Automated Compliance Verification in ATM using Principles from Ontology Matching

Audun Vennesland[1,2], Joe Gorman[1], Scott Wilson[3], Bernd Neumayr[4] and Christoph G. Schuetz[4]

[1] *Department of Software Engineering, Safety and Security, SINTEF, Trondheim, Norway*

[2] *Norwegian University of Science and Technology, Trondheim, Norway*

[3] *EUROCONTROL, Brussels, Belgium*

[4] *Johannes Kepler University Linz, Linz, Austria*

Keywords: Compliance Verification, Air Traffic Management, Ontology Matching, Ontology Engineering.

Abstract: Compliance with standard information models in diverse and complex domains such as Air Traffic Management is an important but highly challenging task. The challenges stem from the fact that the information models are often extensive, the diversity of the domain leads to diverging terminology, and the manual mapping of information elements necessary to assess compliance is very labor-intensive. This work proposes ways in which compliance verification techniques, currently based on manual techniques, can be supported and partly automated by means of a set of basic ontology matching techniques. We have evaluated these techniques in an experiment involving seven datasets consisting of various ATM ontologies that have been transformed to OWL from their original UML representations. A comparative analysis with two other state-of-the-art matching systems shows that some of our proposed matching techniques obtain good quality alignments, especially when they are combined using simple strategies. The evaluation also reveals that identifying equivalence relations is a far easier task than identifying other types of semantic relations.

1 INTRODUCTION

Within Air Traffic Management (ATM), the ATM Information Reference Model (AIRM) is the standard information reference model. Information models targeting information exchange in the ATM network should be compliant with the AIRM in order to foster interoperability. However, assuring such compliance currently requires a significant amount of manual effort both during model development, when modellers investigate potentially re-usable elements in the reference model, and after completion of the model when its compliance with the AIRM has to be verified and maintained for governance purposes.

Several initiatives in the realm of aviation have investigated the feasibility of introducing semantic technologies as a means for improving information management. One of the latest contributions to semantic developments in ATM is the NASA ATM Ontology[1]. Furthermore, the Horizon 2020[2] BEST project[3] looks at ATM information management and represents AIRM as well as other information models (for expressing aeronautical information and weather information) as OWL (Web Ontology Language) ontologies.

Ontology matching, a sub-discipline of ontology engineering, investigates techniques for (semi) automatic identification of semantic correspondences between ontologies. Our hypothesis is that ontology matching techniques lend themselves well to provide automated support for compliance verification, and can reduce much of the human effort that is currently required for compliance verification in ATM. Furthermore, such automated support can motivate re-use of standardised information elements in ATM, preventing interoperability threats and unnecessary use of development resources.

Based on this, we investigate to what extent ontology matching principles can offer automated support for identifying semantic correspondence between different ATM-related models and the AIRM. While the quality of ontology matching systems has improved in recent years, there is no superior system or technique that performs best in all contexts and settings. We therefore suggest an approach where the input ATM

[1] https://data.nasa.gov/ontologies/atmonto/index.html

[2] https://ec.europa.eu/programmes/horizon2020/

[3] http://www.project-best.eu/

Vennesland, A., Gorman, J., Wilson, S., Neumayr, B. and Schuetz, C.
Automated Compliance Verification in ATM using Principles from Ontology Matching.
DOI: 10.5220/0006924000410052
In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018) - Volume 2: KEOD*, pages 41-52
ISBN: 978-989-758-330-8

ontologies are profiled according to a set of ontology analysis metrics before they are matched.

The performance of ontology matching systems is evaluated in the Ontology Alignment Evaluation Initiative (OAEI), an annual evaluation campaign for ontology matching systems. Many systems obtain close to perfect alignment quality in OAEI. However, one may argue that some of the more popular tracks in OAEI are simple and limited in scope. Moreover, many of the participating ontology matching systems are very much tuned in on the ontologies represented in the OAEI tracks as there have been few changes made to the tracks over the last year (Euzenat et al., 2011). Additional benchmarks, preferably involving ontologies describing knowledge models new to the research area are sought. We have developed our own datasets for evaluating the quality of our techniques. The datasets consists of ATM ontologies and ground truth mappings verified by human experts in the ATM field.

This research addresses the following research questions:

- To what extent can ontology matching principles support information modeling and compliance verification processes within ATM?
- Which ontology matching techniques perform better this context?
- Are these techniques capable of capturing the full range of semantic correspondences defined by human experts?

The main contributions from this work are:

- We provide datasets consisting of real ontologies from the ATM domain that can be used to evaluate other ontology matching systems and techniques. The reference alignments in these datasets are verified by experts in the ATM domain.
- We describe matching techniques for identifying equivalence and other semantic relations as well as a set of strategies for combining their computed alignments. The performance of the techniques and combination strategies is compared against well-acknowledged and state of the art ontology matching systems.

2 BACKGROUND

2.1 Semantic Interoperability in SWIM

System Wide Information Management (SWIM) covers a complete change in paradigm of how information is managed along its full lifecycle and across the whole European ATM system [4]. One objective is the establishment of a network-centric information environment in Europe, in contrast to today's information management which is typically based on point-to-point message transfer, limited use of standards, and tightly coupled APIs hindering interoperability. It is further recognised that global interoperability and standardisation are essential.

2.1.1 ATM Information Reference Model

The ATM Information Reference Model (AIRM) is one of the essential elements of realising SWIM. AIRM is a reference model that addresses semantic interoperability through harmonised and agreed definitions of the information being exchanged in ATM[5]. AIRM is formalised in UML (Unified Modeling Language).

The AIRM model is organised into different subject fields, where each subject field includes elements for particular areas of ATM, such as the aircraft, the airport infrastructure, meteorological information, etc.

Semantic interoperability within ATM is accomplished by ensuring that all information being exchanged within ATM is compliant with the intended semantics as defined in AIRM.

2.2 Exchange Models

ATM Exchange models define the structure and content of digital information exchanged between ATM systems. These exchange models have to be compliant with the AIRM.

One such exchange model is the Aeronautical Information Exchange Model (AIXM). AIXM provides a UML data model and associated XML schemas for representing the format of digitally communicated aeronautical information. AIXM defines information related to, among other things, airports and heliports, airspace structures, organisations (including services they provide), geographical points and navigation aids, route information and flying restrictions.

Another exchange model is the ICAO Meteorological Information Exchange Model (IWXXM). IWXXM provides a format for exchanging messages related to actual and forecasted weather reports at aerodromes, weather conditions along the route, significant meteorological information, and advisories related to volcanic ash events and other extreme mete-

[4] http://www.eurocontrol.int/swim

[5] http://www.eurocontrol.int/articles/airm-atm-information-reference-model

orological conditions (e.g. cyclones). As AIRM and AIXM, IWXXM is originally represented in UML.

2.3 Defining Compliance in ATM

Compliance is defined as (ISO/IEC, 2005):

> *"[...] the demonstration that specified requirements relating to a product, process, system, person, or body are fulfilled ."*

Semantic interoperability in the ATM domain is supported by having a compliance framework defining a set of artefact-related and procedural requirements that need to be satisfied in order for an information or data construct to claim compliance with the AIRM (Wilson et al., 2015).

The result of a compliance verification process is a mapping describing the semantic relation between information elements of the model claiming compliance with the AIRM and the corresponding AIRM information element. The semantic relation is either *equivalence* or a *wider semantic meaning* (Wilson, 2017).

2.4 Representing ATM Information Models as OWL Ontologies

The aforementioned BEST project transformed the reference model AIRM and the exchange models AIXM and IWMXX to OWL using the mapping rules specified by the "Ontology Definition Metamodel Specification" developed by Object Management Group (OMG) (Object Management Group, 2014). These rules ensure that the semantics expressed in the UML models are maintained during the transformation to OWL. Furthermore, the BEST project decomposed the ontologies resulting from this transformation into a set of ontology modules following the guidelines of d'Aquin (D'Aquin, 2012) and principles of ontology module extraction from Grau et al. (Cuenca Grau et al., 2008).

3 AUTOMATED COMPLIANCE VERIFICATION

In this section we describe our 3-step process for automated compliance verification as outlined in Figure 1.

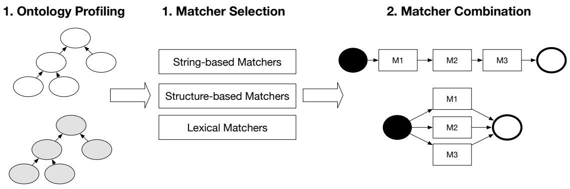


Figure 1: Overall Approach for Automated Compliance Verification.

3.1 Ontology Profiling

After the input ontologies have been pre-processed and parsed to an appropriate representation, the input ontologies are analysed according to a set of profiling metrics. These metrics evaluate the terminological, structural and lexical profile of the ontologies and are computed as an average metric for the ontologies. Such a process helps to select the most optimal matchers and reduce processing at run-time caused by excluding or giving less emphasis to matchers not capable of contributing to the task at hand. Based on these metrics, the set of optimal matchers are identified given the ontologies to be matched.

In the following we describe the profiling metrics we have used in this work.

3.1.1 Compound Ratio (CR)

Compound words are quite common in ontologies. A compound is a word consisting of one or more individual words, such as AerodromeProtectionArea. If the number of occurrences of compounds is high, this suggests that a matcher capable of exploiting such linguistic structures should be employed. This also might suggest that the terminology of the ontology is quite "uniform", where existing concept names are appended (either through prefixing or suffixing), for example when creating sub-classes, instead of using a richer and more fine-grained terminology. In such a case, a string-based matcher could perform well. The Compound Ratio is computed by dividing the number of classes having a compound name by the total number of classes in the ontology.

3.1.2 Annotation Coverage (AC)

Annotation Coverage measures how many concepts that have a natural language definition compared to all concepts in the ontology. If this fraction is high, then a matcher specialised in finding similarity among annotation properties (comments) should be applied. This metric does not indicate whether such a matcher will be successful, but rather that if the score is low such a matcher will probably not contribute much.

3.1.3 Inheritance Richness (IR)

Inheritance Richness (Tartir et al., 2005; Cruz et al., 2012) measures the structural characteristics of the input ontologies as the average number of subclasses per class. Hence, if the Inheritance Richness is high, the concepts in the ontology have many sub-classes, something which could be exploited by a structural matcher. In contrast to the other metrics the IR is an open-ended positive number rather than a fraction.

3.1.4 Relationship Richness (RR)

Relationship Richness (Tartir et al., 2005; Cruz et al., 2012) computes the fraction of relations that are different from subClassOf relations and can suggest to what extent properties can be exploited to infer concept mappings. If an ontology has a Relationship Richness close to zero, that would indicate that most of the relationships are is-a relations, and a structural matcher could be emphasized. On the other hand, if the Relationship Richness is high, this indicates that the ontology has a high fraction of object properties that could be exploited to infer either class equivalence or other semantic relations.

3.1.5 WordNet Synonym Coverage (WNSYN)

One of the strengths of using WordNet in ontology matching is to identify (synonymic) relations between two concepts that other matchers cannot identify, typically through a shared synset among these concepts. So, if the degree of synonymy among the input ontologies is high, then it is likely that a matcher utilising WordNet synonyms could contribute positively in the matcher composition. This metric measures the extent to which a concept is represented by synonyms in WordNet. It is calculated by accumulating the number of concepts for which there exists a synonym and then divide this number by the total number of classes in each ontology. Whenever the concept name is a compound word, each compound part of the word is treated separately. That means that if a compound concept name (e.g. AerodromeProtectionArea) has a compound part (e.g. Protection) for which there is no set of synonyms in WordNet, it is omitted in the accumulation, and the score is reduced.

3.1.6 Profiling Scores for the Datasets

Table 1 shows the profiling scores according to the five introduced profiling metrics (CR, AC, IR, RR, WNSYN) for the seven datasets in our experiment (D1-D7) and the average over all seven datasets (AVG).

So, what do these profiling scores tell us? Well, the *Compound Ratio (CR)* score tells us that most concept names in these ontologies are compound words (90 percent of all concept names in all ontologies involved are compounds), suggesting that there could be a hierarchical structure where a super concept (e.g. Wind) has children with concept names that append their parent (e.g. AerodromeSurfaceWind). This could be utilised by a subsumption matcher that identifies for example that AerodromeSurfaceWind is a specialisation (child concept of) of Wind. Furthermore, it suggests that it could be difficult to straightforwardly utilise lexical resources such as WordNet, since such resources often hold mostly generic, non-compounded terms.

The *Annotation Coverage (AC)* shows that almost all concepts are well defined in the sense that they have a natural language definition associated with them. This means that a matcher that analyses (similarity) between the concepts' definitions should be included in the experimentation.

The *Inheritance Richness (IR)* and the *Relationship Richness (RR)* scores in combination reveal that these ontologies have quite flat structures with few subclasses per class, but that the representation of relations (object properties) between the classes is relatively high. Based on this, we have not included structural matchers that infer equivalence relations from the graph-based representation of the ontologies. However, we have included matchers that exploit object properties as means for inferring similarity between classes should be included.

As earlier mentioned, the fact that most concept names are compounds makes the use of WordNet challenging. However, by splitting each concept names into individual compound tokens, e.g. [Aerodrome] [Surface] [Wind] as in the example used earlier, we then analyse to what extent each individual part has a representation of synonyms in WordNet, resulting in the *WordNet Synonym Coverage (WNSYN)*. This metric represents an extension of the WordNet Coverage used for example in (Tartir et al., 2005; Cruz et al., 2012).

Table 1: Profiling scores for all datasets.

Metric	D1	D2	D3	D4	D5	D6	D7	AVG
CR	.94	.91	.93	.93	.76	.94	.92	.90
AC	1.0	1.0	.99	1.0	1.0	1.0	1.0	1.0
IR	1.13	1.47	1.48	1.46	1.47	1.48	1.46	1.42
RR	.59	.56	.56	.57	.57	.57	.57	.57
WNSYN	.74	.75	.56	.76	.87	.87	.87	.77

Based on the above profiling and discussion, we implemented the matchers presented in the next section.

3.2 Matcher Selection

We have implemented the matchers using the ontology- and ontology matching infrastructures of the OWL API (Horridge and Bechhofer, 2011) and the Alignment API (David et al., 2011).

Some of the matchers identify equivalence relations, while others identify other semantic relations (typically subsumption relations). Table 2 provides a summary of all implemented matchers.

3.2.1 ISub String Matcher

The ISub String Matcher is a string matching algorithm developed by Stoilos et al. (Stoilos, Giorgos and Stamou, Giorgos and Kollias, 2005). The ISub algorithm applies three functions in order to find the similarity between two concept names c_1 and c_2. The first function computes the similarity between the two strings by iteratively finding the common substrings. In the second function it considers the difference as the length of the remaining characters in the two strings. In the third and final function, the Winkler algorithm (Winkler, 1999) is used for improving the results.

3.2.2 Definitions Matcher

The Definitions Equivalence Matcher treats definitions associated with two entities as sets of individual words. Stopwords that carry little meaning, such as 'the', 'a', 'is', etc., are removed before the definitions are processed further. As with the other algorithms relying on set-theoretic similarity scores, this algorithm employs the Jaccard (Jaccard, 1901) set-theoretic similarity measure to compute a similarity score between the two definitions. The Jaccard similarity measure is computed by dividing the intersection of sets with the union of the sets.

3.2.3 Property Matcher

The Property Matcher bases the equivalence identification on similarity of the properties associated with the concepts to be matched. Both object properties and data properties where the concepts to be matched are domains are collected into a single set for each concept and compared with Jaccard.

3.2.4 Range Matcher

The Range Matcher measures the similarity of the sets of range classes of object properties where the concepts c_1 and c_2 being matched represent the domain. If the Jaccard set similarity of the object properties' range classes is above a certain threshold, this matcher considers that the two concepts are equivalent.

3.2.5 WordNet Synonym Matcher

The WordNet Synonym Matcher (WNSyn) computes a similarity score based on how many common WordNet synonyms the two concepts to be matched are associated with. Since the ontology profiling revealed that most concept names are compound words, we split all compounds into a set of compound parts, and synonyms associated with each part represent individual sets of words taking part in the similarity calculation. The synonyms associated with the respective concepts are represented as sets and a similarity score is computed using Jaccard.

While the previously presented algorithms seek to identify equivalence relations, the following algorithms aim to identify other semantic relations.

3.2.6 Closest Parent Matcher

The Closest Parent Matcher determines that one concept c_1 is a subclass of concept c_2 if the superclass of c_1 has a high similarity with c_2, as illustrated in Figure 2. This matcher relies on having a graph representation of the ontologies. We implement such a graph representation using the Neo4J[6] graph database.

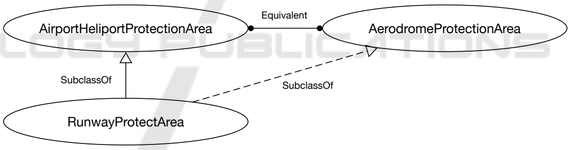


Figure 2: The Closest Parent Matcher.

3.2.7 Compound Matcher

The Compound Matcher identifies subsumption relations between entities reusing principles from the compound strategy from Arnold and Rahm (Arnold and Rahm, 2014). Compound means that several individual words are put together to form another word. Here, parts of compounds in entity names are identified and employed as an indicator of a subsumption relation. So, if one or more compound parts in one concept name c_1 are represented as a subset of compounds in another concept name c_2, the Compound Matcher defines that c_1 subsumes c_2.

3.2.8 Definitions Subsumption Matcher

The Definitions Subsumption Matcher considers both commonality and number of words in the definitions

[6] https://neo4j.com/

Table 2: Overview of Implemented Matchers.

Matcher	Target	Relation Type
ISub	Entity - Name	Equivalence
Definitions Matcher	Entity - Definition	Equivalence
Range Matcher	Entity - Structure	Equivalence
Property Matcher	Entity - Structure	Equivalence
WordNet Synonym Matcher	Entity - Lexical Properties	Equivalence
Closest Parent Matcher	Entity - Structure	Other
Compound Matcher	Entity - Name	Other
Definitions Subsumption Matcher	Entity - Definition	Other

in order to compute if two entities are in a subsumption relation. If the commonality of the definitions is above a certain threshold, we consider the size (number of words) of the definitions as a qualifier for subsumption, where an entity with a smaller definition subsumes an entity with a larger definition. The rationale behind this is that the more specific and detailed the entity is, the more text is required to sufficiently describe it.

3.3 Matcher Combination

The next step of the overall process is to combine the alignments produced by the matchers in the previous step.

3.3.1 Weighted Sequential Combination (WSC)

In the *Weighted Sequential Combination* the initial alignment produced by the first matcher is refined by each following matcher in the sequence. Weight is added to correspondences that are identified by two consecutive matchers. If the correspondence is new, that is, identified only by the current matcher, or if the correspondence is only identified by the previous matcher(s) and not the current one, the correspondence is added to the refined alignment with equally reduced weight. As an example, consider that a matcher m_1 has produced an alignment that is transferred to m_2, the next matcher in the sequence. If the same correspondence (the same two entities and the same relation type) is identified as correct by both m_1 and m_2, the confidence value associated with this correspondence is increased. On the other hand, if a correspondence received by m_2 from m_1 is only identified as a correct correspondence by m_1 and not by m_2, this correspondence is reduced before the alignment is sent further to m_3. The weighting scheme applied in this study is to add (or reduce) 12 percent to the confidence of the correspondence. Maximum confidence is 100 percent (1.0).

3.3.2 Simple Vote

Simple Vote is a parallel combination strategy. Here, all matchers are run in parallel. The alignments they produce are initially treated as equally important, but only those correspondences identified by a predefined ratio of matchers (for example using the majority vote, such as three out of five matchers) are eligible for the final alignment.

3.3.3 Autoweight++

The third combination strategy is an implementation of the Autoweight++ algorithm described in (Gulić et al., 2016). As with Simple Vote, this is also a parallel combination strategy, but a more sophisticated one, since it includes both matcher configuration and combination. The concept of highest correspondences is central in this approach. A correspondence between two concepts c_1 and c_2 is considered a highest correspondence if it has a higher confidence value than any other correspondence that includes either c_1 or c_2. The highest correspondences are used both for automatically configuring the matching algorithms' weight and for combining the individual alignments into an optimal final alignment. When producing the final alignment (from the aggregated set of correspondences in the intermediate common alignment), Autoweight++ takes an iterative approach. It starts by taking the highest correspondences from an intermediate common alignment. Then in the following iterations, the correspondences that do not include concepts taking part in the already established highest correspondences are processed. The algorithm stops when there are no more correspondences above a given confidence threshold.

4 EXPERIMENTAL EVALUATION

In this section we start by describing the datasets included in the experimental evaluation. Then we proceed to explain how the evaluation was performed, before we present our findings from the evaluation.

4.1 Datasets

Table 3 shows some statistics associated with the datasets used in the experiments. Each dataset consists of two ontologies that are matched pairwise. The ontologies used in the datasets include both monolithic ontologies and ontology modules derived from the AIRM, AIXM and IWXXM ontologies (see section 2.4).

A reference alignment represent the correct set of relations between entities in the datasets described in the previous chapter, and act as our comparison base for evaluating the quality of our techniques. The source material for the reference alignments are mapping files in Excel from real compliance verification processes of the exchange models AIXM and IWXXM. These mapping files contain manually developed relations based on human expert verification between the exchange models AIXM/IWXXM and AIRM and have been transformed to reference alignments in the RDF Alignment Format using the Java library Apache POI[7]. The two rightmost columns describes the number of relations in each reference alignment. If a dataset has both equivalence relations (EQ) and other semantic relations (OTH) there are two reference alignments for the dataset, otherwise there is only one.

The datasets, that is, the ontology pairs along with the reference alignments, are available from: https://github.com/sju-best-project/KEOD18

4.2 Evaluation

Typically, evaluation of ontology matching techniques is performed using precision and recall against reference alignments holding the true positive relations based on expert judgment (Euzenat and Shvaiko, 2013). Precision is computed as the ratio of correctly found correspondences (according to the reference alignment) over the total number of identified correspondences. Recall is computed as the ratio of correctly found correspondences over the total number of expected correspondences (as expressed in the reference alignment). An evaluation measure that balances precision and recall is the F-measure. This is often used as an overall measure for representing the quality of an ontology matching technique or complete system.

4.2.1 Baseline Matchers

In order to perform a comparative analysis we have used two ontology matching systems that often rank as top contenders of the OAEI campaign as baseline matchers. LogMap (Ruiz-Jimenez and Grau, 2011) is a system that often competes in several of the OAEI tracks. In our experiments we ran the standalone version (version 2.4) of LogMap. AgreementMakerLight (AML) (Faria et al., 2013) is, like LogMap, a recurring participant at the OAEI, with several top positions in the tracks it competes in. We used the graphical user interface version from April 2016[8]. As with our own matcher implementations we let LogMap and AML compute alignments with confidence thresholds 0.5, 0.7 and 0.9, and only class relations.

4.2.2 Combination Strategies

When using the combination strategies described in section 3.3, we combine the alignments produced by the three best-performing individual matchers across all datasets. For the equivalence relations these are the Definitions Matcher (90 percent confidence), the ISub Matcher (90 percent confidence) and the WordNet Synonym Matcher at 90 percent confidence.

4.3 Results and Findings

In this section we summarise the results and observations from the experiments for each dataset.

4.3.1 Equivalence Relations

Table 4 shows the F-measure scores for all individual matchers, the combination strategies and the baseline matchers for each dataset isolated on equivalence relations.

Dataset 1 is the largest dataset in terms of relations in the reference alignment, with 126 equivalence relations. Many of the relations consists of concepts where the names are identical, but there are also some “traps”, where the semantic meaning deviates despite of name equality. Conversely, there are also relations where the semantic meaning of the two concepts is equal, while their names are different.

The best performing individual matcher is the WordNet Synonym Matcher which obtains an F-measure of 88 percent. The best combination strategy is SimpleVote. Here, the true positive relations identified by the WordNet Synonym Matcher are supplemented with additional true positive relations from the other matchers, resulting in an F-measure of close

[7] https://poi.apache.org/

[8] This was the latest release on github as of April 2018

Table 3: Dataset Statistics.

Dataset	Ontologies	Classes	Object Prop.	Data Prop.	EQ	OTH
D1	AIXM-AirportHeliport AIRM-AerodromeInfrastructure	347	571	162	126	1
D2	IWXXM-Common AIRM-Monolithic	923	1762	494	0	9
D3	IWXXM-Metar AIRM-Monolithic	961	1807	530	11	7
D4	AIXM-Shared AIRM-Monolithic	938	1785	518	21	0
D5	AIXM-Geometry AIRM-Monolithic	922	1764	506	5	2
D6	AIXM-Obstacle AIRM-Monolithic	930	1788	501	11	2
D7	AIXM-Organisation AIRM-Monolithic	925	1776	499	10	0

to 92 percent. In comparison, the best baseline system, AML at confidence 0.9, obtains an F-measure of 94.6 percent.

Dataset 3 is quite challenging as it includes several instances of complex (1..n) mappings among the 11 relations in the reference alignment. The best score is achieved by the ISub string matcher which obtains an F-measure of around 19 percent. ISub identifies two true positive correspondences. None of the combination strategies are able to improve the score, and of the baseline systems only AML is able to identify any true positives (one).

Dataset 4 contains 21 relations in the reference alignment and they include mostly generic, domain-independent concept names. The best performing individual matcher is the Definitions Matcher (confidence 0.9) obtaining an F-measure of around 83 percent. The ISub string matcher manages to identify most relations in the reference alignment at low confidence threshold (hence a high recall), but includes too many false positives, so the resulting F-measure becomes quite low at the lower confidence levels. The overall best alignment quality is achieved by the combination strategy SimpleVote, with an F-measure of 90 percent. The baseline systems AML and LogMap obtains a maximum F-measure of 95 and 92 percent respectively.

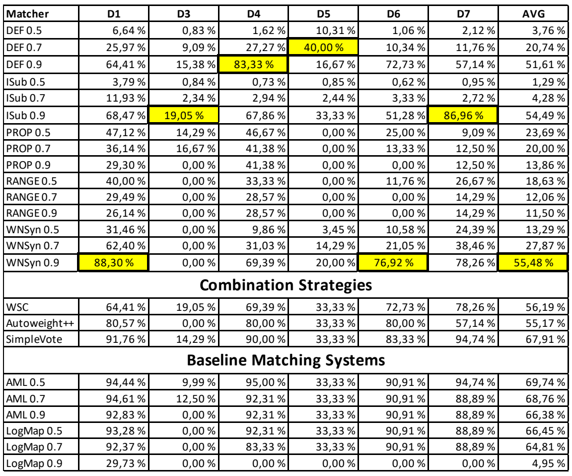
In *Dataset 5* there are 5 relations in the reference alignment. This is a challenging dataset for matchers basing their equivalence identification on string similarity of concept names as in only one of the reference alignment relations such similarity is noticeable. Here, the Definitions Matcher at confidence 0.7 obtains the highest F-measure score of 40 percent, while most of the other matchers are only capable of identifying the aforementioned concept name equality. The Definitions Matcher identifies 3 of the correct relations from its similarity computation of natural language definitions associated with the concepts.

Dataset 6 contains 11 relations. The best F-measure score is obtained by the WordNet Synonym Matcher at confidence level 0.9, which identifies all true positive correspondences, and manages to disregard more false positives than the ISub Matcher that also identifies all correct relations. When combining the alignments with the SimpleVote strategy, we obtain an F-measure of 83 percent.

The reference alignment for *Dataset 7* contains 10 relations. Most of the relations in the reference alignment consists of identical concept names (8 out of 10). Here, the best performance of the individual matchers is achieved by the ISub matcher at confidence 0.9, which identifies all true positive relations but one, resulting in an F-measure of 87 percent. The baseline systems have better precision (less false positives), and AML at confidence 0.5 obtains an F-measure of close to 95 percent.

When we average the F-measure scores across all datasets, the best individual matcher is the WordNet Synonym matcher with an F-measure of 55.48 percent. The best combination strategy is Simple Vote with an average F-measure of 67.9 percent. Not unexpectedly, the baseline matchers obtain overall higher F-measure scores. The best F-measure is achieved by AML at confidence 0.5 (69.74 percent), followed by AML at confidence 0.7. Both these configurations obtain higher F-measure scores than the best combination strategy SimpleVote on third.

Table 4: F-measure scores for all datasets - Equivalence relations. The best performing individual matcher is highlighted in yellow.



Matcher	D1	D3	D4	D5	D6	D7	AVG
DEF 0.5	6,64 %	0,83 %	1,62 %	10,31 %	1,06 %	2,12 %	3,76 %
DEF 0.7	25,97 %	9,09 %	27,27 %	40,00 %	10,34 %	11,76 %	20,74 %
DEF 0.9	64,41 %	15,38 %	83,33 %	16,67 %	72,73 %	57,14 %	51,61 %
ISub 0.5	3,79 %	0,84 %	0,73 %	0,85 %	0,62 %	0,95 %	1,29 %
ISub 0.7	11,93 %	2,34 %	2,94 %	2,44 %	3,33 %	2,72 %	4,28 %
ISub 0.9	68,47 %	19,05 %	67,86 %	33,33 %	51,28 %	86,96 %	54,49 %
PROP 0.5	47,12 %	14,29 %	46,67 %	0,00 %	25,00 %	9,09 %	23,69 %
PROP 0.7	36,14 %	16,67 %	41,38 %	0,00 %	13,33 %	12,50 %	20,00 %
PROP 0.9	29,30 %	0,00 %	41,38 %	0,00 %	0,00 %	12,50 %	13,86 %
RANGE 0.5	40,00 %	0,00 %	33,33 %	0,00 %	11,76 %	26,67 %	18,63 %
RANGE 0.7	29,49 %	0,00 %	28,57 %	0,00 %	0,00 %	14,29 %	12,06 %
RANGE 0.9	26,14 %	0,00 %	28,57 %	0,00 %	0,00 %	14,29 %	11,50 %
WNSyn 0.5	31,46 %	0,00 %	9,86 %	3,45 %	10,58 %	24,39 %	13,29 %
WNSyn 0.7	62,40 %	0,00 %	31,03 %	14,29 %	21,05 %	38,46 %	27,87 %
WNSyn 0.9	88,30 %	0,00 %	69,39 %	20,00 %	76,92 %	78,26 %	55,48 %
Combination Strategies							
WSC	64,41 %	19,05 %	69,39 %	33,33 %	72,73 %	78,26 %	56,19 %
Autoweight++	80,57 %	0,00 %	80,00 %	33,33 %	80,00 %	57,14 %	55,17 %
SimpleVote	91,76 %	14,29 %	90,00 %	33,33 %	83,33 %	94,74 %	67,91 %
Baseline Matching Systems							
AML 0.5	94,44 %	9,99 %	95,00 %	33,33 %	90,91 %	94,74 %	69,74 %
AML 0.7	94,61 %	12,50 %	92,31 %	33,33 %	90,91 %	88,89 %	68,76 %
AML 0.9	92,83 %	0,00 %	92,31 %	33,33 %	90,91 %	88,89 %	66,38 %
LogMap 0.5	93,28 %	0,00 %	92,31 %	33,33 %	90,91 %	88,89 %	66,45 %
LogMap 0.7	92,37 %	0,00 %	83,33 %	33,33 %	90,91 %	88,89 %	64,81 %
LogMap 0.9	29,73 %	0,00 %	0,00 %	0,00 %	0,00 %	0,00 %	4,95 %

4.3.2 Other Semantic Relations

Five of the datasets include other semantic relations than equivalence. Table 5 shows the F-measure scores for all individual matchers isolated on other relations than equivalence. We also experimented with the same combination strategies for the other correspondences experiments also, using the overall 3 best individual matchers and their produced alignments. However, since none of the combination strategies were able to produce better quality alignments than the best individual matcher, we exclude the scores from the combination strategies for these alignments.

In *Dataset 1* there is 1 relation in the reference alignment holding “other semantic relations”. The only relation in the reference alignment is RunwayElement-RunwayElement, which intuitively suggests an equivalence relation. The reason why this seemingly equivalent relation is considered a different semantic relation, is that the natural language definition in ontology 1 is more specific than the natural language definition in ontology 2.

The Closest Parent Matcher with confidence thresholds 0.5 and 0.7 identified the true positive relation, but since these alignments also included a very large number of false positive relations, the precision becomes very poor (6 percent and 9 percent respectively), resulting in very low F-measure scores (0.19 percent).

The Definitions Subsumption matcher at confidence threshold 0.5 identified the true positive relation, but as with the Closest Parent Matcher alignments, the precision and consequently the F-measure were very low due to a very large number of false positives.

The reference alignment in *Dataset 2* contains 9 relations. The best performing matcher is the Definitions Subsumption Matcher at confidence threshold 0.9 which identified 1 true positive correspondence and no false positive ones, resulting in a 100 percent precision. However, since it missed the other 8 correspondences in the reference alignment, the recall was quite low, resulting in an F-measure of 20 percent. The Compound Matcher computed also 1 true positive relation, but 1 false positive one. The Closest Parent Matcher computed 1 true positive relation, and 27 false positives. All the true positive relations identified by these 3 matchers are different relations, so in total 3 out of 9 relations in the reference alignment were identified.

The relations from the reference alignment not identified by any matcher were:

```
AerodromeForecastWeather < CodeSignificantWeatherQualifierType
AerodromeForecastWeather < CodePrecipitationType
AerodromeForecastWeather < CodeWeatherPhenomenonType
AerodromeForecastWeather < CodeObscurationType
AerodromeSurfaceWindTrendForecast < Wind
AerodromeSurfaceWindTrendForecast < TREND
```

where X < Y means that X is a specialisation of Y.

All these correspondences represent complex mappings, and the manual inspection of the ontologies suggests that neither structure nor natural language definitions can help infer any type of semantic relations between these concepts.

Dataset 3 includes 7 relations and the only matcher able to identify any true positives is the Compound Matcher (at all confidence thresholds), which identifies the following two:

```
AerodromeSurfaceWind < Wind
AerodromeRunwayVisualRange < RunwayVisualRange
```

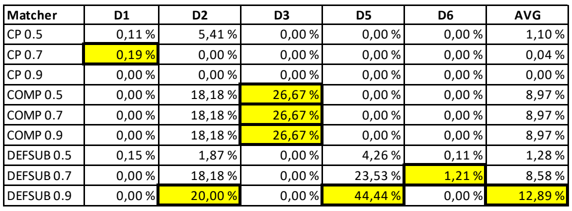
Here, the use of endocentric compounds contributes to the identification of a subsumption relationship. Endocentric compounds consist of a compound head, which represent the base meaning of the compound, and one or more modifiers that serves to narrow the meaning of the compound as a whole (Arnold and Rahm, 2014).

The reference alignment in Dataset 5 contains two relations. Both relations were identified by the Definition Subsumption Matcher at all confidence thresholds, hence a perfect recall. However, the number of false positives increased with lowering the thresholds. The best alignment was thus obtained at 90 percent confidence, with an F-measure of 44.4 percent.

Dataset 6 included also two relations, and the only matcher able to identify a true positive correspondence in this dataset is the Definitions Subsumption matcher at confidence thresholds 0.5 and 0.7. Both alignments identify one true positive correspondence, but contain a large number of false positives, resulting in a very low F-measure score of just over 1 percent for the best performing matcher (at confidence level 0.7).

Looking at the average F-measure across all datasets, the Definitions Subsumption Matcher performs best, with an F-measure of 12.9 percent.

Table 5: F-measure scores for all datasets - Other semantic relations. The best performing individual matcher is highlighted in yellow.



Matcher	D1	D2	D3	D5	D6	AVG
CP 0.5	0,11 %	5,41 %	0,00 %	0,00 %	0,00 %	1,10 %
CP 0.7	0,19 %	0,00 %	0,00 %	0,00 %	0,00 %	0,04 %
CP 0.9	0,00 %	0,00 %	0,00 %	0,00 %	0,00 %	0,00 %
COMP 0.5	0,00 %	18,18 %	26,67 %	0,00 %	0,00 %	8,97 %
COMP 0.7	0,00 %	18,18 %	26,67 %	0,00 %	0,00 %	8,97 %
COMP 0.9	0,00 %	18,18 %	26,67 %	0,00 %	0,00 %	8,97 %
DEFSUB 0.5	0,15 %	1,87 %	0,00 %	4,26 %	0,11 %	1,28 %
DEFSUB 0.7	0,00 %	18,18 %	0,00 %	23,53 %	1,21 %	8,58 %
DEFSUB 0.9	0,00 %	20,00 %	0,00 %	44,44 %	0,00 %	12,89 %

4.4 Conclusions from Experimental Evaluation

The general conclusions are that the identification of equivalence relations is far easier than identification of other semantic relations, such as subsumption. There are several contributing factors to this. First, the reference alignments contain a variety of different semantic relation types (subsumption, part-whole relations, less/more general based on narrower/wider natural language definitions). One such example is the already mentioned relation between AIXM RunwayElement and AIRM RunwayElement which in the reference alignment is of type “less general”. The reason why the first is more restrictive than the latter, is that RunwayElement is described in a generic way in AIRM, where it is defined as “A portion of a runway”, and there was thus a need for making the definition more accurate in AIXM. In AIXM RunwayElement is defined as follows: ’Runway element may consist of one more polygons not defined as other portions of the runway class’.

Secondly, and especially the case when trying to identify such relations between IWXXM and AIRM, it is very difficult to find usable patterns of specialisation in most of the mappings. Most specialisation relations in the mapping files are not identified even if we have implemented matchers that utilise terminological, structural and lexical patterns.

The quality of the equivalence matching is far better. When comparing against two of the top performing ontology matching systems, AgreementMakerLight and LogMap, the performance of our some of our basic matchers is fairly good. In two of the datasets, our matchers obtain higher F-measure than the baseline systems, and when combining the alignments using simple aggregation strategies, we obtain an F-measure on par with the baseline systems.

Finding a complementary set of matchers is essential. The three best performing individual matchers are the WordNet Synonym Matcher, the ISub Matcher and the Definitions Matcher. These matchers infer equivalence relations based on the concept’s name and natural language definition, suggesting that the terminology expressed by these ontologies is quite standardised. However, while these three matchers often identify many of the same relations, the Property Matcher and the Range Matcher supplement with relations where such terminological similarity does not exist through the use of property similarity.

In general, we see that combining the alignments often improves the alignment quality involving equivalence correspondences. Here, the combination strategies extract complementary true positive correspondences from each individual alignment, and also helps reduce the number of false positive correspondences. For other types of correspondences, combining the alignments hurts the quality, resulting in lower F-measure scores than for the best performing individual matchers.

Another observation is that the matchers utilising properties as means for inferring class similarity (Property Matcher and Range Matcher) perform better when using low confidence thresholds, while the other matchers perform better at higher confidence thresholds. The precision is fairly stable, but the recall drops when confidence is increased. The explanation is that at lower confidence, the other matchers produce too many false positives, reducing the precision. Finding a good compromise between precision (i.e. reducing false positives) and recall (i.e. retrieving as many true positives as possible) is the key to good performance of a matching system.

5 RELATED WORK

To the best of our knowledge, automating compliance verification with standard information models using ontology matching is not investigated elsewhere. However, several studies have investigated the use of ontology matching techniques to automate compliance assessment between *business processes* in actual use and those prescribed by standards (Ternai et al., 2013; Ternai, 2015; Gábor et al., 2013; Szabó and Varga, 2014), regulations (Sapkota et al., 2013) and structured end-user requirements (Bakhshandeh et al., 2015). Wong et al. (Wong et al., 2008) investigated how ontology matching could reconcile semantic models extracted from different IT governance standards, with the aim to make it easier for companies to comply with their rules. In their approach they first translated the governance standards into an ontological representation using natural language processing (NLP) techniques and then identified common semantics using ontology matching techniques.

6 CONCLUSIONS AND FURTHER WORK

6.1 Conclusions

Compliance verification in ATM is currently performed by manually inspecting large-sized information models and identifying the semantic relations that exist between them. This is a very laborious process that could be supported by matching techniques. This paper has presented an approach for automating compliance verification which will reduce the manual effort related to the standards compliance process in ATM and motivate reuse of standardised information elements. The approach is based on applying quite basic ontology matching techniques for automatically identifying different types of semantic relations among ontologies describing concepts from the ATM domain. The matching techniques are selected based on the results from an ontology profiling step that reveal terminological, structural and lexical characteristics of the ontologies.

From an experimental evaluation involving seven different datasets we have learned that the proposed matchers identify equivalence relations quite well. A comparative evaluation with two state-of-the-art matching systems show that our individual matchers perform on par with these more sophisticated systems in some of the datasets and even perform better in datasets where concept name similarity is not the deciding factor for equivalence. When the alignments produced by the individual matchers are combined, the alignment quality normally becomes even better. The overall best combination strategy is SimpleVote, a strategy based on aggregating alignment relations based on majority vote.

The evaluation also shows that the identification of other semantic relations than equivalence is more challenging. Due to diversity in semantic relation type, relation cardinality (e.g. 1..n relations), large variations in terminology, structure and lexical characteristics, the implemented matchers struggled to identify such relations.

6.2 Further Work

The “other relations” category includes a variety of semantic relation types. Analysing the other types of semantic relations involved and finding techniques for their identification could lead to more precise matching results for this category. One example is part-whole (meronymy) relations. Investigating patterns in names, definitions and structure that could help reveal part-whole relations (and other possible relations) and distinguish them from equivalence and specialisation relations could lead to better and more accurate alignments.

Ontology matching systems often use external resources to facilitate identification of semantic relations. In this work we have employed WordNet as an external resource, but other more domain-specific sources could possibly enhance the matching results. One such resource for the aviation domain is Skybrary[9], a wiki that contains loads of domain knowledge related to aviation and ATM. Investigating methods on how a resource such as Skybrary could be utilised to support identification of semantic relations is an interesting further work item.

Scalability is not considered in this work, but is an important quality to look at, especially when ontologies are as large as the AIRM ontology (counting over 3000 entities altogether). Some of the matchers required significant run-time, which probably could be substantially reduced by performing a thorough scalability analysis.

ACKNOWLEDGEMENTS

This research has received funding from the SESAR Joint Undertaking under grant agreement No 699298 (the BEST project) under the European Union’s Horizon 2020 research and innovation program. The views expressed in this paper are those of the authors.

REFERENCES

Arnold, P. and Rahm, E. (2014). Enriching ontology mappings with semantic relations. *Data and Knowledge Engineering*, 93:1–18.

Bakhshandeh, M., Pesquita, C., and Borbinha, J. L. (2015). Ontology matching techniques for enterprise architecture models. In *OM*, pages 236–237.

[9] https://www.skybrary.aero/index.php/Main_Page

Cruz, I. F., Fabiani, A., Caimi, F., Stroe, C., and Palmonari, M. (2012). Automatic configuration selection using ontology matching task profiling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7295 LNCS:179–194.

Cuenca Grau, B., Horrocks, I., Kazakov, Y., and Sattler, U. (2008). Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31:273–318.

D'Aquin, M. (2012). Modularizing Ontologies. In *Ontology Engineering in a Networked World*, pages 213–233. Springer Berlin Heidelberg, Berlin, Heidelberg.

David, J., Euzenat, J., Scharffe, F., and Dos Santos, C. T. (2011). The alignment API 4.0. *Semantic Web*, 2(1):3–10.

Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., and Trojahn, C. (2011). Ontology Alignment Evaluation Initiative: six years of experience. *Journal on Data Semantics*, pages 158–192.

Euzenat, J. and Shvaiko, P. (2013). *Ontology Matching*. Springer Science & Business Media.

Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., and Couto, F. M. (2013). The agreementmakerlight ontology matching system. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 527–541. Springer.

Gábor, A., Kő, A., Szabó, I., Ternai, K., and Varga, K. (2013). Compliance check in semantic business process management. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 353–362. Springer.

Gulić, M., Vrdoljak, B., and Banek, M. (2016). CroMatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment. *Web Semantics: Science, Services and Agents on the World Wide Web*, (41).

Horridge, M. and Bechhofer, S. (2011). The OWL API: A Java API for OWL Ontologies. *Semantic Web Journal*, 2(1):11–21.

ISO/IEC (2005). Iso/iec 17000 conformity assessment–vocabulary and general principles.

Jaccard, P. (1901). Distribution de la flore alpine dans la Bassin de Dranses et dans quelques regions voisines. *Bulletine del la Société Vaudoisedes Sciences Naturelles*.

Object Management Group (2014). Ontology Definition Metamodel (ODM) v1.1. Technical report, Object Management Group Inc., Needham, OSA.

Ruiz-Jimenez, E. and Grau, B. C. (2011). LogMap : Logic-Based and Scalable Ontology Matching. In *ISWC 2011*, pages 273–288.

Sapkota, K., Aldea, A., Younas, M., Duce, D. A., and Bañares-Alcántara, R. (2013). Rp-match: a framework for automatic mapping of regulations with organizational processes. In *e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on*, pages 257–264. IEEE.

Stoilos, Giorgos and Stamou, Giorgos and Kollias, S. (2005). A string metric for ontology alignment. In *Proceeding of the International Semantic Web Conference 2005*, pages 624–637. Springer.

Szabó, I. and Varga, K. (2014). Knowledge-based compliance checking of business processes. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 597–611. Springer.

Tartir, S., Arpinar, I., Moore, M., Sheth, a., and Aleman-Meza, B. (2005). OntoQA: Metric-Based Ontology Quality Analysis. *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, pages 45–53.

Ternai, K. (2015). Semi-automatic methodology for compliance checking on business processes. In *International Conference on Electronic Government and the Information Systems Perspective*, pages 243–256. Springer.

Ternai, K., Szabó, I., and Varga, K. (2013). Ontology-based compliance checking on higher education processes. In *International Conference on Electronic Government and the Information Systems Perspective*, pages 58–71. Springer.

Wilson, S. (2017). EUROCONTROL Specification for SWIM Information Definition version 1.0. Technical report, Eurocontrol, Brussels, Belgium.

Wilson, S., Keller, S., Marrazzo, G., and Suzic, R. (2015). AIRM Compliance Framework. Technical report.

Winkler, W. E. (1999). The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*.

Wong, A., Yip, F., Ray, P., and Paramesh, N. (2008). Towards semantic interoperability for it governance: An ontological approach.