating a Materials Sciences benchmark with test cases, controls and supporting ontologies, and performing evaluations of the existing ontology matchers are crucial for the application of automatic ontology matching in the Materials Sciences domain.

The evaluation results have shown that the usage of "LogMap", an ontology matching tool, is recommended for applications that demand high precision. "Agreement Maker Light (AML)", another ontology matching tool, provides a background knowledge ontology matcher, which uses the supporting background knowledge ontologies and boosts the number of correct alignments found by the tool. Accordingly, a mature Materials Sciences background knowledge (semantics bridge) ontology is essential in getting more correct and complete alignments. The usage of the property matcher is currently not recommended until it is improved to include logically correct alignments. Finally,

the fully automated ontology matching is not achieved since all alignments are only equivalences and no sub or super alignments are detected; accordingly, human interference is also needed because the results are not 100 % correct and complete.

1 Introduction

This year, 2020, the worldwide collected data will reach a size of more than 50 zettabytes, (10^9 TB), due to the exponentially increased data collection happened in the last decade. [2] These huge amounts of data are becoming so hard to handle by humans without attempting unintentional errors during storing, transferring or merging, given the fact that over 90 % of these data were not structured correctly in the last years and 80 % are not prepared for reuse. [2–4] So converting these data to information and finally to knowledge is crucial. That requires the presence of computer handling of these data, which will guarantee efficiency in data processing, a noticeable decreased processing time, achieved reusability and interoperability.

Interoperability is when systems, actors or applications provide/accept services to/from other systems, actors or applications and collaboratively use the exchanged services. [5] This could be achieved by semantically arranging the information, conventionally deduced from the data, into ontologies. [6] Using these created ontologies in scientific fields requires highly achieved interoperability between ontologies of the same or similar field. So implementing technologies to find relations between ontologies' entities to achieve interoperability is very important. This can be done manually, but same problems of attempting errors, time consumption and complexity, especially in large-sized ontologies, are faced. That is why many automatic ontology matching tools exist, which find relations between ontologies, enabling interoperability between them. Although these tools are yearly evaluated, updated and adapted to the fields of biomedicine, bioinformatics and medicine, they are not adapted to the Materials Sciences field, since the field is still developing in terms of ontologies' creation. From this point comes the thesis approach presented in this chapter's next section. Nevertheless, the structuring of the thesis will also be shown briefly along with every section's content.

1.1 Thesis Approach

The thesis' main goal is to evaluate automatic ontology matching techniques for Materials Sciences and Engineering. In order to do so, the existing evaluation workflow for automatic ontology matching from domains such as the Bioinformatics and biomedicine are adapted to the Materials Sciences domain.

To this aim, an ontology matching benchmark for Materials Sciences is developed that consists of:

- A set of test cases including Materials Sciences ontologies.
- A manual reference alignment for every test case, which will be needed to evaluate ontology matchers.
- Materials Sciences' background knowledge ontologies, which support some matchers to adapt to the Materials Sciences domain and yield better results.

Moreover, a study on existing automatic ontology matchers is conducted; revealing those that fit the requirements for matching the concepts in Materials Sciences ontologies. Finally, the performance of selected matchers is assessed by adapting the evaluation workflow using the developed benchmark.

After the evaluation, the results provide fundamental data to choose the desired matcher for applications in Materials Sciences. Furthermore, the developed Materials Sciences benchmark can be used and adapted to perform other types of ontology matchers evaluations, e.g. with additional ontology matchers or a differing evaluation workflow. A detailed discussion of the results is provided including suggestions to improve the performance of the matchers in Materials Sciences.

In the thesis, the term Materials Sciences is used as short for Materials Sciences and Engineering as well as the terms ontology matching and ontology matchers are used as short for automatic ontology matching and automatic ontology matchers respectively.

1.2 Thesis Structure

The structure of this thesis is as follows:

In Chapter 2, the related work for the thesis is presented, where similar work approaches are tackled to show their similarities and differences to the thesis' approach, and what of these similarities are adapted to the thesis.

In Chapter 3, the theoretical background of the thesis is discussed as a prelude showing the importance of the thesis approach. As a start, the materials modelling concept by RoMM [7] is presented in Section 3.1. In Section 3.2, ontologies and interoperability are discussed in detail, explained along with the semantics spectrum, examples of upper level ontologies and successful interoperable ontologies. Following, Materials Sciences ontologies used in the thesis will be presented in Section 3.2 as well. In Section 3.3, principles of the ontology matching will be discussed. Finally, in Section 3.4, the concept of ontology matchers' evaluation will be explained.

In Chapter 4, methods used to execute the thesis' approach are presented. In Section 4.1, the matchers chosen and prepared to perform the Materials Sciences benchmark alignments are discussed. Then the benchmark development is presented in Section 4.2. In Section 4.3, the performance evaluation schema created, which are adapted measurements to evaluate the performance of the ontology matchers, will be explained. The performance evaluation workflow for the matching of the benchmark and the implementations, where the performance evaluation schema is applied on the benchmark, will be also explained in Section 4.4.

In Chapter 5, the thesis' results are shown and discussed. In Section 5.1, the results of matching the test cases by all the ontology matchers are presented and discussed. The evaluation results of the matchers using the manual reference alignments created for the Materials Sciences benchmark are shown, compared and discussed in Section 5.2.

Finally, in Chapter 6, the conclusion and suggested future work will be presented.

2 Related Work

The ontology matching term has risen as ontologies were recommended by the "World Wide Web Consortium (W3C)" in 2004. [8] Since then, ontology matching has been a global interest to most industries, research and scientific fields. [9] This interest led to the development of automated ontology matching tools. Consequently, the evaluation of automatic ontology matching tools became a crucial topic, which is the main interest of this thesis. Although the most systematic evaluations of ontology matchers were established the same year as the ontology matching term was introduced, these evaluations have only been applied to the biomedical, bioinformatics, medical and anatomy fields. The "Ontology Alignment Evaluation Initiative (OAEI)" [10] has established benchmarks, evaluation workflows, as well as yearly evaluations of ontology matching systems. The thesis targets the same idea of evaluating automatic ontology matchers but for the Materials Sciences field, where no evaluations have been set, applied or designed. Consequently, no Materials Sciences benchmark has been created yet. Accordingly, for the thesis, adaptations to existing evaluation workflows have been made. In 2004, J. Euzenat and M. Ehrig [11] introduced evaluation schemas for general ontology matching that are used by the OAEI in their yearly evaluations. Nevertheless, ontology matching workflows continued to develop by J. Euzenat and P. Shvaiko [12].

There is no peer work related to the thesis with respect to the Materials Sciences benchmark; however, the OAEI's evaluation schemas and workflows have been used. T. Ashino [13] has mentioned the importance of ontology matching for the Materials Sciences field; he also created a Materials Sciences ontology that has been utilized for the thesis' benchmark. The projects under the "European Materials Modelling Council (EMMC)" [14] have taken into considerations the importance of interoperability to the Materials Sciences field. As a result, the "European Materials Modelling Ontology (EMMO)" [15] has been created, which is an upper and a mid level ontology designed to support interoperability of Materials Sciences ontologies. EMMO has also been a part of the thesis' Materials Sciences Benchmark. MatOnto [16] is a Materials Sciences domain ontology basing on an upper level ontology, which supports information for new Materials Sciences research, the ontology has also participated in the thesis' benchmark. For the ontology matchers that participated in the thesis evaluations, many tools have been tested, which are the same as the ones found in the OAEI history of participated ontology matching tools [17]. Examples of these tools are the "Agreement Maker Light (AML)" created in 2013 [18] and the "Logic-based Methods for Ontology Mapping (LogMap)" [19].

From these similarities of the related work, which have not yet been utilized to the Materials Sciences field, the thesis approach is set. Next chapter, these similarities of the ontology matching workflows and tools evaluated towards other scientific fields are explained. In the Methodology chapter, the adjustments made to the related work similarities to serve the thesis approach targeting the Materials Sciences domain are presented.

3 Theoretical Background

In this chapter, the concepts behind the thesis approach's implementations starting from a view on the materials modelling world to the existing evaluation for the ontology matchers will be presented. The chapter starts with presenting the materials modeling approach by the "Review of Materials Models (RoMM)" [7], which is chosen as an interesting view on the field of Materials Sciences, and an interesting solution for many Materials Sciences industries. RoMM and materials modelling are both explained to serve in understanding the complexity of the Materials Sciences field in Section 3.1. In Section 3.2, the ontologies importance, development, types, examples and evolution as well as interoperability will be explained, followed by the combination between Materials Sciences and Ontologies, and discussing the existing Materials Sciences ontologies in the same section. In Section 3.3, the ontology matchers' tools and the idea of ontology matching will be presented. Finally, the evaluation measurements, which exist in the evaluation workflows designed for evaluating ontology matchers, are shown in Section 3.4.

3.1 Materials Modelling

In material researches, modelling is the mathematical representation of a material's behavior under certain applied actions and conditions. Modelling can be a very robust tool to develop new or to improve existing materials, components, structures or applications. Modelling is highly demanded in industry as it provides insights that experiments cannot provide, by reducing costs, complications, and risks concerning time and money that come as result of the "try and error" experiments. It also reduces time and allows the close look to fine details of a material hardly seen in an experiment, at a nanoscale or even at a femtoscale level of accuracy. Decision-making in R&D departments often entails extensive guessing and verifying activities, based on infinite possible paths, but thanks to materials modelling, it is possible to identify the most promising paths and focus the company's energies on a few attempts. [7]

In RoMM 6, the Review of Materials Models version 6, which is established by the "European Materials Modelling Council (EMMC)", funded under the H2020 NMBP (Nanotechnologies, Advanced Materials, Biotechnology & Advanced Manufacturing & Processing) Programme, a common language is set to guide modelers in creating their materials models. Choosing such a common language is a huge base for an interoperable ontology to build on. Without this common language, unclear expressions are faced by modelers describing their material use case, for which defined concepts and vocabulary are needed for models'

simulation, and as Thomas Davenport said, “People can’t share knowledge if they don’t speak a common language”. [7]

For a material to be described in a model, it needs a lot of complexity whether of the system or of the equations describing its physical or chemical behaviors. However, humans and the computers are lucky, as not all these complex details are needed to describe an experiment. According to Anne F. de Baas [7], only two main parts form a model, a “Physics/chemistry Equation (PE)” and a “Material Relation (MR)”, which describes a specific case of a certain material and its behavior. These two parts, shown in Figure 3. 1, forming a model, are also called a “governing equation”. Earlier, models were identified by their length, time scale and applications. But nowadays this type of identification is changed such that the model is identified by the entity whose behavior is described in the PE, and that is due to the growth of the application areas making them overlapping and hardly be taken as an identifier to the models. The entity identifying the model, according to A. Baas [7], can be either electrons, atoms, molecules/nanoparticles or a continuum. Modelling on the electrons level serves high accuracy but at the same time, it is computationally demanding. Modelling a continuum might be less accurate but computationally not as demanding. From that, the four types of the materials models are defined to be chosen according to the application’s level of accuracy. [20]

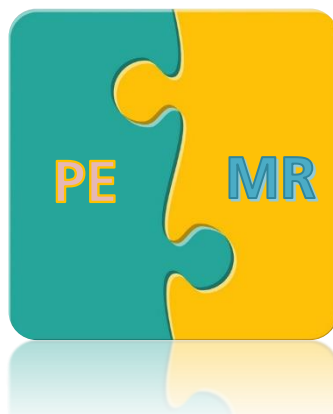


Figure 3. 1: A RoMM Review of Materials Models [7] which is a combination between a Physics/chemistry Equation (PE) and a Material Relation (MR).

3.1.1 The Four RoMM Materials Models

First, the electronic model, which describes the behavior of electrons and quantum particles, is based on physical/chemistry equations such as Schrödinger equations that describe the wave function of electrons as quantum mechanical waves. The output of this model can be a chemical reaction coefficient or force field parameters for atomistic models, etc. [7]

Second, the atomistic model, which starts from a level above the electrons' level, is the model that looks at the behavior of the atoms, and ignores the electrons degree of freedom. Many physics and chemistry equations can be used to describe the atoms' nature, behavior and relations, such as classical mechanics, Newton dynamics, interatomic potentials, etc. When a model uses these related equations, it might be less accurate than the full quantum mechanics in case of electrons, but it has faster simulation schemas, allowing us to represent models that are more complex. The result from this model can be heat transfer, surface and interface energies, etc. [7]

Third, the mesoscopic model, which is considered frozen, is the most unused model for the projects that are done under the NMBP Programme funding in the FP7 (2007 - 2013). That is because this model is used to describe the behavior of nanoparticles, part of molecules or grains. It does not tell exactly whether it looks at an electronic level, an atomic level, a molecular, or even a higher level, which is larger in volume. Since the scale of the material is no longer an identifier for a model, the mesoscopic model is put under more research studies to be developed. The output of that model can be like the magnetic behavior or the thermal stability of a certain model, etc. [7]

The fourth and last RoMM model is the continuum model, the one that is highly used in the NMBP Programme projects performing modelling. This model describes the continuum in a finite volume. It is very useful to every scale it is applied to. For example, if this model is applied to a micro scale phenomenon, it can predict the material decomposition, the defect formation, the crack propagation and the solidification of liquids as a result from the model. Nevertheless, if it is applied to a macro scale phenomenon, it can describe as a result the behavior of a thin film and realistic nano devices with metallic contacts and other valuable variables for the industry need for manufacturing. [7]

3.1.2 Solvers and Multi Scaling

Every model has two main connected parts of a PE and a MR, the equation along with the specific case of the material, which have to be solved together. In that case, a solver is needed, which will solve this whole model and get out the results of the model introduced above. The solvers are numerical methods that may become very complex and computationally heavy. Examples of these solvers are the “Smoothed Particle Hydrodynamics (SPH)” solver, used for the fluid dynamics model equations, the Finite Elements solver used in solving some continuum models, the Monte Carlo solver, and others. [7]

One model on its own is a complete system, but the main idea or the focus of industry and R&D is to have multiple models working together in more than one way in a simulation. This combination of models, giving properties close to the measurements of a specific material better than the information given by a single model, is called multi scaling. A multi scaling workflow is a chain of models linked together either sequentially or concurrently. The sequential multi scaling workflow is called linking, in RoMM vocabulary, and that is when a whole model gives its output to the next model and so on. The other type, which is the concurrent multi scaling workflow, is called coupling in RoMM. The two types of coupling that exist are either iterative coupling or tightly coupled, as shown in Figure 3. 2 and Figure 3. 3.

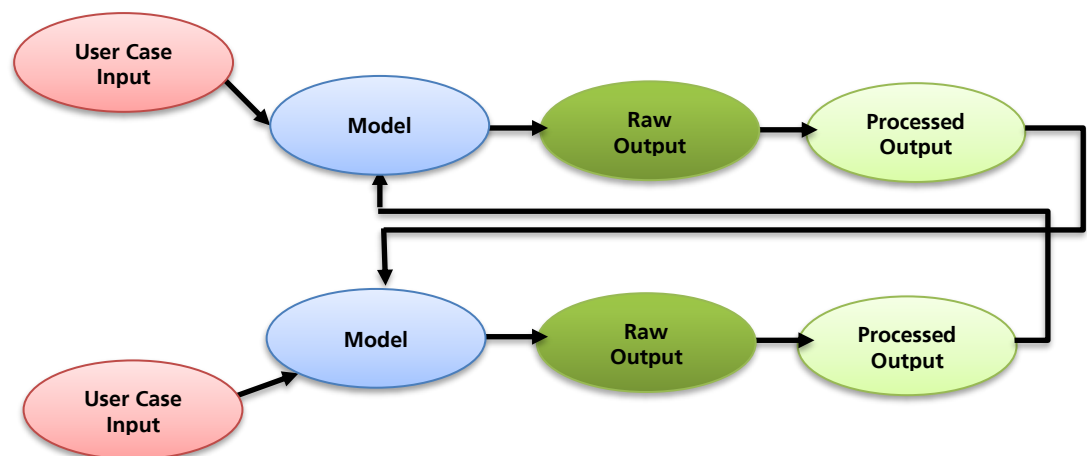


Figure 3. 2: A coupling or a concurrent multi scaling workflow of an iteration type, defined in the Review of Materials Models (RoMM) as a way of forming a chain of materials models, such that each model's processed output is given to the other model.

In Figure 3. 2, the multi scaling workflow is shown as a loop or an iteration, such that each model's processed output is given to the other model. Each model has its own user case input, e.g. values to certain equation's variables, and then the model's raw output is processed to be used as an input to the other model. A user case is given behaviors and properties of a specific material during a manufacturing process or a behavior simulation [21]. The other type of coupling in Figure 3. 3 is the tightly coupled multi scaling workflow, where the models work together as a complete system acting as a single model, then give their combined output, raw output, to be processed afterwards the same way as the linking or sequential multi scaling workflow works.

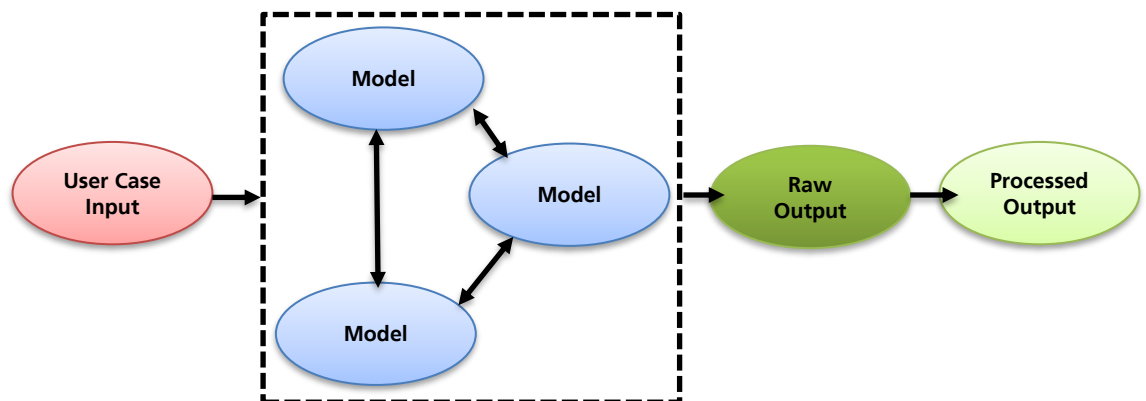


Figure 3. 3: A coupling or a concurrent multi scaling workflow of a tightly coupled type, defined in the Review of Materials Models (RoMM) as a way of forming a chain of materials models such that all models are given the same user case input, then work together to output one combined raw output to be processed afterwards.

Each model is solved by a solver and its output is processed by a post processor, this processed output can be a preprocessing for a next model, as seen in the previous workflows shown in Figure 3. 2, when the processed output from one model is given to another model. The post processor can be tools to visualize the data, a convertor to the output from one unit into another, or an extractor that can extract some datasets from the total output, like averages. In all cases, a post processor does not change a state of a model as the PE does.

The main concern of RoMM is to unify the language between modelers, it is also important to unify the way a simulation or a multi scaling workflow of models will be documented, for example representing a certain user case. "Modelling Data elements templates (MODA)" is a guide established in RoMM, a step by step of how to document a multi scaling workflow, a standardized

description to be used in all materials models created within the European Community. [14]

This common documentation will help increase information of the description modelling of a project. Figure 3. 2 and Figure 3. 3 are shown in the MODA format of a multi scaling workflow description.

3.2 Ontologies and Interoperability

Having MODA, every model has a common way in describing the user case. To ensure the clear understanding of this description, have meaningful information for both computers and humans and allow the exchange between models, metadata is needed, which are data describing and giving information about another data [7]. Metadata can be also used to validate the correctness of data by checking the symmetry between data types, units, etc., of different models.

From what is explained about materials modelling and how much combinations can be obtained from equations, materials relations, solvers, workflows and user cases, it can be imagined how fast the scientific data and information are growing. A huge flow of new scientific information is produced daily due to research, experiments, technologies, discoveries and many other things, making it very hard for humans to manually handle these huge amounts of information alone in an efficient way. A strategy has been developed in order to ensure the integration of all related information already existing and daily added to the scientific world, making it understandable for both computers and humans. This strategy where information is enriched with semantics and cognitively turned to knowledge, a subset of all true beliefs, is called “ontology”.

The semantics is the study of the meaning of the language; linguistically and philosophically, where logic is used to define and relate entities together. Ontologies by now have the highest semantics level as going to be explained in Section 3.2.2; and they are defined as follows.

An ontology is a computer and human understandable knowledge organization system, and according to the computer science definition [22], ontology is a defined machine understandable specification of shared concepts (models). Given a specific domain or a certain scientific field, wanting to represent many types of entities and their relations together, a controlled vocabulary by an ontology is to be used as set of rules to follow in order to create these entities. In philosophy, an ontology is defined to be the study of entities present in reality and the study of relationships that holds in between all these entities. [3]

Recently, the term ontology has become very popular in the field of computer and information science, and has achieved a noticeable success in areas of bioinformatics. [3] Simply, an ontology is a representational artifact that bases

on a taxonomy, combining universals, rules, defined classes and relations between them, such that a taxonomy is a hierarchal representation of entities, anything that exists, with a relation “is_a” between nodes on different hierarchal levels.

Imagining that everyone, every research organization or an industry decided to build their own ontology related to their field of study on their own way. Each ontology is created with different entities’ names, definitions, sub and super entities, relations, etc. Then searching all the present ontologies related to a specific field of study may yield what is needed, however in very different meanings that might be close, far or even contradicting to one another. For such a problem, interoperability is crucial, unifying terminologies and formats between all common fields’ ontologies, ensuring the achievement of FAIR data principles [23] (Findable, Accessible, Interoperable and Reusable) is essential. [24] With interoperability, the retrievability, exchangeability, editability and reusability of data between ontologies are guaranteed. What is the big use of an ontology if it is not malleable for updating, correcting or merging? For that, terms are needed to be understood and common enough between the same fields of studies.

In “Building Ontologies with Basic Formal Ontology” [3], some rules and principles were set for all ontology developers to follow in order to achieve the interoperable ontology.

3.2.1 How to Create an Interoperable Ontology

According to B. Smith [3], going through these following five main steps is essential in order to achieve such an interoperable ontology.

First, decide on the domain specific ontology or the exact field of study for which the ontology is created.

Second, gather information from already existing resources in order to make the ontology as much interoperable as possible; find standard textbooks and the already existing ontologies to identify the general terms, remove redundancies and select the highly common general terms that are relevant to the ontology to use, instead of defining new terms, which might be different from the already existing known ones.

Third, start forming the taxonomy, where terms are arranged in a hierarchal way like a tree with branches each with its hierarchal level, then join them with parent-child “is_a” properties, as shown in Figure 3. 4.

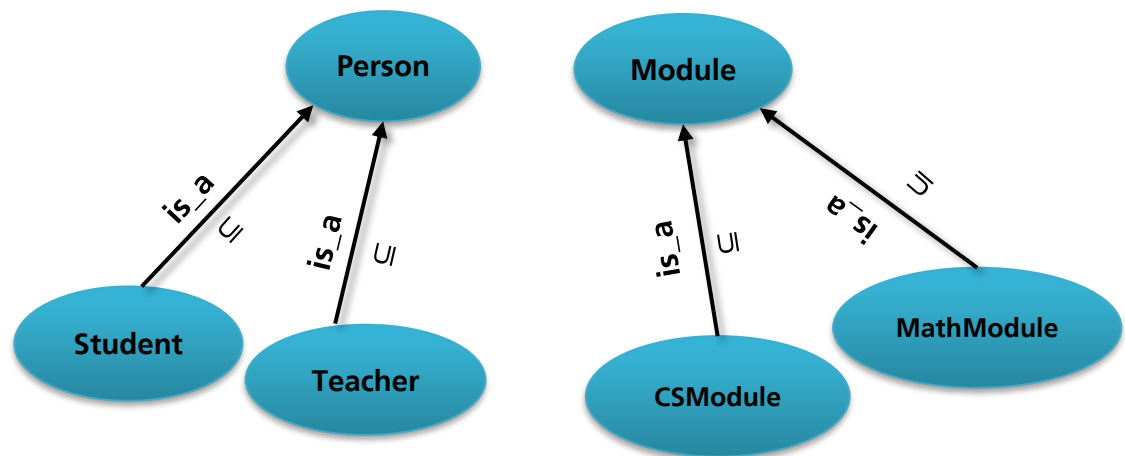


Figure 3. 4: A simple example, part of University Taxonomy, showing different classes being super/sub classes of each other, with a relation “is_a”, shown on solid arrows, between classes to show the hierarchical levels of the classes.

Fourth, turn the taxonomy into an ontology; imagine a spider web, by regimenting the results of the taxonomy, adding more specified properties, non “is_a” properties, between all related classes and ensuring that the hierarchy is logical and scientifically coherent. For example, in Figure 3. 4, the main root nodes in the tree hierarchical structure of the taxonomy are classes called “Person” and “Module”, which are super classes for other classes underneath them in the following tree structure level respectively. To turn this taxonomy into an ontology, instead of “is_a” relation, a “subclass_of” and other descriptive relations are used. As in Figure 3. 5, two types of relations/properties can be included, data properties (shown on solid arrows in the figure) relating classes (blue ovals) to individuals or instances (orange rectangles) of numerical (e.g. integers and floats), text (e.g. strings), date, time, etc. data types or object properties (shown on dotted arrows in the figure) relating classes to other classes.

Nevertheless, also ensure the compatibility with similar and related other ontologies, in terms of entities (classes, properties/relations and individuals), definitions, etc. Finally, ensure human understandability, specially the parts where the entities are defined, such that until today humans are still the end-users of the created ontology so it will always have to be understandable for both computers and humans.

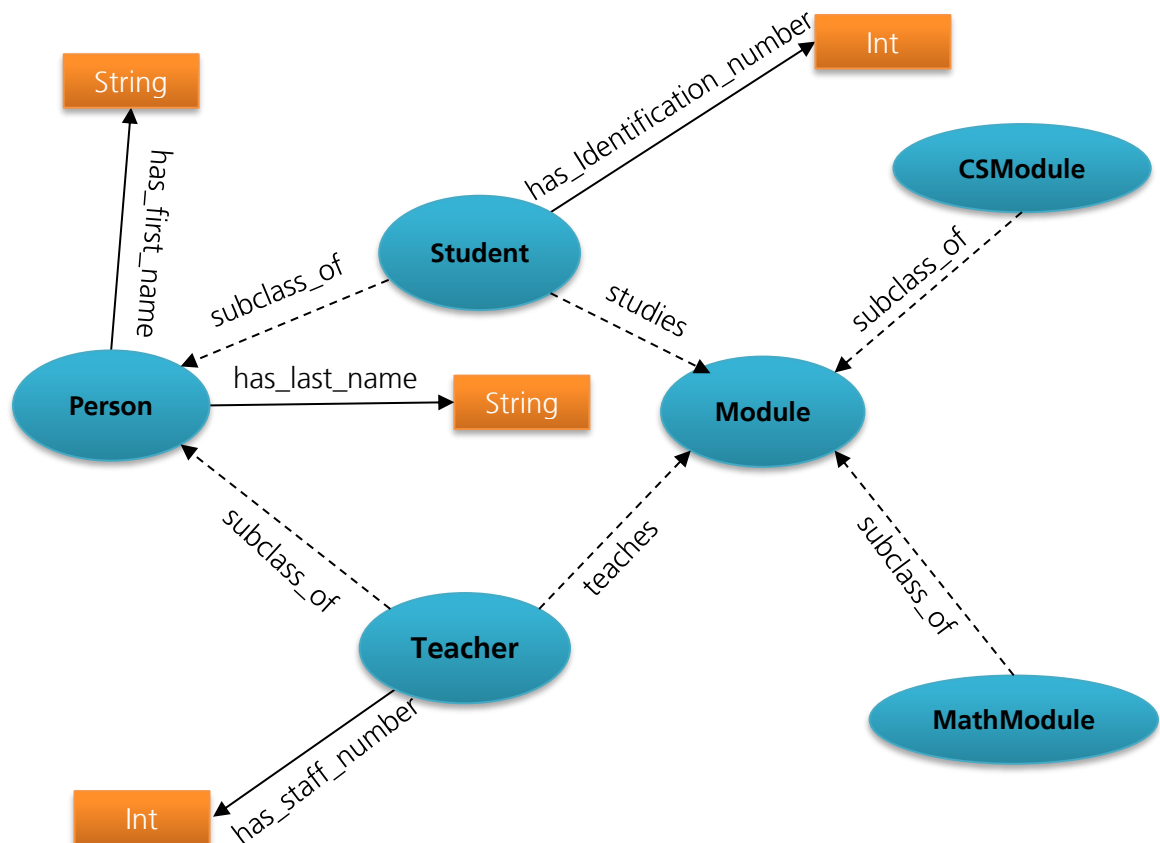


Figure 3. 5: The ontology created from the University taxonomy presented in Figure 3. 4, showing the data (orange rectangles) and object (blue ovals) properties between (class – individual) in solid arrows and (class – class) in dotted arrows, respectively. [25]

Fifth, formalize this regimented representational artifact, or in other words, the ontology created, into a computer language. Where the natural language definitions for all the entities and terminologies chosen for the ontology are transformed into a computer understandable language (e.g. "Web Ontology Language (OWL)" as in the example below), such that the ontology becomes editable and can be used in building applications.

An Example of OWL, as going to be explained in the next section, presenting one object property from Figure 3. 5 between class Module and class MathModule is shown as follows:

```

<owl:Class rdf:ID="MathModule">
  <rdfs:subClassOf rdf:resource="Module" />
</owl:Class>

```

3.2.2 The Semantics Spectrum and its Computer Languages

Since the last and final step in creating an ontology is to convert it from the humans' natural language into a computer understandable language, the discussion of the software to use and the computer languages available and which to choose is important.

The computer languages semantics spectrum, from the text based representation to logical based semantics representation, is shown in the semantics spectrum in Figure 3. 6. Semantics is when agreements about syntax meaning are set. [26].

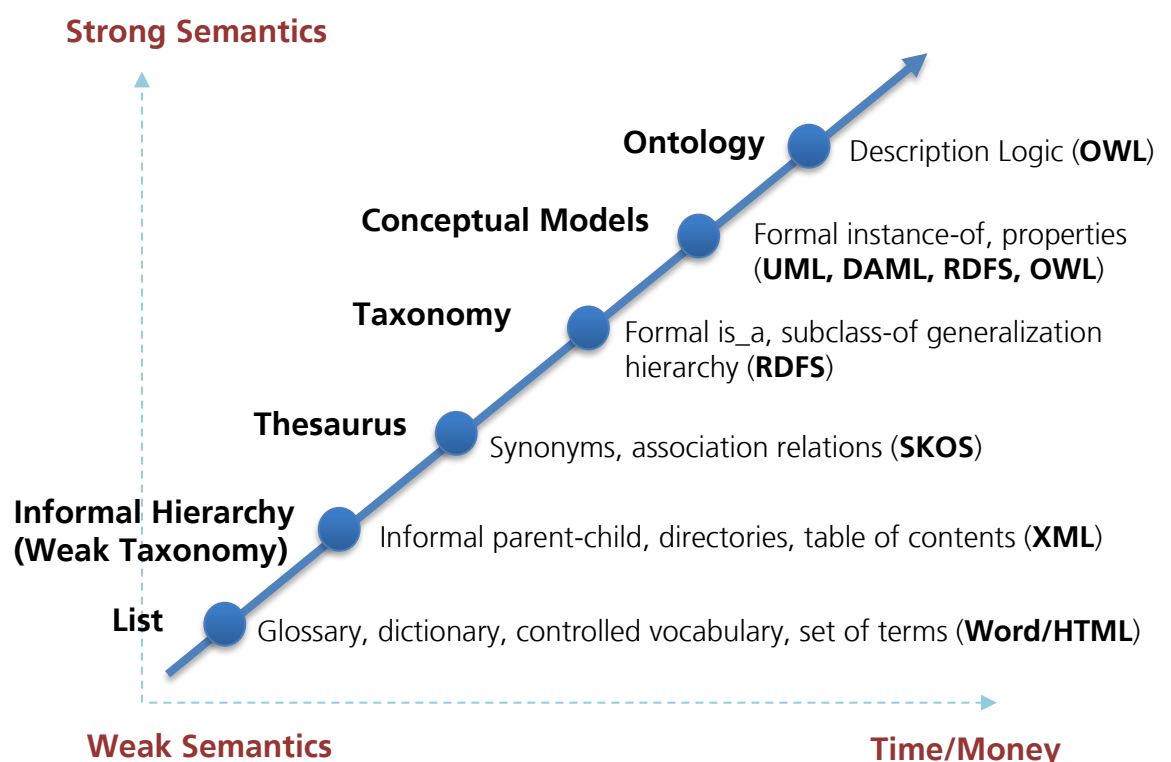


Figure 3. 6: The semantics spectrum of Knowledge Organization Systems [21,27], representing the data handling in terms of resources, e.g. time and money, starting with the lists with the lowest semantics until the highest semantically represented data in terms of ontologies. The highest the semantics level, the highest the time and money needed.

One of the ontology's main goals is to achieve the highest level of semantics or logic, such that the computers can be easily used to reason about the data. Nevertheless, the time and money needed to achieve this highest level of

semantics have to be taken into consideration, since the highest the semantics level, the highest the time and money needed.

In the late 1980s, "Hypertext Markup Language (HTML)" was designed to represent the data on a web page but just the humans can understand or read from the web browser. The computers cannot further process any of these data, such as finding, validating, interpreting or combining. The data is only represented as a list of words and numbers with syntax defining the meaning of the list's terms, using only common symbols and concepts.

In 1998, "Extensible Markup Language (XML)" was then recommended by the W3C (World Wide Web Consortium), adding a one level higher semantics to the HTML allowing the machines to also read and interpret. This is done by utilizing tags. However, it did not include relations between entities, which is why it is considered a weak taxonomy or an informal hierarchy, since only arrows are used to differentiate between parents and children without any description of these arrows. Then, the thesaurus comes after, where the arrows are not only differentiating parents and children but also include synonyms of every entity and further relations. But still another level of higher semantics is needed.

In 1999, the W3C recommended "Resource Description Framework (RDF)" along with the "RDF Schema (RDFS)", allowing simple relations between different types of entities to be represented. These relations arrange the entities in a hierarchal structure forming a taxonomy opening a way for reasoning and adding a higher level of semantics, where the concepts are classified and relations are added to them. However, these are simple relations such as "is_a", "subclass_of", etc. still not expressing types and instances belonging to types, as well as properties or relations, which is a higher level of semantics.

In 2004, OWL was recommended by the W3C, which came as a higher level of semantics in terms of relations, software reasoning, adding more vocabulary along with formal semantics. Rules and knowledge are set to allow meaningful defined relations. OWL uses the term "properties" referring to relations between classes and individuals, such that classes, individuals and properties are all called data "entities". The language bases on the "Description Logic (DL)", a subset of the "First Order Logic (FOL)", one of the main tools to reason with many logical expressions relating entities such as universal (\forall) and existential (\exists) quantifiers, Boolean, Min and Max, etc. DL is used instead of the FOL, as the FOL will need more time to be executed, while DL requires less time and has most of the reasoning needed in building an ontology. [21]

In 2009, W3C recommended OWL 2, which came to meet the challenges that came up when OWL 1 was used. One main challenge is the inability to go deeper in relations. For example, in OWL 1 one can say that someone has four dogs, but one cannot say that he/she has four dogs, two of which are boxers. Also in OWL

1, the barometric pressure is “particular_part_of” the atmosphere has_value of 1,000 millibars, but one cannot say that this value is more than 900 and less than 1,100 millibars. These challenges are addressed in OWL 2. [3]

Prof. Dr. Harald Sack has illustrated the evolution of the spectrum in [26], along with an explanation for the transformation from data over information into knowledge. He also explains the importance of the semantics and why the logic is needed. One of the examples is, “33.6” is a number, with no reference to tell what this number is. It is just data that needs to be understood. By adding “m” to the number, “33.6 m” the data “33.6” can be understood as length in meters, that is the information, by adding the commonly known symbols and concepts. Then, by adding cognition to information, it is converted to knowledge. This can be achieved by having some relations as in the Description Logic (DL) represented as follows, $BaleanopteraMusculus \sqsubseteq Animal \sqcap \forall maxLength. \leq 33.6 m$. Now reasoning that the “33.6 m” is the maximum length of the Baleanoptera Musculus, one type of whales, is possible. Semantics is an agreement on the meaning by having the knowledge or the set of all true beliefs. This data, information and knowledge idea are presented in a “Data Information Knowledge Wisdom (DIKW)” model by Danny P. Wallace, where the last letter “W”, the wisdom, is expected to be achieved by enabling the ability to predict and take decisions and actions accordingly. [6,28]

3.2.3 Upper Level Ontologies and the BFO

As discussed before about semantics and interoperability, data is converted into a human-computer understandable language, which is malleable to exchange between systems, actors or applications of the same domain. Ontologies enable, as a common language used, storing data semantically. In order to ensure semantics and interoperability, a formal/ an upper level ontology has to be chosen as a starting layer for the domain specific ontology. Unifying the upper levels of same domain specific ontologies will result in more common classes and strategies used in building the domain ontology. Figure 3. 7 shows the ontology layers presented in the “Theory and Applications of Ontology”. [27] The idea is that each sub ontology presented in Figure 3. 7 is considered an ontology on its own, and its type is determined by how general this ontology is to the domain. For example, the ontology that represents classes or terms that are commonly used, such as location and time, is considered an upper level, upper or in another word formal ontology. Then comes the middle or a mid-level ontology, that does not have a strong difference between it and the upper ontology, it is still general, but it specifies the terms introduced in the upper ontology more. Upper and mid-level ontologies are both considered as an upper level ontology or a formal ontology in some text books [3], with no differences, as general terms and relations defined for a certain domain to be used on top of the domain’s specific ontologies. An ontology can be built by several upper, mid-level and domain

ontologies. To achieve this, ontologies need to be matched to each other, as is going to be discussed in Section 3.3.

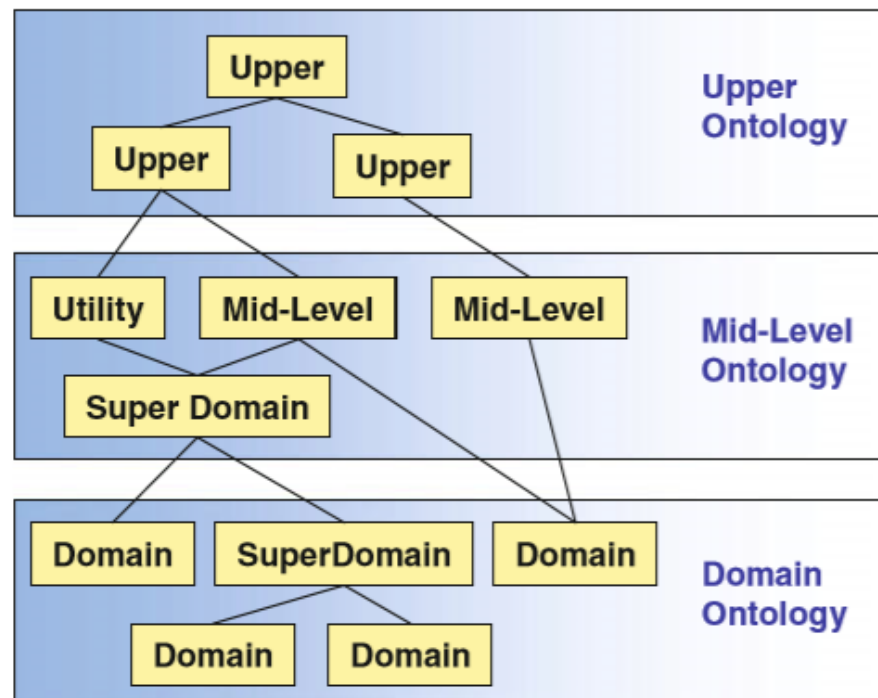


Figure 3. 7: Ontology layers presented in the “Theory and Application of Ontology” [27], showing the three types of sub ontologies, starting from the most general, the upper ontology, the mid-level more specific ontology to the most specific domain ontology.

The upper level ontology, from where the domain ontology is built, represents very general concepts such as time, space, events and so on; independent of a specific domain that has fundamental concepts that specializes terms introduced in the upper level ontology. Many upper level ontologies exist with different perspectives in dividing and describing any matter, such as “Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)”, “General Formal Ontology (GFO)”, “Suggested Upper Merged Ontology (SUMO)”, the KR Ontology, the OpenCyc and the “Basic Formal Ontology (BFO)”.

In this section, the BFO will be discussed briefly, which is one of the upper level ontologies designed for scientifically domain-based ontologies. However, it is not specific such that it can be used by all scientific domain specific ontologies. It is designed small to be consistent with other upper level ontologies used by domain specific ontologies that are also scientific. By that, the BFO can guarantee

cross-interoperability, which is interoperability between multiple domain specific ontologies developed in its terms [23], as well as maintaining interoperability between other domain specific ontologies created in terms of other upper level ontologies. The BFO common terms and relations can be used in different scientific ontologies and at the same time, it is designed to be interoperable with the already existing upper level ontologies such that even the ones that are not created under its terms can still be interoperable with the ones created under the BFO.

The BFO is divided into two main parts, "Continuants" and "Occurrents". The Continuants are entities that continue to exist throughout the whole time [3], while the Occurrents are the process related entities, time dependent events. In the "Building Ontologies with Basic Formal Ontology" [3], the authors describe how an interoperable ontology can be created based on any formal ontology, a set of general steps to follow, and 25 other principles for choosing terms, definitions and relations while creating any domain ontology. Examples of these steps and principles are the five steps explained in Section 3.2.1 to create an ontology. The book also includes all the details regarding every entity in the BFO with many examples.

A lot of scientifically based ontologies, institutional projects and groups have used the BFO in creating their domain specific ontologies. Examples of these ontologies are the "Ontology for General Medical Science (OGMS)", the "Infectious Disease Ontology (IDO)", the "Information Artifact Ontology (IAO)", Gene Regulation Ontology, "Protein Ontology (PRO)", Plant Ontologies and a lot more than that. Examples of the projects and institutions that utilize the BFO in their ontologies are AstraZeneca in Clinical Information Science, U.S. Army Intelligence and Information Warfare Directorate and a lot of other labs and universities.

For Materials Sciences domain ontologies, the BFO is recommended to be used by the Materials community [29], since it is designed for all the scientific domains. However, another upper level ontology can be used, the developing "European Materials Modelling Ontology (EMMO)", as going to be discussed in the following section.

3.2.4 Popular Ontologies and Ongoing Ontologies' Projects

Recently, the term ontology became very usable in the field of computer and information science and very successful in the bioinformatics and biomedical areas. Ontologies such as "Gene Ontology (GO)", IDO, "Plant Ontology (PO)" and a lot more are examples of that success.

The Geno Ontology, or as their creators described it, a controlled vocabulary ontology, is the most successful ontology so far. [3] GO sets this controlled

vocabulary to function as terms ruling the description of the information of the gene products in various models of the living organisms, and that is the key of the ontology success. This controlled vocabulary is now used to describe anything in the same domain, used in literature and appears in published papers. [30]

Other popular and used ontologies are the ones used in the Ontology Alignment Evaluation Initiative (OAEI) [31] yearly evaluations of ontology matchers as going to be discussed later on in this chapter, which are mainly from the biomedical and bioinformatics fields. Examples of these ontologies from the OAEI 2019 are large biomedical ontologies such as the "Foundational Model of Anatomy (FMA)", SNOMED CT, and the "National Cancer Institute Thesaurus (NCI)", anatomy ontologies such as the Adult Mouse Anatomy ontology, disease and phenotype ontologies such as the "Human Phenotype Ontology (HP)" and the "Mammalian Phenotype Ontology (MP)", etc..

The EMMO [14] is an ontology for applied sciences, a successful ongoing upper level and mid-level ontology designed specifically for physics and the Materials Sciences domains. The ontology is under the funding of the EMMC; its main idea is to start building the ontology, unlike other ontologies, from bottom to top, from the scientific application field to the main concepts. One of the ontologies' main goals is to achieve interoperability between all other same domain ontologies, whether these ontologies are built based on EMMO or based on any other upper level ontology. EMMO's goal is to enable the multi scaling modelling, explained in Section 3.1.2, and links them meaningfully using ontologies.

The developing EMMO is written with OWL 2 on Protégé, available on GitHub with a creative commons license as well as an associated Python API. The developers are Emanuele Ghedini from the University of Bologna, Gerhard Goldbeck from Goldbeck Consulting, Adham Hashibon from Fraunhofer IWM, Georg J Schmitz from ACCESS, and Jesper Friis from SINTEF.

3.2.5 Materials Sciences Ontologies

As in any scientific field, the production of data in Materials Sciences exponentially increases over time with a practical concern to store these data to be wisely used over the long-term. To this demand, institutions and funding agencies pressure ensuring the secure exchange and correct storage of data by using semantics. [32] For that, the usage of ontologies in storing Materials Sciences data is crucial.

There are not quite much domain based Materials Sciences ontologies created and openly available for everyone to find on the Web. The reasons are the following: the topic of ontologies / digitization is recently entering the Materials Sciences field. [33] However, scientists aim to increase their data reasoning while

addressing multi scaling for materials modelling using workflows in order to have a clear insight to the functions and behaviors of the materials. [33]

Nevertheless, Materials Sciences is frequently a concern of private industry / applied research, thus they do not submit their information openly available for usage. Ontologies also provide knowledge about the strategic direction of an institution, so submitting this information could cause confidentiality issues for the company.

Example of the Materials Sciences domain ontologies is PLINIUS, which focuses on knowledge about ceramics research. There is also MatOnto [32], which supports information for new Materials Sciences research based on the upper level ontology, the BFO, and aims at providing a common model for exchanging, re-using and integrating Materials Sciences information and experiments. The MatOnto ontology is available on GitHub.

From Japan, there is also Prof. Toshihiro Ashino with Fujita in 2006 and with Oka in 2007, who have performed multiple trial versions of Materials Sciences domain ontology. [13] They did so by adapting multiple Material Sciences databases such as “National Institute of Advanced Industrial Science and Technology (AIST)”, “National Institute of Materials Science (NIMS)”, and the materials data schema defined by MatDB, the ontology is a typical bottom up ontology with no base of any upper level (formal) ontology. [13] The final version of their Materials Sciences ontology is named ASHINO’S ontology later on in the thesis.

3.3 Ontology Matching

Ontologies can be used in many applications such as information integration and knowledge management. In this thesis, one of the main ontologies’ applications [12,34] is targeted, which is ontology matching, where alignment / correspondences between ontologies’ entities (classes, properties and individuals) are found. [12] Accordingly, an alignment is defined to be a set of correspondences between entities (e) mostly classes of matched to be ontologies.

One of the popular applications that needs matching is merging, two separate ontologies are merged to become one ontology that includes all the knowledge from the initial two ontologies [35]. For example, integrating data from different Materials Sciences disciplines to reason about the data. A real world example includes two parties. The first is concerned about materials’ manufacturing and the other works with materials’ characterization, each has its created separate ontology. The meaning of the data from each party must be explicit to be able to reason the dependency of materials’ properties from the process parameters. However, having two ontologies, one for materials’ manufacturing and the other for materials’ characterization, will keep the meaning but separately. For that, merging is needed to be able to automatically analyze the data with the ability

to guarantee the correctness of the meaning. To do so, matching has to come first.

Other applications that need matching are query answering, ontology evolution, ontology integration, data translation and integration, information sharing and a lot others. Applications can be divided into two types, design time and runtime [12,35]. The design time is when matching is a requirement before running the actual system. On the other hand, the runtime is when matching is essential all along the running of the system, which is needed in the peer-to-peer applications such as query answering and information sharing.

So, how does ontology matching work? As shown in Figure 3. 8, two ontologies; "**O1**" and "**O2**", these are the ontologies that alignments, "**A**", needed to be found in between them. Meaning, if "**O1**" has a class "MathModule" and "**O2**" has a class "MathModule" too, then an alignment of "equivalence (=)" is defined between these two classes of the two ontologies. Same for a class "Student" in "**O1**" and a class "Person" in "**O2**", the alignment will be "subclass_of" between these two classes and so on. To do so, matching can have some inputs as per Figure 3. 8. These inputs are the alignment, "**A**", that is needed to be extended, and some input parameters like the threshold and weights according to which limits are set for the accepted alignment's results after matching.

The threshold is a percentage above which alignments are kept based on their weights' percentages. Weights are calculated based on which matching technique is used to find the alignments; the more reliable the technique is, the more weight the alignment gets, which can be decided by the matching tool's developer. For example, alignments based on the exact lexical name of the classes to be matched are weighted higher than the alignments based on similar labels of the classes. So weights are values given to the alignments detected by the matcher to determine how reliable the alignment is. [18] Finally, the resources, which are like external dictionaries and background knowledge, support the matchers to find more alignments from the domain of the matched ontologies.

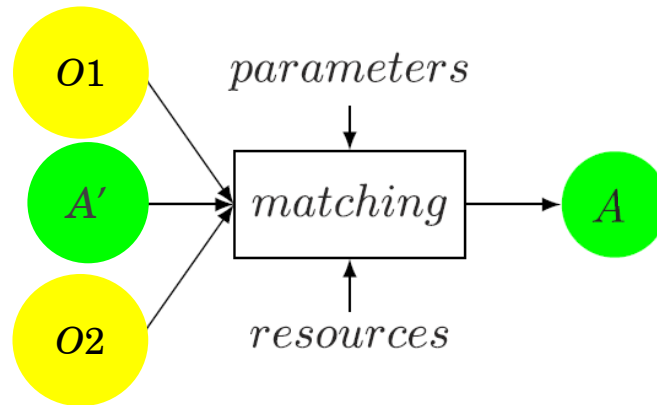


Figure 3. 8: The ontology matching process, showing the inputs to the matching process. The inputs are the two ontologies to be matched (*O1* and *O2*) and helping resources such as the background knowledge ontologies and dictionaries. Nevertheless, the calculating parameters such as the threshold and the weights based on which alignments are decided to be kept or not as final output alignments, and finally an already existing alignment, which is an optional input that is needed to be extended. The output of the matching process is the alignment or an extended alignment, which includes all the alignments between the two input ontologies that their weights are higher than the threshold. [35]

Alignments / correspondences can be processed as one to one (1:1), many to one ($n:1$), one to many (1: m) or many to many ($n:m$) between the entities. [12] The alignment can be presented in a 4-uple, $\langle \text{id}, e1, e2, r \rangle$, which consists of the alignment identifier (id), the classes and/or properties and/or individuals, ($e1$ and $e2$) from the two ontologies, and finally the logical relation r between $e1$ and $e2$ that can be equivalence, sub or super class, disjointness or others accordingly. For example inspired from Section 3.2.1, a 4-uple of $\langle \text{id}_{3,1}, \text{Student}, \text{Human}, \subseteq \rangle$, based on Figure 3. 9 that shows the correspondences between two ontologies (*O1* and *O2*). These logical relations normally take percentages or weights where the threshold is set on to choose from the alignments resulted after the match. [35]

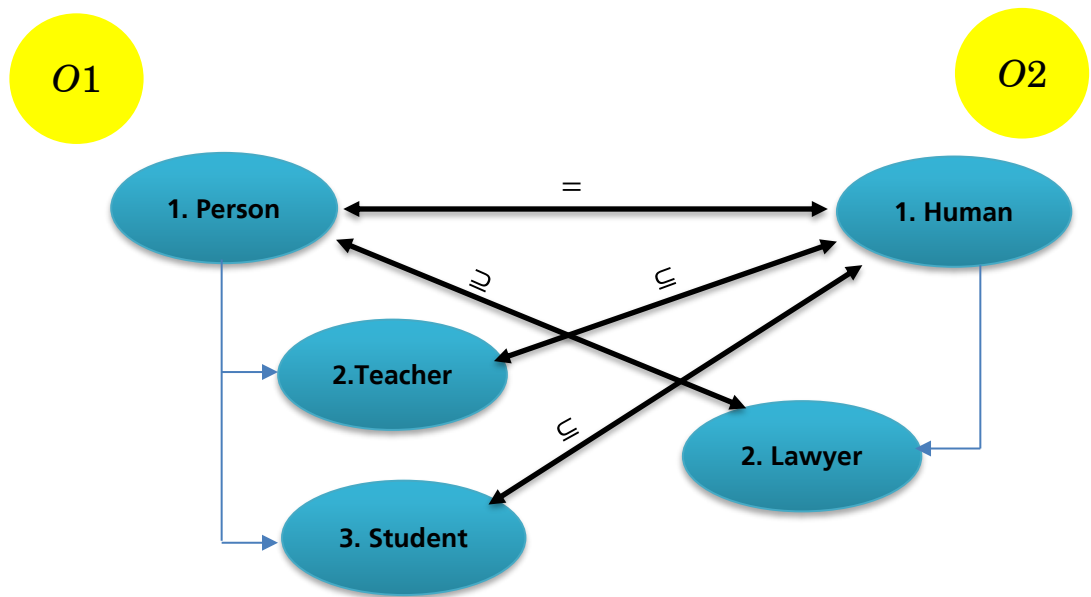


Figure 3. 9: Two ontologies (*O1* and *O2*) matching example. *O1* representing one super class named "Person" of two corresponding subclasses, "Teacher" and "Student". *O2* represents one super class named "Human" with one subclass named "Lawyer". Logical relations (alignments / correspondences) are outputs from the matching process between classes of different ontologies as shown on black arrows between *O1* and *O2*.

3.3.1 Ontology Matchers

Ontology matching can be done either manually or automatically, but creating the alignments manually is often unfeasible. This is due to the size of the ontologies or the complexity of the logical relations between ontologies' entities. Automatic ontology matching is also crucial for dynamic scenarios such as those found in decentralized data spaces, e.g. the "Industrial Data Space (IDS)". This is a standard that ensures the security of exchanging data between companies giving rights to the data owners to be the controllers of the usage of these data. [36] An initiative to implement a decentralized data space for material-intensive value chains is a domain-specific form of the IDS is called "Materials Data Space (MDS)". [37] In such cases, the automation of the process is essential, several ontology matching systems and tools (ontology matchers) exist and are available on the Web to use and adapt. Some of them work properly without errors and others are out of date and need to be adapted to the operating system that it will be working on, debugged and updated. Some have a "Graphical User Interface (GUI)" or a website to match on others have not, and to run them, a development platform has to be established. Most of the ontology matchers that exist use Java to be developed and used on, Maven tools to build the project on using Maven's "Project Object Model (POM)" and a set of plugins that are shared among all projects using Maven. [38]

Every ontology matcher chooses certain input formats of the ontologies that will be matched, such as OWL, RDFS, SKOS, XML or N3. In addition, ontology matchers choose the processing way of their alignment, whether 1:1, $n:m$, or the others. Every ontology matcher has its own matching technique using edit distance, vector distance, substrings, ISUB, WordNet, "Unified Medical Language System (UMLS)" lexicon and many other ways that help in finding literal correspondences between entities of different ontologies. Distances like edit, vector, Levenstein, etc. are different scoring schemas given to literal string correspondences of the alignment to be able to weight how well the alignment is. However, the lexicon indexation such as WorldNet, UMLS lexicon, etc. are to find literal lexicon correspondences between words, these are English lexical databases that include synonyms, adjectives, adverbs, etc. same as a detailed dictionary for words. [39] Some ontology matchers allow the external help of dictionaries and background knowledge ontologies and others not. These design and development differences between some of the ontology matching tools are shown in Table 3. 1.

Approved ontology matchers are the ones that entered one of the yearly model evaluation initiative conferences such as the OAEI, where all participating ontology matching tools are given the same input of ontologies, parameters and resources, then the results are compared in terms of runtime, correctness and completeness in terms of a specific performance evaluation, as will be explained in Section 3.4. The approved ontology matchers are published yearly on the OAEI website. [10]

Most of the ontology matching tools are created and tested to serve the biomedical, anatomy and bioinformatics fields. Examples of these ontology matchers are also presented in Table 3. 1, such as Agreement Maker Light (AML), Agreement Maker, LogMap, SAMBO, "Risk Minimization based Ontology Mapping (RiMOM)", ASMOV, and a lot more. [40–42] Other fields' domains need to adapt the ontology matchers to work perfectly on their ontologies, like the Materials Sciences field. Challenges for ontology matching tools are evolving every now and then during evaluations and usages of the tools. Examples of these challenges are problems that appear during users' involvement in the alignments results and feedbacks after the matching also during the runtime, widening the scale of the background knowledge use in matching ontologies, efficiency of the matching and so on. [35] As a reaction to these challenges, developers of the ontology matching tools improve their tools over time.

Some ontology matching tools such as AML include several matching systems in their own platform, allowing the tool's user to mix between different types of matching techniques or simply use them all.

Table 3. 1: Ontology matching tools, showing differences between the tools in terms of inputs, outputs, having a GUI or not and matching techniques [6,18,35].

Ontology Matcher	Input	Alignments Processing	GUI	Matching Techniques
<i>Agreement Maker</i>	XML, RDFS,OWL,N3	$n:m$ alignments	Yes	Edit distance, substring or/and WordNet
<i>Agreement Maker Light (AML)</i>	OWL	1:1 alignments and optional $n:m$ alignments	Yes	ISUB, Levenstein distance, Jaro- Winkler, WordNet or/and Q-gram
<i>LogMap</i>	RDF/XML, OWL/XML, OWL Functional, OBO, KRSS, and Turtle	1:1 alignments	No, but a website exist	WordNet, ISUB or/and UMLS lexicon
<i>RiMoM</i>	OWL	1:1 alignments	No	Edit distance, vector distance or/and WordNet
<i>ASMOV</i>	OWL	$n:m$ alignments	No	Tokenization, string equality, Levenstein distance, WordNet or/and UMLS lexicon
<i>SAMBO</i>	OWL	1:1 alignments	Yes	n-gram, edit distance, WordNet or/and UMLS
<i>Falcon</i>	RDFS, OWL	1:1 alignments	No	ISUB

3.4 Performance Evaluation of Ontology Matchers

As discussed above, ontology matching tools are evaluated in the OAEI and compared based on the results' run time, completeness (recall), correctness (precision) and a combined measurement between recall and precision, which is the F-Measure.

It is necessary to evaluate the performance of the ontology matching tool to approve it as a qualified ontology matching tool and to be able to choose which tools are qualified for which ontology matching process. For example, which tools perform better for large ontologies matching, which tools get more accurate results while the time factor is not prioritized, which are fast, and so on. Such evaluations are important for the tools' users to choose according to their applications.

The OAEI Evaluation

In 2004, ontology matching researchers introduced campaigns for evaluations that was then known by the name Ontology Alignment Evaluation Initiative (OAEI) in 2005. The OAEI has been running every year since that time until today. Measurements defined by the OAEI are clearly defined to all participants, and results are yearly recorded and published in papers that can be found on the OAEI website. Before the OAEI, there was the "Text REtrieval Conference (TREC)" [43], which is another model evaluation initiative that has been running yearly until today since the year 1992. The conference mainly focuses on the computer sciences research field checking how well computer scientists performed to improve tools and software. These two models for evaluations are examples of the evaluations that exist for data retrieving tools and systems. However, the OAEI is the most famous among all evaluation models because it is designed specifically for ontology matching tools and not for any other semantic data handling tools. The evaluation measurements defined by the OAEI are well known to every ontology matching tools' developers.

For the ontology matching tool's developer(s) to participate with the tool in the OAEI's yearly evaluation, developer(s) should check the tool's own validity by trying out the evaluation's demo benchmark designed also by the OAEI, and then submit the final version of the tool on the OAEI website. By doing that, each tool's developer(s) see(s) if the demo benchmark, applied on the tool, results are above a certain threshold or not. The threshold is specified also by the OAEI yearly based on the demo benchmark, it was 0.9 (90 %) for the OAEI 2019 demo package [44]. If the results are above the threshold that means that, the tool is qualified to enter the yearly evaluation.

The OAEI Benchmark

A benchmark is a set of test cases, input ontologies in a specific domain, and their (manual) reference alignments that are expected from the ontology matching tool to have a close result of alignments to them. Results of the ontology matching tool are compared to the (manual) reference alignment of the same input ontologies (test cases). A benchmark also includes the external resources and parameters such as background knowledge ontologies and threshold.

OAEI's benchmark consists of:

- Test cases (two from the Biomedical, Bioinformatics or Anatomy ontologies, e.g. FMA, SNOMED CT, NCI, HP, etc.)
- A reference alignment for every test case
- Background knowledge ontologies (Biomedical, Bioinformatics or Anatomy ontologies, e.g. "Human Disease Ontology (doid)", "Uber-anatomy ontology (uberon)", etc.)

Figure 3. 9 describes the basic idea of evaluating ontology matchers. Generally, every matching tool takes the same inputs of two ontologies "*O1*" and "*O2*", parameters and resources, and then the tool output alignment "*A*" is compared to the (manual) reference alignment "*R*". Finally, the evaluator computes the performance measurement(s) "*m*"; like completeness and correctness, which are then used to compare between all the ontology matching tools participating in the evaluation. [12]

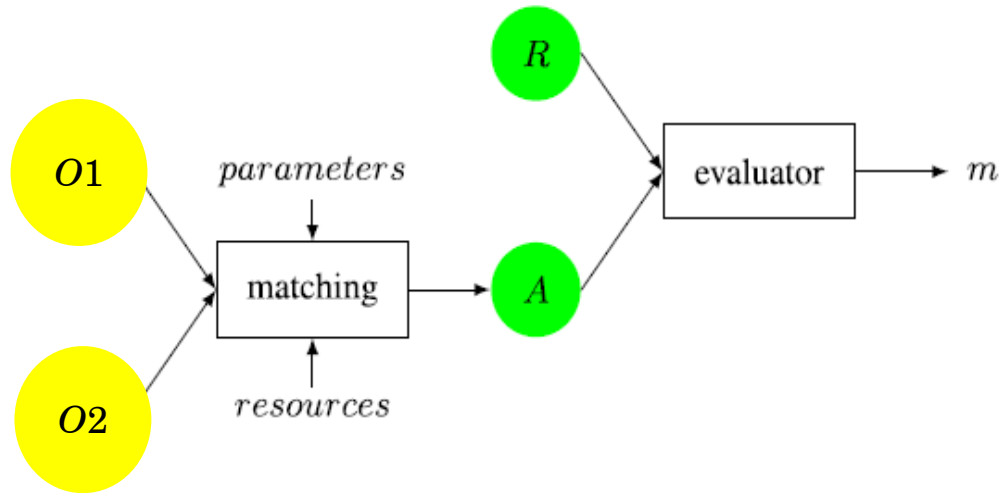


Figure 3. 10: Performance evaluation scheme for ontology matching tools, which shows how the output from the matching tool “A” is evaluated against the reference alignment “R” and finally output “m”, which are the measurements used to compare ontology matching tools with. [12]

The recall or the degree of completeness, defined in eq. (3.1), refers to the output alignment “A” that is checked against the (manual) reference alignment “R”, checking the number of correct alignments in “A” from “R”. Meaning, it is the ratio of the true positive correct results between “R” and “A”, $|R \cap A|$, over the total number of correspondences in “R” that are expected from “A”.

$$Recall = \frac{|R \cap A|}{|R|} \quad (3.1)$$

The precision is the degree of correctness, presented in eq. (3.2), checking the correct alignments or correspondences that “A” got the same as “R”, expressed by $|R \cap A|$. For example, if there is only one alignment in “A” which is correct and found in “R”, then the precision will be 100 % regardless of “A” not having anything else but this single alignment, which is achieved when this intersection is divided by the total number of alignments in “A”. On the other hand, the percentage of the recall is defined by how complete the alignment “A” is; accordingly, eq. (3.1) is multiplied by 100. Therefore, in this example, the recall will have a very low percentage.

$$Precision = \frac{|R \cap A|}{|A|} \quad (3.2)$$

The F-measure is a mix between recall and precision, mainly used to be a final output result for the ontology matching tool to be compared to other ontology matching tools, since precision on its own or recall on its own are not enough or fair to compare tools with. The F-measure defined in eq. (3.3) is a compromise between both measurements. That can be defined by a number α between $[0,1]$, such that if $\alpha = 0$, then the F-measure is equal to only recall, if $\alpha = 1$, then F-measure is equal to only precision and the most used case is $\alpha = 0.5$, where both precision and recall have the same importance in the evaluation. [12] The F-measure is important for applications that are created based on the matching output, these applications choose their priority between precision and recall based on the α value they choose.

$$F\text{-measure} = \frac{Precision \times Recall}{(1 - \alpha) \times Precision + \alpha \times Recall} \quad (3.3)$$

Other measurements that can be built on these commonly used measurements such as Overall, Fallout and Miss can be found in [12]. However, as explained above, the runtime, recall, precision and F-measure are the main evaluation measurements used to evaluate ontology matching tools nowadays, since they conclude the matching results.

All measurements introduced by the OAEI for evaluation and all the yearly evaluations performed for the ontology matchers are mainly done for the bioinformatics, biomedical and anatomy domains. From here, it is important to adapt this evaluation to the Materials Sciences domain and for that, a Materials Sciences benchmark has to be developed.

In this chapter, the background behind the thesis' main goal is presented. The next chapter is going to discuss how this knowledge is used to set the development platform for creating a Materials Sciences benchmark and evaluating it using the chosen and updated ontology matching tools after adapting them to the Materials Sciences domain and setting the evaluating schema for them.

4 Methodology

In this chapter, the methods used in order to perform the ontology matchers' evaluation for Materials Sciences are presented, starting with the chosen ontology matchers to be evaluated and then moving on to the development platform preparations for these chosen ontology matchers. After that, the Materials Sciences benchmark creation is shown with its test cases, manual reference alignments and the background knowledge ontologies. Last but not least, the performance evaluation scheme is discussed, which describes the measurements performed for the evaluation of the chosen ontology matchers using the Materials Sciences benchmark. Finally, the performance evaluation workflow, the experiments' arrangements, how the matchers are combined forming experiments are presented along with the experiments' expected total number of results that will be evaluated and discussed in the coming chapter.

4.1 Ontology Matchers Preparation

As discussed in the previous chapter, many ontology matchers exist and a lot of them join the OAEI's yearly evaluations. Despite this fact, not all matchers are available to use, so choosing ontology matching tools is a very important and critical part in the thesis.

For the evaluation in this thesis, only the participated ontology matching tools at the OAEI are considered, from the OAEI official start in 2005 until today [17]. Some of them are active with code and others were active before but currently they are neither active nor available with code. Examples of the popular but currently inactive ontology matching tools are the "Automated Semantic Mapping of Ontologies with Validation (ASMOV)", the "Framework for Ontology Alignment and Mapping (FOAM)" and SAMBO, which was not only matching but also was able to apply one of the famous ontology matching applications by merging two ontologies after getting the alignments between them. Examples of the active and participating matching tools are like COMA++, "Combinatorial Optimization for Data Integration (CODI)", Blooms, GOMMA and lots of others. Most of them had an explanation of the workflow of the matching technique used but few of them are available to use.

The thesis' criteria is to only consider ontology matchers that are freely available on the web to enable insight into their source code, which makes it possible to adapt and modify the tools. The decision of which ontology matchers are going to be used in the thesis' Materials Sciences ontology matching evaluation and which are going to be excluded started by preparing the development platform

needed for every available on the web and approved OAEI's ontology matching tools [17,31]. As mentioned in the previous chapter, in Section 3.3.1 about ontology matchers, and will be explained in detail in the next section, most of the ontology matching tools work on Java; accordingly, their code can be edited by one of the "Integrated development environment (IDE)s" supporting Java e.g. Eclipse. The tools are also common in using Maven plugins, which assist in building, reporting and documenting the projects.

Excluded ontology matchers include "Agreement Maker (AM)" [12] developed in 2001, which is a tool that has a GUI easing the use of the tool if no adaptation is needed by coding, and its code is available on GitHub [45]. It is one of the leading systems in ontology matching. However, it cannot handle big size ontologies. That was the reason why the tool stopped participating in the OAEI after the year 2012, and a derived version of it was out, which is Agreement Maker Light (AML) that adapted the Agreement Maker and improved it to be able to efficiently match very large ontologies in a few seconds. At the beginning of preparing the Agreement Maker tool in this thesis, it had a problem with a Maven plugin it used, the plugin was out of date, and accordingly the tool did not run. By using the updated Maven plugin, the problem was solved and the tool was able to build and run. The old plugin was Pax and instead, Gradle's building tool's plugins are added and used, both of them are used to run the whole project and run the tool, but the Pax is no longer compatible with current cross development platforms that run on the "Java Virtual Machine (JVM)". Other issues also appeared during the debugging of the tool's code, and solving them was by changing the URLs of the plugins in the Project Object Model (POM) file of the tool. By using Maven, all plugins needed to build the Java project are included in a file with a POM extension, such that every defined plugin has its own URL that calls the plugin once every update from the Maven website, which ensures the up to date plugins among all Java projects building using Maven. However, once those adaptations are made and the matcher is running, it took a long time of 6 hours to even upload medium sized ontologies needed to be matched, with around 500 entities each, which is why the tool was discarded from the evaluation.

Other excluded ontology matching tools are Falcon [46], which has only an offline available version, which does not work, and Blooms [47], which uses a Web search results from Bing to validate and correct the alignments using the Bing Web search API, which is no longer available to generate an "Application Programming Interface (API)" key for the Web search to use, so accordingly the tool does not run, since the Bing Web search API is no longer available.

Another excluded tool is COMA 3.0 [48], which needs MySQL to be running on the machine during the run time of the tool as well as getting access to create a database. After preparing its needed requirements of MySQL and granting access to it, the tool was not able to access the permission of creating the database,

accordingly it did not run either. CODI [49] is also a tool which required a Gurobi's, a powerful mathematical optimization solver license in order to run, which is not available for free to use. In the same time, a license will be needed for every development platform, meaning the same license will not work on cross development platforms, so the decision was not to include the tool in the evaluations since it is not feasible for the work in this thesis.

Based on the above discussed preparations, the considered ontology matchers for the thesis are the only active ones that are able to run with feasibility and practicality.

These chosen tools are:

- All the ontology matchers included in AML that are derived from AM solving its problems of memory and time discussed above.
- The LogMap tool.

The properties of the chosen ontology matchers will be discussed in detail in the following Sections 4.1.2 and 4.1.3.

4.1.1 Development Platform

The development platform needed to be set for the chosen ontology matching tools, which mainly were written in Java, was an IDE, so Eclipse was chosen, accordingly "Java Development Kit (JDK)" was also needed to be set up. The tools also need Apache Maven Project to be set up, since they use its plugins to build the tool. Maven is used for easing the building and managing of any Java-based project, by providing a uniform build system for all projects using Maven by using POM and a set of unified plugins. That means that any project that is built using the Maven facility will not take time to navigate through any other projects also built using Maven. A summary of all Maven features can be found on its website. [50]

Since one of the thesis' goals is to have a Materials Sciences benchmark to perform evaluations afterwards, Protégé was also needed to create and edit the Materials Sciences test cases of ontologies. Protégé is one of the ontology building and editing tools, which is currently the most used tool in building ontologies. Protégé is made with Java to ease the interaction with OWL. As mentioned before in Section 3.2.2, OWL is a complex programming language, where classes, properties (relations), individuals, definitions, etc. can be easily viewed and edited using Protégé's graphical user interface's drop down menus and other facilitating features. [3]

4.1.2 Agreement Maker Light (AML)

In 2013, AML was derived from AM ontology matching tool. It is an automated and efficient system that includes more than two primary ontology matchers and seven secondary ones, allowing the users to mix and choose combinations of them, based on the users' application. [51] The tool also has a graphical user interface as an executable jar file, besides allowing the developers to access, edit and adapt the code, which is available on GitHub. [52] The version of the AML source code that has been adapted for this thesis can be found in Appendix A.1.1.

The GUI allows the user to choose between ontology matchers, upload two ontologies that need to be matched, revise output alignments, remove incorrect alignments, add new alignments, upload reference alignments, evaluate the output alignment from the tool or even from other tools and finally save the alignments.

The Data Structures in AML

The best way to understand how AML works is to discuss similarities and differences between AM and AML, and how AML adapted AM to solve its memory and time difficulties in handling large sized ontologies, e.g. Gene Ontology (GO) of 45,003 classes as January 2019. [53] Both AML and AM have two computational models as presented in Figure 4. 1. The first computational model is responsible for loading the ontologies, either the two ontologies that are inputs to be matched or the background knowledge supporting ontologies. The second computational model is for performing the ontology matching, responsible for aligning the ontologies by the chosen matchers from the tool.

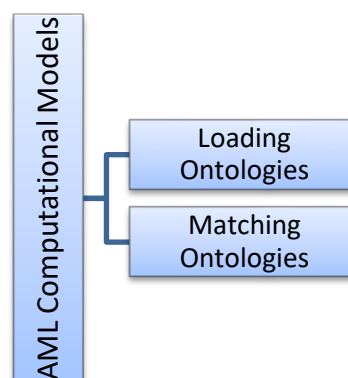


Figure 4. 1: Agreement Maker Light (AML) computational models for performing ontology matching. The first model is responsible for loading the two matched to be ontologies and any resource ontologies. The second model is responsible for performing the matching operation of the two input ontologies.

Also like AM, the AML framework has three key data structures, presented in Figure 4. 2, which are Lexicon, RelationshipMap and Alignment. The Lexicon and the RelationshipMap are mainly created for saving ontologies' information. The classes' names, labels, URIs and synonyms are saved in the Lexicon data structure. However, the structural information like the relations between classes is saved in the RelationshipMap. The Alignment data structure is the one responsible for storing the correspondences between the two input ontologies.

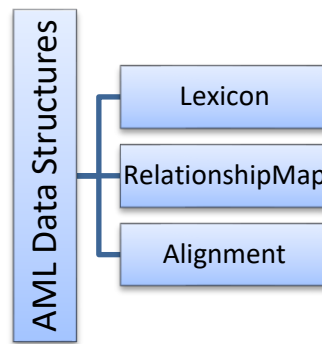


Figure 4. 2: Agreement Maker Light (AML) data structures, which are used to either save the ontologies entities in case of the Lexicon and the RelationshipMap data structures, or save the alignments' results in case of the Alignment data structure.

Like AM, AML loads the ontologies based on Jena2 ontology API, which is a Java programming toolkit, an API responsible for reading the ontologies' entities and save them in the memory as a Jena OntModel. Jena2 is a newer version of Jena that is capable of reading varieties of ontology languages such as RDFS, DAML and the OWLs (OWL 1 and OWL2), while Jena is only capable of reading DAML. [54]

The differences between AM and AML start from here as follows. In AM, the complete OntModel is kept in the memory throughout the matching process, which dramatically increases the runtime of the matching process due to the occupied memory. However, AML stores all the information needed in the matching process from the OntModel in its internal data structures, explained above in Figure 4. 2, and totally removes the OntModel from the memory during the matching process. Given the fact that filling the data structures with the needed information takes less space in the memory than the OntModel takes. Nevertheless, AM also has redundant information by storing the information twice, once in the memory by keeping the OntModel in it and another by filling its data structures with the needed information for matching from the OntModel, same as AML. [51]

Another main difference between AM and AML is in the Alignment data structure they share. AM stores in Alignment during the runtime, all the final outputs of alignments performed between all classes and properties, " n ", of the first input ontology (source ontology) against all classes and properties, " m ", of the second input ontology (target ontology). By that, a space of $O(n \times m)$ in AM is needed from the memory when the algorithm runs. Meaning, if two ontologies of around 50,000 classes are matched to each other and the algorithm checks each class from the first ontology with each class from the second ontology to find alignments, 18.6 GB of the memory will be needed to save alignments found for every single class during the run time of the matching process, and that is why AM have difficulties in handling large sized ontologies in terms of the number of entities. [18,55] On the other hand, AML uses the (1:1) alignment strategy, so the saved alignments will occupy the memory by $(\min(n, m))$. AML made these adaptations since most or almost all similarities between ontologies are detected by the (1:1) strategy, and the $(n:m)$ rarely or almost never gets more alignments in the output results than the ones detected by the (1:1) alignment strategy. [51]

According to these adaptations on AM, AML has the ability to load large ontologies of around 120,000 classes in under 150 seconds on a 4-core CPU server with a RAM of 16 GB, by not keeping the OntModel in memory. [51] Nevertheless, the runtime of the matching process is reduced dramatically by applying the (1:1) alignment strategy. In the same time, the matching quality is preserved or even better according to its results in the OAEI. In the OAEI for the year 2012, AML achieved the highest F-measures in the Anatomy (with an F-measure of 92.4 %), biomedical and bioinformatics tracks (with an F-measure of 85.4 %) not only when compared to AM but also when compared to other ontology matching tools such as YAM++, CODI, GOMMA and a lot of others. [51] AML shows the best results, in the same year, in terms of most evaluation measurements – specially the runtime of matching, which was between 89 to 231 seconds compared to others that match the same ontologies in 30155 seconds. [51]

AML uses weights in most of the ontology matchers it provides that use its Lexicon data structure for the matching process, such as the lexical ontology matcher. As described in Section 3.3, weights are one of the main parameters along with the threshold needed for the ontology matching process. In AML, based on the Lexicon data structure saved about the input ontologies of names, labels, exact and related synonyms, weights are assigned. Meaning, the more reliable the Lexicon is, the more weight percentage it takes. Accordingly, if two classes were matched based on their names, the weight will be 1.0, if they are matched based on their similar labels, weight will be 0.95, if they are exact synonyms of each other, weight will be 0.9 and for other related synonyms the weight will be 0.85. The threshold is like a confidence level set for every weight

for every found alignment, where the final weights' percentages of matched classes after performing all the matching processes designed in AML are kept or discarded based on the threshold set. For example, if the threshold was set to 70 % (0.7), that means that only alignments with a 70 % (0.7) weight percentage or above will be kept and the alignments below will be discarded. In the thesis, the threshold is set to 50 % (0.5) to be able to see all possible alignments detected with AML, as the minimum accepted threshold by AML is 50 % (0.5).

The RelationshipMap data structure, which saves all properties' paths between classes and the distance of each path, is mainly needed in the property matcher in AML.

The Ontology Matchers in AML

The two principal categories of ontology matchers in AML are displayed in Figure 4. 3 and discussed as follows. The first category is the primary, which includes the lexical matcher that checks the Lexicon between entities of the input ontologies and calculate weights, as explained above. The other primary matcher found in AML is the word matcher, which also depends on the Lexicon data structure and measures the similarities between classes by the weighted Jaccard index that can be detected by the names of the classes. [45]

The second category in AML's ontology matchers is called secondary, since its ontology matchers are not responsible for the majority of alignments' result as much as the primary ones do. The property matcher is secondary and uses the properties (relations) between classes to find more alignments, as mentioned above. The property matcher uses the RelationshipMap data structure to determine alignments between properties, and accordingly determining alignments between the classes related by the aligned properties. Another secondary matcher of AML is the background knowledge matcher, which uses the input resource of the background knowledge ontology to get more alignments. This supporting ontology includes classes, and each class has labels with its other descriptions or names, the matcher uses the background knowledge ontology to check if the class it is matching from the first input ontology has a correspondence with another class from the second input ontology based on the intermediate background knowledge ontology. Its idea is like looking up synonyms of words in a dictionary, if the dictionary shows that a word from the first ontology is a synonym to a word from the second ontology, this means that the two words are aligned based on the mid way supporting dictionary, which is the background knowledge ontology. [51]

The structural matcher is also a secondary matcher, which uses both the Lexicon and the RelationshipMap data structures to find correspondences between matched ontologies. As the RelationshipMap was described, it includes the

nodes' and classes' paths and distances, so whenever a lexical correspondence is found between one class from the first ontology with another from the second ontology, the RelationshipMap is used to perform a neighborhood search. The search is done to the ancestors' paths and the descendants' paths of the corresponding aligned classes, such that the ancestors search checks all the classes from the aligned class to the root top main class for both ontologies and see whether there are other lexical correspondences or not, the descendants search is the other way around. This guarantees the logic of aligning, since classes taking similar paths from the root to the leaf nodes might have more correspondences than others. [51]

There are also the obsolete, the cardinality and the coherence matchers, which use the Lexicon data structure saved information other than names, such as labels and synonyms to detect more alignments between classes of the two ontologies.

All of these previously mentioned matchers, whether primary or secondary, use a HashMap cross (1:1) searches, which for example searches where a key from a HashMap is used to directly query another HashMap, which is done using a single for loop with a time complexity $O(n)$. However, AML includes another secondary matcher, called the string matcher, which is timely expensive to choose to match ontologies with, the string matcher makes string comparisons using the ISUB, Levenstein distance, Jaro-Winkler or Q-gram techniques to calculate distances and weights for the aligned classes. To do so, it needs to pass by (n:m) classes of the two ontologies, which requires a nested for loop with a quadratic time complexity of $O(n^2)$, which is why it is a secondary matcher. [51]

Due to all of these features, the variety of ontology matchers the tool has and the success achieved in the yearly OAEI evaluations in medical and biological domains, AML was strongly chosen to be part of the Materials Sciences ontologies evaluation performed in the thesis. The possibility to have support by the developer of the tool, Daniel Faria from the Instituto Gulbenkian de Ciência, Oeiras, Portugal [56], to establish the proper functionality of the tool, made it easier to adapt, fix debugging runtime errors of the tool and deeply understand it. The adaptation of the tool to the Materials Sciences domain is by adding the proper background knowledge ontologies, as is going to be explained later on in this chapter, and the debugging is to fix runtime errors that appears during the adaptation of the tool to the development platform and the usage of out to date plugins of Maven tools.

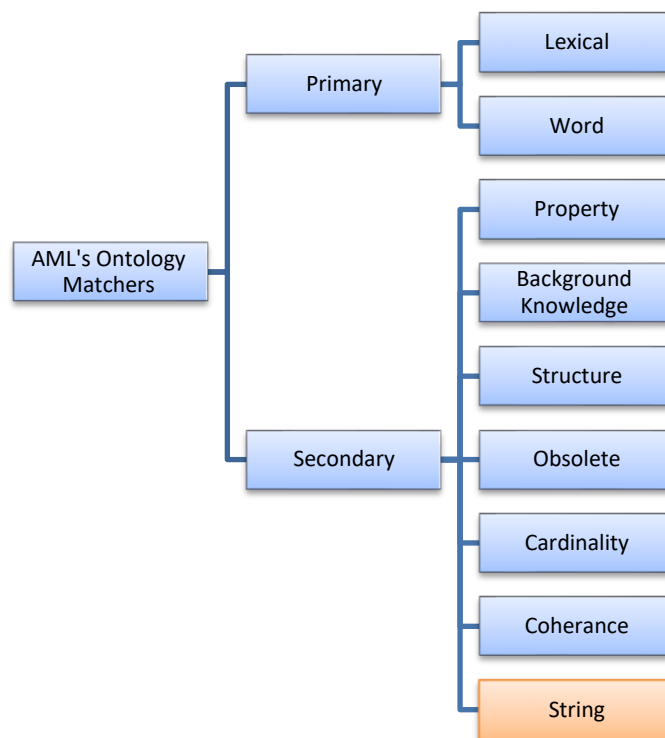


Figure 4. 3: Agreement Maker Light (AML)'s ontology matchers divided according to how essential they are in getting more alignment in the matching results into two categories; primary and secondary, respectively. All of the AML's ontology matchers are with time complexity of $O(n)$ except for the String matcher with a $O(n^2)$ time complexity.

4.1.3 LogMap

“Logic-based Methods for Ontology Mapping (LogMap)”, is the other chosen ontology matcher in this thesis. According to the OAEI yearly evaluations, LogMap is a highly scalable ontology matching system, which can handle large sized ontologies of hundreds of thousands of classes and semantically rich ontologies with many data and class properties. It takes part in the yearly OAEI evaluations mainly matching biomedical, bioinformatics and anatomy domains too, same as AML. [19] It has a high repair algorithm, which is responsible for getting matching outputs of only the highly reliable alignments between classes, with a weight of more than or equal to 90 % (0.9), that makes its results higher in precision than other ontology matching tools, reaching 100 %. [19]

LogMap matcher depends on the lexical information of the classes to get alignments, which is detected either from the classes' URI or the classes' annotations. By default, the annotations in LogMap are the labels saved in "rdfs:label" of the class definition in the ontology file. Users of LogMap can access the code and update the usage of the label annotation to any other type of annotation or have the support from the tool's developers to adapt LogMap accordingly. [57] The code is open source on GitHub, cloned on August 2019 [58]. The steps undertaken to set up LogMap in this thesis after cloning from GitHub can be found in Appendix A.1.2. LogMap same as AML also uses Maven building tool along with Java to run. Although LogMap does not have a GUI that can be used to ease the tool's experience, LogMap provides a Web front-end facility where users submit all the needed inputs online to the webpage and adjust all the options, then submit and receive the output alignments on the submitted email. [59] In the thesis implementations, the LogMap Java version is used in order to preserve the thesis data. The website is also tested to confirm its activity and results, and it has worked same as the Java version showing same results.

LogMap also uses the structure of the ontologies, classes' hierarchy, in the matching process, which is detected after reasoning the input ontologies. This is done by using one of the Description Logic (DL) reasoners like HermiT and Condor that are used by LogMap. A semantic reasoner is a software that infers logical relations of a given set of facts, in the case of ontologies, a set of given entities. Although it is computationally expensive to reason an ontology, it is done once before starting to match. The output of the reasoner will be DL fragments from which the descendants, ancestors and the topological order of all classes in both ontologies can be detected. [60]

4.2 Benchmark Development

After preparing the ontology matchers that are going to be tested in the evaluation, this section presents the creation of the Materials Sciences benchmark, consisting of two input ontologies, "*O1*" and "*O2*", and their reference alignments, "*R*", which will act as standards/controls that the ontology matchers' alignments' outputs, "*A*", will be evaluated against, as explained in Section 3.4. Last part of the benchmark is the Materials Sciences background knowledge ontologies to be used as input to the ontology matchers to support matching the created test cases. These all together are the Materials Sciences benchmark presented below and given in Appendix A.2.

A Materials Sciences benchmark consists of:

- Three Materials Sciences test cases, Appendix A.2.1
 - Two Materials Sciences ontologies

- A manual reference alignment for every test case, Appendix A.2.2
- Materials Sciences background knowledge ontologies, Appendix A.2.3

4.2.1 Preparations of Materials Sciences Ontologies

As discussed in Chapter 3, there are not many domain based Materials Sciences ontologies openly available. However, from the openly available Materials Sciences ontologies discussed in Section 3.2.5, the chosen ontologies are MatOnto, ASHINO's ontology and the developing ontology EMMO. These ontologies are chosen since they do not target very specialized disciplines of the Materials Sciences, e.g. ceramics. They are all created to be general to the Materials Sciences domain. Being from the same level of specialization, they can be matched to each other and used for the evaluation of the ontology matchers in this thesis.

Preparation of the MatOnto Ontology

Before going into details of the test cases creation, the preparation of the Materials Sciences ontologies chosen has to be discussed. First, MatOnto, available on GitHub [16] is cloned, the cloned files are RDF data, saved in Turtle (.ttl) format. The step needed to be done before usage is that Protégé is used to convert the Turtle format into OWL, such that it would be compatible with other chosen ontologies. The ontology is based on the upper level ontology, the BFO. It consists of 847 classes, 96 properties (relations) and 131 individuals (instances of the classes). The ontology has the possibility to choose from its different ontology's partitions. The ontology is divided into smaller ontologies (partitions) in separate files that can be chosen alone. For the test cases creation, all MatOnto ontology's partitions are used.

Preparation of the ASHINO's Ontology

The second Materials Sciences ontology chosen is ASHINO's ontology created by Prof. Ashino [13]. The ontology is not openly available, however, it was provided upon request. The ontology is not based on an upper level ontology, which makes the ontology not easily interoperable with other Materials Sciences ontologies. The ontology is also divided into smaller ontologies (partitions), but on the contrary to MatOnto, ASHINO's ontology has no one file grouping all partitions in one file. So creating the grouped ontology manually is done as a step before usage. The separate ontologies are Environment, Geometry, Material Information, Manufacturing Process, Property, Substance, Unit Dimension, Structure, Equation and Physical Constant. After grouping, the ontology consists of 545 classes, 98 properties and 411 individuals.

Preparation of the EMMO

Last chosen ontology is one of the ongoing Materials Sciences ontologies, the EMMO, which is openly available on GitHub [15], thesis version is downloaded on August 2019. EMMO is an upper and mid level ontology, which adapts the BFO in its creation, accordingly classes are similar to the BFO but modified, representing the same abstraction levels as the other chosen Materials Sciences ontologies. It consists of 103 classes, 57 properties and 2 individuals.

4.2.2 Test Cases Design

One major part of the Materials Sciences benchmark are the test cases. The idea behind the following mixtures of the chosen ontologies forming the test cases is tackling many discussion points of adapting ontology matchers to the Materials Sciences domain. Nevertheless, discussing Materials Sciences benchmark in general is a very important part to be also tackled in the thesis.

A test case is a two-ontology OWL files created from the chosen Materials Sciences ontologies. In this thesis, three different test cases shown in Figure 4. 4 were created to cover principal ontological and Materials Sciences concepts.

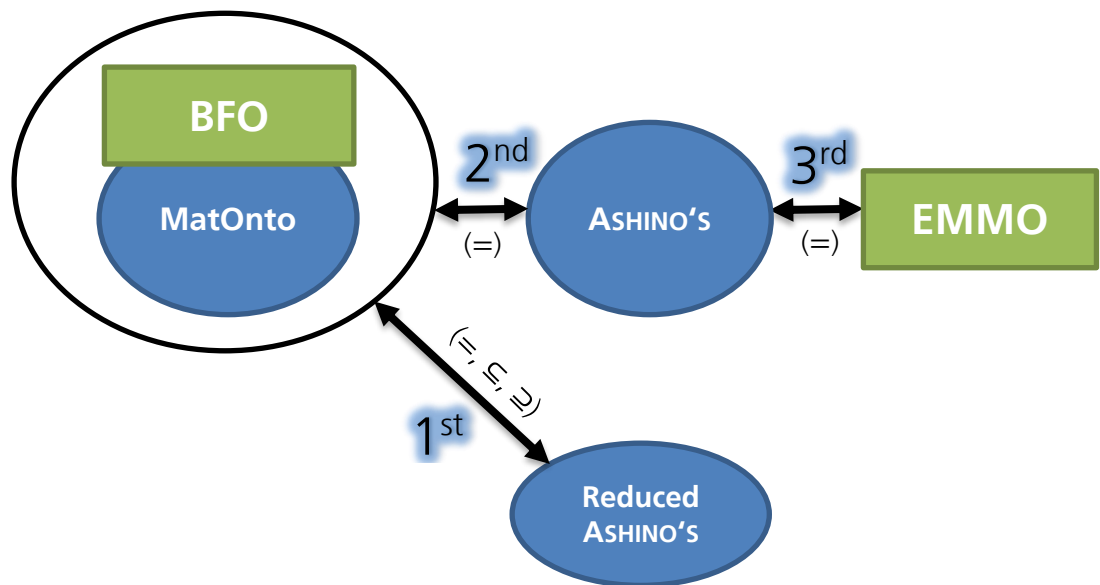


Figure 4. 4: Summary of the three test cases compositions, created from Materials Sciences ontologies to be a part of the thesis' Materials Sciences benchmark, such that the green rectangles show the upper level ontologies EMMO and the BFO and the blue circles show the domain specific ontologies. The 1st test case is between a reduced subset of ASHINO's ontology, named Reduced ASHINO's, and complete MatOnto (BFO + MatOnto). The 2nd test case is between the complete ASHINO's ontology and complete MatOnto (BFO + MatOnto). Finally, the 3rd test case is between the EMMO upper level ontology and the complete ASHINO's ontology.

1st Test Case

The 1st test case is created to evaluate the ability of the ontology matchers to find correspondences of all possible logical relations. The test case consists of a reduced version of ASHINO's Materials Sciences' ontology, named Reduced ASHINO's "**O1**", and MatOnto "**O2**". It can be found in Appendix A.2.1.1. Three of ASHINO's separate ontologies are chosen and combined together, which are Environment, Equation and Manufacturing Process, while the MatOnto was taken as a whole. This test case is designed to demonstrate the behavior of the ontology matchers upon two domain specific Materials Sciences ontologies. It is designed small in terms of the number of entities to be able to evaluate and discuss ontology matchers on the expected possible logical relations (e.g., =, \subseteq , \supseteq , \perp) that would be the best practice to have as alignments between the two ontologies' classes, enabling an easier interoperable merging afterwards. The fact is, all matchers, whether the chosen ones or others, only get the equivalence alignments (=) between classes, and other logical relations such as subset, superset, and not equivalent are not explicitly shown in the matchers results. The test case is designed to discuss how each matcher implicitly shows more logical relations and how to improve each matcher result.

2nd Test Case

The 2nd test case is a bigger scale version of the 1st test case in terms of the number entities, but not in terms of the variety of logical relations, it only consists of the equivalence (=) logical relation. The test case's two ontologies are the complete merged ASHINO's ontology "**O1**" and the complete MatOnto "**O2**", and can be found in Appendix A.2.1.2. Both of the ontologies are domain specific ontologies, MatOnto bases on the BFO as its upper level ontology however, ASHINO's ontology has no base of any formal ontology. This test case is a typical practice if two Materials Sciences domain ontologies are decided to be merged in one ontology to see how far the ontology matchers will get alignments of ontologies' classes of the same domain, and how the ontology matchers will help in improving interoperability between these two ontologies due to the difference of usage of the formal ontologies.

3rd Test Case

The 3rd test case is designed for a case in which a domain ontology has to be merged to an upper level ontology in order to maximize the cross-domain interoperability of this domain ontology. To do so, the most commonly known upper level ontology in the domain ontology's domain is chosen, and then merged to it. Accordingly, a matching should be done first between the domain ontology and the upper level ontology chosen. Finally, based on the output alignments from the matching, classes from the domain ontologies will be directed and added under classes from the upper level ontology.

In the thesis, the test case is designed between the domain Materials Sciences ontology, complete ASHINO's "*O2*", and the upper level ontology EMMO "*O1*", which is an upper and middle level ontology designed specifically for physics and Materials Sciences domains. The EMMO is chosen and not the BFO or other upper level ontologies, since EMMO is mainly designed for the Materials Sciences domain, accordingly no alignments to be expected from other upper level ontologies. The test case can be found in Appendix A.2.1.3.

4.2.3 Manual Reference Alignments

The second part of the Materials Sciences benchmark is the manual reference alignment. The manual reference alignments, "*R*", are the expected result from the test cases' alignments, "*A*". They are compared to the result of the ontology matchers. For every test case, a manual reference alignment is created.

The creation process of a manual reference alignment is very time intensive, since as discussed before; ontologies have both philosophical and computer scientific point of views. In philosophy, ontologies are the expression of the being which explain the being by using classes and adding relations between them [26], while in computer science, ontologies are models or domains described by machine understandable meanings of parts of these models. [26] From the two definitions, creating an ontology from the computer science point of view seems the easier and straightforward task to do, given classes' names of a certain domain and the kind of relations between them, the ontology can be created. However, from the philosophical definition, multiple trials and philosophical questions have to be raised to decide on a single class' name, term, definition, equivalence and a lot of other things. Therefore, the complexity in creating a manual reference alignment "*R*" arises mostly from the philosophical point of view while deciding which and how an entity from the first ontology "*O1*" relates to an entity from the second ontology "*O2*".

For this reason, experts in the corresponding domain have to be consulted. Accordingly, the manual reference alignments for the three test cases were created in close cooperation with three experienced Materials Scientists at Fraunhofer EMI. Some general rules are set during the creation process of the manual reference alignments, which are going to be discussed in detail and why especially they are chosen in the following chapter. Examples of these rules are as follows,

- Singulars and plurals are equivalently matched (e.g. water = waters).
- Only classes alignments are included (No properties or individuals alignments).

The manual reference alignments were created Using the GUI of AML, which helps in viewing the classes and relations of both input ontologies. The data and output file format of the manual reference alignments is RDF. Beside the ontology matchers AML provides, it allows the manual creation of the alignments by simply using the interface to choose the entity name from the first ontology and the entity name from the second ontology and the desired logical relation between these entities and save the chosen alignments in the format of the manual reference alignment, as explained and shown in Appendix A.2.2, Figure A. 1.

For the 1st test case, Reduced ASHINO's and MatOnto, the manual reference alignment created was including all possible types of logical relations between ontologies' classes, subclass (\subseteq), superclass (\supseteq) and equivalence ($=$), although the matchers only get the equivalence type of logical relations of all the alignments. However, the idea is to evaluate ontology matchers upon these logical relations to show how matchers show such types of other logical relations implicitly in their calculations and what should be done to the matchers to be able to output such other types of relations as alignments in the future.

The other two manual reference alignments created for both test cases two and three include only the equivalence logical alignments between the ontologies' classes, to be able to evaluate complete ontologies used as test cases, since the creation process of a manual reference alignment is very time intensive. By the help of AML tool's GUI these manual reference alignments are created by same way as the first manual reference alignment is created.

4.2.4 Background Knowledge Ontologies

As mentioned previously about the AML tool, it provides the matching using the background knowledge matcher. For that, a background knowledge ontology for the Materials Sciences domain has to be created. The creation process for a background knowledge ontology for a certain domain is as important, difficult and sophisticated as the creation process of any other type of ontology. The difficulty of creating any ontology comes from the idea of the philosophy of the decision how to decide on terms that are going to be included and under which category they should be added. [61] For that, the decision of creating an ontology that only includes synonyms of the terms in classes and relations, which is the case for the background knowledge ontology, is as important and sophisticated as other ontologies' creation. The background knowledge ontology acts as a semantic bridge, a midway in the matching process between the two input ontologies. If a class from the first ontology is not lexically aligned to another class from the second ontology, but they are defined to be synonymous of each other in the background ontology, then they will be aligned together as equivalent based on the knowledge provided by the background knowledge ontology.

AML is designed to allow the tool's user to add any background knowledge ontology for usage in the store/knowledge directory of the tool, and then it will automatically appear in the background knowledge matcher's options' list, such that either all the background knowledge ontologies can be chosen from the list or a combination of the ontologies or just one of them.

AML's ontology matchers are primary lexically based. This means that, in order for the background knowledge matcher to learn, for example, that Al class (as an atom) and Aluminum class are synonyms, it has to be declared as annotations (e.g. `rdf:label`) of the same OWL class inside the background knowledge ontology. Declaring Al and Aluminum as two equivalent OWL classes of each other will not work, and will actually preclude AML's other ontology matchers from matching the classes, since the equivalence is already expressed in the input source and target ontologies. Therefore, the background knowledge matcher only attempts to find alignments that have not already been found by other AML matchers, since the background knowledge matcher is a secondary ontology matcher in AML.[62] The background knowledge is only useful when it gets more correct alignments than the ones already detected using the primary ontology matchers, so choosing the optimal background knowledge ontology for the matching of the test cases is very important.

Creating a background knowledge ontology for the thesis' evaluation approach has to be very specific, since creating a full background knowledge ontology for the Materials Sciences domain is neither the aim, nor feasible during this thesis. For that reason, a representative specific case was spot out from the test cases creation. Accordingly, the Periodic Table Background Knowledge Ontology, as can be found in Appendix A.2.3, was created to serve this single specific case in terms of evaluating and discussing the background knowledge ontology matcher for the Materials Sciences domain. This case was the 2nd test case, which was between ASHINO's and MatOnto; the two Materials Sciences domain specific ontologies. Both ontologies include classes *element* \subseteq *pure substance* \subseteq *chemical substance*, the element class is a superclass for all the periodic table's elements, each is defined as classes in both of the ontologies. However, in ASHINO's Materials Sciences ontology, the periodic table's elements are defined only as abbreviations, for example, the Aluminum is only written as "Al", on the other hand in MatOnto the periodic table's elements are written only with names and no abbreviations where related to the classes. Neither one of the ontologies defines that "Al" is an abbreviation for Aluminum and so on. Therefore, the whole periodic table's elements will not be detected by the lexical matchers. The background knowledge ontology can solve this problem acting as a mediator between the test case's two ontologies. Accordingly, the created background knowledge ontology was all about having all elements of the periodic table, and in the same class of each element the (`rdf:label`) includes the element's abbreviation.

Nevertheless, to further discuss the Materials Sciences ontologies acting as background knowledge ontologies, the EMMO was also chosen as a background knowledge ontology since it incorporates a rich hierarchy and is designed for multiscale materials modelling. In consequence, the importance of the choice of the corresponding background knowledge ontology for each ontology matching process can be discussed.

4.3 Performance Evaluation Schema

In Section 3.4, the basic principles of evaluating ontology matchers were explained. The following three measurements " m " are calculated for the evaluation of matchers in this thesis:

- Precision (correctness)
- Recall (completeness)
- F-measure

The runtime of the matchers was also included; however, all the matchers took a range from five to ten seconds in loading and matching the benchmark. For that reason, only the three previously mentioned measurements are used, since they cover all needed evaluation perspectives as the ones covered in the OAEI.

Formulas for the three measurements explained in Section 3.4 are used in calculations. Since the values of " m " are by definition in a range between 0 and 1, the output is given in percentages.

In the thesis, the AML tool is used to calculate the measurements, since it enables the evaluation of any alignment, " A ", against any reference alignment, " R ". The correctness of the AML output values " m " for recall, precision and F-measure was verified manually on a random basis.

4.4 Performance Evaluation Workflow for Matching of Materials Sciences Ontologies

So far in this Chapter, the ontology matchers that are going to be evaluated are chosen, which are the AML's matchers and LogMap ontology matcher. Then, the Materials Sciences benchmark is created, consisting of three test cases, a manual reference alignment for each test case and two background knowledge ontologies. After that, the evaluating measurements are defined. In this section, the workflow of the evaluation process, the experiments' design, is discussed.

For the AML, all previously explained ontology matchers are fixed, except for the property and the background knowledge ontology matchers, since all other matchers are designed to get the lexical alignments between classes. [55] So using them all together was the choice for the evaluation since their result will be directly compared to the LogMap ontology matcher's results, since it depends totally on the lexicon and not the property or the background knowledge.

Table 4. 1 shows experiments (from 1 to 5) executed for every test case using the combination of the chosen ontology matchers from AML's ontology matching tool, such that each experiment is designed by combining ontology matching techniques from AML. The LogMap ontology matcher is used once for every test case to be the experiment number 6, as shown in Table 4. 2.

In the first experiment, AML - L, all AML's lexical matcher are used without both the property and the background knowledge matchers. This experiment is designed to compare the lexical AML's matchers' output with the sixth experiment, LogMap - L, lexical LogMap's output.

In the second experiment, AML - LP, all AML's lexical matcher are used along with the property and without the background knowledge matchers. This experiment is designed to test the effect of the property matcher when used in the second experiment and not the first.

In the third experiment, AML - LB_{pt}, all AML's lexical matcher are used along with the background knowledge matcher (Period Table background knowledge ontology) and without the property matcher. It is designed to test the effect of the background knowledge matcher on the matching results.

In the fourth experiment, AML - LPB_{pt}, is designed with all AML's matchers, the lexical, the property and the background knowledge (Period Table background knowledge ontology). The experiment is designed to be able to evaluate the results of using all AML matchers together.

In the fifth experiment, AML - LPB_{EMMO}, is same as the fourth, but instead of using the periodic table ontology as a background knowledge, EMMO is used as a background knowledge ontology. It is designed to be able to see how the choice of the background knowledge ontology used will affect the results.

The sixth and last experiment, LogMap - L, is designed to be compared to the first experiment comparing lexical matchers techniques designed in LogMap against the ones designed in AML.

Table 4. 1: The AML's ontology matchers' combinations that are designed as experiments for the three Materials Sciences test cases. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology.

Experiment No.	Experiment ID	AML Property Matcher	AML Background Knowledge Matcher	Background Knowledge Ontology Used
1	AML - L	—	—	—
2	AML - LP	✓	—	—
3	AML - LB _{pt}	—	✓	Periodic Table
4	AML - LPB _{pt}	✓	✓	Periodic Table
5	AML - LPB _{EMMO}	✓	✓	EMMO

Table 4. 2: The LogMap ontology matcher that is designed to be experiment number 6 for the three Materials Sciences test cases. In the column "Experiment ID", L stands for lexical.

Experiment No.	Experiment ID	LogMap Ontology Matcher
6	LogMap - L	✓

Accordingly, in total, six experiments are going to be carried out for three different test cases, 1st, 2nd and 3rd test case, and for every experiment with a test case, there are three results of the evaluation (precision, recall and F-measure). So finally, $(3 \times 6 \times 3) = 54$ results will be evaluated, analyzed and discussed in the upcoming chapter.

5 Results and Discussion

In this chapter, the results from applying the evaluation workflow developed for the Materials Sciences domain will be presented and discussed, starting by presenting and discussing the ontology matching results of every experiment out of a total of six experiments, explained in the previous chapter, using the developed Materials Sciences benchmark in Section 5.1. Then in Section 5.2, the evaluation results for every experiment will be presented with a detailed discussion of each experiment. Ways of improvements will be also tackled in the same section.

5.1 Number of Alignments Using the Materials Sciences Benchmark

The experiments, designed as explained in Section 4.4 in Table 4.1 and Table 4.2, address different combination of AML's ontology matchers and the LogMap ontology matcher. The first five experiments are designed using AML's all lexical ontology matchers explained in the previous chapter. Nevertheless, the property and the background knowledge matchers from AML are also used, but with different combinations, as also explained in Section 4.4. The sixth experiment is done using LogMap, which is a lexically based ontology matcher.

The following Table 5.1 displays the number of alignments each experiment [A] achieved in each one of the three test cases; the corresponding alignments are presented in Appendix A.3. For the 1st test case, which is between the Reduced ASHINO's ontology and MatOnto, a total of 13 output alignments can be observed for the first and the third experiments, AML - L and AML - LB_{pt} respectively. 17 alignments are output from the second, fourth and fifth experiments but only one alignment can be observed in the case of the sixth experiment, LogMap - L. The 3rd test case, between the upper level ontology EMMO and the complete ASHINO's ontology, results in similar values of 9 alignments for the first and third experiments, 10 alignments for the second, fourth and fifth experiments and only 1 alignment for the sixth experiment. In contrast to that, the 2nd test case, between two complete Materials Sciences domain specific ontologies, which are the complete ASHINO's and MatOnto, shows a varying number of output alignments for each of the experiments ranging between a total of 88 and 284 output alignments.

Table 5. 1: The table shows the total number of alignments of each ontology matching experiment performed with each of the three test cases of the Materials Sciences benchmark. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology.

Experiment No.	Experiment ID	1st Test Case A 	2nd Test Case A 	3rd Test Case A
1	AML - L	13	101	9
2	AML - LP	17	105	10
3	AML - LB _{pt}	13	280	9
4	AML - LPB _{pt}	17	284	10
5	AML - LPB _{EMMO}	17	105	10
6	LogMap - L	1	88	7

For the 1st and 3rd test case, small and recurring number of output alignments between 1 and 17, as well as 7 and 10, respectively, can be observed. In contrast, the 2nd test case produces large varying numbers of alignments between 88 and 284 output alignments for the individual experiments.

Discussion of the Number of Alignments

The fourth experiment, AML - LPB_{pt}, shows the highest numbers of alignments for all the three test cases, followed by the third experiment, AML - LB_{pt}, which shows the second highest results especially in the 2nd test case, giving an indication that the periodic table background knowledge ontology has dramatically boosted the number of alignments, from 105 to 280, resulted from the matching, in comparison to the second experiment, AML - LP. The reason for this is that the background knowledge is designed specifically to find the correspondence between the periodic table elements that could not be found without the background knowledge ontology, as explained before in Section 4.2.3.

The second and fifth experiments, AML - LP and AML - LPB_{EMMO}, show an equal number of alignments results, which indicates that the EMMO used as a background knowledge ontology in the fifth experiment does not affect the number of alignments resulted from the matching. Giving an indication that the EMMO acting as a semantic bridge between the two matched ontologies of the test cases did not include any synonyms, or other information, supporting the matching process to get more alignments.

Finally, the lexical only matchers' comparison done to the lexical techniques used in the first and sixth experiments, AML - L and LogMap - L. The results show that

the first experiment using AML's lexical matchers has achieved a noticeably higher number of alignments than the sixth experiment designed using LogMap, by around 2 to 13 more alignments.

The sixth experiment, which was designed using LogMap, shows the least number of alignments. This can be expected, as the tool uses a repair algorithm, which is responsible for getting matching outputs of only the highly reliable alignments with a higher weight between matched classes. This will eventually make its results higher in precision than other ontology matchers that do not use this repair algorithm.

The graph represented in Figure 5. 1 also shows the number of alignments resulted for every experiment for every test case explained in this section.

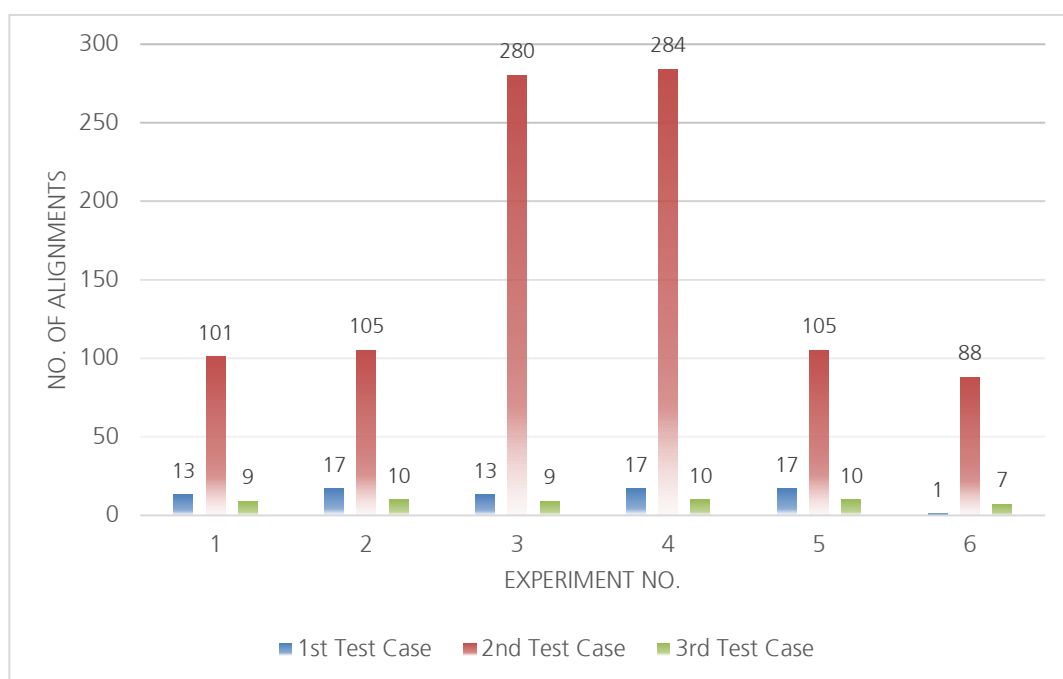


Figure 5. 1: The graph displays the total number of output alignments resulting from each ontology matching experiment performed with every test case of the Materials Sciences benchmark. It could be seen that in the 2nd test case the results are boosted especially in the third and fourth experiments.

From the results shown in the Figure 5. 1, it could be stated that the more matchers are used, the higher the number of alignments. As by using the property matcher in the second and fifth experiments, the results are increased slightly in comparison to the first and sixth experiments, which can be seen clearly

in the 1st and 3rd test cases. In the 1st test case, the number of alignments are increased from 13 to 17 alignments, and from 9 to 10 alignments in the 3rd test case, while results of the 2nd test case are dramatically increased by 196 more alignments when the background knowledge ontology matcher is used along with periodic table background knowledge ontology in the third and fourth experiment. Finally, the fourth experiment with all AML's ontology matchers has the highest number of alignments of 284 in comparison to the other experiments.

The number of output alignments from each experiment alone cannot be a full indicator of how good or bad an ontology matcher is performing. The experiments have to be evaluated in terms of correctness (precision) of their alignments' results "*A*" when compared to the manual reference alignment "*R*" designed for each test case. As well as the completeness (recall) of the alignments "*A*" in comparison to the manual reference alignments "*R*".

5.2 Evaluation of Ontology Matchers using the Materials Sciences Benchmark

After getting the number of alignments $|A|$ resulted from each experiment, these results are compared to the manual reference alignments created for each test case in the Materials Sciences benchmark. In this comparison, the evaluation to the experiments is done, in terms of precision (correctness), recall (completeness) and the F-measure with $\alpha = 0.5$.

The following Table 5. 2 shows the total number of alignments in each manual reference alignment $|R|$ created for every test case, this will be mainly used to evaluate the recall of every experiment as explained in Section 3.4.

Table 5. 2: The table shows the number of alignments in every manual reference alignment created for each test case.

	1 st Test Case	2 nd Test Case	3 rd Test Case
No of Alignments $ R $	23	302	11

5.2.1 1st Test Case

In this test case, the alignments resulting from matching two domain specific ontologies are evaluated such that the manual reference alignment used for evaluating this test case includes the equivalent, sub and super classes' logical relations for alignments ($=$, \subseteq and \supseteq) and no property alignments are included. Given the fact that the experiments' ontology matchers are only capable of detecting the equivalent ($=$) logical relation, as discussed in Section 4.2.2, this test case is designed small in terms of the total number of entities in order to discuss the capability of the ontology matchers to show different kinds of logical relations alignments using only weights of each alignment and equivalence logical relations.

The evaluation results are expected to be very low in terms of both correctness (precision) and completeness (recall). Since the experiments' output alignments, "**A**", precision and recall with only equivalence logical relation will be eventually less than the alignments in the manual reference alignments, "**R**", with equivalence, sub and super class logical relations. The following Table 5. 3 shows the percentage of precision, recall and F-measure of every experiment.

Table 5. 3: The table shows the evaluation results for the alignments resulting from matching the 1st test case with the six experiments created in the Materials Sciences evaluation workflow. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology.

Experiment No.	Experiment ID	Precision	Recall	F-measure
1	AML - L	23.1 %	13.0 %	16.7 %
2	AML - LP	17.6 %	13.0 %	15.0 %
3	AML - LB _{pt}	23.1 %	13.0 %	16.7 %
4	AML - LPB _{pt}	17.6 %	13.0 %	15.0 %
5	AML - LPB _{EMMO}	17.6 %	13.0 %	15.0 %
6	LogMap - L	100.0 %	4.3 %	8.3 %

As expected, the results show low percentages in precision of 17.6 % or 23.1 %, depending on the experiment, except for the sixth experiment, LogMap - L, which resulted in 100 %. That is because the LogMap only presents the equivalence alignments with very high weights above or equal 90 %, so the only alignment detected by the experiment, which is with an equivalence logical relation, is correct. On the other hand, AML's matchers present all possible

equivalence alignments that are higher than or equal to 50 % in weight, so depending on the experiment, 77 % or 82 % of the correspondences were not precise since they are not equivalent, rather they were sub or super alignments.

Although AML's precision results were very low compared to LogMap, AML's matchers were able to detect other logical relations such as sub and super classes implicitly using only equivalence logical relation by the verity of weights percentages assigned to each equivalent alignment result. Alignments that are given high weights were the ones evaluated as precise. However, low weights alignments were not precise but at least detected by the matchers as alignments but with an incorrect logical relation.

The second, fourth and fifth experiments have the lowest precision percentage of 17.6 %. These experiments are designed using the property matcher along with other AML's matchers. This indicates that the property matcher generally results in slightly increasing the number of alignments $|A|$ by 3 to 7 alignments, depending on the experiment. However, these increased alignments are not correct, accordingly the precision decreased by around 6 % when the property matcher is used. This promotes stating that the property matcher is not mature enough to detect meaningful property alignments. That is why the manual reference alignments were designed to include alignments of classes but not alignments of properties. The property matcher is only evaluated for its ability of getting more alignments between classes using the properties aligned between them, as explained in Section 4.1.2. In the thesis the property matcher has equivalently aligned classes with properties which are logically not correct, e.g. the class "unit_dimension" has been aligned equivalently to the property "is_unit_of", which promotes that the property matcher implementation is not designed to cover all cases of alignments. By revising the source code, it is detected that the implementations did not include the domain and range of the properties in the alignment decision, which is the main cause of the immature alignments results achieved by the matcher. For example, the property matcher will detect an equivalence of a property "has_pet" found in both of the two matched ontologies. The first ontology had the domain of a "child" and a range of a "dog" for the property, and the second ontology had a domain of a "patient" and a range of "scan results" for the same property, since "pet" in the second means "Positron Emission Tomography". The challenge for the property matcher is to spot out that although the properties are equivalent, the classes of the domain and range are not equivalent since the logic is not correct, and eventually do not align these classes together. So accordingly, in this example, the property matcher should discard the equivalence alignment "has_pet" as well as discarding the equivalence alignments between domain and range classes of the two ontologies. [34]

For the recall, the results are all low. 13.0 % for the experiments designed with AML's matchers and 4.3 % for the experiments designed with LogMap. This is

expected since the total number of alignments of all experiments $|A|$ are all less than the total number of alignments of the manual reference alignments $|R|$, by 6 to 22 alignments. That is due to the presence of other types of logically related alignments (\subseteq and \supseteq) that the ontology matchers are not able to spot.

The F-measure is a sum-up indication of both the precision and recall. The measure shows that the first and third experiments, AML - L and AML - LB_{pt}, designed without the property matcher and with lower alignments' weights' threshold of $\geq 50\%$, compared to the LogMap with weight's threshold of $\geq 90\%$, are performing best in terms of the F-measure value of 16.7 % among the six experiments. The following graph in Figure 5. 2 shows the results for every experiment in terms of precision, recall and F-measure.

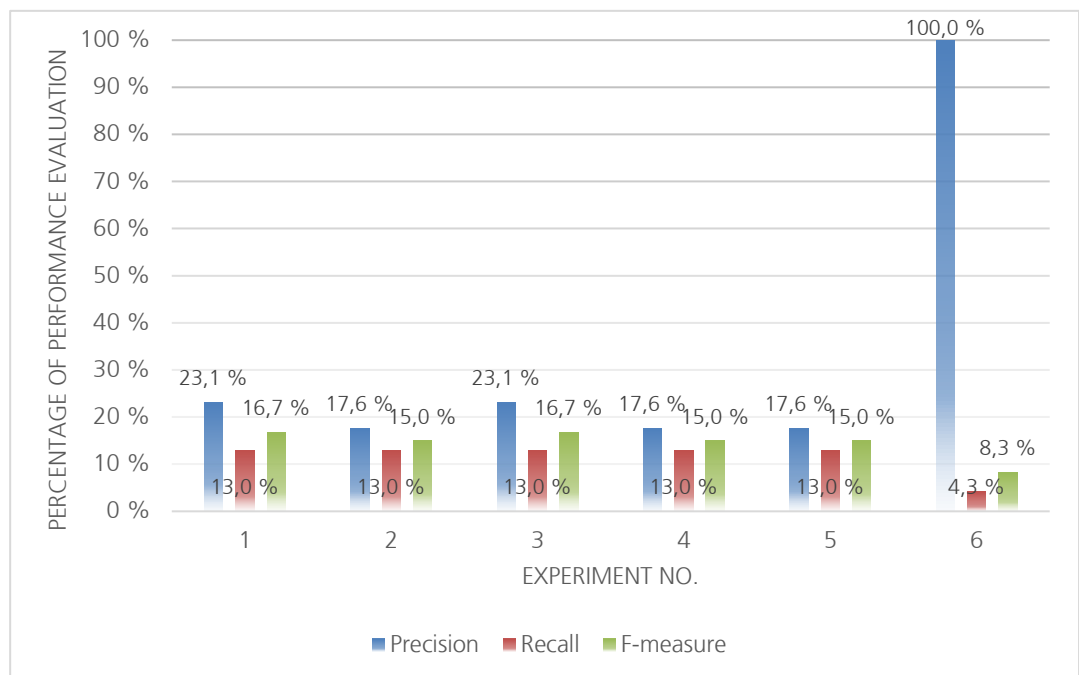


Figure 5. 2: The graph displays the evaluation results for the 1st test case when aligned by the six experiments' matching tools, representing combinations of ontology matchers; evaluation is performed by comparing each result's precision, recall and F-measure.

If the experiments are going to be applied on a bigger scale test case of the 1st test case, in terms of the number of entities, we can expect similar evaluation results growing with the growth of number of the classes and properties of the two matched ontologies. An approach to obtain better evaluation results in terms of the three evaluation measurements " m " is to use the weights assigned to the classes along with the properties between classes to detect other logical alignments such as sub and super class/property of (\subseteq and \supseteq). Also, meaningful

property alignments can be detected by linking the logic between the lexicons of the classes with the lexicons of the properties, taking domain and range into account. Properties can help in finding hierarchies between different classes; together with weights assigned to alignments it can be deduced that lower weights with strong hierarchical property can refer to sub or super alignments logical relation. Also as discussed in Section 4.1.2 about AML, the threshold can be adjusted to accept weights from 50 % to 100 %, so users of the tool can increase the threshold according to the matching application to ensure higher precision values if desired.

5.2.2 2nd Test Case

In this test case, a bigger scale of the 1st test case in terms of the number of entities is designed such that the complete domain specific ontology of ASHINO's is matched towards the complete MatOnto. However, in this case the manual reference alignment is designed to include only equivalence alignments of classes and no property alignments.

It is expected to see the effect of the periodic table background knowledge ontology used in experiment three and four, AML - LB_{pt} and AML - LPB_{pt}, in terms of achieving higher results of the three evaluating measurements "*m*", since the supporting ontology is especially designed for this test case. The following Table 5. 4 shows the evaluation results "*m*" for the performance of the six experiments in the 2nd test case.

Table 5. 4: The table shows the evaluation results for the alignments resulting from matching the 2nd test case with the six experiments created in the Materials Sciences evaluation workflow. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology.

Experiment No.	Experiment ID	Precision	Recall	F-measure
1	AML - L	68.3 %	22.8 %	34.2 %
2	AML - LP	65.7 %	22.8 %	33.9 %
3	AML - LB _{pt}	92.1 %	85.4 %	88.7 %
4	AML - LPB _{pt}	90.8 %	85.4 %	88.1 %
5	AML - LPB _{EMMO}	65.7 %	22.8 %	33.9 %
6	LogMap - L	69.3 %	20.2 %	31.3 %

As expected, the third and fourth experiments have boosted results, compared to the other experiments, in all the three evaluation's measurements from

65.7 % to 92.1 % in terms of precision, from 20.2 % to 85.4 % in terms of recall and from 31.3 % to 88.7 % in terms of the F-measure. The boost in values indicates that the presence of the supporting background knowledge ontology enables the ontology matcher to find more correct alignments between matched ontologies. Since it can act as a mid way, as a dictionary, between the two ontologies. But instead of a general dictionary, it is a scientific background knowledge with as much as possible scientific synonyms of the domain entities (classes, properties and individuals). The graph shown in Figure 5. 3, presents the results of the three evaluation measurements for every experiment, showing how the third experiment acted the best in this test case in terms of the three measurements, with the supporting background knowledge ontology and without the property matcher.

The third experiment is nevertheless better than the fourth in terms of the precision and the F-measure, because of the precision, which was slightly decreased by around 2 % in the fourth experiment due to the property matcher meaningless alignments. That is also why the first experiment has better F-measures than the second and the fifth.

Finally, the sixth experiment has shown the lowest recall percentage of 20.2 % and a lower precision percentage of 69.3 % than the third and fourth experiments. That is because LogMap does not use the background knowledge matcher, so no mid way ontology is used to help in finding more correct alignments. Although the LogMap has a high repair algorithm and eventually higher precision values, the precision of this experiment with this test case is not 100 % like the experiment's results with the other test cases. That is due to the fact that the two matched ontologies are complete domain specific ontologies with terminologies that cannot be verified only lexically, like the LogMap does. However, the verification of science or Materials Sciences in this case is very important as well. For example, LogMap has aligned a class "bohr_magneton" from the first ontology with a class "unit_bohr_magneton" from the second ontology, lexically they had a 90 % weight of equivalent alignment, but scientifically, a physical quantity cannot be aligned equivalently with a unit. That leads to the conclusion that the usage of a background knowledge ontology from the same domain as the matched ontologies is highly required to assist matchers to find more and verify the correctness of alignments not only lexically, but also domain wisely. Nevertheless, the matcher does not also include property matching, which makes the results of precision slightly better than in the second and fifth experiments.

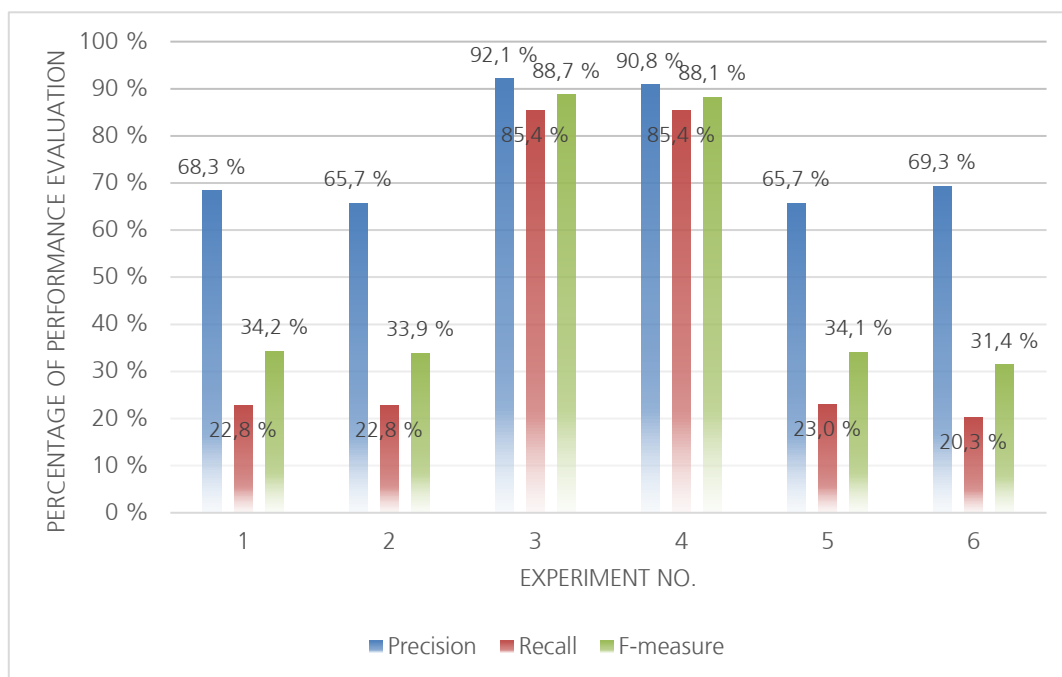


Figure 5. 3: The graph displays the evaluation results for the 2nd test case when aligned by the six experiments' matching tools, representing combinations of ontology matchers. Evaluation is performed by comparing each result's precision, recall and F-measure. Results show high results for the three measurements for the third and fourth experiments.

To improve the results of matching two complete domain specific ontologies using a background knowledge matcher along with lexical matchers, a supportive Materials Sciences background knowledge ontology has to be developed. Any Materials Sciences ontology can be used as background knowledge, such as EMMO, which is used in the fifth experiment. However, the recall results did not improve when compared to the first experiment, since EMMO does not include enough supporting information (e.g. synonyms in case of AML background knowledge matcher) acting as mid way between the matched ontologies' entities. Accordingly, either a specific ontology is to be developed only addressing synonyms between Materials Sciences known entities or during any Materials Sciences ontology creation, creators should take into account adding synonyms of the entities they choose in their ontology.

5.2.3 3rd Test Case

The test case is mainly designed for Materials Sciences domain ontologies that are needed to base on an upper level ontology, in order to be interoperable with other domain specific ontologies, which are already basing on the same or similar upper level ontologies. In this respect, the complete ASHINO'S is matched with EMMO.

It is expected not to get many alignments since the upper level ontology has more general entities than the domain specific ontology. Generic entities are like "set" and "item" in EMMO, and domain specific entities are like "Young's Modulus" and "Shear Modulus" in ASHINO'S. So alignments are going to be just the ones to know where to connect the domain specific ontology to the upper level ontology. The manual reference alignment "*R*" created for this test case also has only the equivalence alignments to be able to evaluate the ontology matchers' abilities.

The following Table 5. 5 shows the evaluation results for the performance of the six experiments' performance in the 3rd test case compared to the manual reference alignment.

Table 5. 5: The table shows the evaluation results for the alignments resulted from matching the 3rd test case with the six experiments created in the Materials Sciences evaluation workflow. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology.

Experiment No.	Experiment ID	Precision	Recall	F-measure
1	AML - L	88.9 %	72.7 %	80.0 %
2	AML - LP	80.0 %	72.7 %	76.2 %
3	AML - LB _{pt}	88.9 %	72.7 %	80.0 %
4	AML - LPB _{pt}	80.0 %	72.7 %	76.2 %
5	AML - LPB _{EMMO}	80.0 %	72.7 %	76.2 %
6	LogMap - L	100 %	63.6 %	77.8 %

Similar to the 1st and 2nd test cases, the evaluation results of the 3rd test case show lower precision results of 80 % for the second, fourth and fifth experiments compared to other experiments. This is due to the property matcher usage in matching. Nevertheless, the highest precision result of 100 % is in experiment six due to the repair algorithm used by LogMap allowing only high weighted alignments of ≥ 90 %.

In terms of recall, all AML's matchers, in the first five experiments, got same higher percentage of 72.7 %. This means that the periodic table background ontology did not affect the test case, which is expected since this ontology includes synonyms of atoms and atoms' abbreviations, which does not help in this case, as there are not atoms and elements entities in both ontologies to be matched. The LogMap shows the lowest recall percentage of 63.6 %, because of the previously mentioned reason of the repair algorithm that allows only highly weighted alignments, that is why the LogMap results are never 100 % complete in the three test cases.

Last but not least, the F-measure shows better overall results of 80 % for the first and third experiments done without using the property matcher due to the immature alignment output resulted from the property matcher, as mentioned before. In this test case, the property "is_property_for" is equivalently aligned with the class "property_name", which is logically not correct. Finally, when the matched ontologies are not from the same abstraction level, upper and domain specific ontologies, evaluation results of the matchers' combinations done in the six experiments are very close, as shown in the following graph in Figure 5. 4. That is due to the low number of alignments, of 7 to 10 in this test case, resulted from matching the upper level and the domain specific ontologies. Since not many equivalences are expected from matching different level ontologies, instead super and sub logical relations are expected to be included.

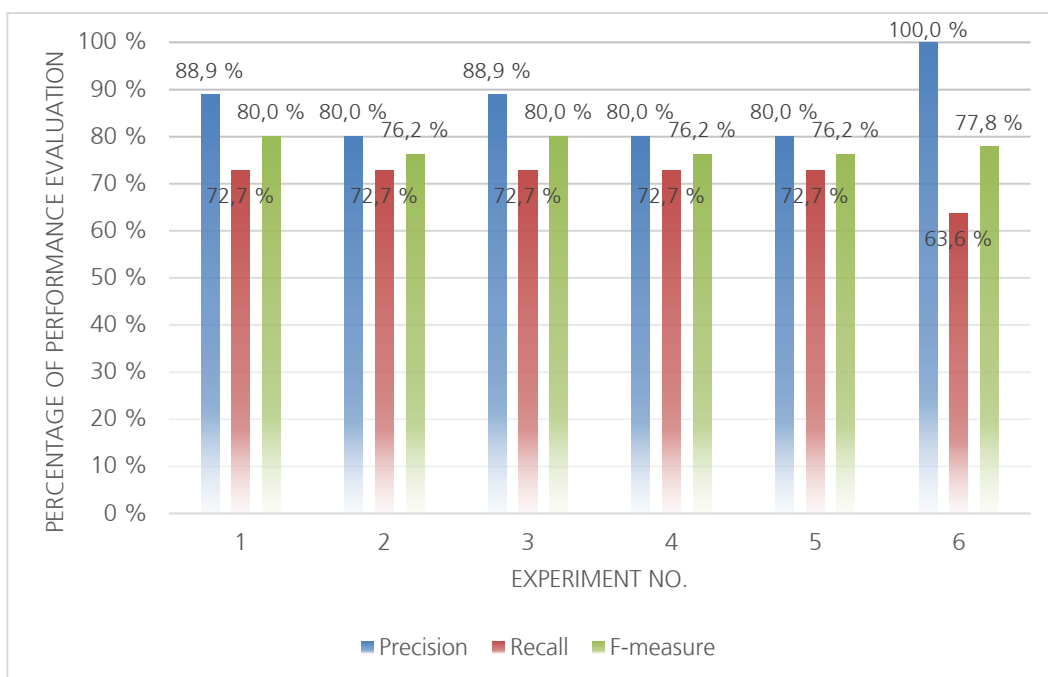


Figure 5. 4: The graph displays the evaluation results for the 3rd test case when aligned by the six experiments' matching tools, representing combinations of ontology matchers. Evaluation is performed by comparing each result's precision, recall and F-measure.

For the approach of matching an upper level ontology with a domain specific ontology, it would be more meaningful to find other logically related alignments between entities of the matched ontologies, not only equivalence but also sub and super logical relations. Since the main goal is to hierarchically integrate the domain specific ontology with the upper level ontology, so the presence of the sub and super classes/properties alignments would work more practically along with equivalence alignments to attach the two ontologies afterwards, knowing where to add entities from the domain specific ontology to the upper level ontology.

6 Conclusion and Future Work

In this chapter, a full coverage conclusion will be presented in Section 6.1, and suggested improvements addressing all discussed challenges in the previous chapter will be tackled in Section 6.2.

6.1 Conclusion

In the Materials Sciences field, the usage of ontologies to document, exchange or reuse data is still developing. Accordingly, not all software and tools designed to process ontologies are tested or applied in the Materials Sciences domain. One type of these tools is the ontology matching tools; these are used to find alignments between different ontologies to be able to use these alignments afterwards in multiple applications such as ontologies merging, query answering, ontology integration, data transformation, etc.

Every year since 2004, the Ontology Alignment Evaluation Initiative (OAEI), has been designing and executing an evaluation for ontology matching tools. But these evaluations are only done using biomedical and bioinformatics domains' benchmarks, since they are the most developed fields using ontologies while dealing with their data [3,10].

In this respect, the thesis approach was designed to adapt the OAEI evaluations performed in the biomedical and bioinformatics domains to the Materials Sciences domain. By first, creating a Materials Sciences benchmark, consisting of: (1) test cases of Materials Sciences ontologies, (2) manual reference alignments that are references to compare ontology matchers' experiments with, and (3) Materials Sciences background knowledge ontologies. After that, a comprehensive analysis of existing openly available ontology matchers was carried out and accordingly, ontology matchers for the thesis were chosen. Then, the ontology matching tools chosen are adapted for building and running by preparing their developing platform. Last but not least, the ontology matching tools were evaluated after being applied on the Materials Sciences benchmark. Finally, the resulting outputs from every ontology matching tool and every evaluation test case were discussed to show which matchers performed the best on which test case.

In the methodology, multiple ontology matching tools were investigated; and the considered matchers for the thesis' evaluations were the Agreement Maker Light (AML) tool and the LogMap tool. The Materials Sciences benchmark was designed to address all ontologies' abstraction levels for possible future Materials

Sciences ontologies' applications. The performance evaluation schema was adapted from the OAEI in terms of correctness, completeness and F-measure, an intermediate measure between correctness and completeness.

The evaluation results have shown that no matcher achieves 100 % precision and 100 % completeness but improved percentages, by around 2.8 times, in terms of the three evaluation measurements' results are achieved when the background knowledge matcher from AML is used along with the supporting background knowledge ontology, in comparison to other matchers. Correctness results were slightly negatively affected by around -5.5 % when the property matcher from AML is used. The LogMap's correctness of the alignments compared to all the other matchers is positively affected due to the tool's repair algorithm. Finally, when the matched ontologies are not from the same abstraction level, upper and domain specific ontologies, evaluation results of the matchers act similarly in this thesis giving similar measurements results.

From the evaluation results, the usage of LogMap is recommended for applications that demand high precision. A supporting Materials Sciences background knowledge ontology is essential in getting more correct and complete alignments results when matching ontologies from the Materials Sciences domain. The usage of AML's property matcher is currently not recommended until it is improved to include logically correct alignments. Finally, the fully automated complete and correct ontology matching is not achieved. That is because none of the results was 100 % complete and 100 % correct, since all alignments are only equivalences ($=$), and no sub or super (\subseteq and \supseteq) alignments are detected by all of the matchers. Accordingly, human interference in adjusting alignments results is essential until this challenge is addressed. Thus, ontology matching tools such as AML with a user interface allowing the manual editing of alignments is very useful.

The thesis approach was to adapt the automatic ontology matching evaluations to the Materials Sciences domain, and for that, a benchmark was created, matching tools were adjusted, matching schemes were adapted and a workflow was defined in order to perform the evaluations. The evaluation results have proven the possibility of matching Materials Sciences different types of ontologies using the matching tools; and that it is possible for the Materials Sciences field to compete and take part in yearly evaluations' benchmarks, same as other scientific fields, in the OAEI or any upcoming ontology matching evaluations.

6.2 Future Work

An updated Materials Sciences benchmark has to be developed to be used in a yearly evaluation for the existing ontology matchers. That is because the topic of ontologies in all fields and also in the Materials Sciences field is rising since the

automation approached by ontologies is necessary, as a lot of actors in the field need to be connected fast and dynamically. So eventually, those automated approaches need to be evaluated.

Materials Sciences test cases created in the thesis were limited to the number of Materials Sciences ontologies that already exist and are available on the web. So for the future, the improvement of Materials Sciences test cases based on the more developed Materials Sciences ontologies is going to improve the reliability of the evaluation, such that the test cases that are used to evaluate the ontology matchers are more representative and statistically evaluable for the Materials Sciences domain.

Other Materials Sciences background knowledge ontologies have to be created including correct supporting synonyms, definitions and other helping entities' correlations that are used by materials scientists, to act as a supporting semantic bridge between matched ontologies.

Manual references alignments created for the thesis' Materials Sciences benchmark were performed by three scientists. So in the future, it would be favorable if more scientists took part in the creation process, to increase the validity of alignments. A survey can be one of the approaches to do so, such that a survey with all entities and all possible logical relations in a form of multiple choice questions is distributed. Then scientists' responses will be collected, analyzed and discussed. Finally based on the survey's results, the manual reference alignments will be created and statistically valid.

Using upper level ontologies, such as EMMO, designed for the Materials Sciences domain ontologies is very important to sustain interoperability between developed and developing Materials Sciences ontologies. Accordingly, the improvement of already existing ontologies or even developing new ones has to be carefully performed in terms of physiology, scientific correctness of the chosen terms, applicability and interoperability with other existing Materials Sciences upper level ontologies such as EMMO and the BFO.

The examined ontology matching tools have to be improved in terms of the property matcher to be able to detect only logically correct property alignments between matched ontologies. Integrating property relations between classes in the lexical matchers is essential to improve finding alignments between different hierarchical classes of the matched ontologies. Finding alignments between properties can help defining their domain and range classes that are aligned by the lexical matcher, as sub or super classes of each other, and consequently finding correct logical alignments between these classes. Nevertheless, AML's property matcher includes the RelationshipMap data structure including all properties between classes, paths of these properties relating classes' root nodes and leaf nodes, and accordingly, this data structure can be used as a hierarchal

reference between lexically aligned classes and be the basis for the matcher improvements.

Nevertheless, the existence of other logically related alignments from the ontology matchers is essential since only equivalence alignments are not enough to automatically perform ontology matching without human interference. Accordingly, human interference is currently needed to be able to manually convert equivalence alignments to sub or super alignments. So upgrading the matchers to include hierarchies, sub and super, in the alignments is an important challenge for the matchers benefiting the output results of the matching and the dependent applications afterwards.

The matching techniques used in the evaluated ontology matchers could be updated to include machine learning in the alignments found. [8] For example, alignments evaluated by the users as wrong should not be detected again by the matching tool during the matching process of the same or similar ontologies. Parallelization is also a technique that can be used to save time for some industries that depend on the runtime matching process in applications like query answering. The parallelization will enable the tool to perform more than one matching process at the same time. Creating separate data structures for every matching process is one of the approaches to upgrade the tools in order to achieve parallelization.

Despite all discussed future work points, one main point should start taking place in the near future, which is the promotion of the participation of the created Materials Sciences benchmark in the OAEI. It would have two positive effects encouraging the improvement of the benchmark as well as the adaption of the ontology matching tools to serve the Materials Sciences domain.

7 List of Tables

Table 3. 1: Ontology matching tools, showing differences between the tools in terms of inputs, outputs, having a GUI or not and matching techniques [6,18,35]. 37

Table 4. 1: The AML's ontology matchers' combinations that are designed as experiments for the three Materials Sciences test cases. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology. 60

Table 4. 2: The LogMap ontology matcher that is designed to be experiment number 6 for the three Materials Sciences test cases. In the column "Experiment ID", L stands for lexical. 60

Table 5. 1: The table shows the total number of alignments of each ontology matching experiment performed with each of the three test cases of the Materials Sciences benchmark. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology. 62

Table 5. 2: The table shows the number of alignments in every manual reference alignment created for each test case. 64

Table 5. 3: The table shows the evaluation results for the alignments resulting from matching the 1st test case with the six experiments created in the Materials Sciences evaluation workflow. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology. 65

Table 5. 4: The table shows the evaluation results for the alignments resulting from matching the 2nd test case with the six experiments created in the Materials Sciences evaluation workflow. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology. 68

Table 5. 5: The table shows the evaluation results for the alignments resulted from matching the 3rd test case with the six experiments created in the Materials Sciences evaluation workflow. In the column "Experiment ID", L stands for lexical, P stands for property, B_{pt} stands for periodic table background knowledge ontology and B_{EMMO} stands for EMMO as background knowledge ontology. 71

8 List of Figures

- Figure 3. 1: A RoMM Review of Materials Models [7] which is a combination between a Physics/chemistry Equation (PE) and a Material Relation (MR). 18
- Figure 3. 2: A coupling or a concurrent multi scaling workflow of an iteration type, defined in the Review of Materials Models (RoMM) as a way of forming a chain of materials models, such that each model's processed output is given to the other model. 20
- Figure 3. 3: A coupling or a concurrent multi scaling workflow of a tightly coupled type, defined in the Review of Materials Models (RoMM) as a way of forming a chain of materials models such that all models are given the same user case input, then work together to output one combined raw output to be processed afterwards. 21
- Figure 3. 4: A simple example, part of University Taxonomy, showing different classes being super/sub classes of each other, with a relation "is_a", shown on solid arrows, between classes to show the hierarchical levels of the classes. 24
- Figure 3. 5: The ontology created from the University taxonomy presented in Figure 3. 4, showing the data (orange rectangles) and object (blue ovals) properties between (class – individual) in solid arrows and (class – class) in dotted arrows, respectively. [25] 25
- Figure 3. 6: The semantics spectrum of Knowledge Organization Systems [21,27], representing the data handling in terms of resources, e.g. time and money, starting with the lists with the lowest semantics until the highest semantically represented data in terms of ontologies. The highest the semantics level, the highest the time and money needed. 26
- Figure 3. 7: Ontology layers presented in the "Theory and Application of Ontology" [27], showing the three types of sub ontologies, starting from the most general, the upper ontology, the mid-level more specific ontology to the most specific domain ontology. 29
- Figure 3. 8: The ontology matching process, showing the inputs to the matching process. The inputs are the two ontologies to be matched (**O1** and **O2**) and helping resources such as the background knowledge ontologies and dictionaries. Nevertheless, the calculating parameters such as the threshold and the weights based on which alignments are decided to be kept or not as final output alignments, and finally an already existing alignment, which is an optional input that is needed to be extended. The output of the matching process is the alignment or an extended alignment, which includes all the alignments between the two input ontologies that their weights are higher than the threshold. [35] 34

Figure 3. 9: Two ontologies ($O1$ and $O2$) matching example. $O1$ representing one super class named "Person" of two corresponding subclasses, "Teacher" and "Student". $O2$ represents one super class named "Human" with one subclass named "Lawyer". Logical relations (alignments / correspondences) are outputs from the matching process between classes of different ontologies as shown on black arrows between $O1$ and $O2$. 35

Figure 3. 10: Performance evaluation scheme for ontology matching tools, which shows how the output from the matching tool " A " is evaluated against the reference alignment " R " and finally output " m ", which are the measurements used to compare ontology matching tools with. [12] 40

Figure 4. 1: Agreement Maker Light (AML) computational models for performing ontology matching. The first model is responsible for loading the two matched to be ontologies and any resource ontologies. The second model is responsible for performing the matching operation of the two input ontologies. 45

Figure 4. 2: Agreement Maker Light (AML) data structures, which are used to either save the ontologies entities in case of the Lexicon and the RelationshipMap data structures, or save the alignments' results in case of the Alignment data structure. 46

Figure 4. 3: Agreement Maker Light (AML)'s ontology matchers divided according to how essential they are in getting more alignment in the matching results into two categories; primary and secondary, respectively. All of the AML's ontology matchers are with time complexity of $O(n)$ except for the String matcher with a $O(n^2)$ time complexity. 50

Figure 4. 4: Summary of the three test cases compositions, created from Materials Sciences ontologies to be a part of the thesis' Materials Sciences benchmark, such that the green rectangles show the upper level ontologies EMMO and the BFO and the blue circles show the domain specific ontologies. The 1st test case is between a reduced subset of ASHINO's ontology, named Reduced ASHINO's, and complete MatOnto (BFO + MatOnto). The 2nd test case is between the complete ASHINO's ontology and complete MatOnto (BFO + MatOnto). Finally, the 3rd test case is between the EMMO upper level ontology and the complete ASHINO's ontology. 53

Figure 5. 1: The graph displays the total number of output alignments resulting from each ontology matching experiment performed with every test case of the Materials Sciences benchmark. It could be seen that in the 2nd test case the results are boosted especially in the third and fourth experiments. 63

Figure 5. 2: The graph displays the evaluation results for the 1st test case when aligned by the six experiments' matching tools, representing combinations of ontology matchers; evaluation is performed by comparing each result's precision, recall and F-measure. 67

- Figure 5. 3: The graph displays the evaluation results for the 2nd test case when aligned by the six experiments' matching tools, representing combinations of ontology matchers. Evaluation is performed by comparing each result's precision, recall and F-measure. Results show high results for the three measurements for the third and fourth experiments. 70
- Figure 5. 4: The graph displays the evaluation results for the 3rd test case when aligned by the six experiments' matching tools, representing combinations of ontology matchers. Evaluation is performed by comparing each result's precision, recall and F-measure. 72

9 List of Abbreviations

AIST: National Institute of Advanced Industrial Science and Technology	32
AM: Agreement Maker	43
AML: Agreement Maker Light	10
API: Application Programming Interface	43
ASMOV: Automated Semantic Mapping of Ontologies with Validation	42
BFO: Basic Formal Ontology	29
CODI: Combinatorial Optimization for Data Integration	42
DIKW: Data Information Knowledge Wisdom	28
DL: Description Logic	27
doid: Human Disease Ontology	39
DOLCE: Descriptive Ontology for Linguistic and Cognitive Engineering	29
EMMC: European Materials Modelling Council	17
EMMO: European Materials Modelling Ontology	30
FMA: Foundational Model of Anatomy	31
FOAM: Framework for Ontology Alignment and Mapping	42
FOL: First Order Logic	27
GFO: General Formal Ontology	29
GO: Gene Ontology	30
GUI: Graphical User Interface	35
HP: Human Phenotype Ontology	31
HTML: Hypertext Markup Language	27
IAO: Information Artifact Ontology	30
IDE: Integrated development environment	43
IDO: Infectious Disease Ontology	30
IDS: Industrial Data Space	35
JDK: Java Development Kit	44
JVM: Java Virtual Machine	43
LogMap. Logic-based Methods for Ontology Mapping	50
MDS: Materials Data Space	35
MODA: Modelling Data elements templates	21
MP: Mammalian Phenotype Ontology	31
MR: Material Relation	18
NCI: National Cancer Institute Thesaurus	31
NIMS: National Institute of Materials Science	32
OAEL: Ontology Alignment Evaluation Initiative	10
OGMS: General Medical Science	30
OWL: Web Ontology Language	25
PE: Physics/chemistry Equation	18

PO: Plant Ontology	30
POM: Project Object Model	35
PRO: Protein Ontology	30
RDF: Resource Description Framework	27
RDFS: RDF Schema	27
RiMOM: Risk Minimization based Ontology Mapping	36
RoMM: Review of Materials Models	17
SPH: Smoothed Particle Hydrodynamics	20
SUMO: Suggested Upper Merged Ontology	29
TREC: Text REtrieval Conference	38
uberon: Uber-anatomy ontology	39
UMLS: Unified Medical Language System	36
XML: Extensible Markup Language	27

10 References

- [1] M. Huschka and M. Dlugosch, "Konzept zur Digitalisierung der Datenverarbeitung aus werkstoffwissenschaftlicher Forschung," *Tagung Werkstoffprüfung 2019: Werkstoffe und Bauteile auf dem Prüfstand*, pp. 341–348, 2019.
- [2] P. Cheesman, "Powering Your Business By Leveraging Your Data. How MDM And Big Data Can Transform Your Business," <https://www.linkedin.com/pulse/20141021044524-3960295-lost-your-data-have-no-idea-who-your-customers-are-scared-you-are-losing-money-here-s-help/>.
- [3] R. Arp, B. Smith, and A. D. Spear, *Building ontologies with basic formal ontology*, The MIT Press, Cambridge, Massachusetts, London, England, 2015.
- [4] D. G. Roche, L. E. B. Kruuk, R. Lanfear et al., "Public Data Archiving in Ecology and Evolution: How Well Are We Doing?," *PLoS biology*, vol. 13, no. 11, e1002295, 2015.
- [5] E. V. Larson, "Interoperability of U.S. and NATO allied air forces: Supporting data and case studies," 2003.
- [6] L. Aroyo, C. Welty, H. Alani, et al., eds., *The Semantic Web – ISWC 2011*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [7] A. F. d. Baas, *What makes a material function? Let me compute the ways: modelling in H2020 LEIT-NMBP Programme materials and nanotechnology projects*, Publications Office, Luxembourg, 2017.
- [8] A. Doan, J. Madhavan, P. Domingos et al., "Ontology Matching: A Machine Learning Approach," in *Handbook on Ontologies*, S. Staab and R. Studer, Eds., vol. 1, pp. 385–403, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [9] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, "Ontology matching: A literature review," *Expert Systems with Applications*, vol. 42, no. 2, pp. 949–971, 2015.
- [10] E. Jiménez-Ruiz, "Ontology Alignment Evaluation Initiative::Home," 2/20/2019, <http://oaei.ontologymatching.org/>.
- [11] M. Ehrig and J. Euzenat, eds., *State of the art on ontology alignment*, 2004.
- [12] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [13] T. Ashino, "Materials Ontology: An Infrastructure for Exchanging Materials Information and Knowledge," *Data Science Journal*, vol. 9, pp. 54–61, 2010.
- [14] P. Mason, C. R. Fisher, R. Glamm, et al., eds., *European Materials Modelling Council: Proceedings of the 4th World Congress on Integrated Computational Materials Engineering (ICME 2017)*, Springer International Publishing, 2017.
- [15] J. Friis, "European Materials Modelling Ontology (EMMO)," <https://github.com/emmo-repo/EMMO>.
- [16] B. Heussler, "MatOnto-Ontologies," <https://github.com/inovexcorp/MatOnto-Ontologies>.
- [17] M. Bergman, "50 Ontology Mapping and Alignment Tools," AI3::Adaptive Information, <http://www.mkbergman.com/1769/50-ontology-mapping-and-alignment-tools/>.

- [18] D. Faria, C. Pesquita, E. Santos et al., "The AgreementMakerLight Ontology Matching System," in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, R. Meersman, H. Panetto, T. Dillon et al., Eds., pp. 527–541, Springer Berlin Heidelberg, 2013.
- [19] E. Jiménez-Ruiz and B. C. Grau, "LogMap: Logic-Based and Scalable Ontology Matching," in *The Semantic Web – ISWC 2011*, L. Aroyo, C. Welty, H. Alani et al., Eds., vol. 7031, pp. 273–288, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [20] G. Barroso, <https://www.lacan.upc.edu/admoreWeb/2018/05/all-models-are-wrong-but-some-are-useful-george-e-p-box/>.
- [21] H. Oberkamp, "From Big Data to Big Analysis: The convergence of Formal Semantics & Data Science in Life Sciences," OSTHUS GmbH, https://emmc.info/wp-content/uploads/2017/12/EMMC_IntOp2017_Cambridge_Oberkampf_OSTHUS.pdf.
- [22] P. Hitzler, M. Krötzsch, and S. Rudolph, *Foundations of semantic web technologies*, CRC Press, Boca Raton, Fla., London, New York, 2010.
- [23] Y. Franc, "FAIR Semantics: From common ontology design practices to a cross-disciplinary semantic ecosystem," 2019.
- [24] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg et al., "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific data*, vol. 3, p. 160018, 2016.
- [25] V. Jayawardana, "Ontology Generation and Visualization with Protégé," <https://medium.com/@vindulajayawardana/ontology-generation-and-visualization-with-prot%C3%A9g%C3%A9-6df0af9955e0>.
- [26] S. Harald, "What are and why are we studying ontologies," <https://www.slideshare.net/lysander07/what-are-and-why-are-we-studying-ontologies>.
- [27] R. Poli, *Computer applications*, Springer, Dordrecht, 2010.
- [28] D. P. Wallace, *Knowledge management: Historical and cross-disciplinary themes*, Libraries Unlimited, Westport, Conn, 2007.
- [29] H. Fliegl, "Platform MaterialDigital," <https://www.materialdigital.de/ausschreibung/>.
- [30] P. Thomas, P. Sternberg, C. Mungall et al., "Gene Ontology Resource," 10/30/2019, <http://geneontology.org/>.
- [31] E. Jiménez-Ruiz, "Ontology Alignment Evaluation Initiative::2019," 12/2/2019, <http://oaei.ontologymatching.org/2019/>.
- [32] K. Cheung, J. Drennan, and J. Hunter, "Towards an Ontology for Data-Driven Discovery of New Materials," 2008.
- [33] S. Stefan, T. Dahmen, F. Fischer et al., "Strategiepapi: Digitale Transformation in der Materialwissenschaft und Werkstofftechnik,".
- [34] V. Mascardi, A. Locoro, and P. Rosso, "Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 609–623, 2010.
- [35] P. Shvaiko and J. Euzenat, "Ontology Matching: State of the Art and Future Challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2013.
- [36] C. Haas and L. Nagel, "Industrial Data Space (IDS) - OPC Foundation," <https://opcfoundation.org/markets-collaboration/ids/>.
- [37] R. Wehrspohn, M. Fütting, and C. Homann, "Materials Data Space: Fraunhofer-Verbund MATERIALS," Apr. 2016.

- [38] Tim O'Brien, Stuart McCulloch, and Brian Demers, *Maven: The Maven Cookbook*, Sonatype, Inc., Mountain View, CA, 2010.
- [39] C. Fellbaum and R. Teng, "WordNet | A Lexical Database for English," 1/13/2020, <https://wordnet.princeton.edu/>.
- [40] Manel Achichi, Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, et al., *Results of the Ontology Alignment Evaluation Initiative 2016: OM: Ontology Matching, Oct 2016, Kobe, Japan*.
- [41] P. Shvaiko, D. Faria, E. Jiménez-Ruiz et al., *Results of the ontology alignment evaluation initiative 2017*, 2017-10-21.
- [42] P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, et al., eds., *Proceedings of the 9th International Conference on Ontology Matching*, CEUR-WS.org, Aachen, Germany, Germany, 2014.
- [43] D. Harman, E. Voorhees, and I. Soboroff, "Text REtrieval Conference (TREC) Home Page," <https://trec.nist.gov/>.
- [44] E. Jiménez-Ruiz, "Ontology Alignment Evaluation Initiative::2019," 12/2/2019, <http://oaei.ontologymatching.org/2019/seals/index.html>.
- [45] I. Cruz, "Agreement Maker," <https://github.com/agreementmaker/agreementmaker>.
- [46] W. Hu, "Falcon-AO: A practical ontology matching tool - Home," <http://ws.nju.edu.cn/falcon-ao/>.
- [47] P. Jain, "BLOOMS," <https://github.com/jainprateek/BLOOMS>.
- [48] E. Rahm, P. Arnold, H.-H. Do et al., "COMA 3.0 | Database Group Leipzig," 1/17/2020, <https://dbs.uni-leipzig.de/Research/coma.html>.
- [49] J. Huber, T. Sztyler, J. Noessner et al., <https://code.google.com/archive/p/codi-matcher/>.
- [50] J. van Zyl, "Maven – Maven Features," 11/20/2019, <https://maven.apache.org/maven-features.html>.
- [51] D. Hutchison, T. Kanade, J. Kittler et al., *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [52] D. Faria, "Agreement Maker Light AML-Project," <https://github.com/AgreementMakerLight/AML-Project>.
- [53] P. Thomas, P. Sternberg, C. Mungall et al., "FAQs," 1/10/2020, <http://geneontology.org/docs/faq/>.
- [54] I. Dickinson, "The Jena 2 Ontology API," 8/26/2005, <https://scis.athabascau.ca/html/lo/repos/comp492/jena/ontology/index.html>.
- [55] R. Meersman, H. Panetto, T. Dillon, et al., eds., *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, Springer Berlin Heidelberg, 2013.
- [56] D. Faria, <https://scholar.google.com/citations?user=cyiaqZ4AAAAJ&hl=en>.
- [57] E. Jiménez-Ruiz, B. C. Grau, W. Xia et al., "LogMap Family Results for OAEI 2014," in *Proceedings of the 9th International Conference on Ontology Matching*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz et al., Eds., pp. 126–134, CEUR-WS.org, Aachen, Germany, Germany, 2014.
- [58] E. Jiménez-Ruiz, B. C. Grau, Y. Zhou et al., "logmap-matcher," <https://github.com/ernestojimenezruiz/logmap-matcher>.
- [59] E. Jiménez-Ruiz, B. C. Grau, W. Xia et al., "LogMap Web Access," 4/2/2019, <http://krrwebtools.cs.ox.ac.uk/logmap/>.

- [60] E. Jiménez-Ruiz, B. C. Grau, Y. Zhou et al., "Large-scale interactive ontology matching: Algorithms and implementation," vol. 242, pp. 444–449, 2012.
- [61] D. Faria, C. Pesquita, E. Santos et al., "Automatic background knowledge selection for matching biomedical ontologies," *PloS one*, vol. 9, no. 11, e111226, 2014.
- [62] S. Spaccapietra, J. Z. Pan, P. Thiran et al., *Journal on Data Semantics XI*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Appendix

A.1. Ontology Matchers

A.1.1. Agreement Maker Light (AML)

Last Updated Pom File:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>aml</groupId>
  <artifactId>aml</artifactId>
  <packaging>jar</packaging>
  <version>2.1-SNAPSHOT</version>

  <name>AgreementMakerLight</name>
  <url>http://somer.fc.ul.pt/aml.php</url>

  <repositories>
    <repository>
      <id>jitpack.io</id>
      <url>https://jitpack.io</url>
    </repository>
  </repositories>
  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
  <build>
    <sourceDirectory>${basedir}/src</sourceDirectory>

    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-shade-plugin</artifactId>
        <version>2.4</version>
```



```

<executions>
  <execution>
    <phase>package</phase>
    <goals>
      <goal>shade</goal>
    </goals>
    <configuration>

<dependencyReducedPomLocation>${project.build.directory}/dependenc
y-reduced-pom.xml</dependencyReducedPomLocation>
    <filters>
      <filter>
        <artifact>*:*</artifact>
        <excludes>
          <exclude>META-INF/*.SF</exclude>
          <exclude>META-INF/*.DSA</exclude>
          <exclude>META-INF/*.RSA</exclude>
        </excludes>
      </filter>
    </filters>
    <!-- Additional configuration. -->
    <transformers>
      <transformer
implementation="org.apache.maven.plugins.shade.resource.ManifestReso
sourceTransformer">
        <mainClass>aml.Main</mainClass>
      </transformer>
    </transformers>
    </configuration>
  </execution>
</executions>
</plugin>
</plugins>
</build>
<dependencies>
  <dependency>
    <groupId>commons-lang</groupId>
    <artifactId>commons-lang</artifactId>
    <version>2.6</version>
  </dependency>

```

```

<dependency>
  <groupId>dom4j</groupId>
  <artifactId>dom4j</artifactId>
  <version>1.6.1</version>
</dependency>
<dependency>
  <groupId>org.semanticweb.elk</groupId>
  <artifactId>elk-owlapi-standalone</artifactId>
  <version>0.4.1</version>
  <classifier>bin</classifier>
</dependency>
<dependency>
  <groupId>org.gephi</groupId>
  <artifactId>gephi-toolkit</artifactId>
  <version>0.8.2</version>
  <scope>system</scope>
  <systemPath>C:\Users\Nasr\Documents\GitHub\AML-
Project\AgreementMakerLight\src\lib\gephi-toolkit.jar</systemPath>
</dependency>
<dependency>
  <groupId>edu.smu.tspell.wordnet</groupId>
  <artifactId>jaws</artifactId>
  <version>1.3.3</version>
  <scope>system</scope>
  <systemPath>C:\Users\Nasr\Documents\GitHub\AML-
Project\AgreementMakerLight\src\lib\jaws.jar</systemPath>
</dependency>
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
<dependency>
  <groupId>com.memetix</groupId>
  <artifactId>microsoft-translator-java-api</artifactId>
  <version>0.6.2</version>
  <type>jar</type>
</dependency>
<dependency>
  <groupId>net.sourceforge.owlapi</groupId>
  <artifactId>owlapi-distribution</artifactId>
  <version>3.4.10</version>
</dependency>

```

```

<dependency>
    <groupId>com.github.firemaples</groupId>
    <artifactId>microsoft-translator-java-
api</artifactId>
    <version>v0.8.6</version>
</dependency>
<dependency>
    <groupId>uk.ac.shef.wit</groupId>
    <artifactId>simmetrics</artifactId>
    <version>1.6.2</version>
<scope>system</scope>
<systemPath>C:\Users\Nasr\Documents\GitHub\AML-
Project\AgreementMakerLight\src\lib\simmetrics.jar</systemPath>
</dependency>
<dependency>
    <groupId>org.swinglabs.swingx</groupId>
    <artifactId>swingx-all</artifactId>
    <version>1.6.4</version>
</dependency>
</dependencies>
</project>

```

A.1.2. LogMap [59]

How to run LogMap:

First Step:

Clone from GitHub by using GitHub Desktop or Command line on (e.g. Gitbash): <https://github.com/ernestojimenezruiz/logmap-matcher.git>

Second Step:

Add manually google translate by navigating to the cloned folder of log map then run the following command:

```
mvn install:install-file -Dfile=lib/google-api-translate-java-0.97.jar -
DgroupId=com.googlecode -DartifactId=google-api-translate-java -
Dversion=0.97 -Dpackaging=jar
```

Third Step:

Navigate to the Jar created file in target in the same cloned folder then run the following command:

```
java -jar logmap-matcher-3.0.jar
```

Or

Go to Eclipse and run the project using maven install to create the jar file then use it.

Fourth Step:

To use it to match, use command line after performing the third step

LogMap can operate as an ontology matching systems (MATCHER) or as a mapping debugging system (DEBUGGER). Additionally it also converts mappings from RDF-OAEI format to OWL.

- LogMap MATCHER facility requires 5 parameters:
 1. MATCHER. To use the matching functionality.
 2. IRI ontology 1. e.g.: <http://myonto1.owl> or <file:/C:/myonto1.owl> or <file:/usr/local/myonto1.owl>
 3. IRI ontology 2. e.g.: <http://myonto2.owl> or <file:/C:/myonto2.owl> or <file:/usr/local/myonto2.owl>
 4. Full output path for mapping files and overlapping modules/fragments. e.g. [/usr/local/output_path/](#) or [C://output_path/](#)
 5. Classify the input ontologies together with the mappings. e.g. true or false

```
For example: java -jar logmap2_standalone.jar MATCHER
file:/home/ontos/cmt.owl file:/home/ontos/ekaw.owl
/home/mappings/output true
```

- LogMap DEBUGGER facility requires 8 parameters:

1. DEBUGGER. To use the debugging facility.
2. IRI ontology 1. e.g.: <http://myonto1.owl> or <file:/C:/myonto1.owl> or <file:/usr/local/myonto1.owl>
3. IRI ontology 2. e.g.: <http://myonto2.owl> or <file:/C:/myonto2.owl> or <file:/usr/local/myonto2.owl>
4. Format mappings e.g.: OWL or RDF or TXT
5. Full IRI or full Path:
 - a. Full IRI of input mappings if OWL format. e.g.: <file:/C:/mymappings.owl> or <file:/usr/local/mymappings.owl> or <http://mymappings.owl>
 - b. or Full path of input mappings if formats RDF or TXT. e.g.: C:/mymappings.rdf or /usr/local/mymappings.txt
6. Full output path for the repaired mappings: e.g. /usr/local/output_path or C:/output_path
7. Extract modules for repair?: true or false
8. Check satisfiability after repair using Hermit? true or false

For example: java -jar logmap2_standalone.jar DEBUGGER
<file:/home/ontos/cmt.owl> <file:/home/ontos/ekaw.owl> RDF
 /usr/local/mymappings.rdf /home/mappings/output false true

- The RDF2OWL converter facility requires 4 parameters:

1. RDF2OWL. To transform from RDF-OAEL format to OWL. Note that the input ontologies are required to check the type of entity of the mapped IRIs.
2. IRI ontology 1. e.g.: <http://myonto1.owl> or <file:/C:/myonto1.owl> or <file:/usr/local/myonto1.owl>
3. IRI ontology 2. e.g.: <http://myonto2.owl> or <file:/C:/myonto2.owl> or <file:/usr/local/myonto2.owl>
4. Full path RDF mappings to be converted: e.g. C:/mymappings.rdf or /usr/local/mymappings.rdf

For example: java -jar logmap2_standalone.jar RDF2OWL
<file:/home/ontos/cmt.owl> <file:/home/ontos/ekaw.owl>
 /usr/local/mymappings.rdf

A.2. Materials Sciences Benchmark

A.2.1. Test Cases

Presenting all entities from every test case.

A.2.1.1. 1st Test Case

Reduced ASHINO's OWL file: is provided in a CD along with the thesis in folder A211.

MatOnto: is cloned from its GitHub source [16]

A.2.1.2. 2nd Test Case

Complete ASHINO's OWL file: is provided in a CD along with the thesis in folder A212.

MatOnto: is cloned from its GitHub source [16]

A.2.1.3. 3rd Test Case

EMMO: is cloned from its GitHub source [15]

Complete ASHINO's OWL file: is provided in a CD along with the thesis in folder A212.

A.2.2. Manual Reference Alignments

Manual reference alignments created for all three test cases are also provided on a CD along with the thesis in folder A22.

Manual Creation of the Reference Alignment Using AML

Figure A. 1 shows an example of creating a manual reference alignment using AML.

- Part (a) of the Figure, the two input ontologies of the test cases are chosen.
- Then in part (b), AML provides the manual creation of alignments by its two options of either adding classes' logical relations (Add Class Mapping) or even more by adding logical relations between the two ontologies' relations (properties), whether an object or data property, by choosing (Add Property Mapping).
- In part (c), the classes' names are chosen from the two ontologies respectively as well as the logical alignment relation between them.
- Finally, in part (d), all classes and the chosen alignments between them are saved and converted by the tool to an RDF format.

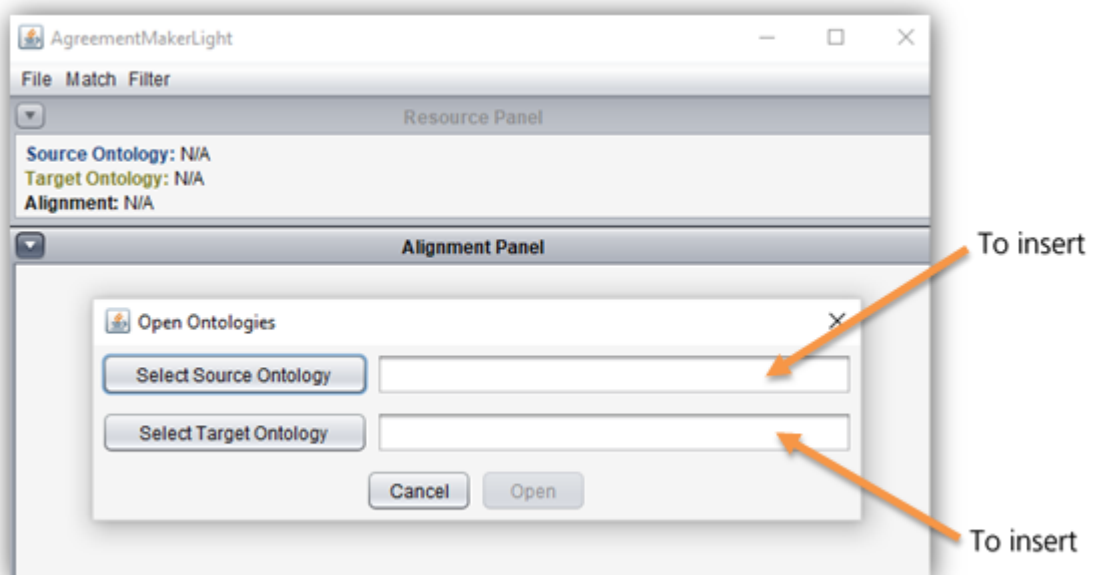


Figure A. 1 (a): AML GUI: choosing the two input ontologies (source *O1* and target *O2*).

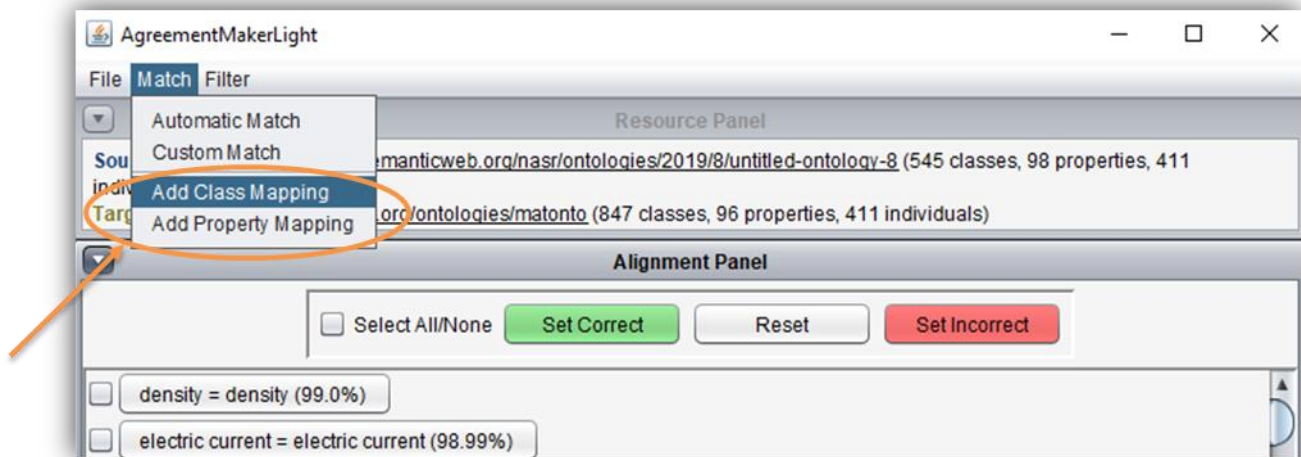


Figure A. 1 (b): AML GUI: choosing the option of adding a logical relation between either the ontologies' classes or properties.

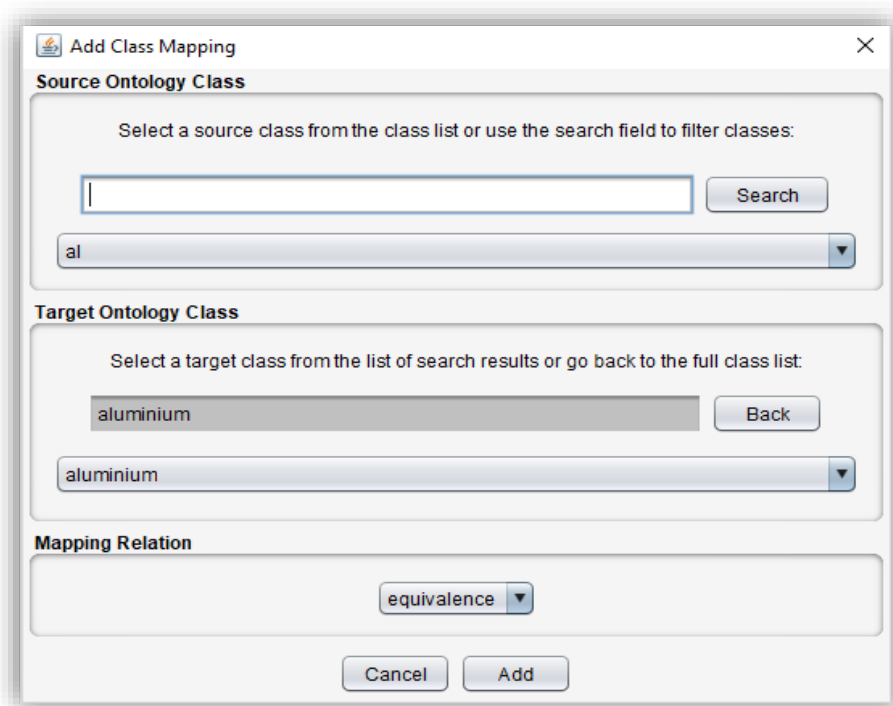


Figure A. 1 (c): AML GUI: the class, Al, is chosen from the first ontology, the class Aluminum is chosen from the second ontology and the logical alignment relation of equivalence is set. Finally, by pressing the Add button, the alignment will be added to the rest of the alignments that are chosen manually by the help of the AML.

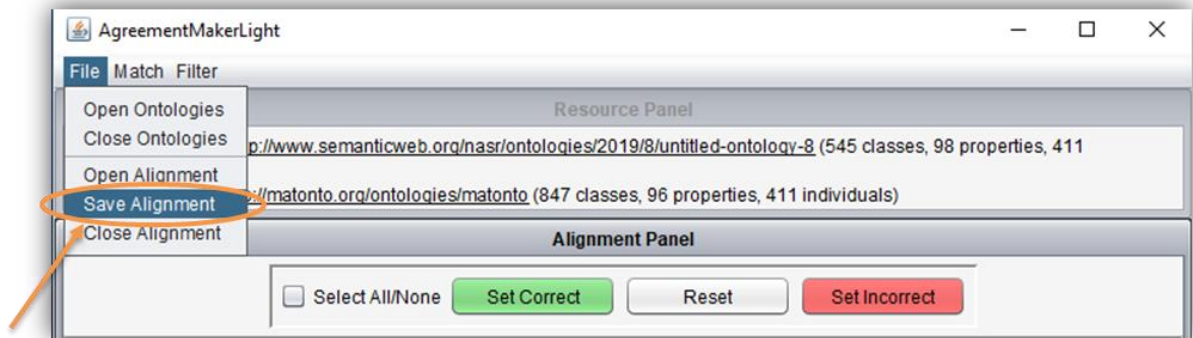


Figure A. 1 (d): AML GUI: saving the whole manually added alignments between the test case's two ontologies.

Figure A. 1: Using AML tool to create manual reference alignments of the created three Materials Sciences test cases.

A.2.3. Background Knowledge Ontologies

Periodic table ontology: is provided on a CD along with the thesis in folder A23.

EMMO: is cloned from its GitHub source [15]

A.3. Alignments of the Experiments performed using the Materials Sciences Benchmark

Presenting alignments' outputs obtained when matching the three test cases with the six experiments created from either AML's ontology matchers or LogMap ontology matching tool

A.3.1 First Experiment

Is provided on a CD along with the thesis in folder A31.

For the 1st Test Case

Is provided on a CD along with the thesis in folder A31 under Test Case 1.

For the 2nd Test Case

Is provided on a CD along with the thesis in folder A31 under Test Case 2.

For the 3rd Test Case

Is provided on a CD along with the thesis in folder A31 under Test Case 3.

A.3.2 Second Experiment

Is provided on a CD along with the thesis in folder A32.

For the 1st Test Case

Is provided on a CD along with the thesis in folder A32 under Test Case 1.

For the 2nd Test Case

Is provided on a CD along with the thesis in folder A32 under Test Case 2.

For the 3rd Test Case

Is provided on a CD along with the thesis in folder A32 under Test Case 3.

A.3.3 Third Experiment

Is provided on a CD along with the thesis in folder A33.

For the 1st Test Case

Is provided on a CD along with the thesis in folder A33 under Test Case 1.

For the 2nd Test Case

Is provided on a CD along with the thesis in folder A33 under Test Case 2.

For the 3rd Test Case

Is provided on a CD along with the thesis in folder A33 under Test Case 3.

A.3.4 Fourth Experiment

Is provided on a CD along with the thesis in folder A34.

For the 1st Test Case

Is provided on a CD along with the thesis in folder A34 under Test Case 1.

For the 2nd Test Case

Is provided on a CD along with the thesis in folder A34 under Test Case 2.

For the 3rd Test Case

Is provided on a CD along with the thesis in folder A34 under Test Case 3.

A.3.5 Fifth Experiment

Is provided on a CD along with the thesis in folder A35.

For the 1st Test Case

Is provided on a CD along with the thesis in folder A35 under Test Case 1.

For the 2nd Test Case

Is provided on a CD along with the thesis in folder A35 under Test Case 2.

For the 3rd Test Case

Is provided on a CD along with the thesis in folder A35 under Test Case 3.

A.3.6 Sixth Experiment

Is provided on a CD along with the thesis in folder A36.

For the 1st Test Case

Is provided on a CD along with the thesis in folder A36 under Test Case 1.

For the 2nd Test Case

Is provided on a CD along with the thesis in folder A36 under Test Case 2.

For the 3rd Test Case

Is provided on a CD along with the thesis in folder A36 under Test Case 3.

List of distribution

Report No. A 02/20

Author: E. Nasr
Title: Evaluation of Automatic Ontology Matching for Materials Sciences and Engineering – Master’s Thesis in Computer Science at the Albert-Ludwigs-Universität Freiburg, Faculty of Engineering

Internal Distribution: M. Huschka
M. Dlugosch
Author: E. Nasr (+ 1 CD)

University:
University of Freiburg

4 copies + 4 CDs (PDF and additional data)

Berichts-/Erschließungsblatt

Bitte Ausfüllanweisung beachten

Nur vom DOKFIZBw auszufüllen

Dieses Berichts-/Erschließungsblatt wird unabhängig vom Geheimhaltungsgrad des Dokuments immer als "offen" behandelt, deshalb sind keine geheimhaltungswürdigen Angaben zu übernehmen.		1 LfdNr	
2 BerichtsNr des Herausgebers/Auftragnehmers (AN) (vollständige Buchstaben-/Ziffernfolge)	3 BerichtsNr des Auftraggebers (AG)	10 DSt und StO (Dokument langfristig verfügbar, Ausleihe)	
4 Titel/Untertitel (VS-eingestuft Titel fingieren "....."; bei mehrbändigen Dokumenten BdNr und zutreffenden Einzeltitel angeben)		11 Geheimhaltungsgrad	
Kurtztitel: (max. 30 Stellen)		<div>Offen</div> <div>VS-NfD</div> <div>VS-Vertr.</div> <div>GEHEIM</div>	
		12 Gesamtseiten-/blattzahl	13 Quellen
		14 Tabellen	15 Statistiken
		16 Techn. Zeichnungen	17 Abbildungen
4a Ins Englische übersetzter Titel/Untertitel		18 Berichtsdatum	
5 Autor(en) (Name, Vorname(n) oder Institution als körperschaftlicher Urheber)		<div>Jahr</div> <div>Monat</div> <div>Tag</div>	
		19 Berichtsart (z.B. Zwischen-/Abschlußbericht, vgl. Feld 27)	
6 Auftragnehmer (AN) (Institution(en), Abteilung, Ort/Sitz der beteiligten AN, SubAN, MitAN)		20 Auftrags-/VertragsNr des AG (vollständige Buchstaben-/Ziffernfolge)	
Fraunhofer-Institut für Kurzzeitdynamik Ernst-Mach-Institut, EMI Freiburg i. Br./Efringen-Kirchen		21 Auftragserteilung/Vertragsabschluss	
		22 Abschlussdatum/Vertragsende	
7 Auftraggeber (AG) / Aufgabesteller (AST) / Fachlich zuständige Stelle		23 Projekt-/Programm-/Konzeptbezeichnung (z.B. ZTL 1979, FAG 1, MBB 1-85-1)	
Ast:		24 Studien-/Aufgabenkennziffer, DateiblattNr (SKZ-, AKZ-, DateiblattNr)	
8 Kurzreferat (Inhaltsbeschreibung des Dokuments)			
9 Schlagwörter (Schwerpunktartige Inhaltskennzeichnung mittels Fachbegriffen, maximal 10 Stellen)		25 Aktenzeichen des AG/Herausgebers oder der fachlich zuständigen Stelle	
		26 Sperrvermerke	
		27 Zusätzliche Angaben/Hinweise	