Applying Semantic Web Technologies to Assess Maintenance Tasks from Operational Interruptions: A Use-Case at Airbus

Ghislain Auguste Atemezing(✉)

MONDECA, 35 Boulevard de Strasbourg, Paris, France
ghislain.atemezing@mondeca.com

Abstract. Airbus, one of the leading Aircraft company in Europe, collects and manages a substantial amount of unstructured data from airlines companies, related to events occurring during the exploitation of an aircraft. Those events are called "Operational Interruptions" (OI) describing observations and the work performed associated by operators in form of short text. At the same time, Airbus maintains a dataset of programmed maintenance task (MPD) for each family of aircraft. Currently, OIs are reported by companies in Excel spreadsheets and experts have to find manually in the OIs the ones that are most likely to match an existing task. In this paper, we describe a semi-automatic approach using semantic technologies to assist the experts of the domain to improve the matching process of OIs with related MPD. Our approach combines text annotation using GATE and a graph matching algorithm. The evaluation of the approach shows the benefits of using semantic technologies to manage unstructured data and future applications for data integration at Airbus.

Keywords: Information retrieval · Tagging system · Graph matching · CA-Manager · GATE · Airbus

1 Introduction

Semantics enable machine-to-machine exchange and automated processing of data to facilitate the integration of business processes and systems [4]. The main goal of having machine readable information is to search, reuse information and develop innovative applications easier. Semantic Web [2] technologies are becoming more and more adopted outside the research communities or academics. The most promising of adopting these technologies is the benefits of using open standards developed by the W3C. Linked (open) data [3] has already shown its application in many domains, such as health, cultural heritage, libraries, etc. What is starting to appear is the use of semantics in aircraft industries where some often tools developed by the semantic web are perceived to be not mature enough by expert domains. This paper describes the challenges of using semantic technologies for matching events occurred during the exploitation of an aircraft

E. Blomqvist et al. (Eds.): ESWC 2017, Part II, LNCS 10250, pp. 3–17, 2017.
DOI: 10.1007/978-3-319-58451-5_1

reported by a company with the official list of maintenance programmed by Airbus during the life-cycle of a given aircraft. The goal is to update the maintenance task (MPD) with new issues if the detection of operational interruptions (OIs) is relevant. The ultimate goal is to anticipate failures reported by the aircraft companies to the manufacturer. The main contributions of this paper are the following: (i) Capture the implicit knowledge of expert domains in RDF for generating gazetteers using SKOS [8] concepts; (ii) propose and implement a semantic annotation workflow to detect relevant OIs concepts and (iii) implement a robust graph matching algorithm to make recommendation to expert for assessment and validation. The results obtained are very promising as we are able to obtain a high level detected OIs (81.52%) with a very limited amount of keywords provided by experts (less than 150 concepts in the gazetteer). The paper is structured as follow: In Sect. 2, an overview of the domain expertise and problem statement are presented, followed by the vocabularies and data model in Sect. 3. Section 4 describes the dataset converted in RDF. The semantic annotation is described in Sect. 5, then the graph matching algorithm is presented in Sect. 6. The experiments and lessons learned are respectively presented in Sects. 7 and 8. Section 9 briefly expose some related work before concluding of the paper.

2 Problem Statement

In this section, we describe the scope of the domain at Aircraft and the main challenge of the provided solution in this paper. We first describe the concepts related to aircraft domain of our study, and then presents the approach using semantic technologies to tackle this real-world use-case.

2.1 Domain Description

Operational interruptions (OIs) are incidents occurring during the use of the aircraft, with an impact on commercial exploitation. All maintenance events are not OIs though: we need a minimal delay time for this to be considered an OI, i.e., if the cause is neutralized very quickly, an OI is not generated). Secondly, the OIs are not only technical incidents due to an aircraft: a crew arriving late because the van had broken down can also be an OI. Also, some elements occurring in the plane does not have a "technical" cause: for example, a collision with a bird (bird strike) will require maintenance, while the plane itself was not involved.

Generally, an OI contains the description of the interruption (what happened: it may be an effect or failure (failure message, vibration, noise, etc.) or a description of what the pilots or ground personnel observed), actions taken to address them (sometimes there are none), and additional metadata as the ATA code[1], the aircraft (type, operator, number, engine, etc.) the "*changeability*" (e.g., is it due to the plane? to ground maintenance teams? etc.), the operational impact

[1] https://en.wikipedia.org/wiki/ATA_100.

(does that re-routed the plane? Did that cancel the flight? etc.), the type of failure, etc. Today, OIs are used primarily for teams studying aircraft reliability to derive statistics and make recommendations.

Scheduled maintenance aims to avoid preemptively that some outages occur, develop into silence and ultimately have a significant impact on the use of the aircraft or its security. Some tasks are performed "before" the effect occurs: inspection of areas, equipment testing, periodic replacement. Not all parts of the aircraft are subject to scheduled maintenance. For example, some devices have sensors, integrated testing means, and thus emit messages in case of failure or when their condition deteriorates - so they perform in these cases, corrective maintenance. Therefore, many OIs concern elements that scheduled maintenance would not have prevented.

On one hand, the scheduled maintenance tasks are described in the Maintenance Planning Document (MPD), with their title, interval, the areas where they occur, the necessary duration, etc. However, the MPD itself does not describe why it is necessary to perform these tasks. On the other hand, the Maintenance Review Board Report (MRBR) falls under analysis by experts of scheduled maintenance, in agreement with the authorities and operators. The objective of these teams is to see if the items to be covered are often detected or not, to adjust the range of preventive actions to more suitable ones. Therefore, they look at reports of scheduled maintenance operations (reports of "Findings"), but need also data from non-scheduled operations to be complete which require the analysis of OIs.

Each MRBR item is a task involving one or more functions, each of which can have multiple causes, which will induce a functional failure in the systems of a plane with effects or consequences on the plane. The effects can be detectable or not. Each row in the MRBR describes a failure. A link exists between MRBR and MPD task by the following rules: (R1)- each MPD task can cover one or more MRBR items (or none) and (R2)- some MPD tasks are derived from further analysis.

This paper solves the following challenge: to identify the indirect link between OIs and MPD task. That is: if the OI describes a cause that would have been avoided by conducting an MPD task, then we associate the corresponding MPD task. Therefore, we seek in OI (description, metadata) elements related to potentially preventable failures of MPD, so as to associate it causes, failures and consequences described in MRBR items. Hence the keywords used by experts to identify potentially interesting OI.

2.2 Our Approach

The main challenge is to assist the expert with a reduced list of OIs candidates that can be matched with some existing tasks in the MPD. The top down approach takes a task, uses the list of keywords and the implicit knowledge of the expert (e.g., experience) and try to match with OI. Currently this is the approach manually used by domain experts by means of formulas in spreadsheets to find and filter sets of OIs based on list of keywords. A second approach consisting a fully automatic process, from OIs annotations and matching without experts in

the loop (bottom-up approach) was not acceptable by the domain experts. We define an *hybrid approach* where the expert can assess and validate the results of the approach, consisting of the following (as shown in Fig. 1):

1. OIs are annotated with keywords from the experts.
2. MP tasks already treated are modeled in RDF with associated keywords.
3. Find OIs candidates matching the set of reference tasks by using any relevance information such as ATA references and aircraft metadata.
4. Experts evaluate and validate the suggested OIs candidates.

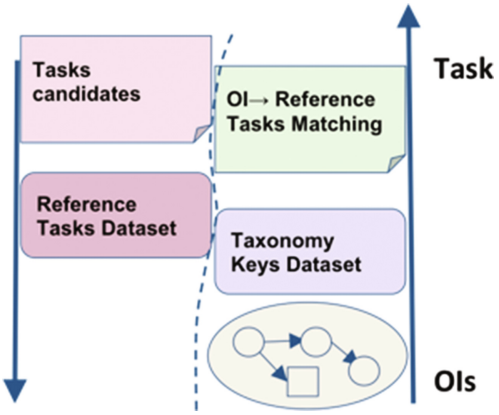

Fig. 1. The hybrid approach used to match the OI descriptions with MPD tasks, using semantic technologies.

3 Vocabularies and Data Model

The legacy data in spreadsheets is converted into RDF for better integration and homogeneity. Two vocabularies is designed to capture the knowledge presented in the diverse data: a vocabulary for operational interruptions (oi-vocab) and a vocabulary for maintenance tasks and analysis experts report (task-vocab). The URI namespaces used for the vocabularies follow the pattern `http://data.airbus.com/def/{vocab-prefix}#`.

3.1 OI Vocabulary

The OI vocabulary[2] is a lightweight model with one main class and three data properties. An operational interruption is modeled as a subclass of an event, reusing the class event:Event of the event ontology[3]. The data properties oi:ataref, oi:description and oi:workPerform are used to respectively capture the ATA reference, the textual description of an OI and the work performed by an operator. Figure 2 depicts the vocabulary and a sample graph generated with the model.

[2] http://data.airbus.com/def/oi.
[3] http://purl.org/NET/c4dm/event.owl.

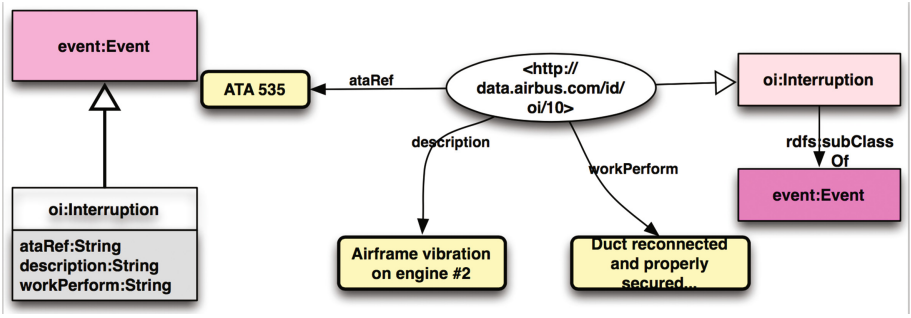

Fig. 2. The left side shows the model and the right side an example of graph based on the OI vocabulary

3.2 Task Vocabulary

A vocabulary[4] for describing the relationships the failures and the maintenance tasks has been created to build a knowledge graph for integrating different silos of documents in Airbus organization. Currently, four main classes are defined: tf:Task, tf:MRBR, tf:Failure and tf:ProgramTask. Figure 3 shows an excerpt of the current usage of the vocabulary.

4 RDF Dataset

One of the main challenge in the generation of RDF data from the different spreadsheets is to create unique URIs to identify instances of the classes from the vocabularies. Each element of the graph should be identified uniquely with a persistent URI. The chosen scheme for the URI for generating RDF dataset is the following: *http://data.airbus.com/id/*{*class*}/{*aircraft-family*}/{*year*}/{*code*} where {class} is one of (oi, task, mrbr); {aircraft-family} is one of (a330, a320); {Year} a four digit for the year and {code} is generated by concatenating a number of the XLS file name + raw number in the file. For example, this URI <http://data.airbus.com/id/oi/a330/2014/07011231177> represents in the Knowledge Graph the OI for an A330 received in 2014, where the description is within the file "isaim-a330-2014-0701to1231.xls" in line 177. Table 1 shows the generated graphs in RDF for the two aircraft families: A330 and A320 for the period from 2013 to 2015. The conversion process to RDF of all the Excel files is performed by the Datalift [12] platform, using custom CONSTRUCT SPARQL queries to transform columns to specific classes and properties.

5 Semantic Annotation

Since the input of the descriptions and the work performed for the each OI is short text in English, an annotation pipeline based on the GATE architecture

[4] http://data.airbus.com/def/tf.

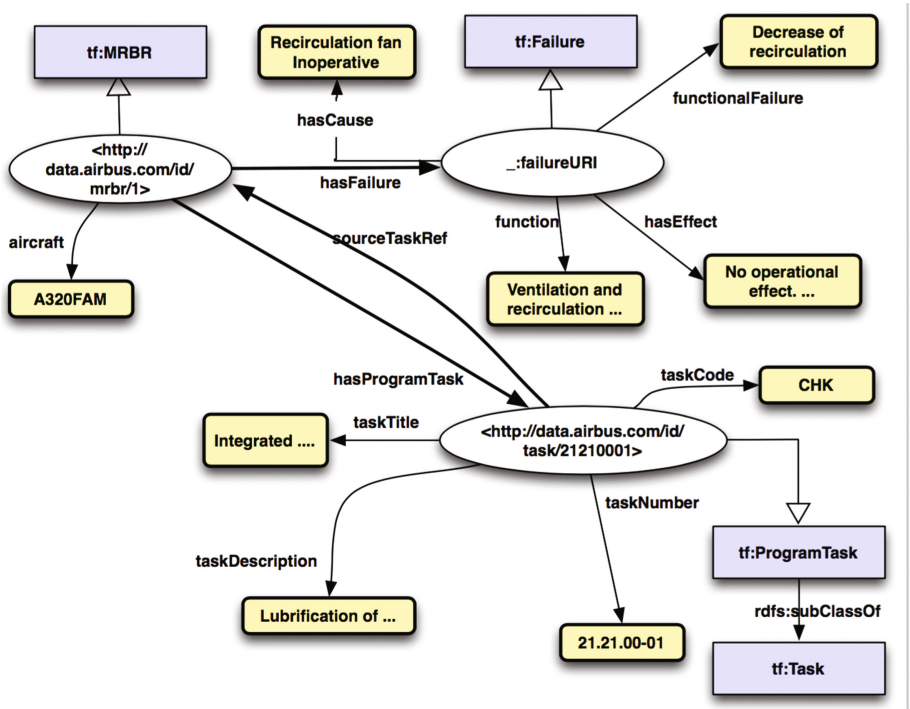

Fig. 3. A sample graph showing the relations between tasks, master documents and failures using the mpd-vocabulary.

Table 1. Overall number of the data converted into RDF by family of Aircraft for the period 2013–2014 (A330) and 2014–2015 (A320).

Class	OI		MRBR- MPD		
Dataset	NbTriples	NbOIs	NbTriples	NbMRBR	NbTask
A320	111,044	27,761	19,481	1,969	880
A330	60,867	21,400	54,457	6,871	4,234

[6,7] is implemented to detect and tag the relevant concepts. CA-Manager is the annotation tool developed at Mondeca to annotate heterogeneous unstructured content with personalized pipelines based on GATE (Fig. 4).

The architecture of the workflow consists of the following modules:

1. skos-ification module, in charge of converting into SKOS concepts the experts keywords.
2. SKOS2Gate module, for creating the gazetteer to be used during the annotation of the OIs.
3. CA-Batch module, for creating multi-thread calls to annotate the text documents.

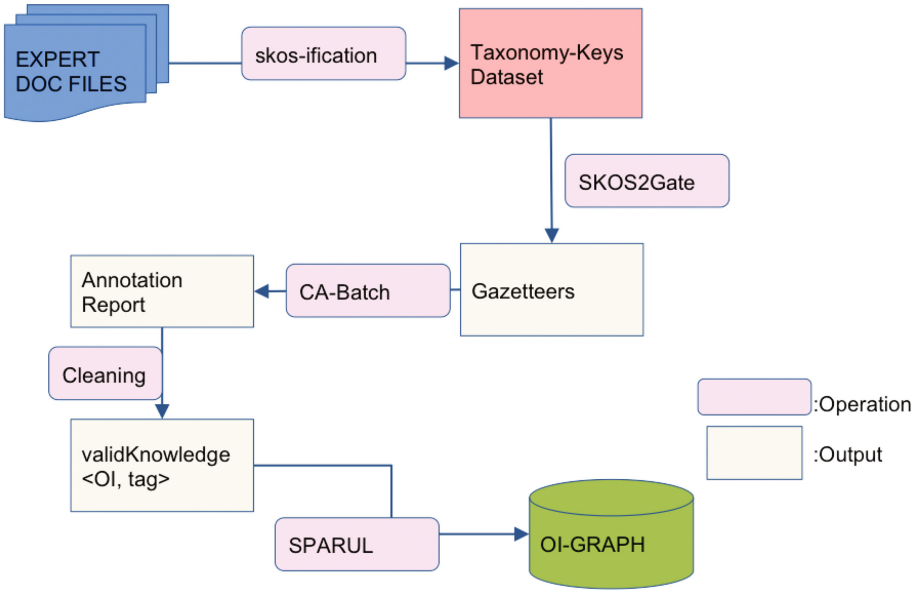

Fig. 4. Workflow of the semantic annotation and knowledge base updates.

4. Cleaning module, in charge of removing non relevant information from the reports generated by the annotator.
5. SPARUL module, which is in charge of updating the dataset in the endpoint using SPARQL updates queries, and enriching the underlying RDF graph.

5.1 SKOS-ification Process

The skos-ification process consists of manually convert the keywords used by the expert of the domain into SKOS concepts that is later used for generating the gazetteers during the annotation. Since the input file are DOCX files, we manually create the RDF file in Turtle. Some rules and hypothesis are made during the process based on different interviews with the experts of the domain (Table 2):

- Ordinal numbers such as "Third", "First", etc. are not taken into account.
- A keyword can not be at the same time in the description AND in the action work performed of the OI. The two sets are DISJOINTS.
- Descriptors SHOULD be short concepts(names), with variants in the case of verbs that are treated as SYNONYMS (see sample below in Turtle format)
- Expressions of type *Not[properly][term]* are modeled as just [term a skos:Concept]. For e.g., in the case of the term "not properly closed", we only model the concept "close" with the variant closed.

Table 2. Number of SKOS concepts manually generated from domain experts file, grouped by description and work performed.

Class	Description		WorkPerformed		Total
Feature	descItem	descAction	workItem	workAction	-
SKOS concept	47	45	22	16	130

```
@prefix workAction: <http://data.airbus.com/id/scheme/workAction/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
workAction:Change a skos:Concept ;
    skos:prefLabel "Change"@en ;
    skos:inScheme workAction:keyword ;
    skos:altLabel "Replace"@en, "Changed"@en, "Replaced"@en , "Replacement"@en .
```

The experiment described in this paper uses 20 tasks with keywords annotated manually by experts. Together with the corresponding task, the resulting file represents a gold standard where the descriptors are linked to appropriate task, as it is depicted in Fig. 5.

The class tf:DescriptionKey is designed to capture the two different types of keywords: the description and the work performed. Once they are converted into SKOS concepts, they are used to identify relevant concepts in OIs text. Furthermore, four properties are used to link to the appropriate namespaces: tf:descItem, tf:descAction, tf:workItem and tf:workAction.

5.2 Content Annotation

The annotation process employed is based on a central component: the Content Augmentation Manager (CA-Manager) [5]. The content augmentation manager (CA-Manager) is in charge of processing any type of content (plain text, XML, HTML, PDF, etc.). This module extracts the concepts and entities detected using text mining techniques with the text input module. The strength of CA-Manager is to combine semantic technologies with a UIMA-based infrastructure[5] which has been enriched and customized to address the specific needs of both semantic annotation and ontology population tasks.

The scenario presented in this paper is built on top of the GATE framework for entity extraction. CA-Manager uses an ontology-based annotation schema to transform heterogeneous content (text, image, video, etc.) into semantically-driven and organized one. The overall architecture of CA-Manager is depicted in Fig. 6. We first create the gazetteer with the SKOS representation of the documents obtained from the experts. We then launch in parallel 10 documents in multi-threads containing OIs. The annotation report contains the valid knowledge section, an RDF/XML document containing the URIs of the concepts detected by the annotator. We then use a python script to map each document with its corresponding URI. Finally, a SPARQL update query is launched to update the dataset containing the OI-Graph.

[5] Unstructured Information Management Architecture (http://uima.apache.org).

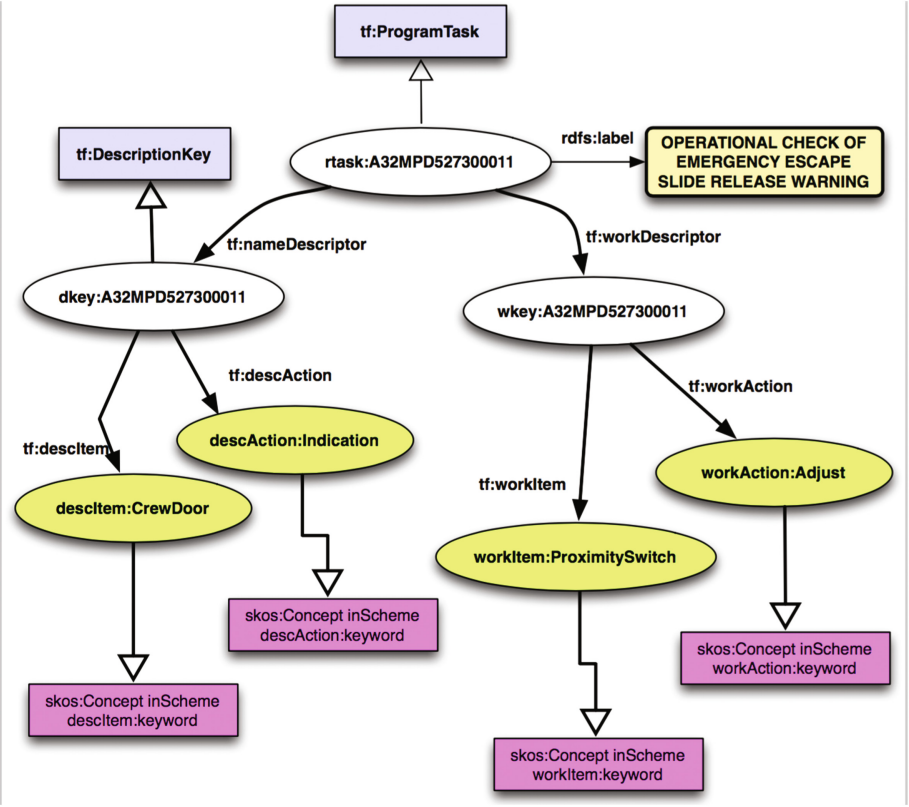

Fig. 5. A sample of SKOS concepts used to model the experts' keywords and linked to a task and maintenance graph

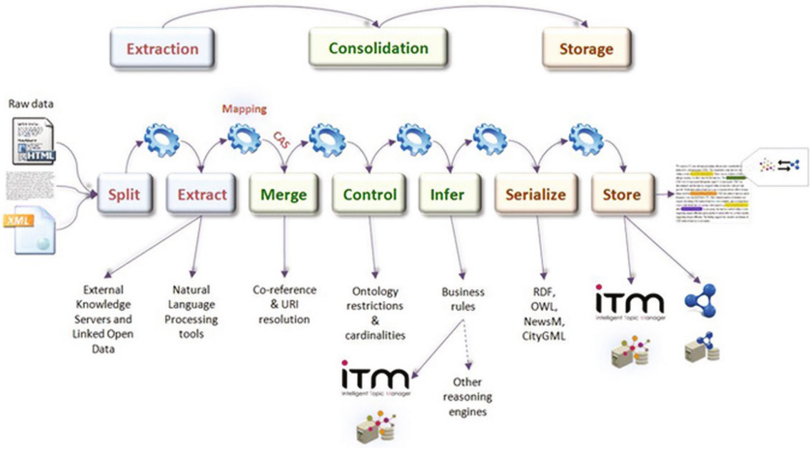

Fig. 6. Pipeline of annotation using CA-Manager

6 Graph Matching Algorithm

Algorithm 1 (Relax version) is the first variant for creating candidates. The initialization sets are presented in lines 1–4, where there are two subsets. We reduce the size of the OIs by taking into account only those with tags in one of the subsets presented (lines 13, 23 and 26) in the initialization phase. The algorithm uses the logical UNION operator to compute the intermediate sets for comparison and the resulting set of candidates in C. In this case, $C = UNION(P, Q, P', Q')$ where $P = UnionOfDescActions$, $Q = UnionOfDescItems$, $P' = UnionOfWorkActions$ and $Q' = UnionOfWorkItems$. All the queries used in the algorithm are based on SPARQL and the computation of the sets involving intersections make use of the LIMES [9] matching tool.

A second approach for a less relax version (LRX-version) is to modify the lines 14, 24 and 27 respectively in Algorithm 1 by doing the adjustments described in Algorithm 2. Depending on how we combine the results makes the difference between the LRX and RLX algorithm (lines 14 and 24), with the corresponding mappings in the modified RLX algorithm in lines 3 and 5.

Algorithm 1. GraphMatching (OI-Graph,MPD-Graph) - RLX Version

1: Input: Set of OIs tagged $< OI, tag >$
2: Input: Set of MPD in RDF with tags $< MPD, tag >$.
3: INITIALIZE $Description = \{DescItems, DescActions\}$
4: INITIALIZE $WorkPerformed = \{WorkItems, WorkActions\}$
5: BEGIN
6: **for** each $j \in MPD$ **do**
7: SELECT the list of tags in $DescActions$.
8: FIND in the set $< OI, tag >$ with at least one $descItem$
9: COMPUTE P = card(OIs with at least one of the MPD tag in $DescActions$) = $|UnionOfDescActions|$
10: SELECT the list of tags in $DescItems$.
11: FIND in the set $< OI, tag >$ with at least one $descAction$
12: COMPUTE Q = card(OIs with at least one of the MPD tag in $DescItems$) = $|UnionOfDescItems|$
13: SELECT OIs in the set (P **OR** Q).
14: COMPUTE R = UNION(P,Q) = P + Q - INTERSECTION(P, Q)
15: **end for**
16: **for** each $j \in MPD$ **do**
17: SELECT the list of tags in $WorkActions$.
18: FIND in the set $< OI, tag >$ with at $least\ one\ workItem$
19: COMPUTE P' = card(OIs with at least one of the MPD tag in $WorkActions$) = $|UnionOfWorkActions|$
20: SELECT the list of tags in $WorkItems$.
21: FIND in the set $< OI, tag >$ with at $least\ one\ workAction$
22: COMPUTE Q' = card(OIs with at least one of the MPD tag in $WorkItems$) = $|UnionOfWorkItems|$
23: SELECT OIs in the set (P' **OR** Q').
24: COMPUTE $R' = UNION(P', Q') = P' + Q' - INTERSECTION(P', Q')$
25: **end for**
26: SELECT OIs in the set $Description$ **OR** $WorkPerformed$.
27: COMPUTE $C = UNION(R, R') = R + R' - INTERSECTION(R, R')$
28: **return** C
29: END

7 Experiments and Results

We perform experiments in both the annotation and the graph matching algorithm. To perform this, we use the datasets for the A330 family of Airbus aircrafts during the year 2013–2014.

7.1 Experiments

We perform the annotation of 21,400 OIs consisting of the descriptions and the work performed in English. We use a single machine on Windows 10, Core i7; 6 Go RAM. The annotation took 1 h 48 min (6434.464 s), with an error rate by CA-Manager of 2.71%. Moreover, 81.52% of OIs were tagged (17,446), where more than 11K OIs have at least two tags. This is important because the graph matching algorithm uses this set of OIs with at least two tags detected. Table 3 gives more details of the annotation process results.

Algorithm 2. GraphMatching - LRX-Version

```
SAME INIT AS ALGO RLX-version
.....
BEGIN
COMPUTE R = UNION(P, P')
......
COMPUTE R' = UNION(Q, Q')
......
COMPUTE C = INTERSECTION(R, R')
return C
END
```

Table 3. Details of the annotation content of the OIs for the A330 family.

Graph	Feature	Number
OI A330-Graph	NbTriples	60,867
	NbOI	21,400
	NbOIs-tagged	17,446
	Max. tags	12
	Min.tag	1
	NbOI-with-one-tag	5,954
	CAM-processed	20,819
	Errors-annotation	581
	Time	6,434.464 s

7.2 Results

We present the results of applying our annotation and matching to assess the possible OIs candidates for three tasks for A330 and the OIs of the same aircraft family during the year 2013–2014 (Table 4).

Table 4. Results of the matching process in terms of size of the candidates using Algorithm 1 with three tasks for A330 and the OIs of the same aircraft family during the year 2013–2014.

Task ID	P	P'	Q	Q'	Inter (P,Q)	Inter (P',Q')	R=Union (P,Q)	R'=Union (P',Q')	Inter (R,R')	Union (R,R')
MPD245000031	545	271	129	0	41	0	633	271	2	**902**
MPD271400031	110	306	0	366	0	250	110	422	7	**525**
MPD235100021	545	271	109	0	19	0	635	271	1	**905**

In a more open criteria, as it is the case with Algorithm 1, we obtain the tasks T1(MPD245000031), T2 (MPD271400031) and T3 (MPD235100021) 902, 525 and 905 potential candidates. That means almost more than 95% of the OIs are not relevant at all to those three tasks. But still many OIs to review by a human expert. So, after some interviews with the domain experts, the LRX-version of the algorithm (see Algorithm 2) is implemented suggesting a more reduced set of candidates, without loosing any good candidate. With this variant, it reduces significantly the number of the candidates (an average of 63%) for reviewing (Table 5).

Table 5. Results of the matching process when using Algorithm 2, the LRX version.

Task ID	P	P'	Q	Q'	Inter (P,P')	Inter (Q,Q')	R=Union (P,P')	R'=Union (Q,Q')	Inter (R,R')
MPD245000031	545	271	129	0	13	0	803	129	**40**
MPD271400031	110	306	0	366	8	0	408	366	**4**
MPD235100021	545	271	109	0	13	0	803	109	**24**

Currently the team in charge of manually detect relevant OIs uses four complex Excel functions to assess the OIs. When presented the solution with the data we received from them, they were satisfied with the results, meaning we were able to use semantics to capture both their expertise and their daily work.

7.3 Evaluation

The experts of this domain are difficult to access. However, we were accompanied by 3 of them during the work to better understand the challenges, to gather relevant datasets. The experts help refining the gazetteers in reviewing some candidates to identify false positives. This sample[6] shows one view of the type of call-for-arms during the duration of the project. Since they are very busy experts

[6] https://github.com/gatemezing/eswc2017/blob/master/sampleEvalForTuningGazetteers_eswv2017.pdf.

of the domain, we didn't have access to a full "evaluation" *ala* research fashion (computing precision and recall for a representative set of OIs). Nevertheless, the domain experts gave us those 3 tasks on purpose (see https://goo.gl/CM3KzB for Turtle transcription of the word file received) - as they can easily check the results - for our scenario to see if we were able to miss some relevant OIs. This work also shows an application of semantic web to solve a real business use case, covering all the process of creation the ontologies, the population of the different knowledge bases from heterogeneous data, etc. The output of this work is installed at Airbus-Toulouse in a virtual machine for internal usage.

8 Lessons Learned

One of the barrier in the adoption of Semantic technologies in industry is the difficulty to explain the key concepts underlying RDF model, vocabulary creation and the benefits of changing the paradigm from traditional data management to semantic repositories. This was one of the challenge of this project at Airbus, starting with a real use case to increasingly find the benefits of semantic technology to solve a real world problem. During the time frame of this work, several meetings where held to explain the "why" at every single steps.

The team in charge of collecting the OIs from diverse companies and experts working on maintenance tasks has to progressively make a mental shift with the new paradigms introduced during this project:

- Data model by implementing vocabularies with Protégé [10] from scratch driven by the existing data.
- The notion of unique identifiers to be used across different data silos to represent the same real world object. Hence, a policy for creating persistent URIs was clearly identified as crucial
- The confusion between the RDF model and the different serializations

This work also permits to identify many datasets that could be useful in other services at Airbus to bring more context and integrate them easily. For example, during some interviews, the experts realized that some implicit knowledge can be derived actually from existing datasets, which otherwise are experts' experience. Moreover, the system proposed would not replace the experts but assist them in their work to speed up the process. This is one of the strong requirement of the proposed approach.

Additionally, the conversion process into RDF reveals some errors in the legacy data. For example, sometimes there were not consistent use of the task code across different XLS files. This shows the benefits of our approach compared to existing one as it helps detecting and cleaning errors in the data. As the result of this work, the expert team is looking at a better combination for the output sets to adjust the proposed algorithm based on their evaluation of the proposed candidates.

9 Related Work

This work falls under contributions and research in the intersection of natural-break language processing with GATE, enterprise semantic data management and matching techniques. The authors in [13] examine semantic annotation, identify a number of requirements, and review the generation of semantic annotation systems. Dadzi and et al. [11] developed an integrated methodology to optimize Knowledge reuse and sharing in the aeronautics domain based on ontologies. They proposed an interface for multiple modalities during search of documents based on their approach. While our approach also used domain ontologies to convert legacy data into RDF, it also combines NLP techniques for annotating textual data, and SPARQL queries for matching purposes. Recently, in [1] it is defined a state of the art in semantic annotation models and a scheme that can be used to clarify requirements of end-user use cases. Moreover, the collection of real world applications from industry in [4] tackle some challenges in the earlier adoption of semantic technologies in industry. Although some issues remind challenging today, a diversity of domains applying semantics is visible worldwide.

10 Conclusion and Future Work

This paper presents a semi-automatic approach using semantic technologies to assist the experts of the aircraft domain to improve the matching process of Operational Interruptions with related Maintenance Programming Task. Our approach combines text annotation using GATE alike system for annotation and a graph matching algorithm for suggesting candidates. Annotation of 21,400 OIs show a high amount of tag detected (82%) with a small amount of expert concepts. The graph matching technique shows that by combining suitable combinations, it is possible to reduce to up to 63% the amount of candidates of OIs to be assessed by domain experts. The first results shed the light for future intensive application of semantic technologies at Airbus for many other aspects, such as data integration, data fusion and semantic recommendation tools.

For future work, we plan to improve the semantic annotation by looking at the better management of noises in the text for annotation. Also, we plan to implement fuzzy annotation in the GATE pipeline to find a trade-off between performance and recall. Currently we are using a small gazetteer, we plan to integrate external sources to have a much bigger scope for the terms to detect.

Acknowledgments. We would like to thank the Airbus team in Toulouse and ATOS colleagues for their valuable input and partnership.

References

1. Andrews, P., Zaihrayeu, I., Pane, J.: A classification of semantic annotation systems. Semant. Web **3**(3), 223–248 (2012)
2. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Sci. Am. **284**(5), 28–37 (2001)

3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semant. Web Inf. Syst. **5**, 1–22 (2009)
4. Cardoso, J., Hepp, M., Lytras, M.D.: The Semantic Web: Real-World Applications from Industry, vol. 6. Springer Science & Business Media, Heidelberg (2007)
5. Cherfi, H., Coste, M., Amardeilh, F.: Ca-manager: a middleware for mutual enrichment between information extraction systems and knowledge repositories. In: 4th Workshop SOS-DLWD Des Sources Ouvertes au Web de Données, pp. 15–28 (2013)
6. Cunningham, H.: Gate, a general architecture for text engineering. Comput. Humanit. **36**(2), 223–254 (1996)
7. Kenter, T., Maynard, D.: Using gate as an annotation tool. University of Sheffield, Natural language processing group (2005)
8. Miles, A., Bechhofer, S.: SKOS simple knowledge organization system reference. W3C (2009). https://www.w3.org/TR/skos-reference/
9. Ngomo, A.-C.N., Auer, S.: Limes - a time-efficient approach for large-scale link discovery on the web of data. In: Proceedings of IJCAI (2011)
10. Noy, N.F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R.W., Musen, M.A.: Creating semantic web contents with protege-2000. IEEE Intell. Syst. **2**, 60–71 (2001)
11. Dadzie, A.-S., Bhagdev, R., Chakravarthy, A., Chapman, S., Iria, J., Lanfranchi, V., Magalhães, J., Petrelli, D., Ciravegna, F.: Applying semantic web technologies to knowledge sharing in aerospace engineering. J. Ind. Manuf. **20**, 611–623 (2009)
12. Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., Hamdi, F., Bihanic, L., Képéklian, G., Cotton, F., et al.: Enabling linked-data publication with the datalift platform. In: Proceedings of AAAI Workshop on Semantic Cities (2012)
13. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: requirements and a survey of the state of the art. Web Semant. Sci. Serv. Agents World Wide Web **4**(1), 14–28 (2006)