

Abdallah Abdulsalam Abdallah

Semantic Integration of Heterogeneous Aircraft Though-Life Documentation

School of Aerospace Transport and Manufacturing
(Integrated Vehicle Health Management (IVHM) Centre)

Doctor of Philosophy
Academic Year: 2020 - 2024

Supervisor: Dr Ip-Shing Fan
Associate Supervisor: Professor Nico Avdelidis

June 2024



School of Aerospace Transport and Manufacturing

PhD

Academic Year 2020 - 2024

Abdallah Abdulsalam Abdallah

Semantic Integration of Heterogeneous Aircraft Though-Life Documentation

Supervisor: Dr Ip-Shing Fan
Associate Supervisor: Professor Nico Avdelidis
May 2024

This thesis is submitted in partial fulfilment of the requirements for the degree of PhD

© Cranfield University 2024. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner

Academic Integrity Declaration

I declare that:

- the thesis submitted has been written by me alone.
- the thesis submitted has not been previously submitted to this university or any other.
- that all content, including primary and/or secondary data, is true to the best of my knowledge.
- that all quotations and references have been duly acknowledged according to the requirements of academic research.

I understand that to knowingly submit work in violation of the above statement will be considered by examiners as academic misconduct.

Abstract

This thesis investigates the challenges of integrating heterogeneous aircraft maintenance records through the development of a novel ontology and prototype application, addressing the complexity and heterogeneity of the data sources, formats, and semantics inherent in aircraft maintenance documentation. This research focuses on the maintenance records of a Boeing 737-400 available at Cranfield to develop Aircraft Maintenance Records Ontology (AMRO). Ontology provides a structured representation of domain knowledge, facilitating improved data consistency by addressing semantic and syntactic inconsistencies. This study aims to minimise manual processes in handling maintenance records, thereby enhancing efficiency, accuracy, and reducing errors related to data inconsistencies.

This study outlines an agile development for ontology-based application (ADOBA) methodology, integrating ontology and application development to ensure continuous feedback and improvement. The developed prototype demonstrates practical application in correcting inconsistencies and enhancing record accuracy using Named Entity Recognition (NER) for component categorisation with the corrected data. The performance of the AMRO model and prototype was evaluated using standard metrics, such as precision, recall, and F-measure, confirming their effectiveness in improving data consistency and operational efficiency.

The key contributions of this research include a systematic review of aircraft operations and maintenance knowledge management, highlighting challenges related to data heterogeneity and integration, introduction of the ADOBA methodology, and development of AMRO based on real-world maintenance records. This thesis demonstrates the potential of semantic technologies in improving data consistency and operational efficiency in aircraft maintenance, setting a foundation for future advancements and standardisation efforts in the aviation industry.

Future work should focus on aligning the AMRO model with international standards, expanding its coverage to include predictive maintenance and lifecycle management, and integrating advanced Natural Language Processing (NLP) models to enhance data interpretation and automation. This research emphasises the importance of robust knowledge management systems to ensure compliance with aviation regulations and improve the overall efficiency of managing aircraft maintenance records.

Keywords:

Knowledge Management, Aircraft Operations and Maintenance (AOM), Ontologies, Knowledge Representation, Aircraft Maintenance Records.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Dr. Ip-Shing Fan and Professor Nico Avdelidis, for their invaluable guidance and support throughout this research.

Special thanks to Abdul Abushalla for his support in providing the maintenance data essential for this research.

I am also deeply grateful to my research colleagues at the Digital Maintenance Repair and Overhaul (DMRO) Lab for their collaboration and insightful discussions that enriched this work.

I extend my heartfelt appreciation to my manager at TUI, Amanda Grafton for her unwavering support and encouragement, which enabled me to balance my professional and academic responsibilities.

I would also like to express my deep gratitude to my parents and family for their constant support, understanding, and encouragement throughout the journey. Their love and belief in me have been an incredible source of strength.

Thank you all for your contributions and support.

Table of Contents

Academic Integrity Declaration i
Abstract ii
List of Figures vii
List of Tables ix
List of Equations xi
List of Abbreviations xii
Definition of Terms xiii
1 Introduction 1
1.1 Aircraft Maintenance Documentation 1
1.1.1 G-DOCB Boeing 737-400 Case Study 1
1.2 Problem Description 2
1.2 Research Aim and Objectives 4
1.4 Research Contributions 5
1.5 Thesis Structure 6
2 Literature Review 10
2.1 Knowledge Management in Aircraft Operations and Maintenance 10
2.2 Review Protocol 14
2.3 Existing KM approaches and solutions relevant for data integration of aircraft maintenance records 18
2.3.1 Development methods for existing KM approaches and solutions 20
2.3.2 What are the critical elements that form KM FOR AOM? 25
2.4 Evaluation of the literature 28
2.4.1 Ontology Development Methods 30
2.5 Semantic Data Integration 32
2.5.1 Ontologies for Semantic Data Integration 34
2.5.2 Semantic Modelling 36
2.5.3 Ontology Based Data Integration 37
2.5.4 Data Heterogeneity Conflicts 39
2.6 AI and Semantic Searching 42
2.6.1 Large Language Models for Resolving Syntactic and Semantic Heterogeneity 43
2.6.2 OCR and AI-Powered Preprocessing in Aircraft Maintenance Records 46
2.7 Reflection from the literature 48
3 Methodology 51
3.1 Research Methodology 51
3.1.1 Dataset Acquisition 53
3.2 Similarity Measures 54
3.2.1 Resnik 54

3.2.2 Levenshtein 55
3.3 Evaluation and Validation Metrics 57
3.3.1 Precision 57
3.3.2 Recall 58
3.3.3 Accuracy 58
3.3.4 F-Measure 59
4 Agile Development for Ontology-Based Applications (ADOBA) 61
4.1 Development Methodology 61
4.1.1 Ontology-based application Development for Aircraft Maintenance Records 67
5 Ontology Modelling and Design of Aircraft Maintenance Records 72
5.1 Modelling Aircraft Maintenance Records Ontology 75
5.2 Maintenance Document Content Analysis 77
5.2.1 Structure and Composition of G-DOCB Maintenance Records .. 77
5.3 AMRO Requirement Specification 86
5.3.1 Aircraft Maintenance Records Ontology 88
5.3.2 Core Structure of AMRO 88
6 Results Evaluation and Validation 98
6.1.1 Prototype Demonstration 98
6.1.2 Dataset Preprocessing 99
6.2 Resolving Semantic and Syntactic Inconsistencies in Task Cards .. 100
7 Discussion 128
8 Conclusion 132
8.1 Future Work 133
8.2 Limitations of the research 134
References 136
Appendix A Ethical Approval Letter 158
Appendix B Systematic Review Protocol 159
Appendix C G-DOCB Documents Indexing 169
Appendix D App Code Snippet for AMRO Ontology 177

List of Figures

Figure 1-1 Use case for the research (Cranfield B737 Maintenance Records) .. 2

Figure 1-2 Challenges of paper-based aircraft operations and maintenance 3

Figure 1-3 Challenges of digital aircraft operations and maintenance 4

Figure 1-4 Objectives of the research 5

Figure 1-5 Thesis overview 9

Figure 2-1 Systematic literature review process [70] 15

Figure 2-2 PRISMA Flowchart Showing the Identification and Screening Processes for the SLR 18

Figure 2-3 Distribution of research context 19

Figure 2-4 Distribution of selected literature 20

Figure 2-5 Distribution of selected papers 25

Figure 2-6 Essential elements of KM 26

Figure 2-7 Semantic web stack [61] 33

Figure 2-8 Transformations of ontologies features to data integration systems requirements [109] 38

Figure 2-9 Types of data heterogeneity 40

Figure 3-1 Methodological approach used in this study 52

Figure 4-1 Proposed ADOBA methodology [128] 65

Figure 5-1 Challenges of building ontology-based applications 75

Figure 5-2 Distribution of G-DOCB maintenance records 78

Figure 5-3 G-DOCB aircraft technical report and AMM task cards 79

Figure 5-4 G-DOCB Certificate of airworthiness, aircraft technical report and component change record 80

Figure 5-5 Concept mapping for document types of G-DOCB maintenance records 86

Figure 5-6 AMRO ontology 89

Figure 5-7 Defining properties for maintenance tasks in AMRO 89

Figure 5-8 Maintenance task dependencies and technician qualifications 90

Figure 5-9 Component inspection and time-limited check dependencies 91

Figure 5-10 ATA chapter classification of aircraft components..... 91
Figure 5-11 Definition and relationships of the InspectionTask class in AMRO 92
Figure 5-12 Representation of the ManHours data property in AMRO..... 93
Figure 5-13 Class and object property hierarchies in AMRO 94
Figure 5-14 Hierarchical relationships of Aircraft Class in AMRO 95
Figure 5-15 Roles and relationships of the MaintenancePersonnel Class in AMRO..... 96
Figure 5-16 Relationships and types of MaintenanceDocuments..... 97
Figure 6-1 Dataset preprocessing 100
Figure 6-2 Sample task cards..... 101
Figure 6-3 Task-based evaluation setup 105
Figure 6-4 Frequency of unique distances 111
Figure 6-5 Matching indicators 112
Figure 6-6 Cumulative distribution function of edit distances 113
Figure 6-7 Word Cloud of the most frequent used words for Defects and Actions Taken in the dataset 124
Figure 6-8 Component entities 124
Figure 6-9 Action entities..... 125

Figure B-1 Search Keywords [4]..... 164

List of Tables

Table 2-1 Selected literature for the review 21

Table 2-2 A representation of the literature containing the various essential elements of KM 27

Table 2-3 Ontology development methodologies 31

Table 2-4 Definitions of ontology 35

Table 2-5 Strengths and Limitations of LLMs classification 44

Table 2-6 Preprocessing Techniques for OCR in AOM 47

Table 4-1 Summary of ADOBA methodology 69

Table 5-1 An example of semantic and syntactic inconsistencies in maintenance task cards 73

Table 5-2 Documents attributes from the aircraft maintenance records 81

Table 5-3 Ontology requirement specification document 87

Table 6-1 Algorithm for ontology-based application 101

Table 6-2 Algorithm for NER 103

Table 6-3 Task-based evaluation steps 106

Table 6-4 Selected results of Corrected Inconsistencies in maintenance task cards for the A1 Check with similarity scores 107

Table 6-5 Summary of Matches 114

Table 6-6 Result of proposed model across selected A Check maintenance task inconsistencies 115

Table 6-7 Overall result of the model 118

Table 6-8 Comparison between AMRO model and existing ontology literature 119

Table 6-9 NER Results for A Check maintenance task Cards with and without AMRO 122

Table 6-10 Comparison between AMRO model and existing ontology models 126

Table B-1 Descriptions of the PICOC Guidelines 161

Table B-2 Inclusion and Exclusion Criteria 163

Table B-3 Search Strings. 164
Table C-1 G-DOCB documents 169

List of Equations

Equation 2-1 40
Equation 3-1 55
Equation 3-2 55
Equation 3-3 57
Equation 3-4 57
Equation 3-5 58
Equation 3-6 59
Equation 3-7 59

List of Abbreviations

ADOBA	Agile Development for Ontology-Based Applications
AI	Artificial Intelligence
AMM	Aircraft Maintenance Manual
AMR	Aircraft Maintenance Records
AMRO	Aircraft Maintenance Records Ontology
AOM	Aircraft Operations and Maintenance
ATA	Air Transport Association
ATM	Air Traffic Management
BERT	Bidirectional Encoder Representations from Transformers
CAA	Civil Aviation Authority
CBR	Case-based Reasoning
EKR	Enterprise Knowledge Resource
ETL	Extract, Transform, Load
FAA	Federal Aviation Administration
GPT	Generative Pre-trained Transformer
IATA	International Air Transport Association
IC	Information Content
KBE	Knowledge-based Engineering
KM	Knowledge Management
KMS	Knowledge Management System
MRO	Maintenance, Repair and Overhaul
NER	Named Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OWL	Web Ontology Language
PICOC	Population, Intervention, Comparison, Outcomes, and Context
RDF	Resource Description Framework
SLN	Semantic Link Network
SLR	Systematic Literature Review

Definition of Terms

Term	Definition
Agile Development for Ontology-Based Applications (ADOBA)	Iterative methodology integrating agile software development principles with ontology development, emphasising user involvement and iterative improvement.
Aircraft Maintenance Documentation	Systematic records generated during aircraft maintenance, ensuring compliance with regulatory requirements, including task cards, technical reports, and maintenance checks.
Aircraft Maintenance Operations	Activities involving inspections, repairs, servicing, and testing to maintain aircraft airworthiness and operational efficiency.
Aircraft Maintenance Records Ontology (AMRO)	Structured representation of domain-specific knowledge for aircraft maintenance records, addressing semantic and syntactic inconsistencies.
Classes	Abstract categories in ontology, representing groups of entities sharing common characteristics.
Data Categorisation	Classification of data into structured categories for improved accessibility and retrieval.
Data Heterogeneity	Variability in data format, syntax, or semantics across multiple sources.
Domain Knowledge	Expert knowledge specific to a field, capturing essential concepts and relationships.

Formalisation	Representing domain knowledge using formal languages and logical structures to ensure unambiguous definitions of concepts and relationships.
Instances	Specific occurrences or examples of classes within an ontology.
Knowledge Management (KM)	Process of creation, storing, and transferring knowledge.
Knowledge Representation and Reuse	Encoding and structuring domain knowledge into reusable formats.
Maintenance, Repair, and Overhaul (MRO)	Specialised services provided to maintain aircraft in operational and airworthy conditions.
Named Entity Recognition (NER)	Natural language processing method used to identify and categorise key information in textual data.
Ontology	Formalised representation of knowledge, defining domain concepts and relationships.
Ontology-Based Application	Software applications utilising ontologies for data categorisation and integration.
Ontology Development Methodology	Methodology guiding the structured development and validation of ontologies.
Semantic	Refers to the meaning, context, or interpretation of data across different systems.
Semantic Heterogeneity	Differences in meaning or terminology usage across data sources.
Semantic Integration	Harmonising heterogeneous data sources by resolving semantic differences.
Semantic Web Technologies	Standardised technologies (e.g., RDF, OWL, SPARQL) for structured semantic data.

“…Where is the knowledge we have lost in information?”

Thomas Stearns Eliot, Choruses from 'The Rock'

1 Introduction

1.1 Aircraft Maintenance Documentation

Aircraft operations and maintenance are subject to stringent regulation to ensure safety. This requires large volume of paperwork as evidence of adherence to due process. These records serve as a vital backbone for the aviation industry, ensuring compliance with the regulations set by aviation regulatory authorities [1]. The digitisation of these records marks a significant evolution from traditional paper-based logs, promising the enhanced efficiency, accuracy, and accessibility of critical data throughout an aircraft's lifecycle. However, this digital transformation possesses a complex challenge of integration and interoperability due to heterogeneous data sources, formats, and semantics [2].

The heterogeneity of this documentation arises from multiple factors; disparate IT systems, varying standards across jurisdictions and organisations, and the evolving nature of aircraft technology [3]. This heterogeneity significantly hinders seamless integration and interoperability of maintenance records.

In this research, a novel domain ontology and prototype that utilises real-world maintenance records is proposed. This provides a significant shift in the management and analysis of maintenance records. The proposed approaches tackle the prevalent challenges in maintenance operations, such as the data inconsistencies [4,5].

1.1.1 G-DOCB Boeing 737-400 Case Study

The use case for this research involves the maintenance records of a Boeing 737-400 aircraft, registered as G-DOCB. The maintenance records encompass various documents essential for ensuring the aircraft's airworthiness and operational efficiency. Figure 1-1 shows the key documents which includes task cards, cabin technical reports, A, B, C and D checks. Owlready2, Natural Language Toolkit (NLTK) and Wordnet are employed for ontology-based data application, providing a structured approach for correcting maintenance records. The Aircraft Maintenance Records Ontology (AMRO) offers a structured

representation of domain knowledge, facilitating improved data organisation and retrieval. Additionally, semantic and syntactic similarity measures were utilised to assess the similarity between different maintenance records. Named Entity Recognition (NER) is employed to identify and categorise key information within the corrected maintenance records including components, defects and actions.

The evaluation and validation of the developed prototype involve task-based evaluation methods, including the use of specific test cases designed to assess the functionality and accuracy of the system. Standard metrics including precision, recall, and F-measure are employed to evaluate the performance of the model and prototype.



Owner/Carrier	Cranfield University (Formerly British Airways)
Registration	G-DOCB
Aircraft	Boeing 737-400
Location	Cranfield Airport (ICAO: EGTC)
Notes	The aircraft is currently in an 'Active Storage' and it is parked at the Hangar Laboratory of the Digital Aviation Research and Technology Center (DARTeC), Cranfield University and primarily used for research purposes.

	Documents/Data Sources	Provider	Requirements		Components/ Data Integration Features	Evaluation and Validation
Specification Docs.	Aircraft Maintenance Manual	B737-400	Procedures References Tools and equipment	Prototype	Owlready2	Task-based Evaluation - Test Cases - Precision, Recall and F-measure
	Aircraft Maintenance Plan Maintenance Planning Document Cranfield Continuing Airworthiness Management Organisation (CAMO) Document	Cranfield Aircraft Maintenance Team	A Check Maintenance Tasks		Natural Language Toolkit (NLTK) and Wordnet	
					Aircraft Maintenance Records Ontology (AMRO)	
					Semantic and Syntactic Similarity Measures	
	Task Sheets and Tech logs		Reports		Named Entity Recognition	
	AMOS System	Swiss Aviation Software	Data Integration			

Figure 1-1 Use case for the research (Cranfield B737 Maintenance Records)

1.2 Problem Description

Aircraft maintenance records are a critical aspect of aviation safety and regulatory compliance, embodying the meticulous documentation of inspections, repairs, and maintenance activities performed throughout an aircraft's lifecycle. The inherent challenge in managing these records stems from their complex nature (Figure 1-2), which is characterised by a wide array of data types, formats, and terminologies. This complexity is magnified by the transition from traditional paper-based logs to digital formats, which introduces variability in how data are

recorded and stored across different platforms and systems. The heterogeneity of these records, combined with the stringent requirements for accuracy and completeness imposed by aviation regulatory authorities, render their management a complex task.

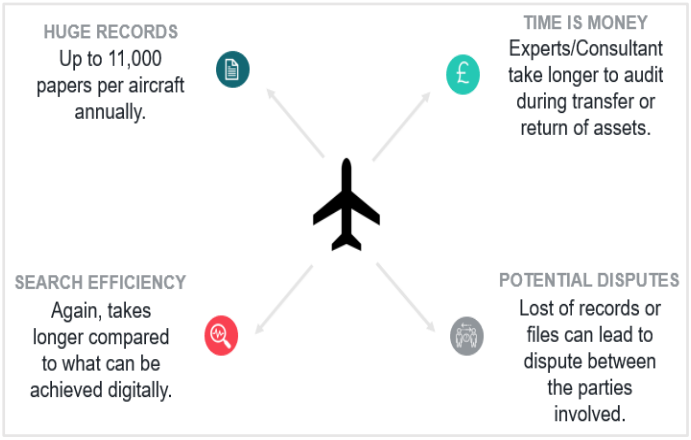


Figure 1-2 Challenges of paper-based aircraft operations and maintenance

The aircraft maintenance process unfolds into a series of structured phases, each contributing to the accumulation of maintenance records. Initially, aircraft undergo routine inspections to identify potential issues, followed by detailed assessments to diagnose the identified faults. The subsequent stages involve the execution of necessary repairs, service or replacements, culminating in post-maintenance testing to ensure airworthiness. Each phase generates significant documentation, from the initial inspection reports to detailed logs of the maintenance actions taken. This documentation process is critical not only for maintaining regulatory compliance but also for ensuring the safety and reliability of the aircraft.

Semantically, the challenge lies in the diverse terminologies and concepts used across different maintenance activities and records, which often leads to inconsistencies in how information is represented and understood. Syntactically, the variability in data formats and structures across different recording systems complicates the integration of records, requiring advanced techniques to harmonise the data. Figure 1-3 shows the representation of the summary of challenges posed by paper-based versus digital maintenance records as discussed in [1].

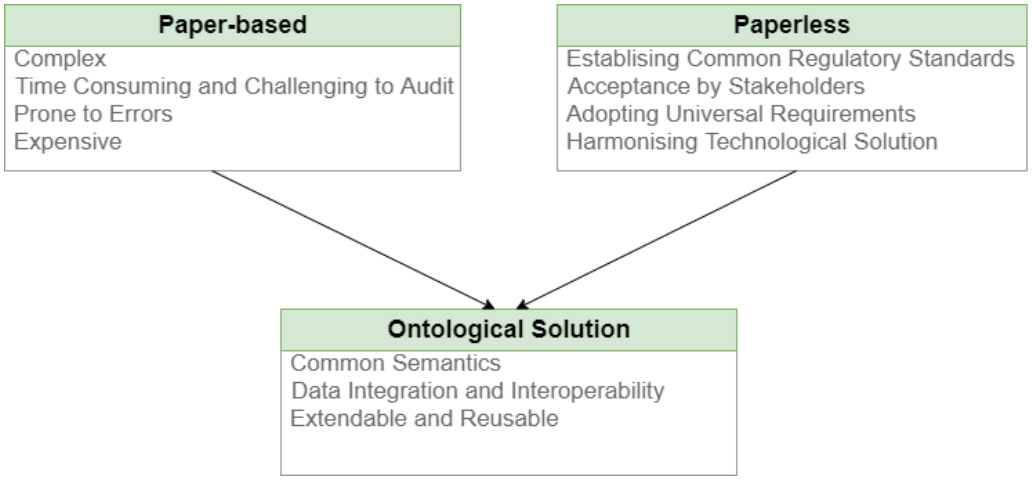


Figure 1-3 Challenges of digital aircraft operations and maintenance

The key part of this research is to minimise manual processes in the handling of aircraft maintenance records, specifically addressing semantic and syntactic inconsistencies. By introducing automation, the research aims to enhance documentation efficiency and accuracy, while reducing the potential for errors related to data inconsistencies.

1.2 Research Aim and Objectives

The aim of this research is to improve the integration of heterogeneous aircraft maintenance records using a novel semantic model - AMRO.

The following research objectives need to be realised to achieve the aim of this research:

I. Establish state-of-the-art in aircraft maintenance knowledge management to identify the research gap.

II. Establish information structure, concepts, and relationships from the document analysis of G-DOCB Boeing 737 maintenance records for the development of an Aircraft Maintenance Records Ontology.

III. To develop Aircraft Maintenance Records Ontology (AMRO), using proposed agile development methodology.

IV. To create prototype demonstration (Ontology-based application, and an NER system) for correction of inconsistencies in maintenance record.

V. Evaluate, validate, and compare the results with the literature.

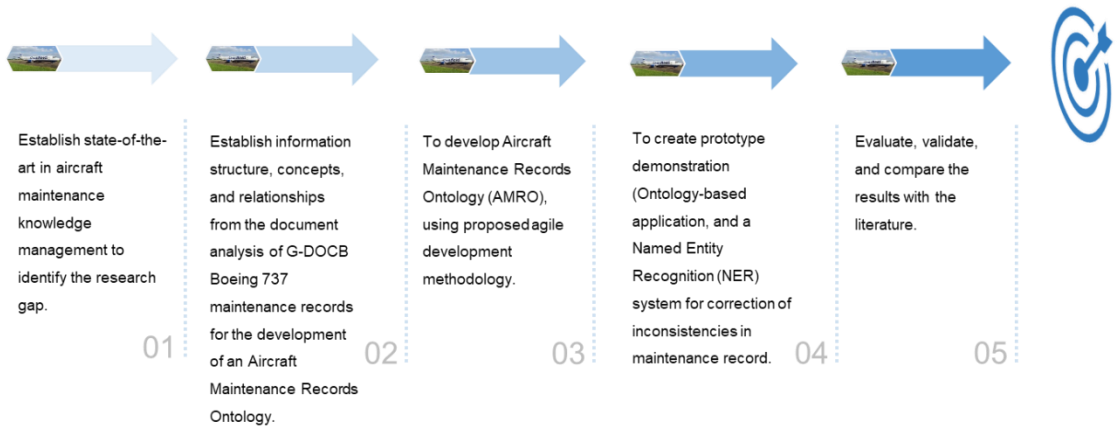


Figure 1-4 Objectives of the research

1.4 Research Contributions

This research on semantic integration of heterogeneous data represents an intersection of multiple facets of knowledge management, information science and software engineering, all contextualised in the domain on aircraft maintenance records. As the aviation industry transitions to paperless maintenance, this result in an ever-increasing amount of operational and maintenance data generated by multiple systems over the lifespan of an aircraft. The diverse and heterogeneous nature of this data makes it difficult to manage and integrate in an efficient manner. In this regard, a significant effort was made to understand the underlying information structure of aircraft maintenance records and IT systems which provided the foundation of building a domain ontology. The novel contributions stemming from this research include:

The systematic review for Aircraft Operations and Maintenance Knowledge Management which aims to consolidate the diverse nature of the research within this domain, highlighting the shift towards applying knowledge management techniques, methods and strategies in aircraft operations and maintenance domain. This is the first systematic review which brings to light the challenges related to data heterogeneity and integration in the domain based on open literature. The review also offers a methodological contribution, developing a new protocol for conducting review in the knowledge management domain, informed by best practices in the fields of information systems and software engineering.

Proposed Agile Development for Ontology-Based Applications (ADOBA) was introduced based on the understanding developed through the systematic review. ADOBA offers an integrated approach for developing ontology-based applications which bridges the gap between ontological and software engineering methodologies, contributing to the practice of ontology-based application development.

Developed an Aircraft Maintenance Record Ontology (AMRO) based using Boeing 737-400 21 years maintenance record available at Cranfield. This ontology formalises the complex domain knowledge of aircraft maintenance records, establishing it as the first semantic model of its kind. In addition, AMRO's potential to be used and reused in developing ontology-based applications which highlights its original contribution to the research domain, paving the way for further contribution.

1.5 Thesis Structure

This thesis comprises 8 chapters, the details of which are given below and also visually represented by **Error! Reference source not found.**:

Chapter 1: Introduction

This chapter sets the stage for this thesis by introducing the critical role of aircraft maintenance documentation in ensuring aviation safety and compliance with regulatory standards. It highlights the evolution from paper-based to digital records and the challenges this transformation brings, such as data integration issues owing to heterogeneous data sources and varying formats. This chapter outlines the motivation for this research, which aims to improve the integration of diverse aircraft maintenance records using a novel ontology - AMRO model. This introduction provides an overview of the research problem, aims, objectives, and contributions of this study to the field of aircraft maintenance records.

Chapter 2: Literature Review

This literature review delves into the existing body of knowledge on aircraft operations and maintenance knowledge management, with a focus on the transition to digital records and the integration challenges it presents. It evaluates various approaches and solutions for managing and integrating maintenance data, emphasising the importance of semantic technologies and ontology-based methods. This chapter discusses the heterogeneity of maintenance records and the semantic and syntactic challenges they pose, providing a comprehensive review of the current research gaps and the need for robust data standards and semantics. Reflection on the literature provides key insights and establishes the foundation for the proposed research methodology.

Chapter 3: Methodology

This chapter details the methodological approach adopted for the research and outlines the steps taken to achieve the research objectives. It begins with an initial investigation phase that includes a systematic literature review and document analysis to understand the structure of aircraft maintenance records. The methodology section describes the development and evaluation metrics used to assess the performance of the AMRO and the prototype. This highlights the iterative and agile development process, emphasising the importance of continuous evaluation and validation to ensure the reliability and accuracy of the developed solutions.

Chapter 4: Agile Development for Ontology-Based Applications (ADOBA)

In this chapter, the Agile Development for Ontology-Based Applications (ADOBA) methodology is introduced as a novel approach to building ontology-based applications. This chapter explains the phases of ADOBA, including planning, requirement specification, knowledge acquisition, ontology development, and concurrent development. It emphasises the iterative nature of the methodology, which allows for continuous feedback and improvement. The integration of ontology and application development is discussed, highlighting the importance of task-based evaluation to ensure the practical utility of solutions.

Chapter 5: Ontology Modelling and Design of Aircraft Maintenance Records

This chapter focuses on the design and modelling of Aircraft Maintenance Records Ontology (AMRO). It discusses the challenges posed by the heterogeneity of maintenance data and lack of interoperability between existing systems. This chapter details the content analysis of maintenance documents and the development of AMRO, which provides a structured representation of domain knowledge. AMRO aims to enhance data integration and retrieval by addressing semantic and syntactic inconsistencies. The core structure of AMRO is presented, demonstrating its potential to support knowledge-based applications in aircraft maintenance.

Chapter 6: Results Evaluation and Validation

In this section, the results of the developed ontology-based application are evaluated and validated. This chapter describes the ontology-based application demonstrator and its prototype, detailing the preprocessing of the dataset and resolution of semantic and syntactic inconsistencies in maintenance task cards. Various evaluation metrics such as precision, recall, accuracy, and F-measure were used to assess the performance of the prototype. This chapter emphasises the importance of rigorous validation to ensure the reliability and effectiveness of the developed solutions in real-world scenarios.

Chapter 7: Discussion

The discussion chapter interprets the results in the context of the research objectives and broader field of aircraft maintenance documentation. It reflects the implications of the findings, the effectiveness of the proposed solutions, and the challenges encountered during the research. This chapter provides a critical analysis of the strengths and limitations of the study and offers insights into the practical applications of the developed ontology and prototype. This study also suggests areas for future research and potential improvements to enhance the integration and management of aircraft maintenance records.

Chapter 8: Conclusion and Future Work

The concluding chapter summarises the key findings and contributions of this study. This reiterates the importance of addressing the data heterogeneity and integration challenges in aircraft maintenance documentation. This chapter highlights the novel contributions of the study, including the development of the AMRO and application of the ADOBA methodology. It outlines the practical implications of this research for the aviation industry and suggests directions for future work to further advance the field. The chapter concludes with reflections on the research journey and the potential impact of this study on improving efficiency of managing aircraft maintenance records.

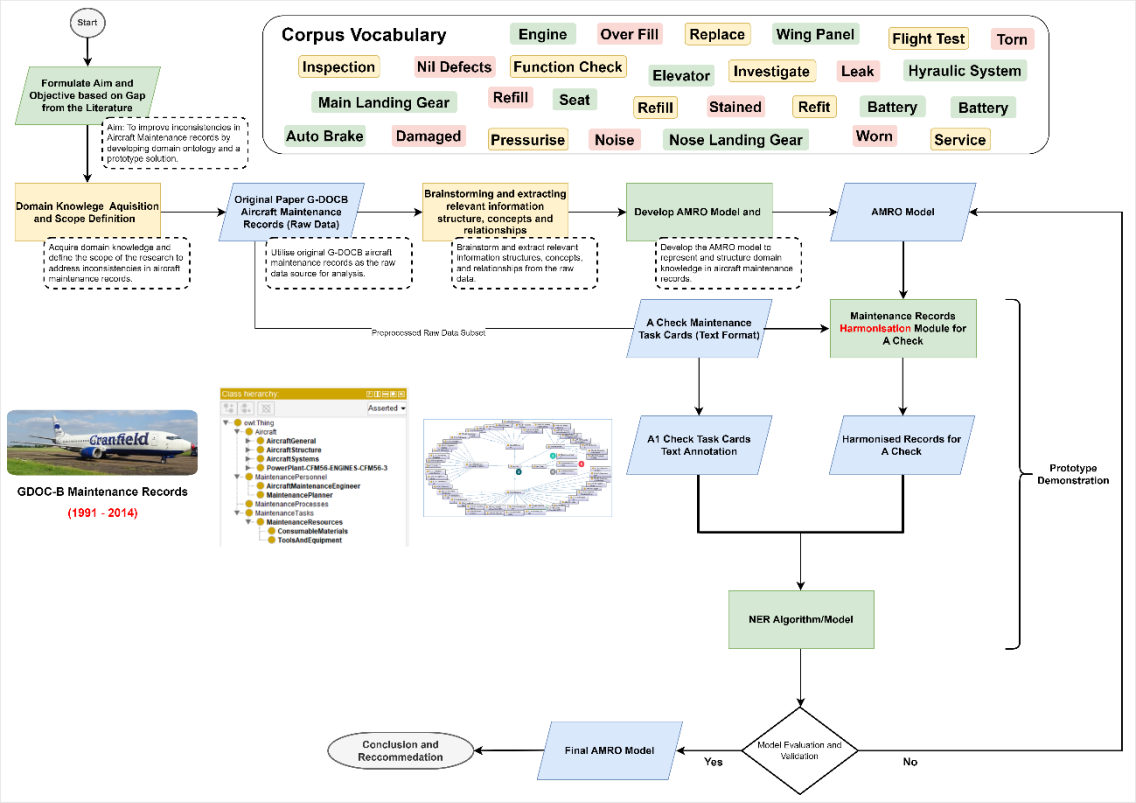


Figure 1-5 Thesis overview

2 Literature Review

2.1 Knowledge Management in Aircraft Operations and Maintenance

The digitisation of aircraft maintenance records marks a pivotal shift in the aviation industry's approach to data management, moving from traditional paper-based systems to advanced digital platforms. This transformation is driven by the need to improve the accessibility, consistency, and integration of maintenance data [2], which is crucial for ensuring the safety and reliability of aircraft operations. Initiatives such as ATA Spec 2500 and IATA Paperless Aircraft Operations[1] have laid the foundation for this shift by providing standardised electronic formats for aircraft transfer records, facilitating the early adoption of digital documents such as spreadsheets, and subsequently, the development of structured databases and maintenance information systems such as AMOS and TRAX. These systems were designed to store and manage maintenance data in uniform formats with the aim of overcoming the limitations of paper records [3].

However, the journey towards fully integrated digital maintenance records has encountered significant challenges. Early systems, while effective in digitising and organising data, often fell short of addressing the semantic richness and complexity inherent in maintenance activities. Data integration and interoperability issues highlighted the limitations of these initial efforts [2,6]. The advent of semantic technologies marked a turning point, offering more sophisticated methods for capturing and utilising maintenance data. Ontology-based approaches, as evidenced in studies [5,7,8], present promising solutions for enhancing data integration by focusing on the semantic heterogeneity of maintenance information. However, these approaches have limitations, primarily due to their narrow scope and the complexity of fully capturing the nuances of maintenance task execution and record keeping.

Despite these advancements, effective integration of maintenance data into cohesive and interoperable systems remains a complex challenge. This challenge is twofold, involving both the development of comprehensive and detailed ontology models that accurately represent the domain knowledge of maintenance processes and the resolution of semantic and syntactic heterogeneity within the data. Both forms of

heterogeneity pose significant barriers to the seamless integration and interoperability of maintenance data.

Almost 2 decades ago, Sripada [6] identified the challenges of information management in the aerospace industry in which he stated that the aerospace industry "poses significant challenges to information management unlike any other industry" and "integrating information from heterogeneous sources continues to be a formidable challenge". Yet, it can be argued that the status quo is no different today and the need for technological solutions is evident.

Knowledge management is concerned with the creation, storing, and transferring knowledge. It is a framework, a management mind-set that includes building on past experiences including libraries, databanks, people, etc and creating new vehicles for exchanging knowledge [9].

The theoretical foundations of knowledge management [10] are deeply rooted into diverse areas including strategic management, organisational culture, behaviour and structure, information economics, and artificial intelligence. There are numerous factors that have resulted in the popularity of knowledge management which include globalization and a drive for sustainable competitive advantage [11]. [12] categorised three elements of a knowledge management system (KMS) as the knowledge source, the knowledge transfer process, and the knowledge recipient. The role of information technology has recently gained the interests of knowledge management researchers, and several methods, frameworks, and techniques have been derived [22], [13]. Information technology is, therefore, effective in the process of capturing, storing, and transferring knowledge.

As the role of information technology is gaining more prominence in the field, information systems are being regarded as a tool to support KM, and rather than a means of processing data [13]. Moreover, the concept of knowledge representation has led to the application of semantic link network (SLN) [14] in KM which offers a means of data modelling based on semantic rules [15]. SLN is therefore increasingly realised as a means to support and enhance management of information. One such example is a formal ontology which is usually created for a specific domain as a form of highly complex knowledge representation artefact used for the development of intelligent knowledge management systems.

Several studies that seek to advance the potential of semantic knowledge representation have been pioneered in aviation research. These can be grouped into 3 critical areas as follows:

i. Maintenance and Support
ii. Operations including Air Traffic Management
iii. Aerospace Engineering and Informatics

While KM systems have been theoretically addressed in the literature, there is limited work on how it can be realised in practice. Based on the scoping search conducted in this research, KM research spans over 3 decades but it's application in the aviation industry was mainly explored over the last decade. The dynamic nature of explicit knowledge which constitute data and information presents a set of challenges such as heterogeneity, integration, storage, and retrieval. These had led to an increase in research on knowledge management (KM) since the early 1990's. In recent years, KM has also been a subject of study in the aviation industry due to digitisation where several solutions and approaches have been developed. Yet, there is an apparent lack of a comprehensive review on how KM supports digital operations and maintenance, as well as the solutions that have thus far been developed, despite the availability of a significant number of studies to help in understanding the application and essential elements that form an effective KM system for aircraft operations and maintenance. This is the key motivation to carry out this review. The core of this review is towards application of KM in the aviation industry, specifically, for aircraft operations and maintenance and sets out to answer the following questions:

1. What are the existing KM approaches and solutions for AOM?
2. How does the KM approaches and solutions differ with regards to development methods?
3. What are the critical elements that form KM FOR AOM?

Building upon these research questions, it is evident that digital transformation in aircraft maintenance necessitates innovative strategies for the effective management of complex and heterogeneous data sources. To address multidisciplinary engineering knowledge, Curran et al. [16] introduced the KNOMAD methodology (Knowledge Nurture for Optimal Multidisciplinary Analysis and Design). KNOMAD systematically facilitates the capturing, normalising, organising, modelling, analysing, and delivering

of multidisciplinary knowledge throughout engineering lifecycles. This structured approach supports the iterative reuse of engineering knowledge and ensures integrated solutions across the design and production domains, thereby enabling significant reductions in production costs and enhanced design effectiveness. Bermell-Garcia et al. [17] emphasise the management of Knowledge-Based Engineering (KBE) applications through the codification and personalisation of knowledge. This framework provides mechanisms for structural design and manufacturing knowledge and manages the lifecycle of knowledge-intensive applications. It addresses challenges such as maintaining quality, accessibility, and traceability, while also supporting a “learning by doing” paradigm. Such frameworks enable more effective deployment of KBE applications as modular software services, providing enhanced capabilities for continuous knowledge updating and customised knowledge deployment, thus significantly improving the efficiency of product development processes.

In practical application scenarios, Viswanath et al. [18] introduced a systematic Multi-Criteria Decision-Making (MCDM) approach for operational maintenance, addressing the timely identification and evaluation of maintenance decisions. This approach employs Boolean Decision Trees for alternative decision identification and the Weighted Sum Method for evaluation, substantially streamlining operational decisions. This method notably reduces decision-making time, ensuring that maintenance personnel can swiftly and accurately identify optimal maintenance actions within tight operational constraints. The identification of the automation potential is another critical dimension of KM, as explored by Verhagen et al. [19]. This study proposes quantifying “information waste” within engineering processes to objectively identify automation opportunities. The approach enhances operational effectiveness by systematically identifying areas where the automation of knowledge-intensive tasks is beneficial. This method emphasises the significance of objectively assessing engineering processes across design, production, and service support phases, thereby allowing for strategic automation implementations.

To further advance knowledge lifecycle management, Li et al. [20] propose a stakeholder-oriented systematic design methodology for Prognostics and Health Management (PHM) systems. This methodology facilitates the comprehensive elicitation of stakeholder expectations, ensures clear traceability, and systematically

integrates multidisciplinary requirements into PHM systems. Such structured stakeholder engagement and requirement traceability significantly enhance the precision and effectiveness of PHM systems, thereby optimising maintenance and operational reliability. Koornneef et al. [4] introduced a web-based decision-support system that integrates real-time dispatch data with maintenance documentation. This innovative system fully automates information retrieval, providing technicians with immediate onsite support for dispatch decisions. These advancements demonstrate substantial improvements in reducing dispatch assessment time and bridging the gap between theoretical knowledge management frameworks and practical maintenance decision-making scenarios.

Despite these significant advancements in knowledge management methodologies and practical solutions, considerable challenges persist, particularly regarding seamless integration and semantic interoperability within heterogeneous aircraft maintenance data. The following section, therefore, critically examines existing knowledge management solutions and methodologies and their potential to resolve semantic and syntactic heterogeneity issues within the aircraft operations and maintenance domain.

2.2 Review Protocol

The review presented in this section involves a series of structured steps as shown in Figure 2-1 to ensure rigor, reliability, and comprehensiveness of the review process (See Appendix B for the full Review Protocol). These steps encompass several key stages, each designed to systematically filter and evaluate the relevant literature on KM in aircraft operations and maintenance (AOM). Furthermore, the “Guideline for doing systematic review” [78] was referred to further understand the processes involved in a SLR such as planning, scoping search, screening, data extraction, quality assessment and synthesis. For the purpose of clarity in drafting the protocol, conducting and final writing of the review, the order of steps outlined by [79] were followed.

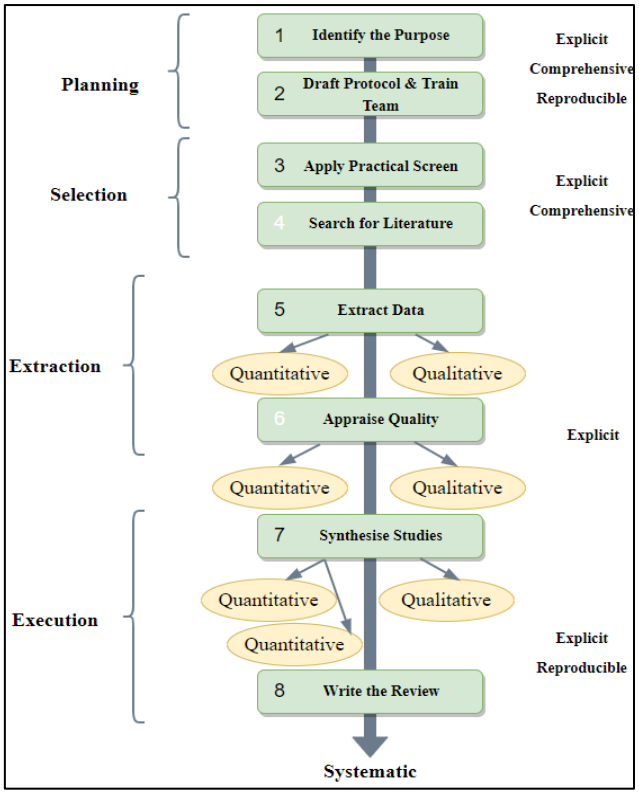


Figure 2-1 Systematic literature review process [70]

The preliminary stage involved conducting an initial scoping search to identify the breadth of the existing literature and refine the search strategy. This step used broad search terms to gather a wide array of studies from identified digital libraries and databases, thereby establishing a foundational dataset of potential primary studies. The aim was to understand the landscape of KM research within the aviation industry and identify the key areas of focus. Figure 2-2 provides a detailed PRISMA flowchart of the systematic review process. The process began with the identification stage, in which 1,018 citations are retrieved from the Scopus database. This initial pool of citations represented all potentially relevant literature identified through the database search.

Following the scoping search, search strings were formulated using the identified keywords AND/OR Boolean operators (AND, OR). These strings were tested and revised to ensure that they captured all the relevant studies. This stage ensured that the search process was both comprehensive and relevant, avoiding unnecessary inclusion of irrelevant studies, while ensuring that no significant studies were omitted.

The screening process involved applying predefined inclusion and exclusion criteria to the retrieved documents. Initially, titles, abstracts, and conclusions were reviewed to filter out studies that did not meet the inclusion criteria. Given the complexity and technical nature of the KM and AOM literature, this stage extended to a brief review of conclusions to capture studies with less detailed abstracts. This dual-approach screening ensured that only studies directly relevant to the research questions were included. The screening stage involved removing duplicate entries, which reduced the number of citations to 996.

To ensure the reliability of the screening process, a "test-retest" approach was employed. The first reviewer (R1) screened all documents, followed by a second screening of randomly selected documents. This method assessed the consistency in the selection process. Additionally, the second reviewer (R2) checked the screening results for R1 to ensure accuracy and reduce potential bias. This dual-check mechanism enhanced the reliability and validity of the review. At this stage, the 996 citations underwent practical screening, an initial assessment to filter out clearly irrelevant studies. Following practical screening, titles and abstracts of the remaining citations were carefully examined. This step further refined the selection process, resulting in 149 articles that were deemed relevant based on their titles and abstracts. The articles were then moved to the full-text screening phase.

During full-text screening, 117 articles were excluded because they did not meet the review’s inclusion criteria upon closer inspection. This left a smaller subset of articles that proceeded to the quality appraisal stage. Each article was evaluated for methodological quality and relevance, ensuring that only the highest quality studies were included in the final synthesis. This stage resulted in the exclusion of 9 additional articles. The inclusion stage culminated in the synthesis of 23 articles. These articles passed all previous screening and appraisal stages, confirming their relevance and quality.

The data extraction process involved using a predefined form to extract relevant information systematically from the full content of the selected studies. This form captured critical data points such as the study context, KM approaches, development methods, and supporting evidence. The extracted data was verified using the R2 value

to ensure consistency and accuracy. This step was crucial for maintaining the integrity and reproducibility of review findings.

Quality appraisal was conducted using a structured questionnaire based on a Likert scale (Agree, Disagree, Undecided). This questionnaire assessed the methodological quality of each study, focusing on aspects such as research aims, context definition, process documentation, and the credibility of the findings. Studies scoring below 7 out 9 were excluded to ensure that only high-quality studies contributed to the synthesis.

The synthesis stage involved aggregation of data from the included studies. Due to the anticipated diversity in study designs and outcomes, a critical realist approach was adopted for synthesis by explanation rather than quantitative meta-analysis. This approach involved organising findings based on key concepts and review questions, presenting them in tabulated formats, taxonomic maps, and clusters. This method allowed for a comprehensive understanding of the various KM approaches and solutions within the AOM.

Each of these steps was designed to methodically filter, evaluate, and synthesise the vast body of literature on KM in AOM, ensuring that the final review is comprehensive, reliable, and valuable for advancing knowledge in the field. The detailed descriptions of each step and the full review protocol is presented in Appendix B.

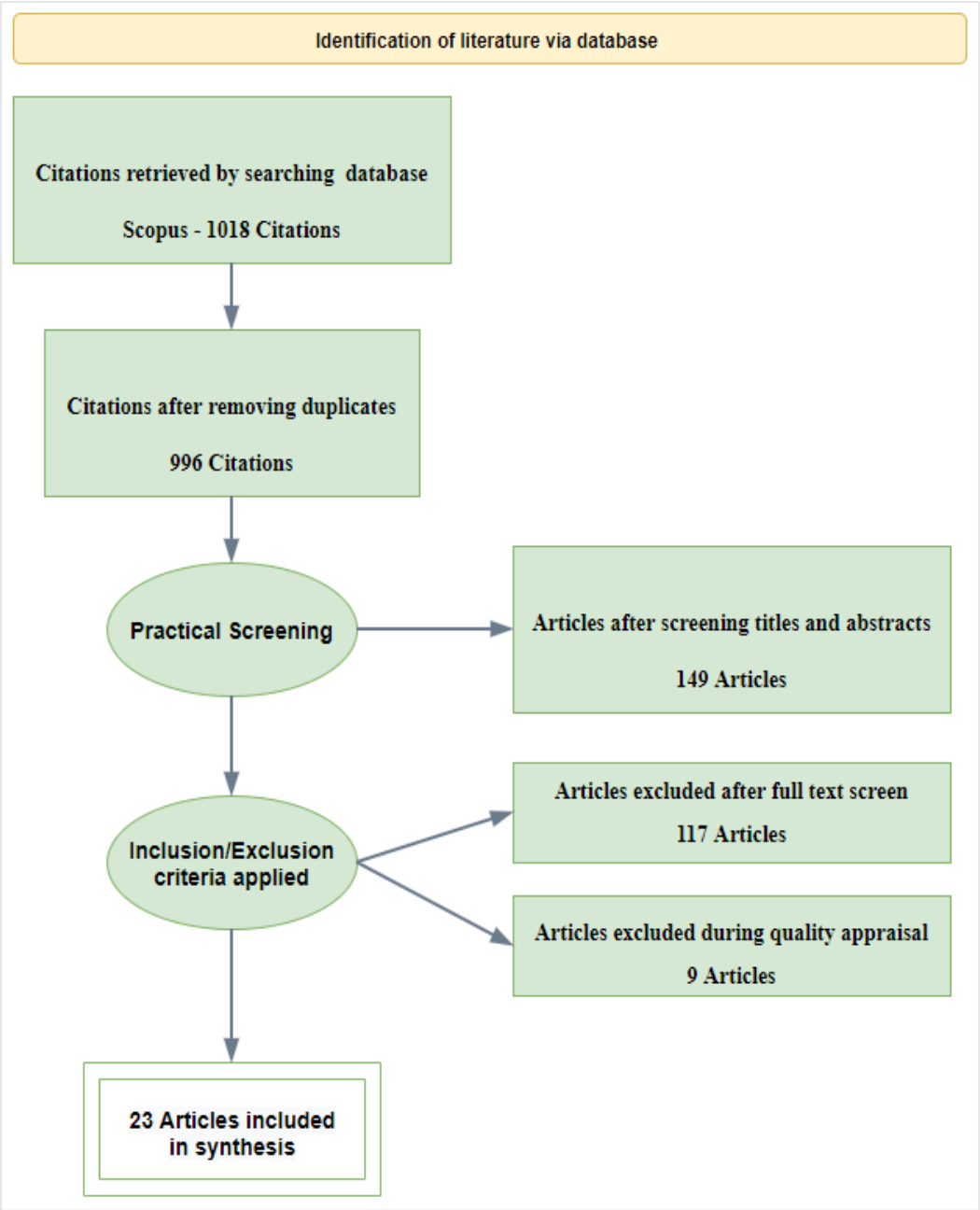


Figure 2-2 PRISMA Flowchart Showing the Identification and Screening Processes for the SLR

2.3 Existing KM approaches and solutions relevant for data integration of aircraft maintenance records

The data on the primary studies in Table 2-1 shows that there have been several KM approaches applied across different areas of aircraft operation and maintenance. The most common approach found was the use of ontology (N = 15) in developing knowledge management systems and integrating heterogeneous operations and

maintenance data. This is followed by knowledge management frameworks for development and evaluation of KM solutions (N = 4), semantic modelling of enterprise knowledge resource (EKR) (N = 1), case-based reasoning (CBR) (N = 1), metadata model of maintenance process knowledge (N = 1) and a method to homogenise knowledge models (N = 1). Additionally, majority of the primary studies (5 or 23%) were in air traffic management (ATM) S1,S2,S6,S12 and S19 in Table 2-1. All the studies related to ATM were observed to have emphasis on semantic heterogeneity of air traffic data and systems. This is followed by aircraft maintenance records comprising (N = 4 or 18%) (Figure 2-3). Although, aircraft route planning, aviation domain ontology, health management of aerospace were found to have lesser number of studies, further research on the usefulness and practicality of the approaches and solutions is needed within all the areas observed.

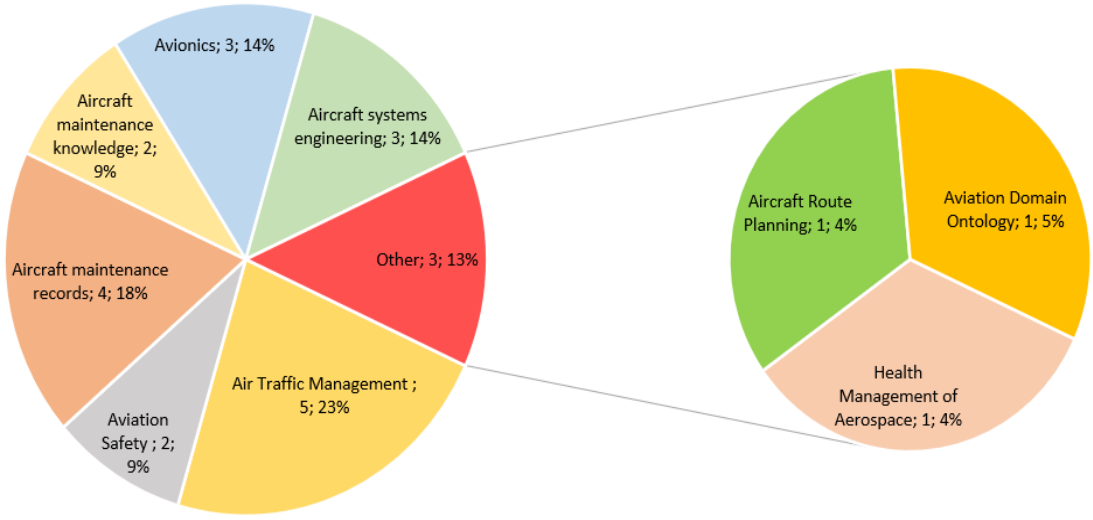


Figure 2-3 Distribution of research context

For example, few of the studies (S4, S5, S6, S7, S9, and S11) included the development and evaluation of a prototypical solution or evaluation of the proposed approached. Although the search for the review papers was between 1990 and 2021, the earliest literature retrieved was from 2008 which suggests that the application of KM approaches in the aviation domain is relatively recent as shown in Figure 2-4 below.

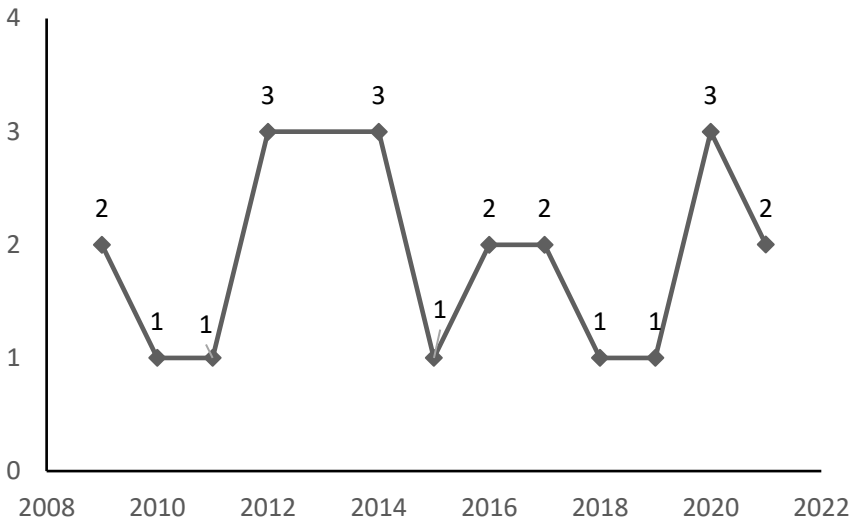


Figure 2-4 Distribution of selected literature

2.3.1 Development methods for existing KM approaches and solutions

This review shows a mixed application of the development methodologies with the distributions shown in Figure 2-5. Notably, only two studies (S1 and S9) explicitly used 'Methontology', a structured approach proposed by Gomez-Perez [24], for building, reusing, and re-engineering ontologies. This methodology ensures rigor and quality and highlights the importance of a standardised process. However, other studies such as S2, S3, and S6, did not specify their methodological approach, indicating a gap in standardisation. This lack of detail can affect the quality and interoperability of resulting ontologies. Conceptual studies often focus on theoretical frameworks without detailing practical methodologies, while empirical studies provide real-world applications but frequently omit methodological specifics. Studies such as S13, S14, and S19, S20 have focused on broader KM frameworks, where traditional ontology development methodologies are not applicable, which contribute to overarching strategies rather than specific ontological solutions. These findings underscore the need for explicit methodologies in ontology development to ensure reproducibility, consistency and high-quality research outputs.

Table 2-1 Selected literature for the review

	KM Approaches or Solutions	Context	Methodology	Year	Contribution	Authors
S1	Ontology of flight safety messages	Air Traffic Management	Methontology	2021	Conceptual	[25]
S2	Ontology based on flight and weather information exchange models	Air Traffic Management		2021	Conceptual	[26]
S3	Aviation domain ontology for semantic integration of information sources	Aviation Domain Ontology		2020	Conceptual	[27]
S4	Ontology for aviation safety data integration	Aviation Safety Data Integration		2019	Empirical	[28]
S5	Ontology for contextualisation of aircraft maintenance documentation	Aircraft maintenance documentation		2017	Empirical	[4]

S6	Semantic techniques to integrate heterogeneous air traffic management data	Air Traffic Management		2016	Empirical	[29]
S7	Ontology for identifying relevant maintenance documents	Aircraft maintenance documentation		2016	Empirical	[30]
S8	Ontology for generic representation of maintenance knowledge	Aircraft maintenance record		2014	Conceptual	[8]
S9	Knowledge-based engineering (KBE) framework	Aerospace product design systems	Methontology	2014	Empirical	[31]
S10	Enterprise Knowledge Resource (EKR) for engineering task representation	Aircraft maintenance documentation		2014	Conceptual	[7]

S11	Metadata model of maintenance process knowledge	Aircraft maintenance process knowledge modeling	2020	Empirical	[32]
S12	Ontologies for Decision-Making in Air Traffic Management	Air Traffic Management	2020	Conceptual	[33]
S13	Avionics Analytics Ontology	Avionics	2018	Conceptual	[34]
S14	Avionics Analytics Ontology	Avionics	2017	Conceptual	[35]
S15	Ontologies for NextGen avionics systems	Avionics	2015	Theoretical	[36]
S16	Case-Based Reasoning (CBR) for safety cases	Safety Cases in Avionics	2012	Theoretical	[37]
S17	Ontology for reuse of knowledge in maintenance, repair and overhaul (MRO) services	MRO services	2012	Conceptual	[38]

S18	Ontology for integration of aerospace health management	Health Management of Aerospace	2012	Conceptual	[39]
S19	Ontology-Based-Control-Room-Framework (ONTOCOR)	Air Traffic Management	2011	Conceptual	[40]
S20	Methodology to interoperate existing heterogeneous design models	Aircraft systems engineering	2010	Conceptual	[41]
21	Ontology for representation of fault knowledge	Aircraft maintenance	2009	Empirical	[42]
22	Ontology for Route Planning	Aircraft Route Planning	2009	Empirical	[43]
23	Ontologies for aviation data management	Aviation Domain Ontology	2016	Theoretical	[44]

Greater emphasis on standardised methodologies such as Methontology could enhance the reliability and integration of KM systems in the aviation sector. Future research should prioritise detailed methodological documentation to improve the rigour and effectiveness of ontology development.

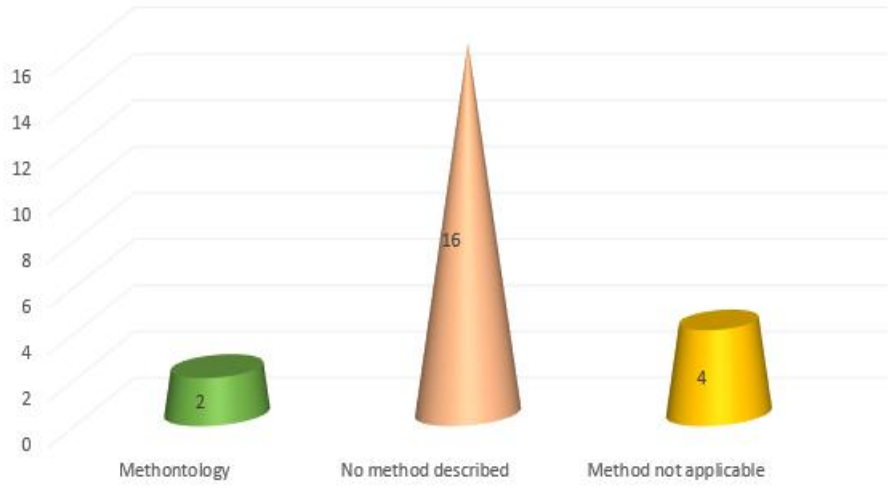


Figure 2-5 Distribution of selected papers

2.3.2 What are the critical elements that form KM FOR AOM?

The critical elements (Figure 2-6) that form an effective KM have been broadly discussed in the literature. It is evident that knowledge representation and reuse is widely considered as a critical element of KM. A shown in Table 2-2, it is found in 15 or 65.21% of the primary studies which mainly emphasise on modelling [94,101,102], semantics [99,101,110,114] and interoperability [8,26,44]. In this regard, [103] stated the structuring of knowledge in the form of an ontology can be used as the ‘semantic backbone’ for knowledge-based applications and the ‘semantic truthfulness’ of information results from a known ontology [106]. The interests of several researchers in this area have therefore resulted in the development of standardised data models [116] to enable data exchange and interoperability.

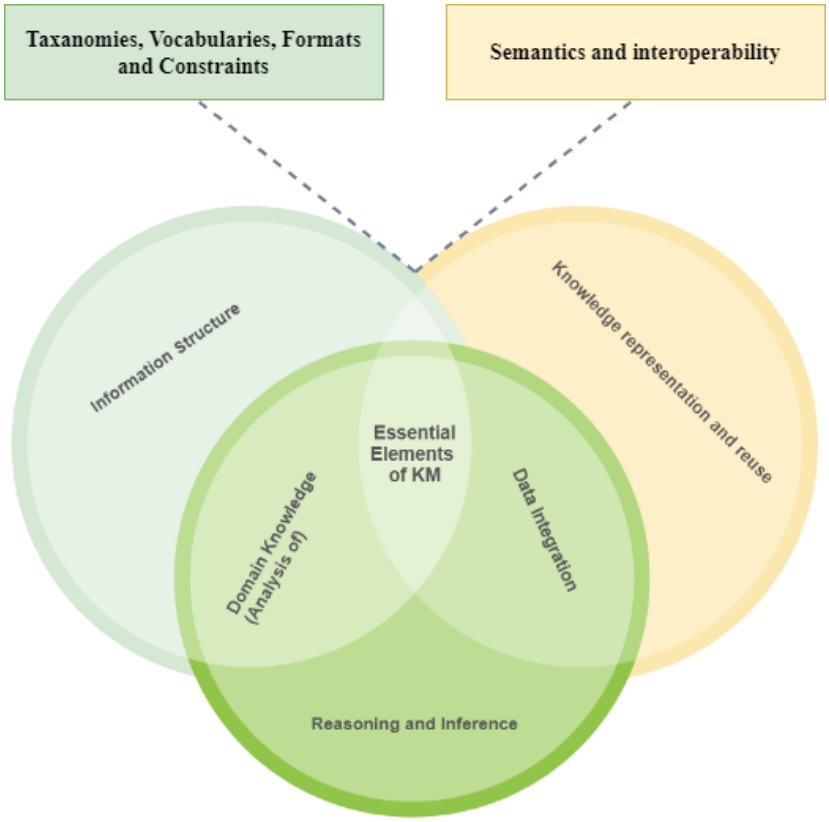


Figure 2-6 Essential elements of KM

Reasoning and inference capabilities are vital to transforming raw data and information into actionable knowledge. In AOM, these capabilities enable KM systems to predict potential failures [45] recommend preventive measures and support maintenance documentation [46,47]. By leveraging this capability, the KM system can analyse patterns, identify trends, and make logical deductions from vast amounts of data generated by aircraft operations and maintenance activities. The findings point to the increasing role of reasoning and inference capabilities of KM systems, particularly in the context of maintenance data, where the application of semantic link networks and formal ontologies are used to enhance the management of information and support decision-making processes in AOM.

Data integration involves the seamless merging of information from heterogeneous sources, ensuring that a unified and consistent view is available to all stakeholders. In the aviation sector, this involves integrating data from

various systems such as flight operations, maintenance records, part inventories, and even external sources such as weather reports and air traffic control information.

Table 2-2 A representation of the literature containing the various essential elements of KM

S/N	Essential Elements of KM	Sources	Frequency	%
1	Information structure: including taxonomies, vocabularies, formats, constraints	S1, S3, S4, S5, S7 S10, S22	7	30.43 %
2	Domain knowledge (Analysis of)	S1, S2, S5, S7, S8, S9, S18, S22	8	34.78 %
3	Knowledge representation and reuse: including modelling, semantics, and interoperability	S1, S2, S4, S6, S8, S9, S10, S11, S13, S17, S18, S20, S21, S22, S23	15	65.21 %
4	Data integration	S1, S2, S3, S4, S5, S6, S8, S23	8	34.78 %
5	Reasoning and inference	S1, S8, S9, S12, S13, S14, S15, S16, S18, S19, S23	11	47.82 %

The challenges in integrating data from heterogeneous sources and the need for technological solutions highlight the critical role of data integration in KM. The seamless merging of information from various systems and sources, as highlighted by [48,49], ensures a unified and consistent view that supports

informed decision-making. This supports the argument that standardised models and frameworks facilitate data integration and interoperability.

Domain knowledge encompasses specialised understanding and insights specific to the aviation industry, including aircraft operations, maintenance practices, safety regulations, and engineering principles. This involves not only the collection and documentation of explicit knowledge, such as procedural manuals and regulatory documents, but also tacit knowledge held by experienced professionals. This review underscores the significance of domain-specific knowledge in AOM, reflecting the complexity and specialised nature of the aviation industry. Effective data integration facilitates a holistic view of operational and maintenance documentation, enabling informed decision-making. It addresses the challenge of semantic heterogeneity, in which different systems may use varying terminologies and formats by applying standards and protocols for data exchange and interoperability.

The information structure is the backbone of any KM system, providing a framework for organising, storing, and retrieving knowledge. In the context of AOM, an effective information structure includes taxonomies, vocabularies, formats, and constraints tailored to the aviation industry's unique requirements. For instance, taxonomies can classify different types of aircraft components, maintenance tasks, and operational guidelines, whereas vocabularies standardise the terminology used across different departments and stakeholders. Formats and constraints ensure that the data captured and stored in the KM system are consistent, reliable, and adhere to industry standards, such as those set by the International Civil Aviation Organization and International Air Transport Association. This directly relates to the identified challenges and the need for technological solutions to efficiently manage and integrate diverse data sources.

2.4 Evaluation of the literature

In this study, we have demonstrated the multifaceted nature of aircraft operations and maintenance knowledge management research. It is shown that a trend began to emerge in applying KM approaches and solutions in the domain within

the last 2 decades. This is driven by the need for practical solutions to the immense data generated for operations and maintenance. The results indicate that data heterogeneity and integration is the main challenge leading to the identified studies. This was anticipated as the search for literature specifically focused on the explicit dimension of knowledge *i.e.* knowledge that can be codified or documented. In doing so, we developed a protocol which incorporates guidelines from information systems and software engineering for carrying out systematic review within the knowledge management domain.

The synthesis of the primary studies confirms the use of ontology as a key approach to representing knowledge and to a broader extent, developing KM solutions. However, less than half or 8 [4,28–32,42,43] of the studies were beyond a conceptual level and included the development of prototype to demonstrate practicality of the solutions. It was therefore, challenging to understand the practicality and strength of the solution. Moreover, only 3 of the studies [25,28,29] focused on industrial context while others were considered academic. This consideration was made based on the environment, sources of information and end-users of the solution. Thus, presenting the need for more empirical studies based on industrial context.

With respect to development methodologies, 2 of the studies [25,31] explicitly described the use of ‘methontology’ while other studies that involved an ontological solution did not describe the use of a methodology. Gomez-Perez [50] noted that ontology development methodologies are crucial in building, reusing, and re-engineering ontologies. Yet, these findings shows that development methodologies are often overlooked albeit important in standardising and ensuring rigour and quality of the process. There are notable exceptions, however, in which we considered a development methodology as ‘not applicable’ such as [34,35,40,41] that focused on KM frameworks and methods.

This review highlights that knowledge representation and reuse emerge as pivotal elements in KM, particularly within the context of aircraft operations and maintenance. This focus underscores the critical role of structuring knowledge in ways that facilitate easy access, interpretation, and application across diverse

operations and maintenance scenarios. The emphasis on modelling, semantics, and interoperability as components of knowledge representation and reuse not only aligns with technological advancements in the aviation industry but also reflects the growing complexity of AOM activities. Such complexity demands robust systems capable of integrating and making sense of the vast amounts of data generated from various sources.

2.4.1 Ontology Development Methods

Ontology is the foundational pillar of knowledge representation, enabling systematic categorisation and formal representation of knowledge within a specific domain. As the demand for semantically rich data representation and integration grows, the significance of robust ontology development methodologies cannot be overstated. Ontology development involves process of developing, refining and reusing ontologies. Given the diversity and intricacies of knowledge domains, the challenge lies not only in the representation of such knowledge, but also in the systematic approach to its construction. Consequently, several ontology development methodologies have been proposed over the years, each with its strengths, limitations, and application scenarios. From Methontology's comprehensive lifecycle approach [24] to the NeOn methodology's emphasis on reuse [51], the landscape of ontology development has varied and evolving. More recently, there has also been proliferation of agile methodologies for ontology development such as Simplified Agile Methodology for Ontology Development [52] and OntoSoft Process [53] for ontology-based application.

However, despite the abundance of methodologies, choosing an appropriate approach remains challenging. It necessitates a clear understanding of both the domain's requirements and the specific nuances of the available methodologies with respective set of principles, processes, and objectives. Table 2-3 illustrates a comparative view of ontology development methodologies based on key criteria in the ontology engineering process. These include:

Table 2-3 Ontology development methodologies

+ Yes; - No; P Partial

	Iterative Process	Specification	Conceptualisation	Formalisation	Implementation	Evaluation	Knowledge Acq.	Maintenance	Documentation
Methontology [24]	+	+	+	+	+	+	+	+	+
NeOn [51]	+	+	+	+	+	+	P	-	P
On-to-Knowledge [54]	+	+	P	+	+	+	-	P	-
SAMOD [52]	+	+	+	P	+	+	+	P	P
AMOD [55]	+	+	+	+	+	+	+	+	+
OntoSoft [53]	+	+	P	P	+	+	P	P	+

Each methodology is assessed based on its support for an iterative process, specification, conceptualisation, formalisation, implementation, evaluation, knowledge acquisition, maintenance, and documentation. It is crucial for ontology development to include an iterative process, as it enables incremental improvements and refinement. This approach is well-supported by methodologies such as SAMOD, AMOD, and OntoSoft, which emphasise continuous development cycles and iterative refinement.

Specification involves defining the ontology requirements, including its scope and competency questions. All the methodologies covered in the table provide robust mechanisms for specification, ensuring that the ontology accurately addresses the needs of its intended domain. Conceptualisation refers to the organisation of

domain knowledge [50]. Methodologies, such as Methontology, SAMOD, AMOD, and OntoSoft, strongly support this phase, facilitating the creation of a comprehensive and coherent conceptual model.

Formalisation is the transformation of a conceptual model into a formal ontology using languages such as OWL. This criterion is consistently well supported across all methodologies, ensuring that ontologies are precise, machine-readable, and interoperable. Implementation involves practical development and integration of ontologies within applications or systems. All methodologies in the table emphasise this phase, reflecting the importance of creating functional and usable ontological models.

The evaluation process is intended to verify that the ontology satisfies its specified requirements and effectively answers its competency questions. Methontology, SAMOD, AMOD, and OntoSoft involve thorough testing and validation processes in this regard. Knowledge acquisition entails gathering information from domain experts and other sources to support in the ontology development process. Methontology and AMOD support this activity, while methodologies such as NeOn and OntoSoft offer only partial support, indicating the need for clearer processes. Maintenance involves updating the ontology to reflect changes in the domain it represents, ensuring its relevance and accuracy over time. AMOD and OntoSoft provide robust support for this aspect, while other methodologies offer varying levels of support. Documentation is crucial to making the ontology understandable and usable by others. AMOD and OntoSoft, for example, have comprehensive documentation practices that ensure the ontology development process and structure are documented. Thus, highlighting the need for a systematic and iterative approach to ontology development, to achieve high-quality, reliable, and reusable ontologies.

2.5 Semantic Data Integration

In data management, semantic data integration plays a vital role with the aim of harmonising and combining heterogeneous data from various sources while the meaning and context are highly considered. It involves the use of semantic technologies (Resource Description Framework, Web Ontology Language,

SPARQL Protocol and RDF Query Language, RDF Schema) as described in the semantic web stack in Figure 2-7 [56], methods and standard facilitate the integration and interoperability of the disparate data, enabling a unified understanding and utilisation of the information [57–59]. This technique ensures the effective exchange, queried, and reasoned over, promoting better decision-making and insights. The semantic data integration process has been described by [60], to have the following:

i. Data sources Identification and Extraction
ii. Ontology Development
iii. Schema Mapping and Transformation
iv. Data Integration and Fusion
v. Querying and Analysis

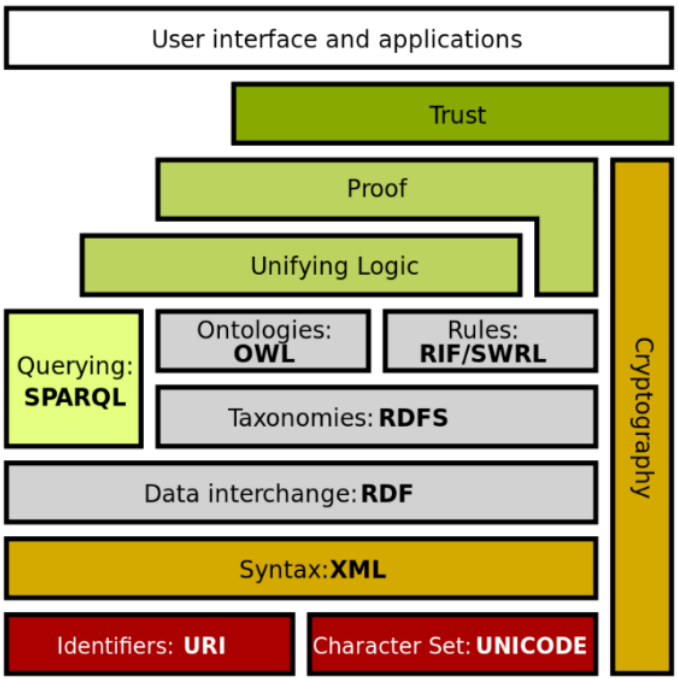


Figure 2-7 Semantic web stack [61]

The Semantic Web Stack shown in Figure 2-7 above is a hierarchical structure of technologies designed to provide a universal framework for data sharing and reuse across various systems, applications. The stack includes Uniform Resource Identifiers (URIs) and Unicode, which serve as unique and global identifiers for resources and a universal character encoding standard,

respectively. These fundamental components guarantee consistency and interoperability across different systems. XML offers a versatile syntax for data representation, while the Resource Description Framework (RDF) facilitates data interchange through structured, graph-based representations. Furthermore, RDF Schema (RDFS) and the Web Ontology Language (OWL) add additional layers of meaning by allowing the definition of vocabularies, taxonomies, and intricate relationships within data. Thus, these layers contribute to the formation of a rich, interconnected data web, enabling sophisticated querying and manipulation of data using SPARQL.

The upper layers of the Semantic Web Stack are dedicated to enhancing data processing and establishing trust and dependability. Rule Interchange Format (RIF) and Semantic Web Rule Language (SWRL) empower the formulation and exchange of rules for inferring novel information, while the Unifying Logic tier guarantees consistent reasoning across diverse data sources. The Proof level validates the correctness of data and deductions, guaranteeing precise and dependable outcomes. The Trust layer, underpinned by cryptographic technologies, ensures data integrity, confidentiality, and authenticity, fostering confidence in the data and its source. This comprehensive, layered strategy ensures that the Semantic Web Stack provides a robust and scalable architecture for data-driven applications, facilitating seamless data integration, interoperability, and sophisticated reasoning capabilities.

2.5.1 Ontologies for Semantic Data Integration

The concept of ontology has been explored by many researchers from different perspectives depending on the area of study. This includes philosophy, computer science, artificial intelligence, health informatics, information science, and knowledge management and representation [62]. The earliest roots of ontology can be traced back to philosophy, which describes concepts of reality and existence, including the rules in which they are governed [63]. However, it has since been adopted in the computer and information sciences as an important concept for representing the structure of data and enabling reasoning.

The earliest definition of modern ontology applications stems from artificial intelligence (AI) research in the 1970s [50,63]. This has led to the development of the theoretical foundation of “formal ontology,” which, from a philosophical perspective, is considered within the framework of formal logic. This was evident in the work of philosopher Edmund Husserl on logical investigations [40], in which a clear difference between formal logic and ontology was outlined. The latter is concerned with formal ontological structures and the interconnectedness of entities, objects, properties, and their relations, whereas the former is concerned with formal logical structures and the interconnectedness of truth. In this context, ‘formal’ refers to expressions of reality or that exist in a machine-readable format rather than a natural language[63,64].

Advances in computer science and artificial intelligence led to the adoption of this concept in knowledge-based research. Although several definitions have been presented in the literature (see Table 2-4), the most notable definition from the literature was given by Tom Gruber [65,66], who described ontology as an ‘explicit specification of a conceptualization.’ Ontologies were used to represent the conceptualizations of different domains as a foundation for a knowledge base. These include the vocabulary of terms and how key concepts are interrelated with the key emphasis of semantics, which together culminate in and represent the structure of the domain in the form of a formal ontology. A critical underpinning of such an ontology is that it constitutes the requirements of a specific domain. Hence, demonstrating the important difference between formal logic and formal ontology, as drawn by Smith [67] .

Table 2-4 Definitions of ontology

Author(s)	Definition of ontology
Gruber 1993 [66]	‘Explicit specification of a conceptualization’
Borst 1999 [68]	‘Formal specification of a part of a conceptualization’

Guarino and Giaretta 1997 [69]	‘An explicit , partial account of a conceptualization’
Van Heijst et al. 1996 [70]	‘Explicit knowledge level specification of a conceptualization, which may be affected by the particular domain and task it is intended for.’
Alberts 1993 [71]	a body of knowledge concerning a particular task or domain describes a taxonomy of concepts for that task or domain that define the semantic interpretation of the knowledge

Ontologies are the bedrock and play fundamental role in the semantic data integration realm by providing a knowledge structure, formal representation within a specific domain [72,73]. They define properties, concepts, relationships and axioms, which establishes a common understanding and vocabulary for the integrated data [74–77]. By creating a standardised framework for data semantic annotation and organisation, ontologies enable the harmonisation of heterogenous data sources and support alignment of shared structure data elements [77–80]. In solving heterogeneity and facilitating the seamless integration of data from various sources the alignment plays a vital role [81–84]. Data elements are enriched through the ontologies with additional semantic information, allowing for precise interpretation and enabling more sophisticated querying and reasoning [85–89]. Consequently, ontologies serve as a crucial tool in achieving semantic interoperability and unlocking the potential of integrated data for an effective decision-making and analysis [48,49,90].

2.5.2 Semantic Modelling

This is the forefront of knowledge representation and organisation, offering a structured and precise approach to encoding meaning and relationships within a particular domain [91–96]. It allows the representation of complex knowledge in a machine-interpretable format, which enhances the understanding and facilitating effective data integration by utilising semantic technologies such as

RDF and OWL. It does not only capture the concepts definitions and their interconnections but also connects to their logics and reasoning rules governing them, enabling automated inferences and advance querying [97]. The standardisation of data representation, fostering interoperability and enabling collaborative efforts across diverse systems and applications is promoted by an established ontologies, taxonomies, and vocabularies through semantic modelling [98,99]. Specifically, semantic modelling is the foundation of Semantic Web, fostering meaningful exchange of data, knowledge sharing and innovative applications in different domains [100–103].

2.5.3 Ontology Based Data Integration

Ontology based data integration is the method that uses ontologies to facilitate the integration of distributed and heterogeneous data sources. It provides a common and formalised concepts representation, relationships, and constraints within a specific domain, which enable a standardised procedure to structure and think about data as shown in Figure 2-8 [57,58,104–106]. This approach gained significant traction in the field of data integration because of its ability to harmonise various data sources and enhance interoperability. Ontology-based data integration have the following key components:

i. Definition and purpose of ontology-based data integration: This involves the utilisation of ontologies to provide a semantically and standardised rich data representation and its relationships within a particular domain. It has a primary purpose of facilitating the integration of heterogeneous data sources to enable seamless information retrieval, querying and analysis across different datasets [59,73,106].
ii. Ontologies as the backbone: Ontologies are a fundamental component of ontology-based data integration. They provide a formal representation of domain knowledge, properties, defining and relationships within a particular domain [49,60,107,108]. The use of ontologies assists to achieve a standardised and shared understanding of the domain, supporting integration efforts.

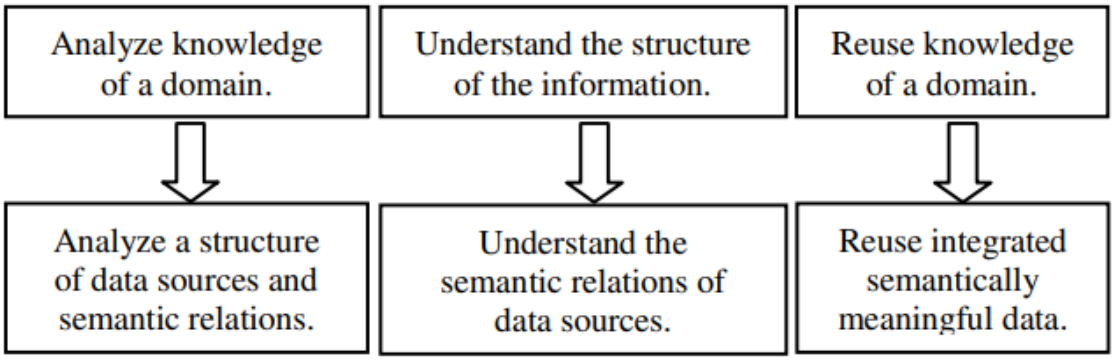


Figure 2-8 Transformations of ontologies features to data integration systems requirements [109]

iii. Semantic mapping and mediation: Ontology-based data integration involves creating semantic mapping between terms and concepts in different sources of data and the concepts defined in the ontology [104,110]. These mappings enable mediation between heterogeneous data sources allowing queries and the translated operations and executed in a manner that respects the domain ontologies.

iv. Query rewriting and semantic interoperability: User queries or applications are rewritten based on the domain ontology to ensure semantic interoperability [80,111,112]. This procedure ensures that queries are translated into a standardised representation that is executed across different sources of data while considering the domain semantics.

v. Integration process and architecture: This process involves the aligning and integrating data from various sources based on the ontology's conceptual model. This may occur at the instance level or schema, with the goal of providing a unified data view [76,92,113]. The architecture usually typically includes ontology components mapping, query rewriting and mediation to achieve a successful integration. This plays crucial role in addressing the challenges various data and heterogeneous data sources posed, allowing an effective and utilisation of data efficiently in different application domains.

2.5.4 Data Heterogeneity Conflicts

Data heterogeneity [114–117] refers to the diverse inconsistencies and disparities that arise across various data sources stemming from differences in how data are structured, represented, and interpreted. Specifically, data sources can exhibit heterogeneity in terms of their syntax (how data are formatted and presented), schema (the structure or organization of data), and semantics (meaning or context ascribed to data) as shown in Figure 2-9. These variations introduce significant challenges particularly for data interoperation, as integrating or harmonising data from disparate sources becomes a complex with potential mismatches and misunderstandings [64]. Furthermore, the literature on data heterogeneity is vast and multifaceted, offering several classifications for the types of heterogeneity encountered. Some prominent classifications include [63,118], and so on, each shedding light on specific dimensions or nuances of data discrepancies and conflicts. The classification posited by Euzenat et. al. [119] was considered in this research.

Syntactic Heterogeneity

Syntactic Heterogeneity revolves around machine-readable aspects [59,120] in data formats, structures, representation syntax, and even the languages in which the data or knowledge are expressed. Such heterogeneity occurs when two systems or data sources adhere to different syntactic rules and conventions. Variations may be evident in aspects such as date formats, data delimiter styles, or when two ontologies differ in their representation, richness and complexity [59], as in the case of RDF's basic resource descriptions versus OWL's detailed and expressive ontology constructs. The direct comparison or integration of these systems is challenging because of these disparities. To address syntactic heterogeneity, parsers, converters, and translators are often deployed to bring varied representations into a unified format, ensuring consistency and coherence. While syntactic differences may be straightforward to identify, addressing them remains pivotal for seamless data integration. The concept of syntactic

heterogeneity between two data elements can be define as shown in Equation 2-1:

Let D represent data:

$$SyntacticHeterogeneity(x,y) \leftrightarrow (D(x) \wedge D(y) \wedge \neg SameRepresentationFormat(x,y)) \quad \textbf{Equation 2-1}$$

Where x and y are subsets of D

Terminological Heterogeneity

Terminological conflicts emerge when different terms or identifiers are used to denote the same concept or entity, respectively. For instance, in the domain of aviation services, one database might use the abbreviation "Lav" to describe the onboard toilet facility, while another might utilise the full term "Lavatory." These terminological variances can obstruct seamless data integration, as automated systems may not immediately discern that "Lav" and "Lavatory" in different databases refer to the same onboard facility. Among the various approaches to addressing these discrepancies, ontologies have been identified as a particularly effective strategy [121]. By developing a structured and semantically rich representation of a domain, ontologies offer a unifying model in which synonyms, alternative labels, and concept hierarchies are clearly articulated.

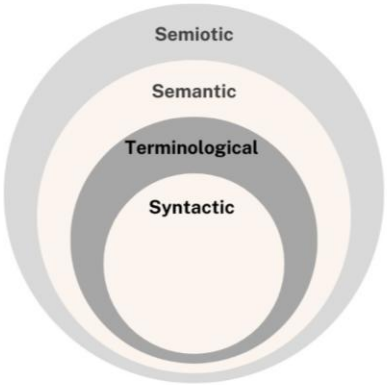


Figure 2-9 Types of data heterogeneity

Semantic heterogeneity

Semantic heterogeneity [59,110,122] refers to inconsistencies in how identical terms or concepts have different meanings or interpretations across various systems, data sources, or contexts. It arises from diverse representations, definitions, structures, or granularities in data sources, often posing challenges to data integration and interoperability. The two systems may present the same terminology but have distinct meanings or interpretations. For example, the term 'landing gear inspection' on one maintenance task card might detail procedures to check the hydraulic systems, whereas another card might emphasise tire and brake examinations. These discrepancies can be deceptive as they may not be immediately discernible, leading to significant variances in how maintenance tasks are executed. Resolving such heterogeneity demands an in-depth understanding of the context of each source, often necessitating tools such as semantic mapping and ontological methods to bridge these interpretational gaps.

Within the sphere of semantic heterogeneity, mismatches often emerge from distinctions in ontology content. These discrepancies can be broadly categorised into two groups [123,124].

I. Metaphysical differences
II. Epistemic differences

Metaphysical differences focus on how ontology represents the real world through entities, properties, and relationships [123]. Specifically, ontologies can differ in terms of coverage, detailing which parts of the world or domain they focus on, granularity, indicating the depth of their descriptions, and perspective, highlighting unique viewpoints on a domain that may be temporal or spatial. For instance, one set of maintenance task cards might focus specifically on the aircraft's propulsion system (coverage) with an in-depth breakdown of each component (granularity), whereas another set might provide an encompassing overview of the entire aircraft (perspective). The presence of such differences explains why multiple ontologies in a domain may have varied foundational structures or contradictory claims regarding the same entities.

Conversely, Epistemic differences pertain to the specific assertions made about these entities and go beyond how the information is structured (the ontology's metaphysical aspect) and dive into the specifics of what is claimed or known about the entities in question [123]. For example, two different ontologies, both designed for aircraft maintenance, might represent the task of "Wing Inspection." One ontology may specify that this inspection predominantly involves checking for surface cracks, whereas the other might highlight the importance of detecting corrosion near the wingtips. Both ontologies recognise the task, but they offer different nuances or focus areas regarding the specifics of the task. These variations represent epistemic differences between the two ontologies.

However, for the purpose of this work, the primary focus is on Metaphysical differences, as they pertain to the structural aspect of information. Addressing Epistemic differences, which go beyond just structure and delve into specific claims or knowledge, will be considered outside the scope of our current discussion.

Semiotic Heterogeneity

Semiotic heterogeneity [117] refers to disparities observed in the human interpretation of entities, even when these entities possess identical semantic interpretations. This form of heterogeneity arises primarily because of the contextual nature of human understanding, which influences how entities are processed based on their intended use or application. Context-specific interpretation means that two entities with the same semantic content might be interpreted differently depending on the situation or the particular use case.

2.6 AI and Semantic Searching

AI and domain ontologies have significant transformative potential within AOM. The growing complexity of operational and maintenance data highlights the suitability of AI-driven systems, particularly Large Language Models (LLMs) such as Generative Pre-trained Transformer (GPT) and Bidirectional Encoder Representations from Transformers (BERT), enhanced through the integration of

domain ontologies. These ontologies are useful in facilitating improved semantic search and data integration. They provide essential semantic consistency and enhance knowledge retrieval capabilities within increasingly complex operational contexts. Krzanowski and Polak [125] noted that AI systems inherently possess an ontological gap in understanding the real world. This gap arises from the different ways in which AI and humans conceptualise reality. For AOM, this gap is significant because AI systems must process complex, domain specific data, such as Task cards and maintenance manuals. AI models such as GPT and BERT can analyse vast amounts of unstructured data but often struggle to fully understand the relationships between concepts [126] unless those relationships are formally defined. Ontologies that formalise the structure of domain knowledge by defining key concepts and their relationships play a crucial role in filling this gap. By incorporating a domain specific ontology into AI systems in AOM, AI models can explain the interactions between concepts and relationships in a manner that approximates human understanding. Thus, providing enhanced semantic reasoning and inference capabilities.

2.6.1 Large Language Models for Resolving Syntactic and Semantic Heterogeneity

LLMs are advanced artificial intelligence systems that leverages deep neural network architectures known as Transformers, which utilise self-attention mechanisms to handle context-rich textual data efficiently. These models are pre-trained on extensive text corpora, enabling them to understand linguistic nuances, semantics and syntax [127]. There are three primary architectural classification of Transformer-based LLMs which include encoder, decoder and encoder-decoder models.

Encoder models comprise solely the Transformer's encoder stack, which processes input text bidirectionally, considering the full context, to produce rich language representations. These models excel in language understanding tasks, such as classification and extraction, by learning deep contextual embeddings, although they do not inherently generate long-form text. Decoder models employ

only the decoder stack, functioning auto-regressively to predict the subsequent word in a sequence. These models are designed for text generation and sequence prediction, capable of producing coherent outputs, but they primarily consider context in a unidirectional manner, from past to future. Encoder-decoder models, integrate both components, an encoder initially processes the input sequence into a context representation, and a decoder subsequently generates an output sequence, often for tasks such as translation, summarisation, or question answering, based on that representation [128–130]. Table 2-5 shows strengths and limitations of LLMs classification.

Table 2-5 Strengths and Limitations of LLMs classification

Type	Example	Strengths	Technical Limitations
Encoder	BERT	Contextual embedding, classification accuracy	Limited generation, Fixed-length inputs
Decoder	GPT-4	Fluent text generation, Broad knowledge	Resource-intensive, susceptible to hallucinations
Encoder-Decoder	BART	Versatility, structured output generation	Requires task-specific fine-tuning

Syntactic Heterogeneity in AOM Data and LLMs

In data integration, LLMs can help bridge inconsistencies using their flexible natural language understanding. Since LLMs are pretrained on diverse text, they can ingest information in various formats including free-text fields, semi-structured logs, or even structured code, and convert or normalise it into a common form. Recent research has shown that an LLM can be prompted to parse heterogeneous semi-structured documents and output them as unified, queryable tables without task-specific customisation [131]. Thus, an LLM can act as a universal parser, interpreting different input schemas or notations and generating a standardised representation. This significantly reduces the problem

of rigid schema matching. additionally, encoder-based LLMs provide high quality embeddings that encode syntactic context, allowing even differently formatted sentences conveying the same content to end up close in vector space, which aids downstream alignment of data fields. Decoder-based models can be directed via prompts or fine-tuning to produce outputs in a required format or syntax, effectively translating diverse inputs into a consistent schema. Together, these capabilities mean LLMs can mitigate syntactic heterogeneity by abstracting away format differences and focusing on content [132].

Semantic Heterogeneity in AOM Data and LLMs

Semantic heterogeneity arises when different sources use varying terminology or conceptual frameworks for the same or overlapping entities [133]. LLMs are highly effective at bridging these meaning gaps. Because they learn context-dependent language patterns, LLMs can recognise that two differently worded expressions refer to the same concept. For instance, an LLM can infer that “engine swap” and “powerplant replacement” are semantically similar even if the phrasing differs, due to its broad training knowledge. In practice, LLMs have been applied to ontology matching, aligning concepts across different ontologies or databases; a direct approach to handling semantic discrepancies. Babaei et al. demonstrate that an LLM-based framework can match and even surpass traditional ontology alignment systems in identifying equivalent concepts across heterogeneous schemas [134]. This indicates that LLMs capture synonymy and contextual meaning, helping to map data elements that are semantically related but lexically dissimilar. Additionally, LLMs can perform zero-shot reasoning on unseen terminology by drawing parallels to known language usage [135]. In AOM, where one data source might label a part “APU” and another uses “Auxiliary Power Unit,” an LLM can use context to understand both as the same item. By producing a consolidated representation or translating terms, LLMs effectively reduce semantic ambiguity and inconsistency.

Ontologies Supporting LLMs to enhance semantic reasoning and Inference

While LLMs are powerful, they do not inherently possess structured domain knowledge. This is where ontologies, formal representations of concepts and relationships complement LLMs. Ontologies provide a semantic backbone that can guide or constrain LLM interpretations in specialised domains like AOM. Researchers note that AI language models often face an “ontological gap” in understanding real-world relationships [125]. For example, GPT or BERT can analyse vast maintenance manuals but might not reliably infer how specific components relate functionally or hierarchically, unless those relations are explicitly defined . By incorporating a domain ontology, this gap can be filled. The ontology formalises key AOM concepts including aircraft systems, part-whole relationships, maintenance actions, etc. and their interrelations. LLMs can then be augmented with this knowledge. For instance, via prompt engineering, fine-tuning, or coupling with a knowledge graph to ensure outputs adhere to known relationships and terminology [126,136]. This leads to enhanced semantic consistency. The LLM is less likely to use inconsistent names or violate domain constraints if it knows the ontology. Moreover, ontologies enable inference that complements LLMs’ statistical reasoning [137]. An LLM can propose an answer or classification based on context, and an ontology can then be used to check or refine that answer against formal rules e.g., ensuring a component mentioned exists on that aircraft model, or inferring missing details through logical relationships.

2.6.2 OCR and AI-Powered Preprocessing in Aircraft Maintenance Records

Aircraft maintenance operations have historically produced large volumes of paper documents, work orders, logbook entries, task cards, component change forms and inspection reports, etc [138]. Converting these paper or scanned records into digital text is a crucial step in modernising maintenance knowledge management. Digital text enables searchability, data integration, and analytics that are impossible with siloed paper files. Industry initiatives like ATA Spec 2500 and IATA’s Paperless Aircraft Operations have pushed for standard electronic formats for maintenance recordsfile. However, a vast amount of legacy data still

exists on paper or in non-searchable scans, and even new data, which often starts as unstructured images. OCR is the key technology for extracting machine-readable text from document images, and recent advances in AI have greatly improved OCR accuracy and capability [139]. Artificial Intelligence-Based OCR can enhance efficiency compared to error-prone manual processes [140] and can be useful for maintenance documents, preserving not just the text but also the structure and meaning needed for downstream analysis.

Digitising paper-based aircraft maintenance documentation is vital for knowledge management systems. These documents ranging from printed manuals and task cards to handwritten logs must be converted to structured text using OCR. However, challenges such as degraded scans, handwriting variability, and document layout complexity necessitate AI-powered preprocessing steps to ensure high OCR accuracy [141].

Table 2-6 Preprocessing Techniques for OCR in AOM

Technique	Purpose	Application in AOM
Noise Removal	Filters speckles, ink artifacts, scanner dust	Enhance old maintenance records
Binarisation	Covers black and white printouts	Improves OCR on faded printouts
Skew Correction	Aligns rotated or tilted text	Correct images of maintenance records
Layout Analysis	Detects columns, Forms and paragraphs	Separate tables from text in manuals and job cards
Text Region Detection	Locates and isolates text blocks	Finds handwritten entries or annotations in mixed-content scans

Segmentation	Splits text into lines or characters	Prepares clean input for OCR, especially for handwritten or form-based documents

Preprocessing improves OCR effectiveness by cleaning and structuring document images. This includes noise reduction, binarisation, skew correction, layout analysis and segmentation as shown in Table 2-6. These techniques, often AI-assisted e.g. CNN-based text detection or adaptive thresholding, ensure that OCR engines receive clean, well-aligned and text-focused input. Once recognised, post-processing with domain-specific language models or ontologies helps correct OCR errors and standardise terminology.

2.7 Reflection from the literature

This review systematically synthesises diverse KM approaches and solutions specifically for the AOM industry. By integrating findings from various studies, this study offers a comprehensive overview that has been previously spread across individual research efforts. This synthesis not only makes it easier for practitioners and researchers to understand the landscape of KM in AOM but also highlights the evolution of KM practices in response to technological advancements and changing industry needs. A gap between the theoretical advancements in KM and their practical applications within the AOM sector was identified. This is highlighted by the lack of studies demonstrating the real-world applicability and effectiveness of KM approaches for AOM.

The literature reveals a significant research gap in the lack of a standardised methodology for integrating heterogeneous maintenance data, which leads to challenges in data interoperability and consistency across systems. Ontology-based approaches have been identified as promising solutions to overcome semantic heterogeneity. Key findings emphasise the importance of data integration, knowledge representation, and reasoning capabilities in KM systems

for AOM. Existing ontologies, although valuable, often lack the expressiveness and coverage required to effectively integrate maintenance records from diverse sources. This aligns with the requirements of AMRO, which aims to improve the semantic interoperability of maintenance data, as outlined in the ontology requirement specification document (see Table 5-3). The functional and non-functional requirements, including the use of OWL are also supported by the findings in the literature, emphasising the need for standardised knowledge representation and data formats for effective integration.

As Lehmann et al. [142] highlighted in their study on the integration and assessment of ontology coverage in the aviation industry, a lack of coverage can impede the use of semantic capabilities. By identifying ontology as a key approach to knowledge representation [4,8,25,26,30,42,44] and the need for standardised development methodologies focusing on developing practical ontology-based application [52,55,143], this study lays a path forward for addressing the challenges of data heterogeneity and integration in aircraft operations and maintenance.

The review addresses the first research objective by reviewing the state-of-the-art in aircraft maintenance KM systems. Through this review, key gaps in the existing literature were identified, specifically for data integration, semantic interoperability and application of ontologies in AOM. This provides a foundation for the research gap identified, which is critical for the next steps in developing a domain ontology. The second research objective, which focuses on establishing the information structure, concepts, and relationships from the document analysis of Boeing 737 maintenance records, is addressed by highlighting the importance of structured data and ontological modelling in aviation maintenance. Thus, setting the groundwork for this objective by demonstrating the need for organised information frameworks to represent the complexity of maintenance records. Finally, the third research objective is understood through the review of different ontology development methodologies discussed in the literature. This objective is addressed by presenting a theoretical basis for developing AMRO and how such ontological frameworks have been applied within the AOM sector. The

literature also informs the choice of an agile methodology, which will be explored further in the development of AMRO in the subsequent sections.

3 Methodology

This section presents the methodological approach used in this study. Figure 3-1 illustrates the phases of the methodology used to achieve the overall aim and objectives of this research.

3.1 Research Methodology

The initial investigation phase laid the groundwork for this research by exploring the current landscape of KM in aircraft operations and maintenance. This phase involves an initial review to understand existing practices and identify the challenges associated with integrating heterogeneous aircraft maintenance records. A systematic review protocol was developed that adhered to guidelines for conducting comprehensive literature reviews in information systems. This protocol ensures a structured approach for gathering and synthesising relevant studies. The systematic review focused on KM for aircraft operations and maintenance, identifying the key elements that contribute to effective KM solutions. This review highlighted significant gaps and provided insights into the essential components of an effective KM system. The findings from this phase were synthesised to outline the critical elements required for a robust KM solution by setting the stage for the subsequent development of the ontology model.

In the second phase, the development of an ontology model and a prototypical demonstration was carried out. This began with a thorough review of existing ontology development methodologies to identify their strengths and limitations. Based on this review, a new methodology, Agile Development for Ontology-Based Applications (ADOBA), is proposed. ADOBA integrates the iterative and user-centric principles of agile development with the structured phases of traditional ontology development. The next step involved a detailed document analysis of aircraft maintenance records and extracting key concepts and relationships to form the basis of the AMRO model. This analysis is critical for effectively categorising maintenance data. Subsequently, an ontology-based application was developed to demonstrate the practical application of the ontology. The prototype application utilised ontology to correct inconsistencies in

maintenance records, thereby improving data categorisation and retrieval processes.

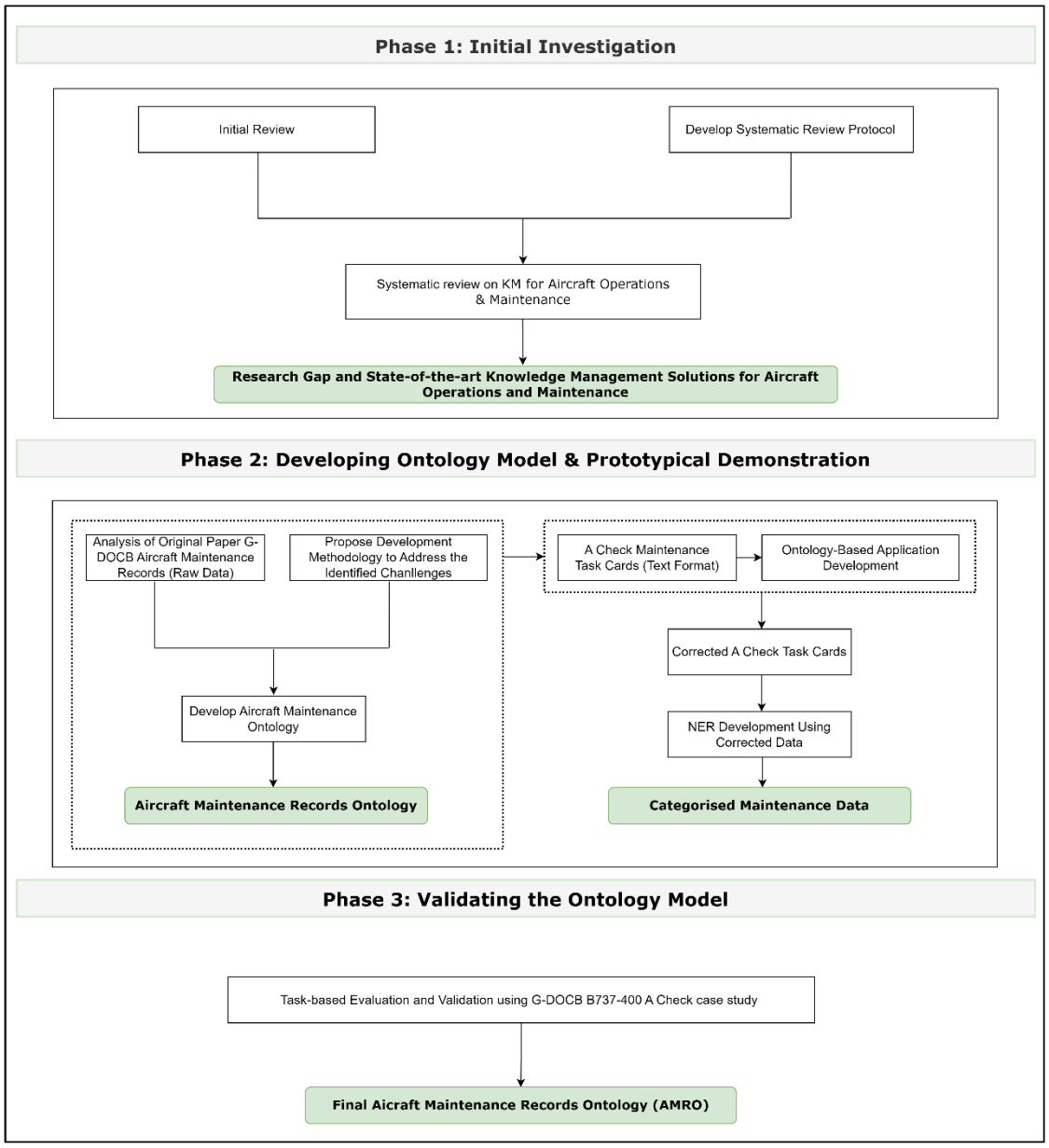


Figure 3-1 Methodological approach used in this study

The content analysis process employed in this research was based on inductive content analysis methodology [144]. This method was chosen due to its suitability for systematic analysis of qualitative data, allowing for the extraction and organisation of information from a comprehensive dataset of aircraft maintenance records. The process includes 3 phases as follows:

Preparation Phase

The preparation phase began with the collection of all available maintenance records, including both paper and scanned documents, to ensure a comprehensive dataset. Following this, a representative sample of documents was selected from the collection, covering various document types, such as task cards, technical reports, and airworthiness certificates. The selected documents were then divided into manageable sections, typically ranging from three to ten pages, to facilitate a detailed analysis. An initial review of these sections was performed to assess their variability and complexity and to determine whether they should be analysed in parallel or sequentially based on their similarity. This approach ensures that the data are well prepared for subsequent detailed analysis.

Organising Phase

The organising phase involves open coding, where documents are carefully read to understand the types of documents and their attributes. Key concepts were identified, such as special checks, routine maintenance task cards, daily ramp checks, D checks, airworthiness certificates and maintenance review certificates. These concepts and attributes were collected on coding sheets.

Reporting Phase

In the reporting phase, each identified category and subcategory was detailed with supporting excerpts from documents to maintain data richness. Concept mapping was used to illustrate the document types, and nucleotide attributes were grouped and defined. The reporting provided a reliable representation of domain knowledge, which is essential for developing the domain ontology.

3.1.1 Dataset Acquisition

This research focuses on a comprehensive dataset comprising aircraft maintenance documentation for the Boeing 737 aircraft managed by Digital Aviation Research and Technology Centre (DARTeC) at Cranfield from 1991 to 2014. The documents are stored in paper formats and their respective scan

copies. These documents contain a wealth of information crucial for understanding the maintenance data, patterns, challenges, and evolution of practices over more than two decades. Boeing 737, one of the most common and long-serving commercial aircraft, presents a unique case study examining maintenance records and document's structure.

To facilitate efficient analysis of the maintenance records, the documents' structure and attributes were extracted into an excel document. The process of extracting structure and attributes from the maintenance records of the Boeing 737 aircraft involves a detailed analysis of documents to identify key information components. Special attention is paid to the structure and context of the information to ensure that the extracted concepts accurately represent the content and purpose of each document.

3.2 Similarity Measures

The final phase focuses on evaluation and validation of the developed ontology model to ensure its accuracy and effectiveness. A case study was conducted using A Check maintenance records for a Boeing 737-400. This case study aims to validate the practical utility of an ontology-based application and Named Entity Recognition (NER) system. The evaluation involved comparing the original and corrected maintenance records to assess the improvements in data consistency and categorisation accuracy. The successful validation of the ontology and its application demonstrated its potential to enhance data integration and decision-making processes in aircraft maintenance operations.

3.2.1 Resnik

This is a node-based similarity measure that is widely applied in calculating semantic similarity between ontology concepts based on information content (IC). Resnik similarity was originally developed for the WordNet and applied to ontologies that are widely used in natural language processing [112,145]. The fundamental idea behind the Resnik similarity [98,99] is measuring the closeness between two terms as a simple *IC* of their most informative common ancestor (MICA) with the following steps:

I. Compute IC
II. Find the most Informative Common Ancestor; and
III. Compute the Resnik similarity considering the most informative common ancestor, as the most related and similar concept connected through that ancestor.

Equation 3.1 shows the mathematical representation of the Resnik similarity measure.

$$sim_resnik(s,t) = IC(LCS(s,t)) \quad \textbf{Equation 3-1}$$

Where:

For each attribute *s* in *S* and *t* in *T*:

LCS(s,t) represents the most specific common ancestor or Lowest Common Subsumer of concepts and;

For each concept *c* in either *S* or *T*:

$$IC\ (c) = -\log\ (p(c)) \quad \textbf{Equation 3-2}$$

Where:

p(c) represents the probability of occurrence of concept *c* in the provided Corpus.

Store *IC* for each concept in a table for easy access.

3.2.2 Levenshtein

This is a similarity measure also known as edit distance, that is used as a metric for quantifying the difference between two strings. This similarity measure calculates the minimum number of a single character needed to transform one string into another [146]. This is useful in ontology mapping, concept matching, or related applications for understanding the similarities between the terms [147]. The Levenshtein similarity measure for ontology can be adapted by the following steps and mathematical representations:

i. Extract Names or Labels

Let $\{a_1, a_2, \ldots, a_m\}$ be the set of names or labels from the Ontology. Let $\{b_1, b_2, \ldots, b_m\}$ be the entries extracted from the aircraft maintenance dataset.

ii. Calculate the Levenshtein Distance

The Leveshtein distance between 2 strings a and b can be represented mathematically using dynamic programming approach. Let a = $a_1, a_2 \ldots a_m$ and b = $b_1, b_2 \ldots b_n$. The Levenshtein distance $d(a,b)$ is defined as the minimum number of single character insertions, deletions or substitutions required to transform string a and b.

We define a distance matrix D where $D[i][j]$ is the distance between the first i characters of a and the first j characters of b. The dimensions of this matrix are $(m+1) \times (n+1)$, where m and n are the lengths of a and b, respectively.

Initialisation

The base cases are:

$$D[i,0] = i \quad for\ i = 0, 1, \ldots, m$$

$$D[0,j] = j \quad for\ j = 0, 1, \ldots, n$$

Recurrence Relation

For I = 1,2,…m and j = 1,2,…n, the matrix is filled with the following recurrence relations:

$$D[i][j] = \min \begin{cases} D[i-1][j] + 1 & (deletion) \\ D[i][j-1] + 1 & (insertion) \\ D[i-1][j-1] + \delta\ (a_i, b_j) & (substitution) \end{cases}$$

Where:

$$\delta(a_i, b_j) = \begin{cases} 0 & if \quad a_i = b_j \\ 1 & if \quad a_i \neq b_j \end{cases}$$

$\delta\ (a_i, b_j)$ is the value of substituting a_i with b_j , which is 0 if they are the same and 1 if they are different.

Therefore, the final Levenstein distance between the two strings a and b is given by $D[m,n]$

iii. Normalised the Distance

The normalised Leveshtein distance between 2 strings a and b is given by:

$$Normaliseed\ Distance = \frac{lev(a,b)}{max(length(a),length(b))}$$ **Equation 3-3**

Where:

$lev(s,t)$ is the distance between strings a and b

$max(length(a),length(b))$ is the length of the longer string between a and b

iv. Apply Thresholds

Define a threshold T such that 2 strings a and b are considered similar if:

$Normaliseed\ Distance\ (\mathrm{a},\mathrm{b}) \leq T$

3.3 Evaluation and Validation Metrics

In this section, the metrics for evaluation and validation the results from the experiment of the thesis are explained together with their mathematical representations.

3.3.1 Precision

This is the metric that access the relevance of the retrieved concepts or entities as compared to reference or the standard [77,148–151]. It measures the proportion of the retrieved concepts that are relevant concepts. The precision is from 0 to 1, a higher precision value indicates a higher portion of the retrieved relevant concepts are correct. This is a crucial metric in evaluating the performance of ontology related tasks such as ontology matching, ontology alignment or entity recognition within ontology. It assesses the reliability of the procedure or algorithm in identifying the relevant concepts which fundamental in ontology and semantic technologies.

Equation 3-4 is the mathematical representation of precision:

$$Precision = \left(\frac{Tp}{Tp+Fp}\right) 100$$ **Equation 3-4**

where Tp indicates true positives from the ontology mapping, and Fp refers to the false positives from the ontology mapping.

3.3.2 Recall

This metric measures the comprehensiveness of the method in the retrieving the relevant concepts or entities from a reference. It measures the proportion of the correctly retrieved relevant concepts from the entire relevant concepts that is supposed to be retrieved. The recall value ranges from 0 to 1, a higher recall value means the proportion of the retrieved relevant concepts are high [149,152]. This is a crucial metric in evaluating the performance of ontology related tasks such as ontology matching, ontology alignment or entity recognition within ontology particularly when assessing the concept retrieval comprehensiveness.

Equation 3-5 is the mathematical representation of recall:

$$Recall = \left(\frac{Tp}{Tp + Fn}\right) 100 \qquad \textbf{Equation 3-5}$$

where Tp indicates true positives from the ontology mapping, and Fn refers to the false negatives from the ontology mapping.

3.3.3 Accuracy

This is a fundamental metric that measures the overall correctness or the reliability of a method in identifying and classifying concepts, entities, or relationships within an ontology. This shows the proportion of the instances that are correctly classified out of the total evaluated instances. The accuracy is from 0 to 1, a higher accuracy value means higher proportion of the instances are correctly classified. This is important metric in ontology because it shows the overall efficacy of the method in correctly classifying the identify concepts, entities, or relationships. The goal of every ontology method is achieving high accuracy[149,152].

Equation 3-6 is the mathematical representation of accuracy:

$$Accuracy = \frac{Number\ of\ True\ Positives\ +\ Number\ of\ True\ Negative}{Total\ Number\ of\ Instances}$$ **Equation 3-6**

3.3.4 F-Measure

This is a metric that combines recall and precision to provide a balanced assessment of the method that identify relevant concepts or entities from a reference. This is useful when both recall and precision are considered simultaneously. This ranges from 0 to 1 with higher values indicating a better balance between precision and recall. The F-measure reaches its maximum value of 1 when precision and recall are both 1 [149,152].

$$f - measure = 2\left(\frac{Precision\ \times\ Recall}{Precision\ + Recall}\right)$$ **Equation 3-7**

Where:

Precision is the proportion of correctly retrieved relevant concepts out of all retrieved concepts.

Recall is the proportion of correctly retrieved relevant concepts out of all relevant concepts.

4 Agile Development for Ontology-Based Applications (ADOBA)

4.1 Development Methodology

Agile Development for Ontology-Based Applications was proposed as a new integrated approach for building ontology applications. The methodology supports both the development on the domain ontology and the application which it is designed to interface. The key feature of the methodology is the iterative evaluation of both the ontology and corresponding application in an incremental sequence. The phases of development include:

I. Planning: The development process begins the planning phase where the definitive objectives, scope and requirements of the ontology-based application, domain coverage and assumptions are clearly defined. These include understanding the domain area, tasks, resources, target users and development schedule. The culmination of the planning phase results in the planning document.

II. Requirements Specification and Knowledge Acquisition: In this phase, an in-depth analysis is carried out to understand the domain. Notably, the integrated approach incorporates the knowledge acquisition activity from Methontology, primarily because the bulk of acquisition emerges during the requirement specification phase, as described in [24]. However, more knowledge can be discovered as the development of unfolds, reinforcing the use of an iterative approach and end-users involvement echoing the tenets of DSDM [53,153]. The outputs resulting from this phase are twofold: the ontology specification document and software requirement specification. These documents highlight the connection between the ontology's blueprint and the application's capabilities.

III. Ontology Development: This phase includes a sequence of distinct yet interconnected steps which include conceptualisation, formalisation, integration and evaluation. Within the purview of ADOBA, the specification phase is sequentially positioned post-planning covering specific

requirements of both ontology and the application. While Methontology recognises maintenance as a post-deployment corrective and updating phase, its manifestation within the ADOBA is inherently flexible, rendering the ontology receptive to dynamic requirements changes throughout its lifecycle. Hence, supporting the ontology development based on evolving prototypes [24]. During conceptualization, the domain knowledge is synthesised into a structured model, detailing concepts, terminologies, properties, and relations. The subsequent formalization phase serves to translate this conceptual model into a definitive formal structure. Integration, on the other hand, supports reuse of existing ontologies which can speed up the development process. During implementation, a computable model is built by leveraging on ontology development tools and languages. In this research, protégé employed to build the ontology using OWL.

a) Conceptualisation

The conceptualisation phase of ADOBA involves identifying the essential domain concepts, relationships, and terminology relevant to the specific domain that the ontology aims to represent. The outcome of this phase is the concepts and attributes to be incorporated into the ontology. ADOBA prioritises iterative refinement of this conceptual model through continuous feedback from domain experts. Rather than being a singular event, the conceptual model evolves as development advances, with new concepts added or refined as necessary. Concepts are logically grouped, and the relationships between terms are mapped to provide a structured foundation that will inform the formalisation phase.

b) Formalisation

The formalisation phase includes translating the conceptual model into a formal representation. ADOBA employs iterative cycles to refine formalisation, allowing for adjustments as new requirements or insights arise. By translating the concepts into formal languages, it ensures that

the ontology can be integrated with software applications, facilitating reasoning and inference.

c) Integration

In the integration phase, ADOBA emphasises the reuse and integration of existing ontologies where applicable to enhance the efficiency and consistency of ontology development. This phase supports the dynamic integration of new or external ontologies as they are identified, supporting the evolving nature of both ontology and application. ADOBA encourages integration not only with ontologies from external sources, but also within the development of the application itself, ensuring that the ontology is continuously aligned with the software it is intended to support. The integration process in ADOBA is iterative, feedback from the implementation phase may lead to new ontologies being integrated, and changes may be made as the requirements evolve.

d) Implementation

The Implementation phase within ADOBA is centred on the integration of the formalised and integrated ontology into the software application. A distinctive feature of ADOBA is the concurrent execution of implementation along with ontology development. As ontology undergoes formalisation and integration, preliminary versions are evaluated within functional prototypes, enabling developers to identify deficiencies, issues, and opportunities for enhancement. This iterative methodology ensures the continuous improvement of the ontology based on user feedback and the evolving requirements of the application. In ADOBA, ontology is not regarded as a static construct, rather, it is embedded within the application and subjected to real-world testing from the outset of development.

e) Evaluation

The evaluation phase of ADOBA is an iterative process that persists throughout the development lifecycle. Rather than constituting the final step, evaluation in ADOBA is dynamic and continuous. As ontology evolves, it is consistently tested for accuracy, consistency, and

applicability in real-world contexts. ADOBA prioritises task-based evaluation, wherein the ontology's efficacy is assessed based on its support of the application's domain-specific tasks. The evaluation is conducted not only at the completion of ontology development but also during each iteration of the development cycle, ensuring that any issues or deficiencies in the ontology are identified and addressed. This iterative evaluation process is crucial for maintaining the ontology's flexibility, adaptability, and alignment with user needs, thus facilitating continuous improvement as the application progresses.

The phases of conceptualisation, formalisation, integration, implementation, and evaluation are interconnected, with each phase contributing to the next in an iterative and adaptable process. ADOBA support the concurrent development of both ontology and application, ensuring that feedback from each phase informs and refines the others. This iterative approach facilitates continuous improvement and adaptation to evolving requirements and ensures that the ontology remains aligned with the application's needs throughout its lifecycle. By integrating these phases with agile principles, ADOBA offers a robust methodology for the development of ontology-based applications that are both effective and adaptable.

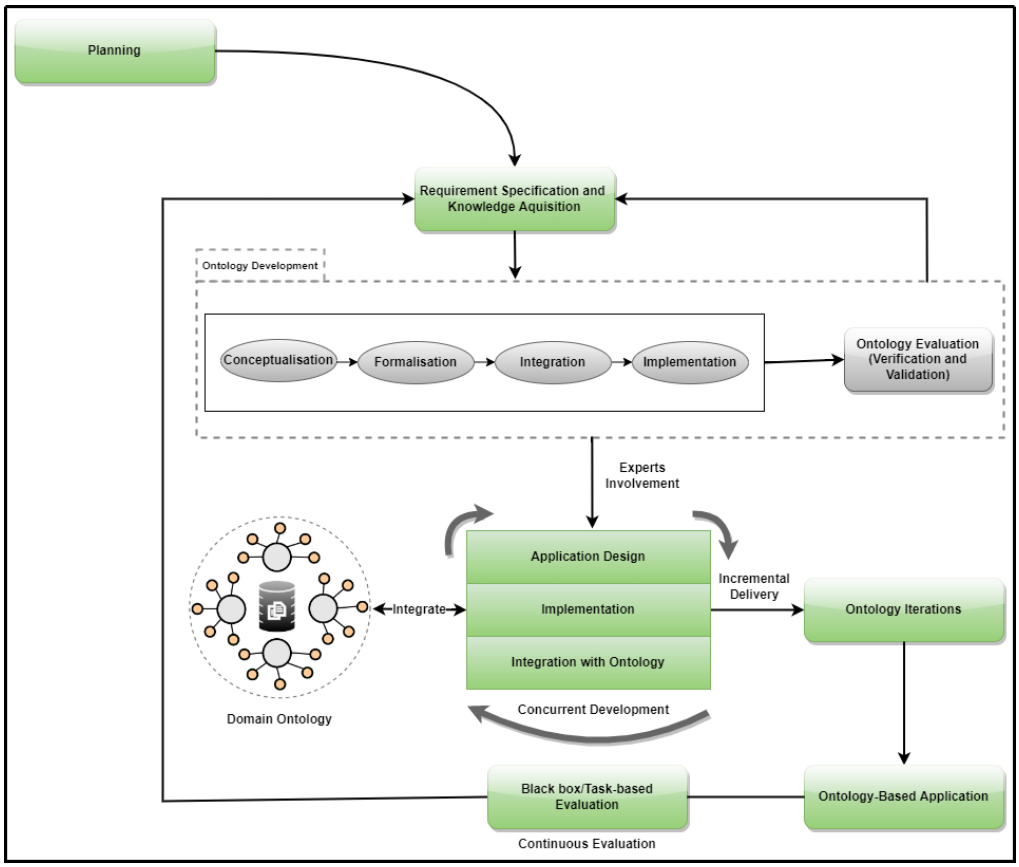


Figure 4-1 Proposed ADOBA methodology [143]

IV. Concurrent Development:

Concurrent Development is a pivotal stage in the development of ontology-based applications, particularly when implemented within a hybrid agile methodology. This phase is characterised by an emphasis on constructing a high-level functional prototype. This prototype is developed iteratively, ensuring that it aligns closely with the requirements specification, and demonstrates its practicality for the business case. In this phase, the design, implementation, and integration are carried out concurrently.

The process follows an iterative approach, in which developers constantly test the prototype and refine it based on end-user feedback and newly identified requirements. Therefore, the involvement of the end-users is sustained throughout this stage. This consistent user feedback loop is instrumental in ensuring that the prototype is not just compliant with requirements but is also practically functional and usable.

While design decisions are typically established during the requirements specification and knowledge acquisition phases, the actual realisation and integration of ontology occur during current development. The ontology, which is the foundation upon which the application is built, is integrated into the prototype during this phase. This ensures that the ontological specifications are validated in a real-world application context.

Recurrent oversight in many projects underestimates the intricate interdependence between ontology and its integration within applications, particularly when these applications are designed as information systems. The ontology, while a standalone component, is tied to the application. Its efficiency, accuracy, and relevance can affect the effectiveness of the application. As such, in the concurrent development phase, the ontology is not treated as a separate entity, but is deeply integrated and tested within the application's context.

The hybrid agile methodology highlights this interdependence, ensuring that the ontology and application evolve concurrently, reinforcing each other.

V. Black box or Task-based Evaluation:

In the application development process, ontology-based applications reside at the intersection between software and ontological engineering. The task-based [154] or black-box evaluation phase provides a critical point for assessing and analysing this integration, merging traditional software evaluation methods with the fundamental requirements of ontological accuracy and relevance.

From the perspective of software engineering, the evaluation of a typical application is largely centred on its functional correctness, performance, and user experience. However, in the context of ontology-based applications, a new dimension of evaluation comes into play; the effectiveness of the application in leveraging the ontology to perform domain-specific tasks. In this phase, the functionalities of the ontology-

based application are tested, emphasising the pivotal tasks that manifest the real-world applicability of the ontology. This includes complex data integration tasks and seeks to ascertain the actionable utility of the ontology beyond its theoretical foundation. What distinguishes this from traditional software testing is the application's performance, based on established test cases that are specifically designed to assess the ontology's domain-specific criteria [37].

Whereas traditional ontology validation and verification processes in ontological engineering prioritise formal correctness and specification alignment, the task-based evaluation emphasises the practical, actionable, domain-aligned utility of the ontology within the application. This involves navigating the domain specificity, ensuring that the application not only works but works meaningfully within the intended context of the domain. Traditional methodologies such as Methontology, NeOn, and On-to-knowledge have been oriented towards the development and evaluation of ontologies, treating them as distinct entities.

ADOBA incorporates the structured phases of Methontology with the adaptable user-centric principles of DSDM [155]. This offers a methodology that covers both software and ontological engineering paradigms, thereby ensuring that ontology-based applications are technically robust and semantically meaningful.

4.1.1 Ontology-based application Development for Aircraft Maintenance Records

The key to developing an efficient ontology-based application lies in the accurate representation of concepts and relationships and semantics in the domain ontology. Such ontology, while challenging to develop, promise a standardised knowledge base that can be invaluable for domain applications particularly for semantic data integration.

In this study, Agile Development for Ontology-Based Applications (ADOBA) is presented as a structured approach to modelling aircraft maintenance record data ontology (see section 5.1) to address the aforementioned challenges. The summary of the ADOBA methodology is presented in Table 4-1 below.

Table 4-1 Summary of ADOBA methodology

Phase	Description	Activities	Evaluation Points
Planning	Establish the project scope, objectives, and allocate resources.	Define the ontology goals. Allocate team and resources. Create a timeline.	Ensure alignment with project goals. Confirm resource allocation. Validate the timeline.
Requirement Specification and Knowledge Acquisition	Gather detailed requirements and domain knowledge necessary for the ontology.	Engage with domain experts. Review maintenance records. Define key concepts and relationships.	Verify the completeness of requirements. Check for accurate representation of domain knowledge.
Ontology Development	Construct the ontology in a structured process.	Conceptualisation: Define concepts and relationships. Formalisation: Translate into a formal language. Implementation: Build the ontology model	Verification: Assess technical correctness. Validation: Ensure real-world relevance.
Experts Involvement	Continuous engagement of domain experts throughout the development process.	Validate the ontology's accuracy and relevance. Provide domain-specific insights and knowledge.	Ensure ontology's validity and applicability through expert feedback.

Application Design	Design the application and architecture incorporating the ontology.	Application Design Define ontology application within the system.	Appraise design against user requirements and system feasibility.
Implementation	Develop the application's codebase where the ontology will be utilised.	Implement the application functionalities. Develop the integration components for the ontology.	Test functionalities for adherence to design.
Integration with Ontology	Combine the developed ontology with the application to ensure the system can employ structured knowledge effectively.	Embed the ontology into the application. Establish data flow and usage.	Evaluate the integration for data consistency and functionality.
Black Box/Task-Based Evaluation	Perform real-world task-based tests to assess the performance and utility of the application without examining the internal ontology workings.	Execute end-to-end system tests. Validate against predefined tasks and user scenarios.	Evaluate application effectiveness in real-world scenarios.
Ontology-Based Application	The outcome is an application that utilises the ontology for managing	Plan for ongoing maintenance and updates.	Measure performance and application impact on operational efficiency.

aircraft maintenance records data.

5 Ontology Modelling and Design of Aircraft Maintenance Records

In the context of integrating Aircraft Maintenance Records, there are three primary challenges that directly affect maintenance operations and, subsequently, the airworthiness of aircraft. These challenges stem from the nature of the data involved, historical evolution of aircraft maintenance record-keeping practices, and technological landscape of MRO IT systems aircraft maintenance domain.

Heterogeneity of Data Sources

The first significant challenge is the heterogeneity of the data sources. Aircraft maintenance records are obtained from a wide array of sources, each with its own format, structure, and level of detail. The traditional paper-based records, digital logs, and data captured by modern MRO IT systems must be integrated. The disparity in these formats leads to a scenario in which crucial data are siloed, making it difficult to achieve a comprehensive view of an aircraft's maintenance history. This heterogeneity not only complicates data retrieval and analysis but also poses a risk of misinterpretation of critical maintenance information.

Lack of Interoperability Between Systems

The lack of interoperability between the existing MRO IT systems is closely related to the challenge of heterogeneous data sources. Each system, which is often developed independently to serve specific operational needs, operates in isolation without a standardised mechanism for data exchange. This lack of interoperability results in significant barriers to the seamless integration of maintenance records, necessitating manual interventions that are both time consuming and prone to errors.

Semantic and Syntactic Ambiguity in Maintenance Records

In the practical context of maintaining aircrafts, this semantic and syntactic ambiguity manifests as a significant barrier to accurately understanding and actioning the myriad of information contained in maintenance records. Semantic inconsistencies are based on the expected use of terminology within the domain of the aircraft maintenance records. Syntactic ambiguity compounds this issue by introducing inconsistencies in how information is structured and presented across AMR.

To illustrate the correction of semantic and syntactic inconsistencies, consider the following real-world example in Table 5-1 involving two aircraft maintenance task cards:

Table 5-1 An example of semantic and syntactic inconsistencies in maintenance task cards

Task Card 1	Task Card 2 (with Semantic and Syntactic differences)
ID: 00071 Aircraft Registration: GDOCB Title: Inspect Flight Deck Seat Harnesses ManHours: 0.4 Revision: 1.0 Task Description: "Inspect (Detailed) the captain seat harnesses, crotch straps, and	ID: 00072 Aircraft Registration: GDOCB Title: Examine Flight Deck Seat Belts ManHours: 0.4 Revision: 1.0 Task Description: "Inspect the captain's sit harnesses and crotch straps for signs of weariness and fastening."

shoulder belts (as applicable) for wear, condition, and security."	

Let *T* be Taskcard.

Let Syntactic (*T*) be a predicate indicating that Taskcard has a syntactic difference.

Let Term (*T, n)* be a predicate indicating that Taskcard has *n* semantic differences.

Let Term (*T, x*) be a predicate indicating that Taskcard T contains term *x*.

Assertion 1 (Syntactic difference 00072):

Syntactic (T_{00072}, "Examine")

Task Card 00072 has a syntactic difference due to the spelling "sit".

Assertion 2 (Semantic difference in title for 00072):

¬Term (T_{00071}, "Examine") ∧ Term (T_{00072}, "Examine")

The title of the task card 00071 does NOT contain the word "Examine" AND the title of the task card 00072 does contain the word "Examine".

Assertion 3 (Semantic difference in title for 00072):

¬Term (T_{00071}, "weariness") ∧ Term (T_{00072}, "weariness")

The description of the task card 00071 does NOT contain the word "weariness" AND the description of the task card 00072 does contain the word "weariness".

5.1 Modelling Aircraft Maintenance Records Ontology

Modelling ontologies for aircraft maintenance records are a critical component in advancing knowledge management capabilities in the aviation industry. The foundation of MRO systems lies in domain-specific modelling of aircraft maintenance knowledge [32]. The heterogeneity of the maintenance data from multiple sources presents a formidable challenge and manifests in a range of structural, formatting, lexical, and semantic disparities. As noted by Bouquet et al. [156], these discrepancies can be categorised into several distinct layers of heterogeneity: the syntactic layer concerning the representation formats, the terminological layer involving the variability in nomenclature, the conceptual layer that encompasses the depth and breadth of content, and the semiotic layer pertaining to the interpretation of the information.

The use of ontologies is becoming more prevalent in the integration of aircraft maintenance records. However, this task has inherent challenges that can impede the development of ontology-based applications as demonstrated by Figure 5-1.

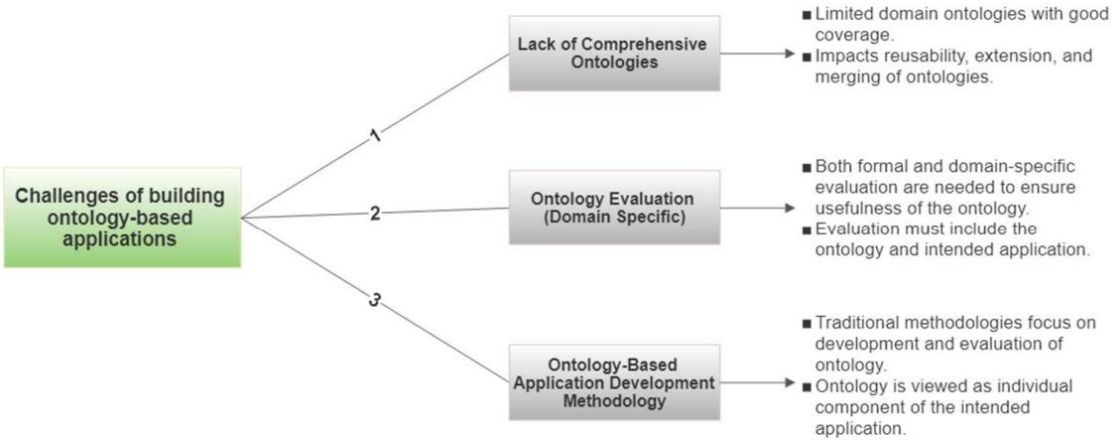


Figure 5-1 Challenges of building ontology-based applications

Despite the growing recognition of ontology value within the aviation industry, as evidenced by recent studies [44,157,158], there is limited work on aircraft maintenance record ontologies, which limits the potential for reusability, a concept emphasised by Pinto and Martins [118], where ontologies are envisioned

as modular elements that can be readily used, expanded upon, or combined. Furthermore, the aspect of domain coverage, which is crucial for the applicability of ontology, remains underexplored. Domain coverage refers to the extent of the domain concepts that an ontology encompasses for its intended application. While aiming for exhaustive coverage might seem ideal, priority should instead be to achieve the widest possible scope and detail to ensure the ontology's practical utility [50,62,159]. Current ontologies in AMR tend to offer modular yet less expressive models, which may not suffice for broader or more in-depth applications. Lehmann et al. [142] have also observed that insufficient coverage or the complete absence thereof can significantly impede the effective use of semantic technologies within the aviation sector.

Evaluating the quality and richness of ontologies is critical to ensuring their usefulness. The usefulness of an ontology is largely determined by the depth of its content and thoroughness of its development [159]. However, the practical application of an ontology depends on its accurate reflection of the real-world domain and the specific challenges it aims to address. Given that ontologies are constructed based on a consensus among domain stakeholders, it is crucial that their evaluation is adaptable to the diverse needs of various applications. Obrst et al. [160] highlighted these ontologies, even within the same domain but expressed in different languages, such as OWL or CycL, require distinct evaluation approaches. This emphasises the importance of thoroughly evaluating ontologies intended for knowledge-based applications. The general lack of interest in such evaluations, as pointed out by Fernandez-Lopez et al. [24], risks the efficacy of knowledge applications that rely on these ontologies. The absence of a systematic evaluation strategy is seen as a barrier to the success of ontology-driven knowledge technologies, as argued in [161]. The literature suggests various evaluation methods [160–162], some of which fall under established ontology development frameworks such as Methontology [24]. Therefore, it is imperative to prioritise both formal and domain-specific evaluations when developing ontologies for the aircraft maintenance record domain.

Although methodologies such as NeOn, On-to-Knowledge, and Methontology offer structured lifecycle stages for ontology development, they traditionally concentrate on developing the ontology as a standalone component detached from any specific application. This dichotomy has led to the utilisation of separate methodologies for the development of ontology and its corresponding applications. Sure et al. [54] addressed this gap by introducing the knowledge meta-process, a methodology that encompasses both ontology and application development within the On-to-Knowledge framework, facilitating the development of ontology-based knowledge applications. In a move towards more dynamic development practices, recent research efforts have explored the inclusion of agile methodologies into ontology creation, including the Simplified Agile Methodology for Ontology Development [52] and OntoSoft Process [53]. These advancements highlight the growing consensus regarding the necessity of blending ontology development with software engineering principles to streamline the creation of ontology-based applications.

5.2 Maintenance Document Content Analysis

This section presents the results of the analysis of aircraft maintenance records used in this study.

5.2.1 Structure and Composition of G-DOCB Maintenance Records

G-DOCB maintenance records include a wide spectrum of document types, each representing a unique operational or maintenance event during aircraft lifecycle. The granularity of the data is suitable for this research, ranging from routine maintenance task cards and daily ramp checks to comprehensive D checks, which occur approximately every to 6-10 years. The diversity of documentation provides a nuanced perspective on the varying frequencies and scopes of maintenance activities required throughout the aircraft's service period.

The structure of the maintenance records shows the complex interplay between different maintenance activities. Figure 5-2 shows a visual representation of the most common maintenance documents in the 107 boxes reviewed.

G-DOCB maintenance records cover a broad range of document types, each of which represents a distinct operational or maintenance event that occurs during the aircraft's lifecycle. The level of detail in these records is suitable for this research, ranging from routine maintenance task cards and daily ramp checks to extensive D checks that take place approximately every 6-10 years. The variety of documentation provides a comprehensive understanding of the varying frequencies and scopes of maintenance activities that are required throughout the aircraft's service period.

The grouping of the maintenance records illustrates the complex relationship between different maintenance activities. As shown in Figure 5-2, the most common maintenance documents in the 107 boxes reviewed are visualised. The attributes are presented in Table 5-2.

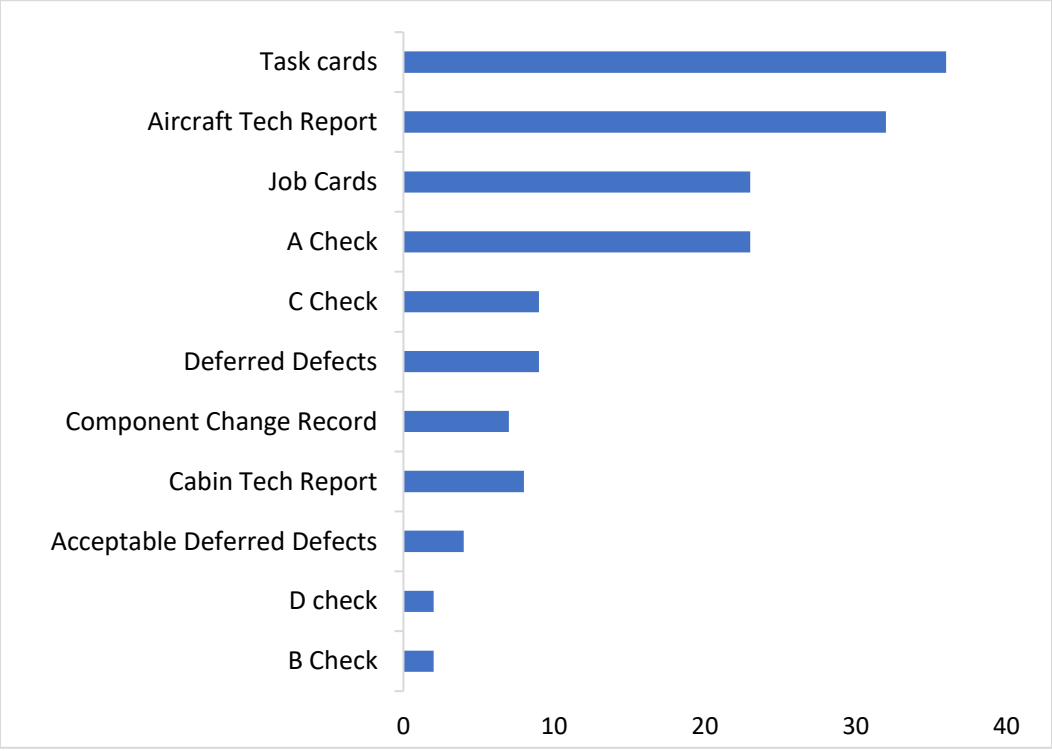


Figure 5-2 Distribution of G-DOCB maintenance records

This G-DOCB maintenance record serves as an important resource for ontology development for the aircraft maintenance record domain. The richness and variety of maintenance records help facilitate a thorough representation of domain knowledge, ensuring that the resulting ontology comprehensive and capable of supporting sophisticated knowledge-based applications in aircraft maintenance. Examples of these maintenance records are shown in Figure 5-3 and Figure 5-4.

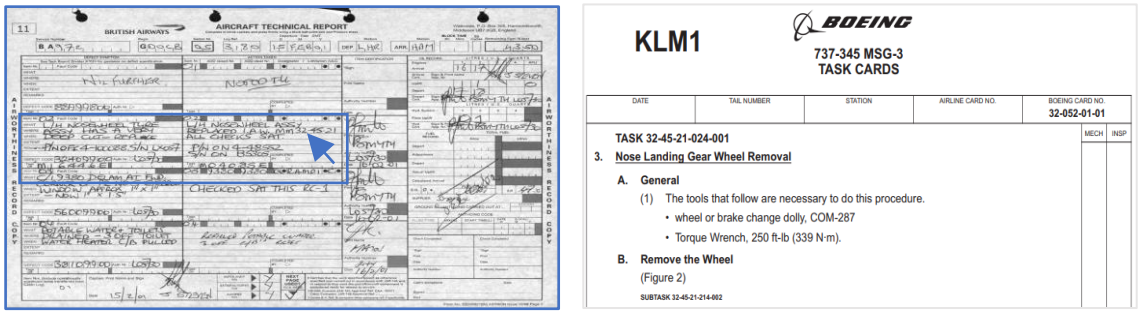
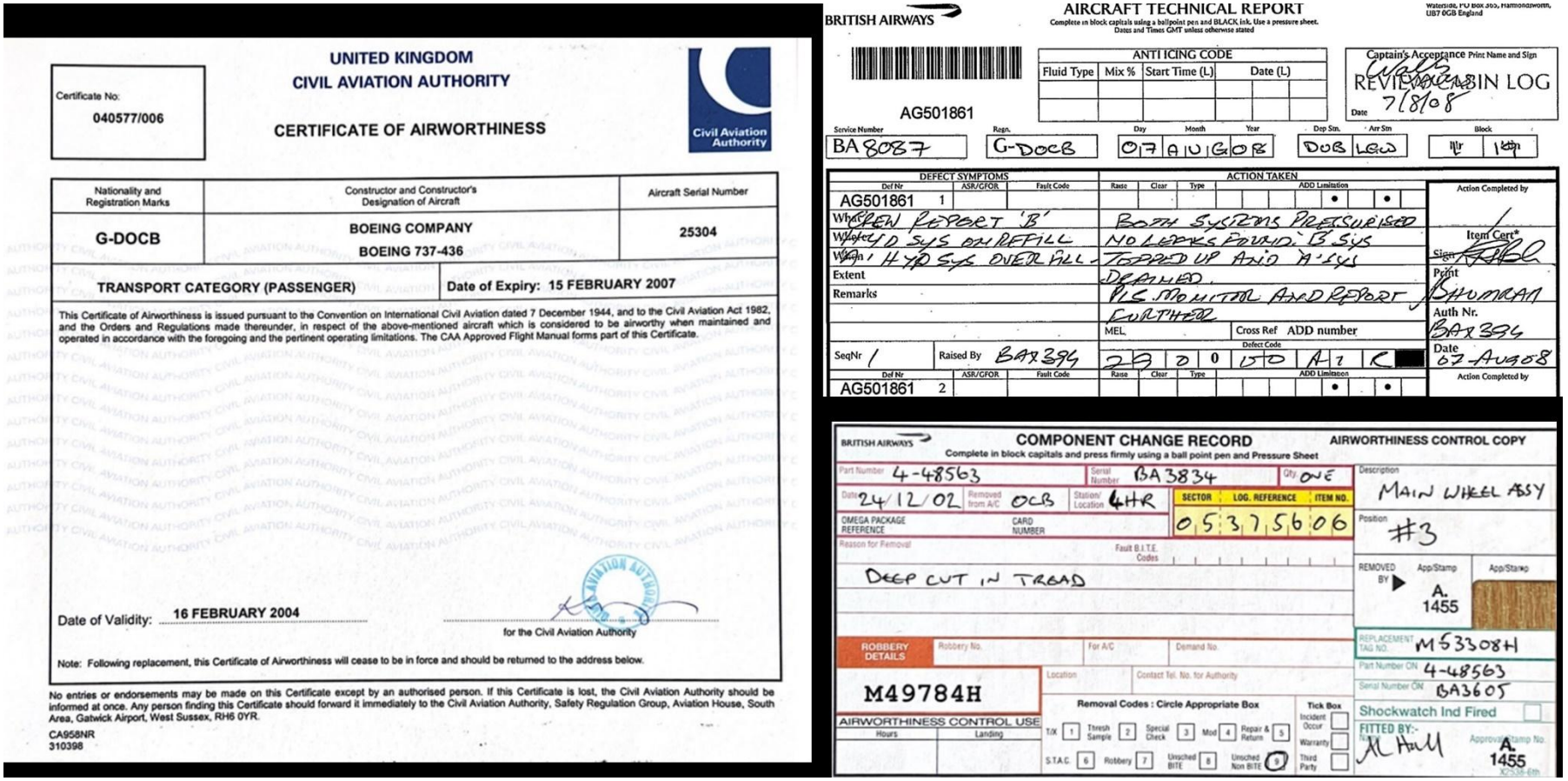


Figure 5-3 G-DOCB aircraft technical report and AMM task cards

G-DOCB Documentation Categories

The concepts are categorised to reflect the various aspects of maintenance activities. This classification includes categorising documents according to their type (e.g., task cards, technical reports), nature of maintenance activities (e.g., routine checks, ramp checks), and compliance-related aspects (e.g., airworthiness certificates and CAA approvals).

These documents comprise a vast collection of paper-based maintenance records, including task cards, technical reports, airworthiness certificates, and logs of deferred defects. Given Boeing 737's long service history and the period covered, the dataset represents a significant portion of the aircraft's maintenance lifecycle.



UNITED KINGDOM
CIVIL AVIATION AUTHORITY

Certificate No: 040577/006

CERTIFICATE OF AIRWORTHINESS

Civil Aviation Authority

Nationality and Registration Marks	Constructor and Constructor's Designation of Aircraft	Aircraft Serial Number
G-DOCB	BOEING COMPANY BOEING 737-436	25304
TRANSPORT CATEGORY (PASSENGER)	Date of Expiry: 15 FEBRUARY 2007	

This Certificate of Airworthiness is issued pursuant to the Convention on International Civil Aviation dated 7 December 1944, and to the Civil Aviation Act 1982, and the Orders and Regulations made thereunder, in respect of the above-mentioned aircraft which is considered to be airworthy when maintained and operated in accordance with the foregoing and the pertinent operating limitations. The CAA Approved Flight Manual forms part of this Certificate.

Date of Validity: 16 FEBRUARY 2004

for the Civil Aviation Authority

Note: Following replacement, this Certificate of Airworthiness will cease to be in force and should be returned to the address below.

No entries or endorsements may be made on this Certificate except by an authorised person. If this Certificate is lost, the Civil Aviation Authority should be informed at once. Any person finding this Certificate should forward it immediately to the Civil Aviation Authority, Safety Regulation Group, Aviation House, South Area, Gatwick Airport, West Sussex, RH6 0YR.

CA958NR
310398

BRITISH AIRWAYS

AIRCRAFT TECHNICAL REPORT

Complete in block capitals using a ballpoint pen and BLACK ink. Use a pressure sheet. Dates and Times GMT unless otherwise stated

Waterside, PO Box 365, Harmondsworth, UB7 0GB England

AG501861

ANTI ICING CODE

Fluid Type	Mix %	Start Time (L)	Date (L)

Captain's Acceptance Print Name and Sign

REVIEWED IN LOG

Date 7/8/08

Service Number	Regn.	Day	Month	Year	Dep Stn.	Arr Stn	Block
BA 8087	G-DOCB	07	AUG	08	DUB	LGW	1hr 15m

DEFECT SYMPTOMS			ACTION TAKEN							
Def Nr	ASR/GFOR	Fault Code	Raise	Clear	Type	ADD Limitation				
AG501861 1										
When GREEN REPORT 'B'			BOTH SYSTEMS PRESSURISED							
When G/D SYS ON REFILL			NO LEAKS FOUND. B SYS							
When H/D SYS OVER FILL.			TOPPED UP AND A SYS							
Extent			DRAINED.							
Remarks			PLS MONITOR AND REPORT FURTHER							
			MEL			Cross Ref ADD number				
						Defect Code				
SeqNr 1	Raised By BAX394		29	0	0	170	A	1	R	
Def Nr	ASR/GFOR	Fault Code	Raise	Clear	Type	ADD Limitation				
AG501861 2										

Action Completed by

Item Cert*

Sign

Print BHUMRAH

Auth Nr. BAX394

Date 07 Aug 08

Action Completed by

BRITISH AIRWAYS

COMPONENT CHANGE RECORD

Complete in block capitals and press firmly using a ball point pen and Pressure Sheet

AIRWORTHINESS CONTROL COPY

Part Number	Serial Number	Qty	Description
4-48563	BA3834	ONE	MAIN WHEEL ASSY

Date 24/12/02 | Removed from A/C OCB | Station/Location LHR | SECTOR | LOG. REFERENCE | ITEM NO. | Position #3

0 5 3 7 5 6 0 6

OMEGA PACKAGE REFERENCE | CARD NUMBER

Reason for Removal: DEEP CUT IN TREAD

Fault B.I.T.E. Codes

REMOVED BY App/Stamp A. 1455 | App/Stamp

ROBBERY DETAILS | Robbery No. | For A/C | Demand No.

M49784H

Location | Contact Tel. No. for Authority

AIRWORTHINESS CONTROL USE | Hours | Landing

Removal Codes : Circle Appropriate Box

T/X 1 | Thresh Sample 2 | Special Check 3 | Mod 4 | Repair & Return 5 | S.T.A.C. 6 | Robbery 7 | Unsched BITE 8 | Unsched Non BITE 9

Tick Box: Incident Occur | Warranty | Third Party

REPLACEMENT TAG NO. M53308H

Part Number ON 4-48563

Serial Number ON BA3605

Shockwatch Ind Fired

FITTED BY:- Name AL Hall | Approval Stamp No. A 1455

Figure 5-4 G-DOCB Certificate of airworthiness, aircraft technical report and component change record

Table 5-2 Documents attributes from the aircraft maintenance records

S/N	Attribute	Synonym	Description
1	Location	Station/Location	The specific place where maintenance or checks are conducted.
2	Aircraft Registration	A/C Regn, Regn	Registration number identifying the aircraft.
3	Type of Check	Check	Type of maintenance inspection being performed.
4	Aircraft Type	N/A	Model or type of aircraft.
5	Revision	N/A	Revision level of documents or parts used in maintenance.
6	Work Centre	N/A	Specific area or workshop where maintenance tasks are performed.
7	Man-Hours	Hours	Total number of manpower hours dedicated to a task.
8	Position	Position	Specific position on the aircraft where work is performed.
9	Part Number to Remove	Part Number ON	Identification number of a part that needs to be removed.
10	Order	N/A	Reference number or identifier for the maintenance order.
11	Serial Number	S/No, Serial Number ON	Unique identifier for a specific part within an aircraft.

12	Completed	COMPLTD	Indicates whether a task or checklist item has been completed.
13	Card Clearance	N/A	Status indicating maintenance clearance or approval.
14	Country	N/A	Country in which the maintenance is being performed.
15	Certificate Reference Number	N/A	Reference number of a certification relevant to the maintenance.
16	Company Name	N/A	Name of the company performing the maintenance.
17	Company Address	N/A	Physical address of the maintenance company.
18	Company Phone	Contact	Contact phone number of the maintenance company.
19	SITA Code	N/A	Specific airline or airport code used for communications.
20	Work Order/Contract	N/A	Contract or order under which the maintenance work is being done.
21	Item	Item No	Specific item or part being referenced in documentation.
22	Description	N/A	Description of maintenance tasks or part specifications.

23	Part Number	Part Number	Manufacturer's part number for a specific aircraft component.
24	Eligibility	N/A	Criteria that qualify a part or task for use or performance.
25	Quantity	Qty	Number of items or parts involved in the maintenance task.
26	Status/Work	N/A	Current status of the maintenance task or part condition.
27	Remarks	Remarks	Additional notes or comments regarding the maintenance performed.
28	Purchase Order	P/O	Reference number for the purchase order associated with parts.
29	Tested/Repaired	N/A	Status indicating whether an item has been tested or repaired.
30	FAA Repair Station	N/A	Designation indicating a repair station certified by the FAA.
31	Date	Date	Relevant date for the task, inspection, or documentation entry.
32	Name	Name	Name of the person or entity relevant to the documentation.

33	Release to Service	N/A	Authorization for the aircraft to return to service after maintenance.
34	Other Regulation	N/A	Reference to any other regulation applicable to the maintenance.
35	Signature	Sign	Signature of the authorised personnel approving the maintenance.
36	Document Control Number	Doc Control No.	Number used to track and control maintenance documentation.
37	Removed from Aircraft	Removed from A/C	Indicates if a part was removed from the aircraft.
38	Sector	Sector	Sector or segment of the aircraft being worked on.
39	Log Reference	Log Ref	Reference entry in the maintenance log book.
40	OMEGA Package Reference	OMEGA Package Reference	Reference to a specific maintenance package or protocol.
41	Card Number	Card Number	Identification number for a maintenance task card.
42	Removed By	Removed By	The person who performed the removal of a part.

43	Reason for Removal	Reason	The rationale or cause for removing a part or component from the aircraft.
44	Fault B.I.T.E Codes	Fault B.I.T.E Codes	Codes used to diagnose or identify faults in aircraft systems.
45	Robbery Number	Robbery No.	Placeholder or identifier used in specific contexts, unclear usage.
46	For Aircraft	For A/C	Indicates that the specific task or part is intended for the aircraft.
47	Demand Number	Demand No.	Reference number associated with a specific maintenance demand.
48	Landing	Landing	Indicates landing-related checks or maintenance.
49	Removal Code	Removal Code	Code specifying the reason or method for part removal.
50	Replacement Tag Number	Replacement Tag No	Tag number used to track replacements.
51	Shockwatch Indicator Fired	Shockwatch Ind Fired	Indicates if a shockwatch indicator was triggered during handling.
52	Fitted By	Fitted By	Identifies the technician who installed or fitted a part.

53	Approval/Stamp Number	Approval/Stamp No.	Number associated with official approval or certification stamps.

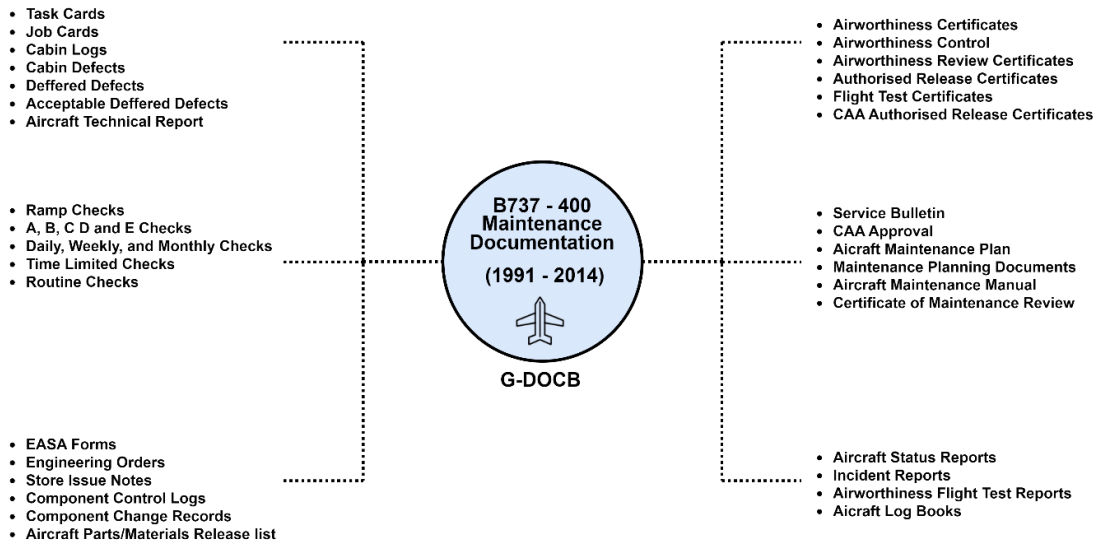


Figure 5-5 Concept mapping for document types of G-DOCB maintenance records

5.3 AMRO Requirement Specification

The AMRO specification describes the structured approach for developing an ontology aimed at integrating heterogeneous aircraft maintenance records for a Boeing 737-400. AMRO development aligns with the Agile Development for Ontology-Based Applications (ADOBA) proposed in Chapter 4, emphasising iterative development and evaluation to adapt swiftly to domain-specific needs. The ontology requirement specification document presented in Table 5-3 was derived based on [141].

Table 5-3 Ontology requirement specification document

AMRO Requirements Specification Document

1	**Purpose**
	Provide a reference ontology model which represents the structure of aircraft maintenance records to improve inconsistency and integration of heterogeneous aircraft maintenance records.
2	**Scope**
	The model is built around the documentation of Boeing 737-400 aircraft covering aircraft maintenance records, with focus on A Check as the use case.
3	**Implementation Language**
	Web Ontology Language (OWL)
4	**Intended Users**
	• Aircraft Leasing Companies • MRO Organisations
5	**Intended Uses**
	• Improve inconsistency in heterogeneous maintenance records • Support Integration of maintenance records between diverse systems • Regulatory compliance checks
6	**Functional Requirements – Competency Questions (CQs)**
	CQ1: What are the maintenance tasks related to a component? CQ2: What maintenance actions have been performed on a component? CQ3: Which components were replaced during A Check? CQ4: What are the defects associated to a component?

CQ5: Which components have been replaced?

7	**Non-Functional Requirements**

- The ontology must be based on Air Transportation Association (ATA) Chapters

5.3.1 Aircraft Maintenance Records Ontology

AMRO is a semantic model that encapsulates the complex details and processes of the aircraft maintenance record domain, including tasks, components, personnel, and documentation (Figure 5-6). The ontology design begins with a comprehensive analysis of the domain, incorporating explicit knowledge and existing standards to ensure that the ontology encapsulates the full breadth of the maintenance record context. At its core, AMRO is developed using OWL to define a structured hierarchy of classes and subclasses, which covers the primary entities of aircraft maintenance records.

5.3.2 Core Structure of AMRO

AMRO is structured into classes that reflect the real-world entities of the aircraft maintenance records domain. This structure was modelled in accordance with the industry-standard Air Transport Association (ATA) chapters, which categorise all aspects of aircraft systems and structures. Classes such as *ATA-71-PowerPlant* and *ATA-72-Engines* represent specific systems, whereas classes such as *AircraftMaintenanceTechnician* represent personnel involved in the maintenance tasks. Each class is equipped with object properties to express relationships, such as *requiresDocument* and *requiresConsumable*, which link documents and consumables, respectively. This level of granularity in the relationships allows precise querying and data manipulation within the ontology. The use of disjoint classes ensures that there is no overlap between entities, where such an overlap is logically or practically impossible. For example, *AChecksDocuments* and *CChecksDocuments* are disjoint, reflecting the exclusivity of these types of checks in maintenance records.

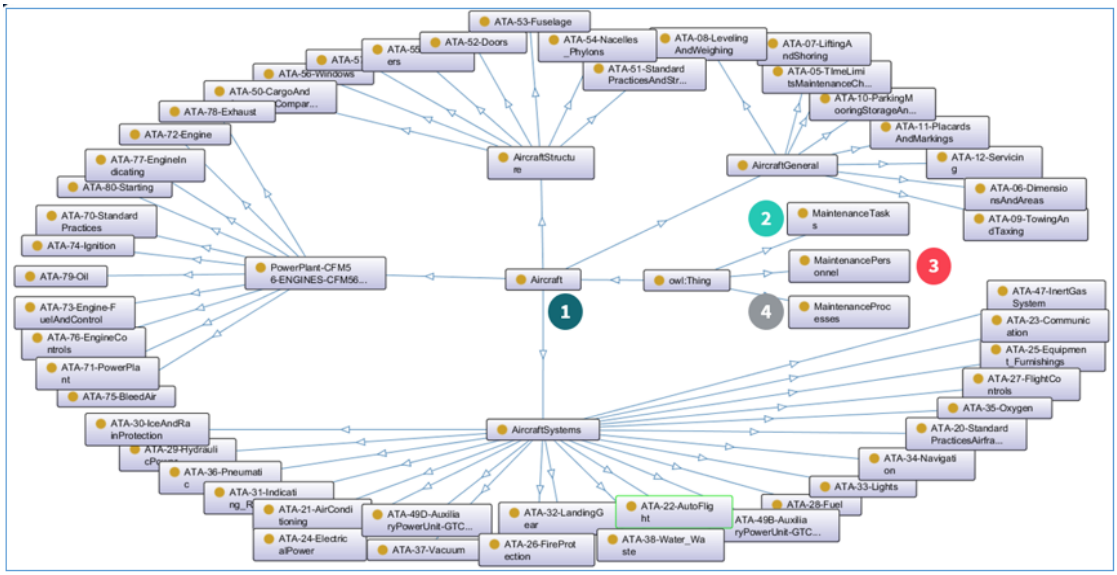


Figure 5-6 AMRO ontology

Subclasses and their relationships were carefully defined to reflect the hierarchical nature of aircraft maintenance. For example, *AChecksDocuments* is a subclass of *ScheduledMaintenanceProceduresDocuments*, which establishes that A-checks are a type of scheduled maintenance procedure. Inverse properties such as *isDocumentOf* and *requiresDocument* allow ontology to capture the bidirectional nature of the relationships between tasks and their resources or documentation. This bidirectionality is essential for tracing maintenance *activities* and their outcomes.

```
<InverseObjectProperties>
    <ObjectProperty IRI="#isDocumentOf"/>
    <ObjectProperty IRI="#requiresDocument"/>
</InverseObjectProperties>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#requiresConsumable"/>
    <Class IRI="#MaintenanceTasks"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#requiresDocument"/>
    <Class IRI="#MaintenanceTasks"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
    <ObjectProperty IRI="#requiresConsumable"/>
    <Class IRI="#MaintenanceConsumables"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#requiresDocument"/>
    <Class IRI="#MaintenanceDocuments"/>
</ObjectPropertyRange>
```

Figure 5-7 Defining properties for maintenance tasks in AMRO

While OWL is adept at defining class hierarchies and properties, first-order logic can be used to formalise the complex relationships and rules that govern the AMRO as shown in Figure 5-8, Figure 5-9 and Figure 5-10:

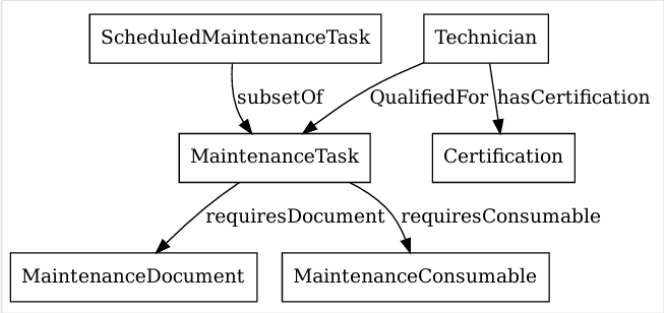


Figure 5-8 Maintenance task dependencies and technician qualifications

Maintenance Task Requirement

$$\forall x(\text{MaintenanceTask}(x) \rightarrow \exists y(\text{MaintenanceDocument}(y) \wedge \text{requiresDocument}(x,y)))$$

This asserts that each maintenance task must be associated with at least one maintenance document.

Hierarchical Structure of Maintenance Documents

$$\forall x(\text{AChecksDocuments}(x) \rightarrow \text{ScheduledMaintenanceProceduresDocuments}(x))$$

All A-check documents are scheduled maintenance procedures documents.

Maintenance Documents

$$\forall x \forall y \forall z((\text{requiresDocument}(x,y) \wedge \text{requiresDocument}(x,z)) \rightarrow y=z)$$

This ensures that maintenance documents are uniquely linked to their respective tasks.

Scheduled Task Subset

$$\text{ScheduledMaintenanceTask}(x) \rightarrow \text{MaintenanceTask}(x)$$

Scheduled maintenance tasks are a specific subset of all maintenance tasks.

Technician Qualification

$$\forall x \forall y(\text{QualifiedFor}(x,y) \rightarrow (\text{MaintenanceTechnician}(x) \wedge \text{MaintenanceTask}(y) \wedge \exists z(\text{Certification}(z) \wedge \text{hasCertification}(x,z))))$$

A technician must possess the appropriate certification for a given maintenance task.

Document and Consumable Requirements for Maintenance Task

$$\forall x(\text{MaintenanceTasks}(x) \rightarrow (\exists y(\text{MaintenanceDocuments}(y) \wedge \text{RequiresDocument}(x,y)) \vee \exists z(\text{MaintenanceConsumables}(z) \wedge \text{RequiresConsumable}(x,z))))$$

- For every maintenance task, there exists a requirement for a document or a consumable.
- **Consumables Required for Maintenance Tasks**
- $\forall m(\text{MaintennaceTask}(m) \rightarrow \exists c(\text{Consumable}(c) \wedge \text{RequiredFor}(c,m)))$
- For each maintenance task m, there exists a specific consumable c required to perform m.

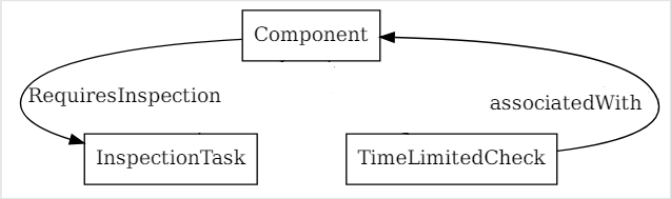


Figure 5-9 Component inspection and time-limited check dependencies

Component Inspection Requirement

$$\forall c(\text{Component}(c) \wedge \text{RequiresInspection}(c) \rightarrow \exists i(\text{InspectionTask}(i) \wedge \text{Inspects}(i,c)))$$

For any component c that requires inspection, there exists an inspection task i to inspect c.

Time Limited Checks and Component Type

$$\forall c(\text{Component}(c) \wedge \text{RequiresTimeLimitedCheck}(c) \rightarrow \exists t(\text{TimeLimitedChecks}(t) \wedge \text{AssociatedWith}(t,c)))$$

For every component c that requires a time limited check, there exists a time-limited check t that is associated with c.

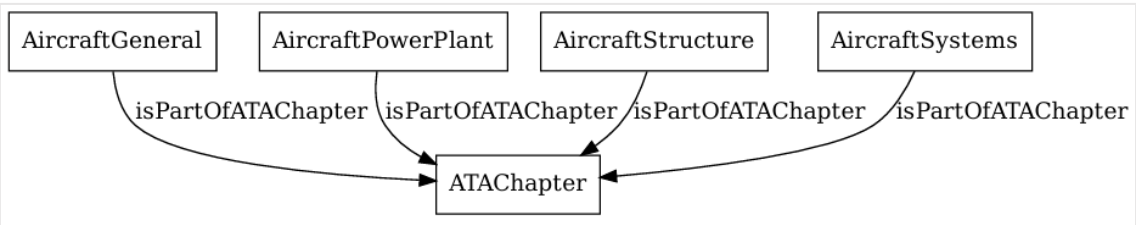


Figure 5-10 ATA chapter classification of aircraft components

ATA Chapter Classification for Aircraft Components

$$\forall x(\text{AircraftGeneral}(x) \vee \text{AircraftPowerPlant}(x) \vee \text{AircraftStructure}(x) \vee \text{AircraftSystems}(x) \rightarrow \text{isPartOfATAChapter}(x))$$

For any item x, if x is classified under AircraftGeneral, AircraftPowerPlant, AircraftStructure, or AircraftSystems, then x is part of an ATA chapter.

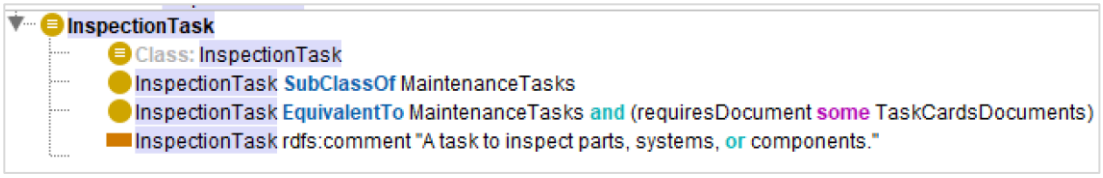


Figure 5-11 Definition and relationships of the InspectionTask class in AMRO

The InspectionTask class in AMRO represents specific maintenance activities focused on inspecting aircraft parts, systems, or components. This class is crucial for categorising and defining tasks that involve tasks such as general visual inspection.

- Class: InspectionTask is defined as a class, representing a distinct concept within the AMRO.
- SubClassOf: InspectionTask is a subclass of MaintenanceTasks. This hierarchical relationship indicates that an InspectionTask is a type of MaintenanceTask, inheriting its properties and characteristics.
- EquivalentTo: The InspectionTask class is equivalent to a combination of MaintenanceTasks and the condition that it requires some TaskCardsDocuments. This equivalence relationship ensures that InspectionTask instances meet specific criteria, including the necessity of associated task card documents.
- rdfs: The comment for InspectionTask provides a descriptive note: "A task to inspect parts, systems, or components." This annotation offers additional context and clarification the classes in AMRO..



Figure 5-12 Representation of the ManHours data property in AMRO

The ManHours data property in the ontology represents the amount of labour required for a specific task, measured in hours. This property is essential for capturing the amount of time associated with various maintenance activities, allowing for more precise planning and resource allocation.

- Domain: The domain of the ManHours property is TaskCardDocuments. This indicates that the ManHours property is applicable to instances of TaskCardDocuments, which represent detailed instructions for maintenance tasks.
- DataProperty: ManHours is defined as a data property, meaning it links instances of TaskCardDocuments to a specific datatype value, in this case, a numerical representation of hours.
- SubPropertyOf: The ManHours property is a sub-property of owl:topDataProperty. This relationship signifies that ManHours is a more specialised property under the generic topDataProperty category, inheriting characteristics common to all data properties.
- Range: The range of the ManHours property is specified as xsd:decimal. Therefore, the value assigned to ManHours is expected to be a decimal number, allowing for precise representation of labor hours, including fractional values.

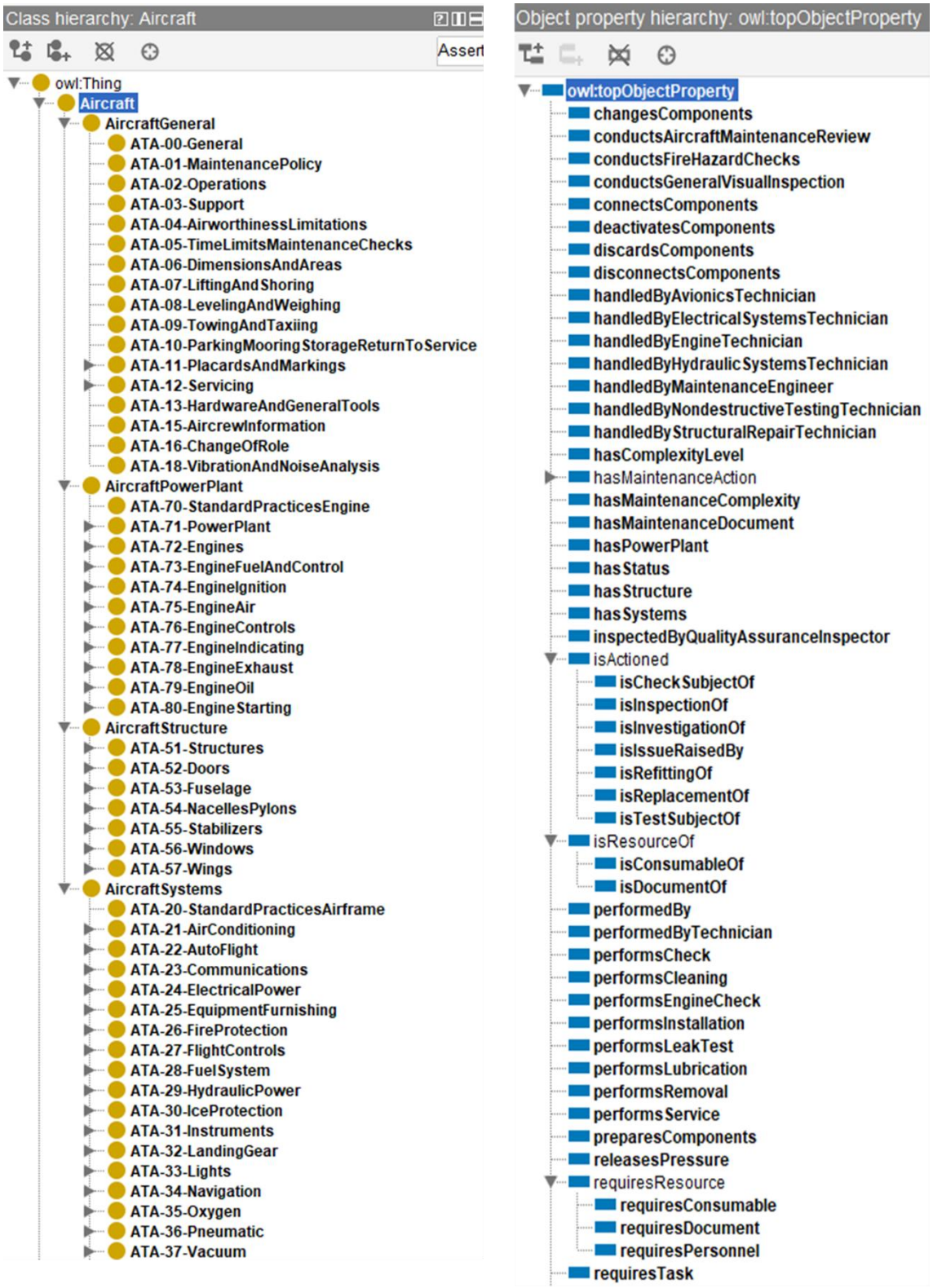


Figure 5-13 Class and object property hierarchies in AMRO

AMRO includes detailed class and object property hierarchies to model the various aspects of aircraft maintenance records as shown in Figure 5-13. These hierarchies ensure that all relevant components, systems, and maintenance actions are accurately represented and interconnected.

- Aircraft Class: The Aircraft class is a parent concept in the ontology, encompassing various aspects of an aircraft, including its general information, power plant, structure, and systems as shown in Figure 5-14.
- Subclasses: The hierarchy under the Aircraft class is organised according to the ATA chapters, which standardise the classification of aircraft parts and systems. This organisation ensures consistency and comprehensiveness in capturing maintenance documentation.

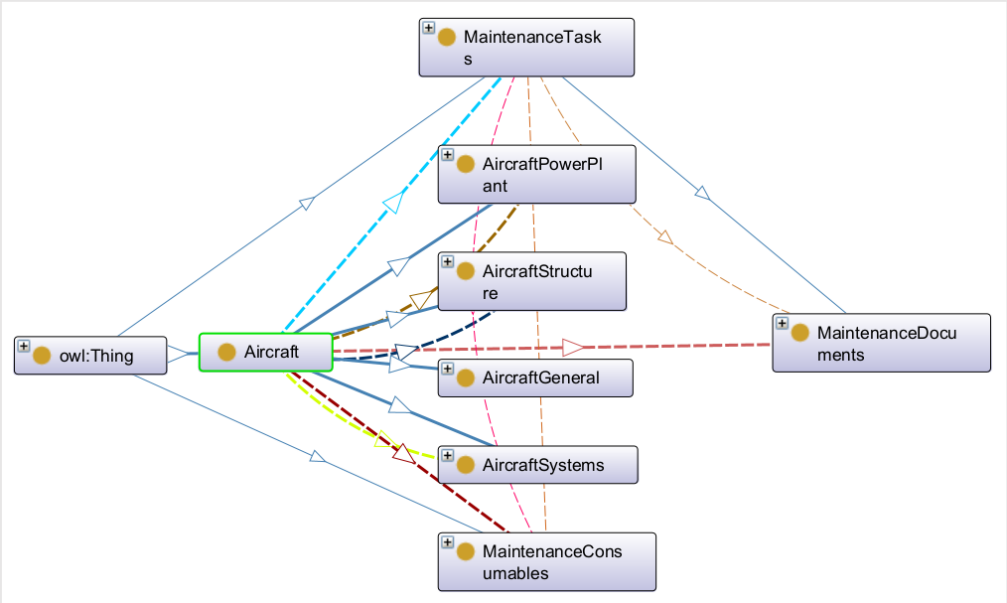


Figure 5-14 Hierarchical relationships of Aircraft Class in AMRO

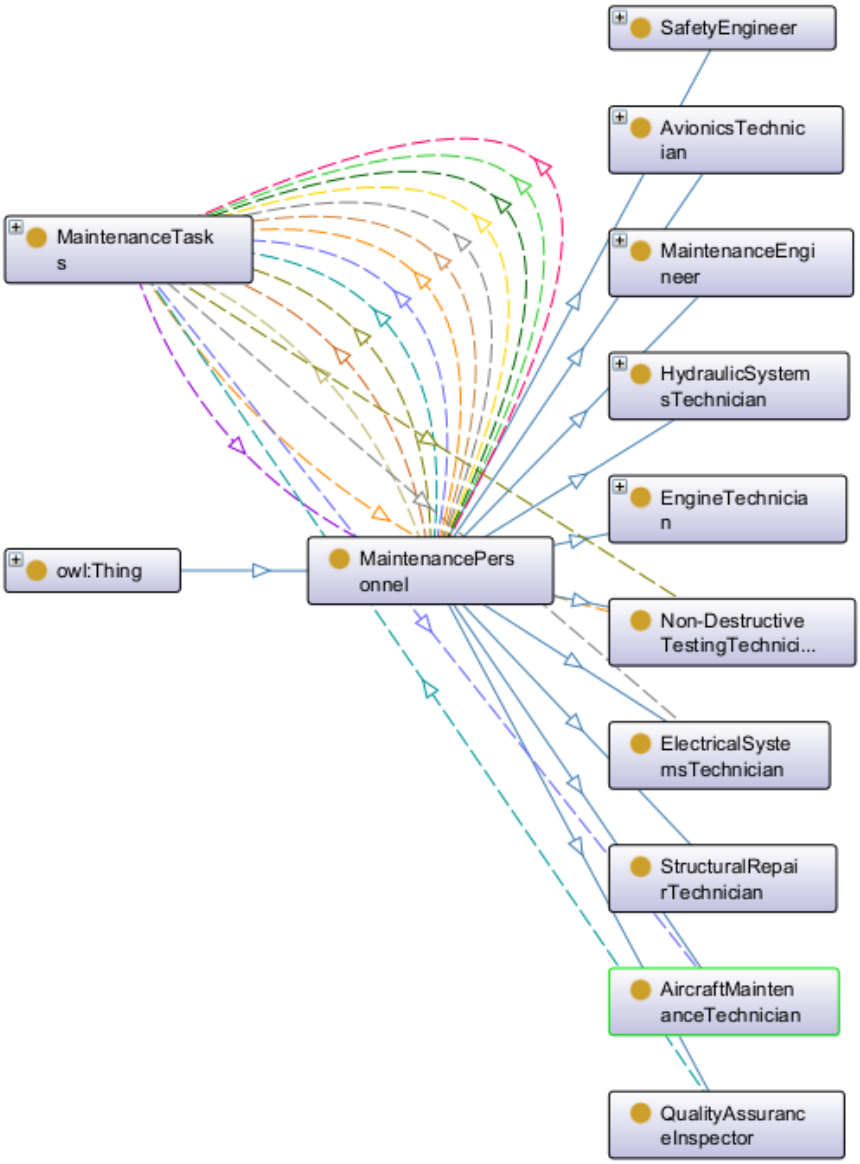


Figure 5-15 Roles and relationships of the MaintenancePersonnel Class in AMRO

Figure 5-15 shows MaintenancePersonnel class and its relationships within the AMRO. This class connects to different types of technicians and engineers responsible for specific maintenance tasks. It shows object properties that define the relationships between MaintenancePersonnel and other classes in the ontology. These properties ensure that all relevant data points about personnel and their tasks are connected logically and coherently.

MaintenancePersonnel to MaintenanceTasks: Indicates the specific maintenance tasks associated with each personnel role.

MaintenancePersonnel to Specific Technician Roles: Links personnel to their specialised roles, ensuring clear responsibility assignment and expertise utilisation.

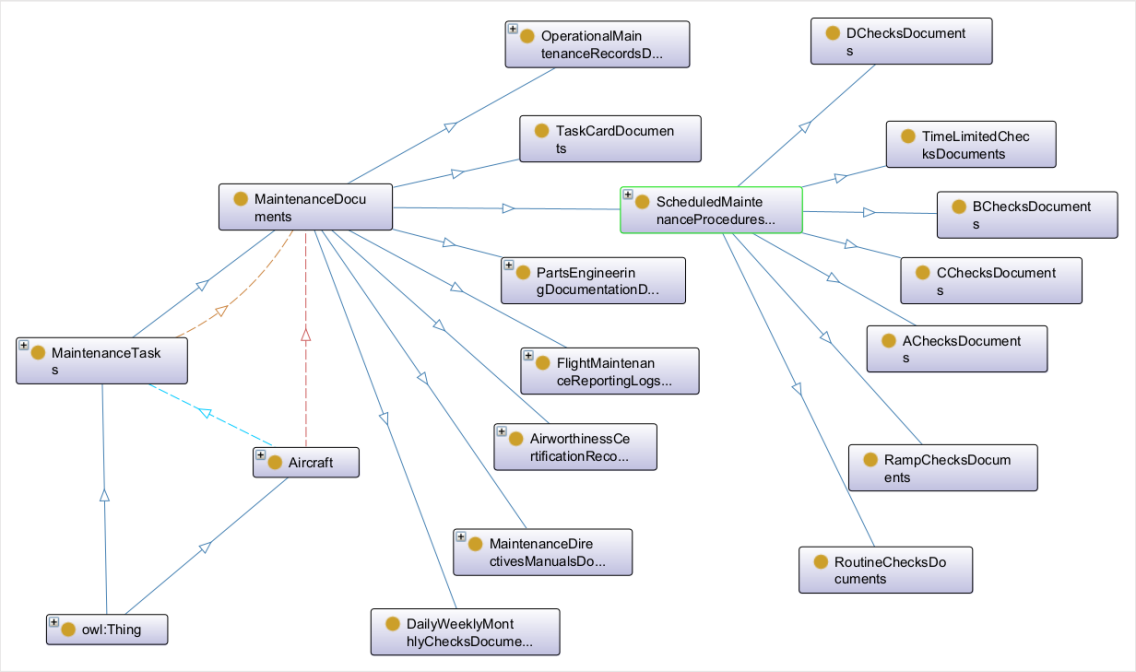


Figure 5-16 Relationships and types of MaintenanceDocuments

Figure 5-16 show different types of maintenance documents are interconnected. It is crucial for the documentation aspect of aircraft maintenance and how various documents support and record maintenance activities.

- MaintenanceDocuments to MaintenanceTasks: Indicates the specific maintenance tasks associated with each document.
- MaintenanceDocuments to Aircraft: Connects maintenance documents to the relevant aircraft.

The proposed AMRO was developed using Protege which provides a structured representation for managing maintenance records. By organising classes hierarchically and defining clear object and data properties, the ontology enhances standardisation, and efficiency. A prototype utilising AMRO is presented in Chapter 6.

6 Results Evaluation and Validation

6.1.1 Prototype Demonstration

The implementation of the ontology-based application demonstrator involved utilising Python along with various specialised libraries to manipulate ontologies, handle data, and compute semantic and syntactic similarities. The primary libraries used include owlready2 for ontology operations, pandas for data manipulation, NumPy for numerical operations, and NLTK for natural language processing tasks, specifically for computing semantic similarities.

Initially, the ontology was loaded and configured using owlready2, which facilitates the manipulation of OWL ontologies directly within Python. This step involved defining the classes and properties within the ontology that correspond to the conceptual model of the maintenance domain. This ontology setup allowed structured data to be effectively integrated and linked based on the semantic relationships.

Following the ontology configuration, the dataset underwent preprocessing where data normalization tasks are performed. This includes converting all text to lowercase, removing extraneous whitespaces and special characters, and standardising the formats for dates and numerical values. This normalization is crucial for maintaining consistency in the data comparison and similarity calculation.

The core of the implementation revolved around computing the semantic and syntactic similarities. The semantic similarity between the dataset entries and the ontology was measured using the Resnik similarity measure, which was implemented using the WordNet interface of the NLTK library. This measure assessed similarity based on the shared information content between terms. For syntactic similarity, the Levenshtein distance was calculated, which helped to identify and correct typographical errors by determining the minimum number of edits required to transform one string into another.

Once similarities were computed, the application proceeded to correct the data inconsistencies by aligning the dataset entries with the most semantically and syntactically similar terms from the ontology. These corrected data were then used to populate the ontology with instances, linking them according to the defined relationships. This step not only helped in cleaning the data, but also enriched the ontology with real-world instances, enhancing its utility and applicability.

The reasoning capabilities of owlready2 are leveraged by running the reasoner to validate the logical consistency of the ontology after the integration of new data. This reasoning step ensured that all relationships and properties were correctly applied, and that the ontology can infer new knowledge where applicable.

Finally, the corrected data, along with details of the corrections (such as edit distances or similarity scores), were compiled back into the original dataset format. This not only provided a clean version of the data but also offered transparency regarding the derivation of corrections.

6.1.2 Dataset Preprocessing

Figure 6-1 illustrates the process of converting paper-based aircraft maintenance records into a structured digital format using several techniques. It begins with the Original Paper G-DOCB Aircraft Maintenance Records (Raw Data), which are then scanned to create digitally scanned copies. These digital images were processed using Optical Character Recognition (OCR) to convert the text within the images into machine-readable text. The text data was cleaned and preprocessed using techniques such as text normalisation, lowercasing, and regex-based removal of unwanted characters. Finally, in the Data Extraction and Structuring phase, the cleaned data are organised into a structured, tabular format, as illustrated by the sample table containing columns such as TCN, Task Number, Revision, Maintenance Task, and Description.

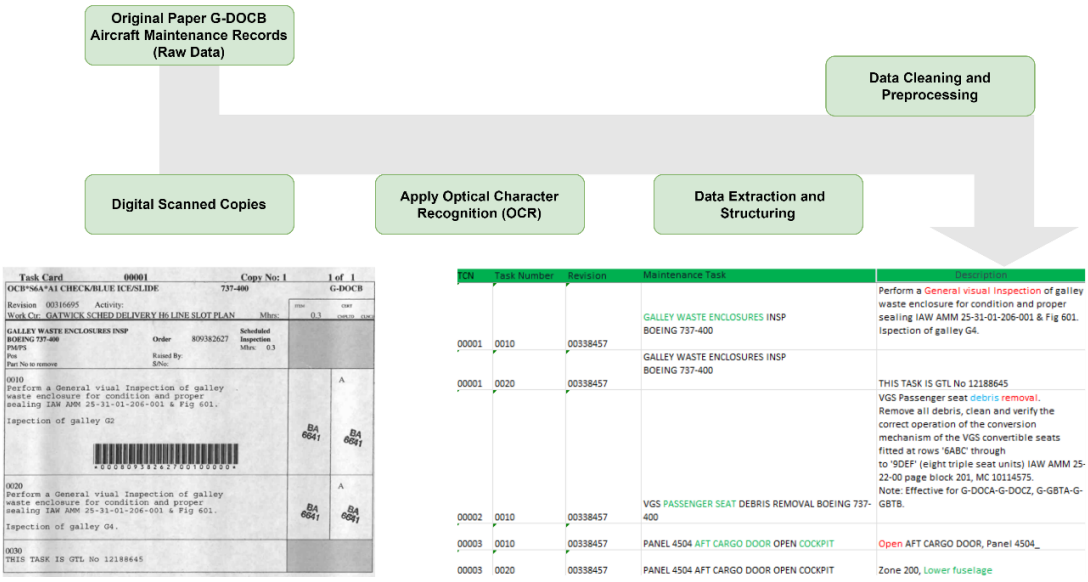
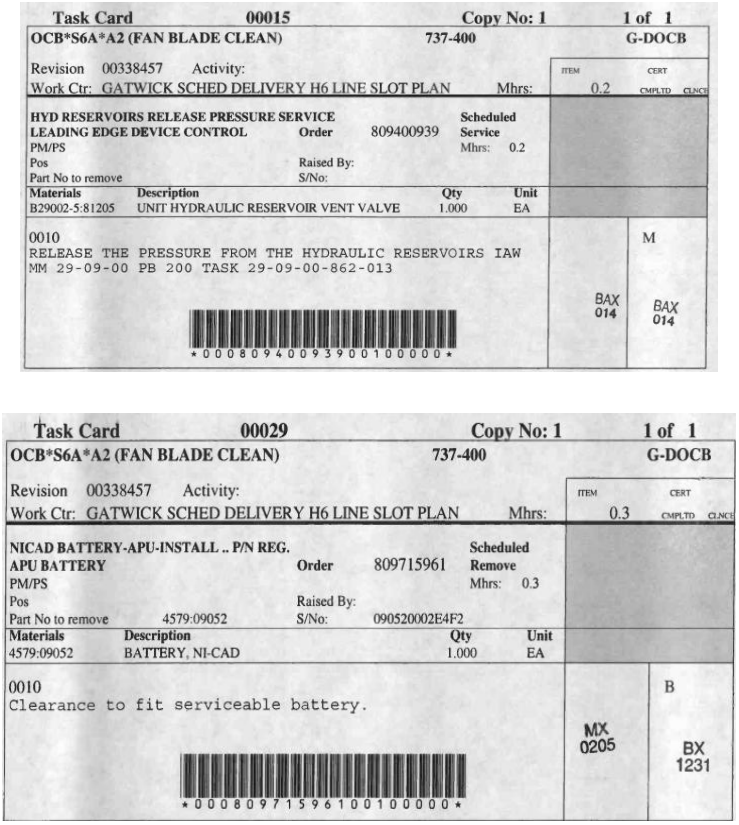


Figure 6-1 Dataset preprocessing

6.2 Resolving Semantic and Syntactic Inconsistencies in Task Cards

The section describes the practical application of AMRO to real-world aircraft maintenance data, specifically for the Boeing 737-400. Through A Check task card of G-DOCB maintenance records, we represent the application of AMRO in resolving semantic and syntactic issues within aircraft maintenance records.

The data represented in Table 6-4 demonstrate how AMRO can be used to standardise diverse forms of maintenance data. The correction of maintenance task descriptions from a raw format into a clear, semantically rich, and syntactically correct structure is presented. This demonstrates the efficacy of the ontology-based approach to achieve tangible improvements in maintenance data quality and usability.



Task Card 00015 Copy No: 1 1 of 1
OCB*S6A*A2 (FAN BLADE CLEAN) 737-400 G-DOCB
Revision 00338457 Activity:
Work Ctr: GATWICK SCHED DELIVERY H6 LINE SLOT PLAN Mhrs: 0.2
HYD RESERVOIRS RELEASE PRESSURE SERVICE
LEADING EDGE DEVICE CONTROL Order 809400939 Scheduled Service
PM/PS Mhrs: 0.2
Pos Raised By:
Part No to remove S/No:
Materials Description Qty Unit
B29002-5:81205 UNIT HYDRAULIC RESERVOIR VENT VALVE 1.000 EA
0010
RELEASE THE PRESSURE FROM THE HYDRAULIC RESERVOIRS IAW
MM 29-09-00 PB 200 TASK 29-09-00-862-013
BAX 014
M
BAX 014

Task Card 00029 Copy No: 1 1 of 1
OCB*S6A*A2 (FAN BLADE CLEAN) 737-400 G-DOCB
Revision 00338457 Activity:
Work Ctr: GATWICK SCHED DELIVERY H6 LINE SLOT PLAN Mhrs: 0.3
NICAD BATTERY-APU-INSTALL .. P/N REG.
APU BATTERY Order 809715961 Scheduled Remove
PM/PS Mhrs: 0.3
Pos Raised By:
Part No to remove 4579:09052 S/No: 090520002E4F2
Materials Description Qty Unit
4579:09052 BATTERY, NI-CAD 1.000 EA
0010
Clearance to fit serviceable battery.
MX 0205
B
BX 1231

Figure 6-2 Sample task cards

Table 6-1 Algorithm for ontology-based application

1	**Inputs** S: Source data (csv) T: OWL ontology. A set of manually curated matches between S and T
2	**Pre-processing** Convert all characters to lowercase. Remove special characters and whitespaces. Standardise date formats and numerical values.
3	**Extract essential attributes** For S, identify columns or attributes. For T, extract class, property names, and other descriptors.

4	**Information Content Computation** For each concept c in either S or T: $IC\ (c) = -\log\ (p(c))$ Where: p(c) represents the probability of occurrence of concept c in the provided Corpus. Store IC for each concept in a table for easy access.
5	**Semantic similarity (Resnik Measure)** For each attribute s in S and t in T: $sim_resnik(s,t) = IC(LCS(s,t))$ Where: LCS(s,t) represents the most specific common ancestor or Lowest Common Subsumer of concepts
6	**Syntactic similarity (Levenshtein Distance): Definition** Compute $lev(s,t)$, derived from the Levenshtein distance as represented in below: $sim_levenshtein(s,t) = 1 - \frac{lev(s,t)}{max(length(s),length(t))}$

Name Entity Recognition

The implementation of a rule-based Named Entity Recognition (NER) for processing aircraft maintenance documentation represents a significant advancement in the field of text analysis and data extraction in the aviation industry. The core objective of this model is to accurately identify and categorise key entities, such as components, defects, and actions, from technical logs, thereby facilitating a more efficient and informed maintenance decision-making process.

The foundation this implementation is built on SpaCy, a powerful and flexible natural language processing library. The implementation utilises SpaCy's PhraseMatcher in conjunction with a carefully constructed gazetteer, a repository of categorised terms and phrases relevant to aircraft maintenance

documentation. This combination enables the precise identification of domain-specific entities, a task with which generic NLP models often struggle.

Table 6-2 Algorithm for NER

Inputs	1. file_path - Path to the Excel file with unannotated records 2. gazetteer_path - Path to the JSON file with entity patterns
Outputs	1. annotated_records_path - Path to the output JSON file with annotated records 2. frequency_visualization - Visual charts of entity frequency
Begin	
Step 1:	▪ Preprocess the input data ▪ Function: PreprocessData(file_path) ▪ Description: Load the Excel file, extract relevant data, and clean the text. ▪ Output: preprocessed_text_data (List of strings)
Step 2:	▪ Load the gazetteer for entity patterns ▪ Function: LoadGazetteer(gazetteer_path) ▪ Description: Load the entity patterns from a JSON file. ▪ Output: gazetteer_data (Dictionary)
Step 3:	▪ Initialize the phrase matcher for entity recognition ▪ Function: InitializePhraseMatcher() ▪ Description: Create a new PhraseMatcher object for SpaCy. ▪ Output: matcher (PhraseMatcher object)
Step 4:	▪ Add entity patterns to the matcher ▪ Procedure: AddPatternsToMatcher(matcher, gazetteer_data) ▪ Description: Add the patterns from the gazetteer to the matcher.
Step 5:	▪ Identify entities in the preprocessed text ▪ Function: FindMatches(preprocessed_text_data, matcher) ▪ Description: Find and categorise matches in the text using the matcher. ▪ Output: categorised_matches (Dictionary)

Step 6:	▪ Annotate the original records with identified entities ▪ Function: AnnotateRecords(preprocessed_text_data, categorised_matches) ▪ Description: Annotate the preprocessed text with the identified entities. ▪ Output: annotated_records_path (String)
Step 7:	▪ Perform frequency analysis on the identified entities ▪ Function: FrequencyAnalysis(categorised_matches) ▪ Description: Count the frequency of each identified entity. ▪ Output: entity_counter (Dictionary of Counters)
Step 8:	▪ Visualise the frequency of entities ▪ Function: VisualiseFrequency(entity_counter) ▪ Description: Generate and display visual charts for entity frequency. ▪ Output: frequency_visualization (Visualization artifacts)
End	

The development of the AMRO incorporates ADOBA methodology described in Chapter 4. Task-based evaluation is central to ADOBA methodology, as outlined by Porzel and Malaka [154] and further adapted by Pittet and Barthelemy [164]. Task-based ontology evaluation is predicated on assessing the performance of an ontology against specific tasks within the application for which it was designed. This section discusses the evaluation of the AMRO using a task-based approach within the ADOBA methodology.

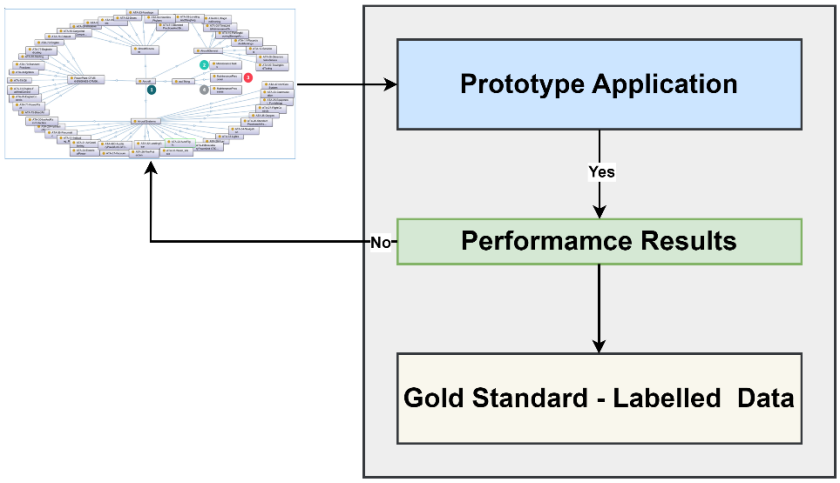


Figure 6-3 Task-based evaluation setup

The evaluation focused on specific maintenance tasks such as "LEAK CHK ELCT MTR HYD PMP" and "INSP LEFT ENGINE INLET." The original and corrected forms of these tasks were compared to assess the effectiveness of the ontology in improving the data consistency. The evaluation involved calculating semantic and syntactic similarities for each task, with results indicating the degree of improvement post-correction. The edit distances and match indicators provided quantitative measures of these improvements.

The detailed breakdown of these results is presented below, highlighting specific tasks, their original and corrected forms.

Table 6-3 Task-based evaluation steps

S/N	Step	Description
1	Task Definition	The evaluation tasks were derived from actual A-Check maintenance task cards of Boeing 737-400. These tasks were chosen because they require accurate semantic interpretation and syntactic correction, making them suitable benchmarks for ontology performance.
2	Ontology	Aircraft Maintenance Records Ontology (AMRO)
3	Prototype Application	Maintenance records were pre-processed to standardise the text. This included converting text to lowercase, removing extraneous characters, and normalising the date formats and numerical values. Such preprocessing ensures consistency and prepares the data for subsequent semantic and syntactic analyses. Semantic similarity between maintenance tasks and the ontology was computed using the Resnik measure via the WordNet interface of the NLTK library. Syntactic similarity was calculated using the Levenshtein distance, identifying and correcting typographical errors by measuring the minimum edits needed to transform one string into another.
4	Gold Standard and Evaluation Metrics	Manually label matches between source data and ontology to establish a reference set of correct annotations. These matches represent the expected output, which is free of semantic and syntactic inconsistencies. Quantitative metrics, including precision, recall, and F-measure, were used to assess performance.

Table 6-4 Selected results of Corrected Inconsistencies in maintenance task cards for the A1 Check with similarity scores

S/N	Original Maintenance Task	Preprocessed Maintenance Task	Corrected Maintenance Task	Labelled Data	Edit Distance	Match Indicator
1	LEAK CHK ELCT.MTR.HYD PMP.CASE DRN FILT HYDRAULIC SYSTEM	LeakChkElctmtrhydPmpcaseDrnFiltHydraulicSystem	LeakCheckElectricMotorHydraulicPumpcaseDrnFiltHydraulicSystem	LeakCheckElectricMotorHydraulicPumpcaseDrnFiltHydraulicSystem	25	1
2	INSP. LEFT ENGINE INLET ENGINE NR 1 L/H	InspLeftEngineInletEngineNr1Lh	InspectLeftEngineInletEngineNr1Lh	InspectLeftEngineInletEngineNr1Lh	10	1
3	EQUIP COOLING SYST CHK VLVS - OP CHECK AIR SYSTEM	EquipCoolingSystChkVlvs-OpCheckAirSystem	EquipCoolingSystemCheckValves-OpCheckAirSystem	EquipCoolingSystemCheckValves-OpCheckAirSystem	15	1
4	GEN.VISUAL INSP.(EXT.)OF AFT CARGO DOOR CARGO DOORS	GenvisualInsp(ext)ofAftCargoDoorCargoDoors	GeneralVisualInspection(ext)ofAftCargoDoorCargoDoors	GeneralVisualInspection(ext)ofAftCargoDoorCargoDoors	18	1
5	LEAK CHK. APU SURGE CONT.VLV.AIR FILTER. Scheduled AIR	LeakChkApuSurgeContvlvairFilterScheduledAir	LeakCheckApuSurgeContValveairFilterScheduledAir	LeakCheckApuSurgeContValveairFilterScheduledAir	13	1
6	NE4930303-pax o2 unit used.BOEING 737-400	Ne4930303-paxO2UnitUsedBoeing737-400	Ne4930303-PassengerOxygenUnitUsedBoeing737-400	Ne4930303-PassengerOxygenUnitUsedBoeing737-400	17	1
7	INSPECT LEFT ENG HPT BLADES	InspectLeftEngHptBlades(borescopEngineNr1Lh	InspectLeftEngineHptBladesborescopEngineNr1Lh	InspectLeftEngineHptBlades(borescopEngineNr1Lh	11	1

	(BORESCOP ENGINE NR 1 L/H					
8	FUSE SKIN AT NO3 VHF ANTENNA -EXT DVI BOEING 737-400	FuseSkinAtNo3VhfAntenna-extDviBoeing737-400	FuselageSkinAtNo3VhfAntenna-extDviBoeing737-400	FuselageSkinAtNo3VhfAntenna-extDviBoeing737-400	12	1
9	IGN SYSTEM - ENG NO! - ENGINE START ENGINE NR 1 L/H	IgnSystem-EngNo!-EngineStartEngineNr1Lh	IgnitionSystem-EngineNo!-EngineStartEngineNr1Lh	IgnSystem-EngineNo!-EngineStartEngineNr1Lh	16	1
10	EXTERNAL ZONAL (GV) FWD GALLEY SVC DOOR SERVICE DOORS	ExternalZonal(gv)FwdGalleySvcDoorServiceDoors	ExternalZonal(gv)FwdGalleyServiceDoorServiceDoors	ExternalZonal(gv)FwdGalleyServiceDoorServiceDoors	12	1
11	FDR BEACON ASSY - DATE LIFE CHK INDEPENDENT INSTRUMENTS	FdrBeaconAssy-DateLifeChkIndependentInstruments	FdrBeaconAssembly-DateLifeCheckIndependentInstruments	FdrBeaconAssembly-DateLifeCheckIndependentInstruments	14	1
12	TRAILING EDGE FLAPS TO UP POSN RETRACT FLAPS	TrailingEdgeFlapsToUpPosnRetractFlaps	TrailingEdgeFlapsToUpPositionRetractFlaps	TrailingEdgeFlapsToUpPositionRetractFlaps	12	1
	NO2 ENG BORESCOPE PORT PLUG S15 REMOVE ENGINE NR 2 R/H	No2EngBorescopePortPlugS15RemoveEngineNr2Rh	No1engborescopeportplugsi5installEnginenrilh	No2EngineBorescopePortPlugS15RemoveEngineNr2Rh	13	0
	OIL COOLER BYPASS VALVE RH ENG INSPECT AC GENERATION	OilCoolerBypassValveRhEngInspectAcGeneration	OilCoolerBypassValveRhEngInspectAcGeneration	OilCoolerBypassValveRightHandEngineInspectAcGeneration	9	0

2A2 CHECK - MSG3 Scheduled BOEING 737-400	2a2Check-Msg3ScheduledBoeing737-400	AircraftReviewScheduledBoeing737-400	2a2Check-MaintenanceSteeringGroup3ScheduledBoeing737-400	15	0
NO1 ENG BORESCOPE PORT PLUG S15 REMOVE ENGINE NR 1 L/H	No1EngBorescopePortPlugS15RemoveEngineNr1Lh	No1engborescopeportplugsi5installEnginenrilh	No1EngineBorescopePortPlugS15RemoveEngineNr1Lh	11	0
PNL 8207(ACCESS DOOR 0/B T/E FLAP) CLOSE FLAPS	Pnl8207(accessDoor0bTeFlap)CloseFlaps	Panel8707(accessDr0bTeFlap)closeFlaps	Panell8207(accessDoor0bTeFlap)CloseFlaps	10	0

Table 6-4 shows the result performance of the corrected inconsistencies for the maintenance task of an A1 Check by using AMRO and Levenshtein Distance measures. The Original Maintenance Task shows a wide variation of words commonly used by technicians, reflecting the practical usage of onsite operations but posing significant challenges for standardisation and automation. These initial entries are critical as they represent the raw data input into the system, which must be accurately interpreted and standardised to maintain operational integrity and regulatory compliance.

The ***Preprocessed Maintenance Task*** provides a first look into the normalisation process. Here, variations and abbreviations are expanded, and special characters and stop words are removed to create the initial preprocessed text for further processing. This step is crucial for reducing variability and preparing the text for semantic analysis. While the preprocessing does not completely standardise the language, it significantly reduces the complexity of the text, which is beneficial for the subsequent semantic mapping. The preprocessing reflects an intermediate state that balances the need to retain original meaning while simplifying the structure for further refinement.

The ***Corrected Maintenance Task*** shows the results after applying the AMRO model. This step transforms the preprocessed tasks into a form that adheres to standard terminology and formats predefined in ontology. The corrections involve more than simple text edits; they reformat the data according to standardised maintenance terminologies, which is essential for ensuring that the tasks are both technically accurate and universally understandable within the industry. For instance, changing “LeakChk” to “LeakCheck” and inserting necessary changes that ensures clarity and reduces the risk of misinterpretation. This is crucial for further utilisation of the maintenance records for data integration or advanced data applications.

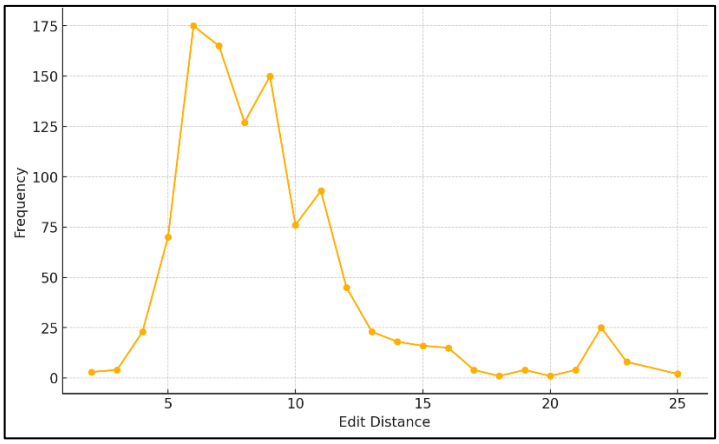


Figure 6-4 Frequency of unique distances

Figure 6-4 shows the occurrence of each unique edit distance within the dataset, highlighting the commonality of specific edit adjustments needed in the maintenance tasks. For instance, the most frequently occurring edit distance is 6, appearing 175 times, which indicates that minor adjustments are most common. The second most occurrence is the edit distance of 7, recorded 165 times, followed closely by distances of 9 and 8, appearing 150 and 127 times respectively. This suggests a concentration of edit needs in the lower range, with a steep decline as the edit distance increases. Additionally, 11 occurs 93 times, showing a significant drop in frequency. As the edit distances grow larger, their occurrences become increasingly sparse, highlighting that extensive modifications are much less common. For example, edit distances of 22 and 25 appear only 25 and 2 times respectively, demonstrating the rarity of needing such substantial corrections. This pattern illustrates the typical scope of corrections applied, with a clear emphasis on less extensive edits dominating the correction process which indicates the high performance of the proposed model.

The analysis of the dataset shows that the corrections performed less on records containing specific part numbers or component numbers. These elements often carry no semantic weight outside their referential context in the aviation industry, thus posing a challenge for ontology-based systems that rely on contextual understanding and natural language processing. For instance, the entries such as "NO2 ENG BORESCOPE PORT PLUG S15 REMOVE ENGINE NR 2 R/H" in Table 6-4 did not match correctly, possibly due to the semantic ambiguity or

incorrect parsing of technical identifiers such as 'S15' and 'NR 2 R/H', which are critical but not semantically rich.

The ***Edit Distance*** and ***Match Indicator*** quantitatively and qualitatively evaluate the effectiveness of the corrections. 'Edit Distance' measures how many edits are needed to transform the preprocessed text into the standardised format, providing an objective metric of change magnitude. A smaller distance would typically indicate minor modifications, suggesting that the preprocessing was highly effective, whereas a larger distance points to significant initial inconsistencies. The 'Match Indicator', showing values of 1 for successful alignment with 'Labelled Data', demonstrates AMRO model effectiveness in achieving a high degree of semantic accuracy. This validation confirms that the corrected tasks align well with established standards in the model.

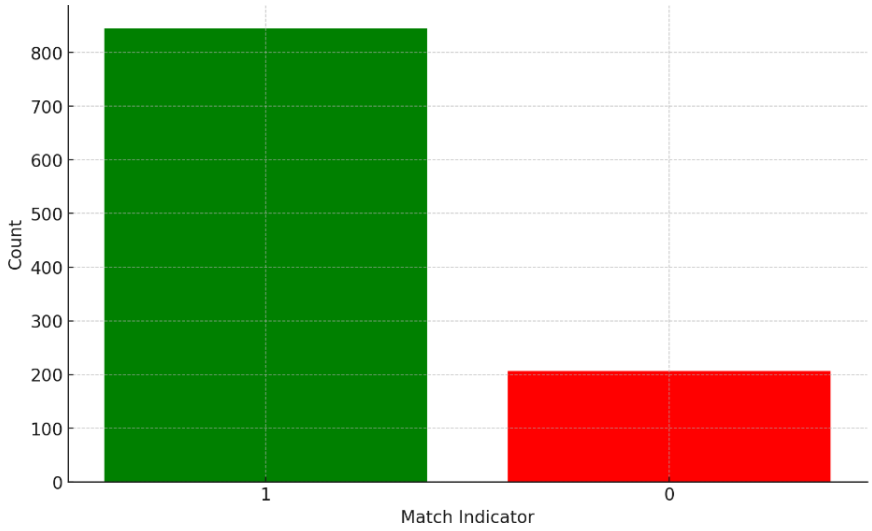


Figure 6-5 Matching indicators

Critical evaluation of the maintenance record correction process for a Boeing 737-400 reveals a complex landscape of success and challenges inherent in applying semantic technologies to technical documentation. Figure 6-5contains a total of 1052 records analysed, 840 were corrected successfully, yielding a success rate of approximately 80% while 212 which is 20% were not corrected. This indicates a robust performance of the Aircraft Maintenance Record Ontology (AMRO) model in standardising maintenance tasks but also highlights areas for

improvement, particularly where part numbers and component identifiers are involved.

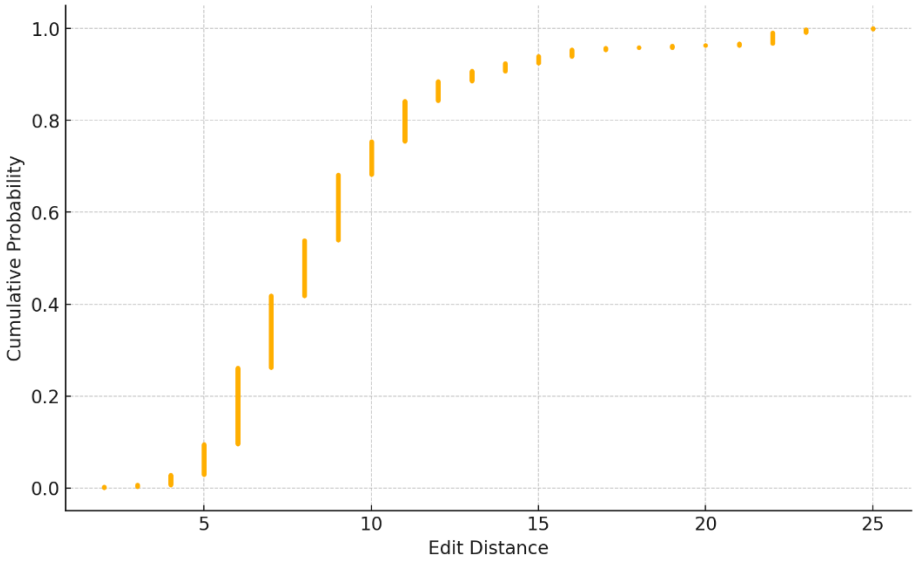


Figure 6-6 Cumulative distribution function of edit distances

The cumulative distribution function of the edit distances illustrates the cumulative percentage of records at or below each edit distance, providing insights into the overall distribution and concentration of editing needs within the dataset. The CDF plot reveals that approximately 25% of the tasks require an edit distance of 6 or less, highlighting that a significant fraction of tasks need only minimal corrections. This value rises sharply to about 50% at an edit distance of 8, indicating that half of the tasks are corrected with moderate adjustments. The curve continues to ascend steadily, reaching 75% at an edit distance of 10, showcasing that most tasks fall within a relatively tight range of edit distances. As the curve approaches the upper tail, it gradually flattens, signifying that only a small percentage of tasks require more substantial edits. Specifically, the plot reaches nearly 100% at an edit distance of 25, confirming that very few tasks exceed this level of correction. This CDF effectively underscores the extent to which the majority of maintenance tasks are clustered around lower edit distances, with only a minor proportion extending into higher levels of editing complexity.

The Match Indicator column provides a straightforward assessment of the ontology's performance, where a '1' indicates a successful match and a '0' suggests a failure. This binary outcome helps in rapidly assessing the effectiveness of the corrections but does not account for the partial successes or near-misses that might provide valuable insights for refining the AMRO. For example, the entry with the original task "2A2 CHECK - MSG3 Scheduled BOEING 737-400" was pre-processed to "AircraftReviewScheduledBoeing737-400" and corrected to "2a2Check-MaintenanceSteeringGroup3ScheduledBoeing737-400". Despite the semantic enhancement, it received a Match Indicator of '0', suggesting a mismatch with the labelled data, perhaps due to the complexity or specificity of the task.

Critically, while the high success rate of corrections is significant, the incorrect matchings indicate a need for the ontology to better handle the peculiarities of technical jargon and component identifiers. These elements are often crucial for the accurate execution of maintenance tasks and can significantly impact safety and operational efficiency if not correctly interpreted. Enhancements in the ontology could involve deeper integration with component databases or the development of context-aware parsing algorithms that better understand and process alphanumeric identifiers.

Table 6-5 Summary of Matches

S/N	Metric	Values
1	Total Records	1052
2	Correct Matches	859
3	Incorrect Matches	193
4	Mean Edit Distance	8.97
5	Median Edit Distance	8.00
6	Mode Edit Distance	6.00

7	Standard Deviation of Edit Distances	3.79

Table 6-5 shows the overview of the result validation which includes the essential statistics from the dataset of 1,052 maintenance tasks. It shows that 845 tasks were accurately aligned with the predefined format in the AMRO ontology, denoting a successful correction rate, whereas 207 tasks failed to meet the criteria, reflecting a need for process improvements. The average editing requirement is quantified by a mean edit distance of 8.97, closely aligned with the median of 8, indicating a central tendency toward moderate editing. The most common edit distance observed is 6, as highlighted by the mode, suggesting that six corrections per task is a frequent occurrence. The spread of edits, with a standard deviation of 3.79, implies a moderate variation in the number of corrections needed across different tasks. Overall, these statistics demonstrate a generally effective correction process but also reveal a significant spread in the complexity of the corrections required.

Table 6-6 Result of proposed model across selected A Check maintenance task inconsistencies

S/N	Maintenance Task	Precision	Recall	F-Measure	Support
1	AftLavatoryDrainValve-LeakTestBoeing737-400	1.00	1.00	1.00	5
2	AirExhaustMufflerInstallBoeing737-400	1.00	1.00	1.00	1
3	LhlwrwingslataccessPanel6233closeLeadingEdgeDeviceControl	0	0	0	3

4	CrewSmokeHoods-DateLifeCheckPortableOxygen	1.00	1.00	1.00	2
5	EquipCoolingSystemCheckValves-OpCheckAirSystem	1.00	1.00	1.00	2
6	Ext-InspectFwdEntryDoorstopPadFuselage	1.00	1.00	1.00	2
7	FanBladesNo1EngineCleanEngineNr1Lh	1.00	1.00	1.00	2
8	No1EngineBorescopePortPlugS14InstallEngineNr1Lh	0	0	0	3
9	GeneralVisualInspection(ext)OfAftCargoDoorCargoDoors	0.50	1.00	0.67	1
10	HydraulicSystemsRestoreBoeing737-400	1.00	1.00	1.00	2
11	InspectLeftEngineIgnitorLeadsEngineNr1Lh	1.00	1.00	1.00	1
12	LandingGearControlLeverCleaningBoeing737-400	1.00	1.00	1.00	0.50
13	NicadBattery-Apu-RemoveApuBattery	1	1.00	1	1
14	OutboardFlapBearingLeftWingFlaps	1.00	1.00	1.00	7
15	Panel1301OxygenServiceDoorCloseCrewOxygen	0	0	0	2

16	ShroudDrains-Apu-InspectionApuInstallation	1.00	1.00	1.00	1
17	VisualCheckLeftEngineInletCowlEngineNr1Lh	1.00	1.00	1.00	1
18	UndercarriageGroundLocksInstallBoeing737-400	1.00	1.00	1.00	2
19	LandingGearControlLeverCleaningBoeing737-400	1.00	1.00	1.00	5
20	InspectApuOilMagneticChipDetectorApuOil	1.00	1.00	1.00	3

Table 6-6 shows the performance result of the proposed model across selected heterogeneous Boeing 737-400 aircraft A Check Maintenance Tasks. This process utilises AMRO model designed to address inconsistencies based on similarity measures. The analysis consists of a summary table containing 20 randomly selected tasks and their respective precision, recall, and F-measure metrics to gauge the effectiveness of the model.

These metrics offer a comprehensive view of the model's performance. Notably, 258 distinct maintenance tasks with a total support of 784 achieved scores of 1.0 for precision, recall, and F1-score, demonstrating the accuracy of AMRO in correcting and capturing the dataset. High-frequency tasks such as "AftEntryDoorDrainPath-CleanWasteWaterSystem" and "AftLavatoryDrainValve-LeakTestBoeing737-400" illustrate the model's robustness, as their substantial support correlates with superior performance metrics.

However, the result shows 84 distinct maintenance tasks with total support of 191 have 0.0 scores across all metrics, such as Lhlwr Wing Slat Access Panel 6233 Close Leading Edge Device Control and No.1 Engine Borescope Port Plug S14 Install. The presence of numerical values in these tasks appears to correlate with their zero precision, suggesting that the ontology model struggles to accurately

correct maintenance tasks containing numerical values. This issue indicates significant gaps in the current similarity measures and highlights the need for further refinement of the AMRO model to handle such cases effectively. Also, 36 distinct maintenance tasks with total support of 77 maintenance show moderate scores, with precision ranging from 0.5 to 0.75, reflecting the success in identifying and correcting the maintenance tasks but also highlighting the need for further refinement of AMRO Model.

The high accuracy of the proposed AMRO model is due to the in-depth domain knowledge representation for aircraft maintenance records. However, to enhance overall performance, addressing data standardisation and adhering to universal requirements [3] such ATA SPEC 2500 is crucial.

Table 6-7 Overall result of the model

Metrics	Accuracy	Precision	Recall	F-Measure	Support
	0.82				1052
Macro Average		0.73	0.78	0.74	1052
Weighted Average		0.78	0.82	0.79	1052

Table 6-7 shows a comprehensive summary of the overall performance metrics for the proposed AMRO model, which was used to correct the heterogeneous A-check maintenance records. The metrics assessed included accuracy, precision, recall, F-measure, and support, at both the macro and weighted average levels. This analysis provided key insights into the overall efficacy of the AMRO model.

The overall accuracy of the AMRO model was 0.82, indicating 82% of the successful correction of A Check maintenance task cards. This suggests a high level of reliability in the model's ability to perform the intended tasks. The precision, recall, and F-measure metrics offer a more in-depth evaluation of model performance. The macro-average precision, recall is 0.78 and F-measure

were 0.73, 0.78, and 0.74, respectively. These macro averages suggest that on an unweighted average across all tasks, the model performs reasonably well. Since precision is slightly lower than recall, this implies that during the data correction task, the AMRO model produces more false corrections (incorrectly identified as correct) than it misses actual errors (fails to correct).

In terms of weighted averages, the precision was 0.78, recall was 0.82, and F-measure was 0.79. These weighted metrics consider support, which is the number of actual occurrences of each class, thus giving more weight to more frequent tasks. The improvement in these weighted metrics over the macro averages suggests that the model performs better on more common maintenance tasks, which is advantageous for the operational reliability. A higher weighted recall indicates that the AMRO model is particularly adept at correcting maintenance tasks that are more frequent. The analysis of these metrics indicates that, while the model is generally effective, there are notable discrepancies between the macro and weighted averages. This suggests a performance disparity between the frequent and infrequent tasks. However, less-common tasks in the A-check maintenance task cards are also noted to include numerical values which add to the complexity of semantic and syntactic corrections using the proposed AMRO model. Table 6-8 contains a comparative analysis proposed AMRO model which is assessed against 3 other ontology models using precision, recall and F-measure metrics.

Table 6-8 Comparison between AMRO model and existing ontology literature

Authors/Models	Dataset	Precision	Recall	F-Measure
Vennesland, Audun, et al. [165]	NASA Air Traffic Management Ontology (ATMONTO)	100.00%	31%	48%

Pirró, Giuseppe, and Domenico Talia. [166]	Ontology Alignment Evaluation Initiative (OAEI)	82.00%	83.00%	82.00%
Jung, Jason J. [167]	Heterogeneous Information Systems	79.00%	73.00%	76.00%
Proposed AMRO Model	Boeing 737-400 Maintenance Records (A Check)	78.00%	82.00%	79.00%

The AMRO model demonstrates a balanced result in terms of recall, with a score of 82.00%, and F-measure, with a score of 79.00%, although its precision of 78.00% is marginally lower than the other models. Vennesland, Audun et al. attained the highest precision at 100.00%, but the recall was extremely low at 31%, resulting in a lower F-measure of 48%. This suggests that the model demonstrates a good capacity for accurately identifying relevant instances but poorly capture all relevant instances. Pirro and Domenico achieved balanced scores across all metrics with a precision, recall and F-measure of 82%, 83% and 82%, respectively. This consistency indicates a well-rounded and effective model capable of identifying and capturing relevant instances. Jung attained precision 79.00% and recall 73.00%, resulting in an F-measure of 76.00%, suggesting a slight imbalance but overall robust performance in identifying and capturing relevant instances. On the other hand, AMRO model's advantage is in its balanced approach, particularly in recall, where it outperforms Jung's model and closely matches Pirro's model. This balance is important for practical ontology-based applications where both precision and recall are important.

The substantial difference in recall for Vennesland, Audun et al. highlight a critical issue; while the model is precise, it misses most relevant instances, reducing its practical applicability in scenarios requiring comprehensive data coverage. For Pirro and Domenico, the high and balanced metrics across precision, recall, and

F-measure suggest a robust model that can be relied upon for accurate and comprehensive ontology mapping. Jung shows a moderate trade-off between precision and recall, indicating reasonable reliability but with areas of improvement in capturing all relevant instances. The AMRO model's metrics addresses imbalance in the performance metrics for precision and recall, achieving a good F-measure. This balance is crucial for its intended application in correcting data inconsistencies in aircraft maintenance records, where both accuracy and completeness are essential. The practical applicability of these models in data correction and ontology alignment is significant. The proposed AMRO model's balanced metrics make it highly suitable for ensuring quality and reliability in data correction tasks within aircraft maintenance records. High precision ensures accurate corrections, while high recall ensures that most inconsistencies are identified and addressed. Pirro's and Jung’s models are also practically applicable but may be suited to different contexts. Pirro's model, with its high balance, is ideal for general ontology alignment tasks, ensuring both accuracy and comprehensiveness. Jung's model, with slightly lower recall, could be more appropriate for environments where precision is prioritised over completeness.

The AMRO model presents a balanced performance in terms of precision, recall, and F-measure, making it a suitable for correcting inconsistencies in aircraft maintenance records. Its performance closely aligns with the high precision and recall of other models evaluated, indicating its ability to effectively identify and correct inconsistencies with both accuracy and comprehensiveness. The comparison of these models indicates the significance of maintaining a balanced performance in ontology models, particularly in domains that demand high data quality and reliability such as aircraft maintenance, where both precision and recall have direct implications on operational safety and efficiency.

Table 6-9 NER Results for A Check maintenance task Cards with and without AMRO

Entity Type	Entity	Occurrence Count with AMRO	Occurrence Count without AMRO	Difference in Occurrence Count (AMRO – Without AMRO)
Actions	check	88	66	22
Components	door	251	240	11
Actions	service	84	78	6
Components	leak check	13	8	5
Components	fuselage	37	34	3
Components	horiz stab lh sliding seal	6	3	3
Actions	install	105	102	3
Components	cooling system	2	0	2
Components	fwd entry door	21	19	2
Actions	general visual inspection	2	0	2
Components	aft cargo door	8	7	1
Components	hydraulic reservoirs	2	1	1
Components	fwd cargo door	6	7	-1
Components	eng - lh fan cowl	3	12	-9
Components	eng - rh fan cowl	3	12	-9
Components	eng borescope port plug	3	16	-13

Table 6-9 illustrates the results of Named Entity Recognition (NER) applied to the A Check Maintenance Task cards, comparing outcomes with and without the use of the AMRO model to correct inconsistencies. The NER model shows several notable drawbacks without the use of AMRO. The Occurrence Count without

AMRO shows discrepancies and inconsistencies in entity recognition, highlighting the challenges faced by the model with non-standardised data. The higher error rate in this scenario is indicative of the model's difficulty in accurately classifying entities due to variations in terminology, abbreviations, and format inherent in the maintenance records. For example, entities such as "eng - lh fan cowl" and "eng borescope port plug" display significant disparities in occurrence counts compared to maintenance record using AMRO. These inconsistencies indicate the limitations of the NER model in effectively identifying and categorising entities without a standardised maintenance record.

The use of AMRO in correcting the A Check Maintenance Task cards significantly enhances the performance of the NER model, as evidenced by the Occurrence Count with AMRO and the Difference in Occurrence Count. The utilisation of AMRO shows clear improvements in accuracy, with increased occurrence counts for numerous entities, such as "check," "door," and "service." This improvement indicates a more comprehensive capture of relevant entities using the NER model. Furthermore, the role of AMRO in standardising terminology and abbreviations facilitates a more consistent and reliable entity recognition. This standardisation reduces the possibility of entities being overlooked or misclassified, thereby addressing the issue of wrong identification. Entities previously unrecognised, such as "cooling system" and "general visual inspection", are now accurately identified, although in small numbers, reflecting the enhanced precision offered by AMRO. The application of these results within the aircraft maintenance domain holds significant potential for enhancing the operational efficiency and reliability. Accurate NER facilitated by AMRO can streamline the processing of maintenance records by automating the categorisation of tasks, components, and actions. In addition, the improved accuracy of entity recognition enables the development of predictive maintenance strategies. Furthermore, the use of the AMRO system ensures compliance with regulatory standards, streamlining documentation, and auditing processes. This enhancement contributes to improved safety records and adherence to regulations. In knowledge management, the consistent use of an ontology, such as AMRO, aids in the training of new personnel by providing a

clear, standardised vocabulary of terms and procedures. This consistency is essential for effective knowledge transfer and operational continuity, ensuring that the maintenance practices are uniformly understood and applied.

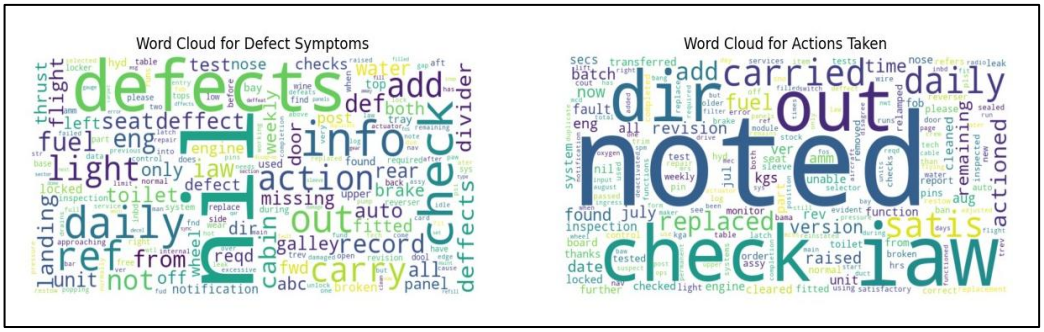


Figure 6-7 Word Cloud of the most frequent used words for Defects and Actions Taken in the dataset

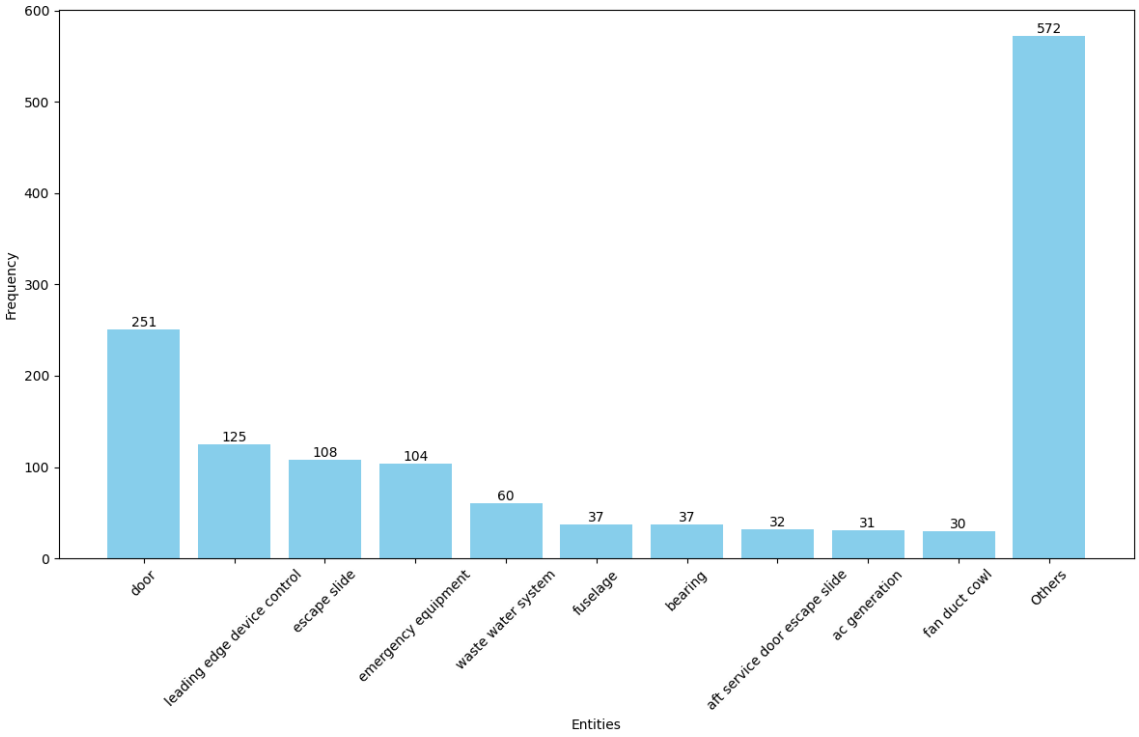


Figure 6-8 Component entities

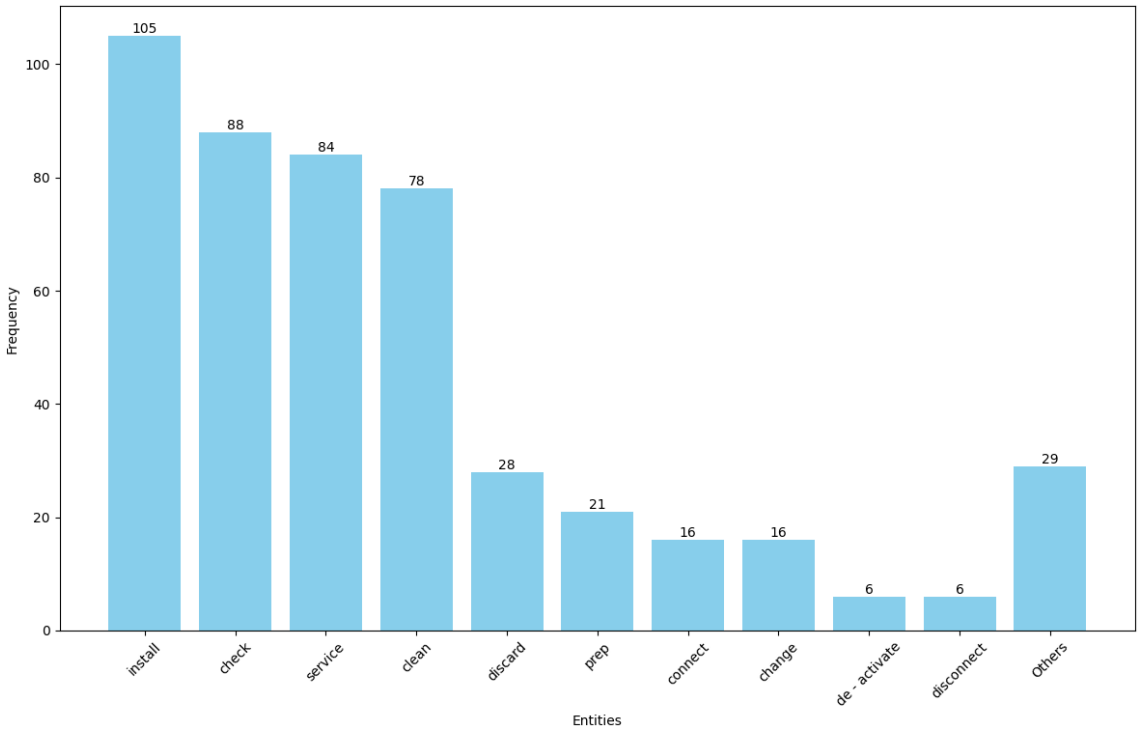


Figure 6-9 Action entities

Table 6-10 Comparison between AMRO model and existing ontology models

Author	Dataset Sources	Approach	Application	Major contribution
H Koorneef and Verhagen [30]	Maintenance documents Airworthiness Directives are chosen due to access restriction to maintenance documentation	Ontology-based	Maintenance Documentation (stored in network file folder)	Ontology for commercial aircraft domain Improved retrieval of maintenance documents
Yuchang Wu [8]	ATA SPEC 2200	Ontology-based	Maintenance records heterogeneity and inconsistency	Ontology-schema methodology
Palacios, Luis, et al. [168]	Post Flight Report (PFR) Electronic Log Book (ELB) Expert Knowledge	Ontology-based	Avionics Maintenance	Ontology for Failure Diagnosis Support
Sharad Dixit [5]	Maintenance records	Ontology-based	Maintenance records heterogeneity	Syntactic and semantic rules-based approach Language Model Approach

			(semantic heterogeneity)	Preliminary ontology for aircraft maintenance records
W. Verhagen and Curran [7]	Service Bulletin Airworthiness Directive	Enterprise Knowledge Resources with ontology	Maintenance Task Support (Detailed inspection of the main track downstop of the leading edge slats of the Boeing B737 aircraft)	The use Enterprise Knowledge Resources with semantic model to improve maintenance task execution and record keeping.
AMRO	B737-400 Maintenance Records	Ontology-based	Maintenance records heterogeneity (semantic and syntactic heterogeneity)	Domain Ontology for Aircraft Maintenance Records Syntactic and semantic correction of heterogeneous maintenance records

7 Discussion

The primary objective of this research was to propose and develop an Aircraft Maintenance Records Ontology, which sought to address significant challenges associated with data heterogeneity, integration, and consistency in aircraft maintenance documentation. The first objective of this study was to establish state-of-the-art aircraft maintenance knowledge management to identify research gaps. A systematic literature review revealed substantial advancements in the field, particularly the adoption of ontologies and semantic technologies to address data heterogeneity issues. Despite these advancements, this review highlighted persistent challenges, particularly in the practical implementation and integration of heterogeneous data sources. It did not only identify the theoretical frameworks and methodologies currently in use, but also established the specific gaps addressed in this research, which includes the lack of practical solutions for standardising heterogenous maintenance records.

The second objective was to extract information structure, concepts and relationships from the Boeing 737 maintenance records. This process involved conducting an extensive document analysis of the records, which included categorising the key concepts, terminologies and relationships inherent within the data.

Drawing upon the existing literature on ontology development, the synthesis of the literature reinforces the importance of document analysis in establishing the foundational structure of AMRO. Document Analysis was crucial for ensuring that the developed ontology accurately reflected domain specific knowledge and requirements. The extraction and structuring of this information laid a solid foundation for developing AMRO, facilitating a comprehensive representation of the aircraft maintenance domain.

The synthesis of the literature highlights the lack of standardised methodologies for ontology-based application development. This gap highlights the need for an agile methodology which offers flexibility and adaptability, particularly in the context of a continuously evolving field, such as aircraft maintenance.

The third objective aimed to develop the AMRO using an agile development methodology, specifically the proposed ADOBA methodology. This iterative methodology allowed for continuous feedback and refinement, which was crucial for aligning the ontology with the dynamic needs of aircraft maintenance operations. This facilitated the timely identification and correction of issues, but also enabled the integration of evolving domain knowledge, thereby enhancing the overall robustness and applicability of the ontology. The successful application of ADOBA in the development of AMRO demonstrated the efficacy of agile methodologies in ontology development.

Evidence from the literature demonstrates the need for practical applications to validate the theoretical frameworks discussed. The potential of AMRO to address real-world applications, such as correcting inconsistencies in maintenance records, is emphasised as a critical next step in demonstrating the utility of the ontology and NER system.

The fourth objective was developing a prototype demonstration, this involved the development of an ontology-based application to correct inconsistencies in maintenance records and further utilised NER for component categorisation with the corrected data. The prototype shows practical application of AMRO in real-world scenarios, particularly in addressing data inconsistencies and enhancing record accuracy. This phase of research was crucial in translating theoretical understanding into functional application that could be employed by industry practitioners. The successful implementation of the prototype showed the potential of semantic technologies to improve data consistency and operational efficiency in aircraft maintenance, thereby corroborating the practical relevance and impact of this research.

The fifth objective was to evaluate, validate, and compare the outcomes with existing literature. This involved testing of AMRO and the prototype against established metrics. The evaluation confirmed the effectiveness of the developed solutions in improving data consistency. Comparing these results with those from the existing literature further validated the contribution made in this research. The metrics indicated significant improvements in data accuracy, showcasing the

potential of AMRO and the prototype to establish new benchmarks in aircraft maintenance records management.

The performance of the developed AMRO model in terms of precision, recall, and F-measure metrics is balanced, which indicates its ability to handle data inconsistencies effectively. This aligns with the findings of other models, such as those proposed by Pirro and Domenico [166], which also emphasise balanced metrics for effective ontology mapping. Therefore, the findings validate the importance of developing robust domain ontologies to overcome the data heterogeneity challenges in AOM.

The lack of methodologies has created a notable gap in the ontology domain, often leading to inconsistencies and reduced applicability of ontological solutions in practice. An agile approach (ADOBA) is proposed, which emphasise reuse and structured development processes. The findings align with the methodologies that a structured approach to ontology development can significantly enhance the practical applicability and robustness of ontological models. By integrating comprehensive lifecycle approaches and continuous testing, these methodologies ensure that the developed ontologies are technically sound and relevant to real-world applications. This implication is crucial for advancing the state-of-the-art in ontology development, particularly in specialised domains, such as AOM.

The inclusion of ontologies in knowledge management (KM) systems has significant implications for the efficiency of these systems in managing complex operational data. The findings indicate that ontologies play a crucial role in structuring knowledge for easy access, interpretation, and application across diverse maintenance scenarios. This is especially important in AOM, where the complexity and volume of data require robust KM solutions. State-of-the-art KM systems that employ ontologies, as reviewed in this research, have demonstrated improved data integration, reasoning, and inference capabilities. These capabilities are essential for transforming raw data into actionable knowledge, allowing for advanced data analytics such as predictive maintenance.

The findings of this research facilitate seamless data integration by correcting inconsistencies in heterogeneous systems, which is crucial for aircraft through-life management and support. This integration is particularly beneficial for lessors who need to manage the asset data stored in multiple systems by different lessees over the lifespan of an aircraft. Also, the research approach to correcting inconsistencies in maintenance records supports advanced data analytics applications.

Ensuring compliance with stringent aviation regulations is a significant application of this study. The ontology developed provides a systematic approach to maintaining comprehensive and accurate maintenance records, aligned with standards such as ATA Spec 2500. This alignment simplifies audits and inspections and helps airlines and maintenance organisations for certification and compliance.

Finally, this study contributes significantly to knowledge management in the aviation industry by creating a comprehensive knowledge base for aircraft maintenance records. The ontology serves as a central repository for maintenance knowledge and captures best practices and expertise. It can be reused, thereby supporting continuous improvements in maintenance data management.

8 Conclusion

This research explores the critical role of knowledge management in the context of aircraft operations and maintenance (AOM), emphasising the importance of knowledge representation, reuse and integration. The primary focus was the development and evaluation of domain ontology to address the heterogeneity and interoperability challenges inherent in aircraft maintenance records (AMR).

The synthesis from the literature forms the foundation for this research. It highlights the challenges in data integration, interoperability and the need for standardised frameworks, which supports the first objective of understanding the limitations of current aircraft maintenance KM systems. The second objective, which focuses on establishing an information structure based on the analysis of Boeing 737 maintenance records, is reinforced by the literature, demonstrating the importance of structured data and ontology-based approaches to address semantic heterogeneity and ensure effective data management. The third objective, to develop AMRO using an agile methodology, is supported by a review of existing ontology development approaches, emphasising the need for an iterative approach tailored to the evolving needs of the AOM domain.

This research began by highlighting the deficiencies in existing KM approaches within the AOM industry, particularly the lack of comprehensive ontologies tailored to maintenance records. The systematic review shows that while ontologies are increasingly recognised as fundamental for KM, their application in the AMR domain is still in its early stage. This gap necessitates the need for a novel information model that can effectively manage and integrate heterogeneous maintenance data.

The proposed AMRO model shows a robust result, demonstrating balanced performance in terms of precision, recall and F-measure. The comparative analysis with other models such as those by Pirró, Jung, and Vennesland et al., highlighted the AMRO model's ability to correct inconsistencies in maintenance records while ensuring accuracy and comprehensiveness. This balance is

particularly crucial for aircraft maintenance applications, where both operational safety and efficiency are significant.

The practical implications of the AMRO model are important; by ensuring high precision, the model facilitates accurate data corrections, whereas high recall ensures that most inconsistencies are identified and addressed. This dual capability makes the AMRO model highly suitable for real-world applications, providing a reliable approach for enhancing the data quality and reliability in aircraft maintenance records.

Furthermore, the thesis also discusses the challenges and methodologies related to ontology development, emphasising the need for standardised and modular approaches. The review of various ontology development methodologies, including Methontology, NeOn, and agile methodologies, such as SAMOD and OntoSoft, provided a comprehensive understanding of the strengths and limitations of each approach. This analysis informs the development of the AMRO model following ADOBA methodology, ensuring that it is grounded in rigorous methodological principles.

This thesis contributes to the field of KM in AOM by providing a novel, balanced, and practically applicable ontology model for AMR. The AMRO model does not only address the current gaps in ontology-based solutions for aircraft maintenance but also sets a foundation for future advancements in this critical area.

8.1 Future Work

Future work should aim to contribute to standardisation efforts in the aviation industry by aligning the AMRO model with international standards for data exchange and interoperability such as SPEC 2500. In addition, the current version of the AMRO model, while comprehensive, can be further expanded to cover additional aspects of aircraft maintenance; this includes predictive maintenance and lifecycle management. These efforts will facilitate wider

adoption and integration with other systems to ensure the practical utility of the model.

To further improve the practical applicability of the AMRO model, it is important to optimise AMRO model for scalability to handle larger datasets efficiently. Future work should also explore the integration of the AMRO model with advanced LLMs such as GPT and BERT. This hybrid approach can leverage the structured knowledge of the AMRO model and the linguistic capabilities of NLP models to enhance data interpretation, automate the extraction of information from unstructured text, and improve the overall efficiency and accuracy of the maintenance documentation.

In summation, the proposed AMRO model needs to be expanded to provide broader domain conceptualisation. It is hoped that it serves as a foundation for further development because of the need to standardise aircraft maintenance records.

8.2 Limitations of the research

This study demonstrated the importance of knowledge representation, reuse, and integration in managing the complexities of AOM. Through the development of the AMRO, the study addresses data heterogeneity and interoperability challenges in aircraft maintenance records. AMRO provides a structured approach to harmonise diverse data formats and improve consistency and accuracy. However, the application of ontologies in this domain is still in its early stages, highlighting the need for more advanced and scalable solutions for real-world applications. Although AMRO demonstrated balanced performance in terms of precision, recall, and F-measure, the trade-offs between these metrics reveal the challenge of optimising both accuracy and comprehensiveness.

Despite AMRO’s success in addressing some core challenges, the research acknowledges that the alignment of ontology with broader international standards, such as ATA Spec 2500 and IATA Paperless Aircraft Operations, remains incomplete. This limits its immediate utility for widespread adoption in

environments that require strict regulatory compliance. Additionally, the scope of AMRO's application is focused primarily on resolving historical data inconsistencies, with limited attention given to predictive maintenance and lifecycle management, which are increasingly critical in modern data-driven aircraft maintenance strategies. Although NER has been employed to categorise data more effectively, the integration of more advanced NLP techniques has not been fully explored. Future work should aim to incorporate these advanced models and extend the ontology capabilities to support predictive analytics and broader industry standards, thus paving the way for more comprehensive and scalable solutions in the aircraft maintenance domain.

References

1. IATA. Guidance Material for the implementation of Paperless Aircraft Operations in Technical Operations. 2017.

2. Lamb K. Challenges of Digitalisation in the Aerospace and Aviation Sectors. Centre for Digital Built Britain, University of Cambridge; 2018.

3. Abdallah AA., Fan I-S. Emerging Challenges of Digital Aircraft Operations and Maintenance: A Knowledge Management Perspective. TESConf; 2020;

4. Koornneef H., Verhagen WJC., Curran R. Contextualising aircraft maintenance documentation. International Journal of Agile Systems and Management. Inderscience Publishers; 2017; 10(2): 160–179. Available at: DOI:10.1504/IJASM.2017.086159

5. Dixit S., Mulwad V., Saxena A. Extracting Semantics from Maintenance Records. arXiv preprint arXiv:2108.05454. 2021;

6. Sripada SM. Information management challenges from the aerospace industry. Proceedings of the 28th international conference on Very Large Data Bases. 2002. pp. 1006–1007.

7. Verhagen WJC., Curran R. An ontology-based approach for aircraft maintenance task support. 20th ISPE International Conference on Concurrent Engineering, CE 2013 - Proceedings. 2013. pp. 494–506. Available at: DOI:10.3233/978-1-61499-302-5-494

8. Wu Y., Ebrahimipour V., Yacout S. Ontology-Based Modeling of Aircraft to support Maintenance Management System. 2014.

9. O'Dell CS., O'dell C., Grayson CJ., Essaides N. If only we knew what we know: The transfer of internal knowledge and best practice. The Free Press; 1998.

10. Baskerville R., Dulipovici A. The theoretical foundations of knowledge management. Knowledge Management Research & Practice. Springer; 2006; 4(2): 83–105.

11. Alavi M., Leidner D. Knowledge management systems: issues, challenges, and benefits. Communications of the Association for Information systems. 1999; 1(1): 7.

12. Wiig KM. Knowledge Management Methods: Practical Approaches to Managing Knowledge. Schema Press; 1995; 3.

13. Davenport TH., Prusak L., others. Working knowledge: How organizations manage what they know. Harvard Business Press; 1998.

14. Metaxiotis K., Ergazakis K., Psarras J. Exploring the world of knowledge management: agreements and disagreements in the academic/practitioner community. Journal of knowledge management. Emerald Group Publishing Limited; 2005;

15. Gao T., Chai Y., Liu Y. A review of knowledge management about theoretical conception and designing approaches. International Journal of Crowd Science. Emerald Publishing Limited; 2018;

16. Curran R., Verhagen WJC., Van Tooren MJL., Van Der Laan TH. A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD. Expert Systems with Applications. 2010. pp. 7336–7350. Available at: DOI:10.1016/j.eswa.2010.04.027

17. Bermell-Garcia P., Verhagen WJC., Astwood S., Krishnamurthy K., Johnson JL., Ruiz D., et al. A framework for management of Knowledge-Based Engineering applications as software services: Enabling personalization and codification. Advanced Engineering Informatics. April 2012; 26(2): 219–230. Available at: DOI:10.1016/j.aei.2012.01.006

18. Dhanisetty VSV., Verhagen WJC., Curran R. Multi-criteria weighted decision making for operational maintenance processes. Journal of Air Transport Management. Elsevier Ltd; 1 May 2018; 68: 152–164. Available at: DOI:10.1016/j.jairtraman.2017.09.005

19. Verhagen WJC., De Vrught B., Schut J., Curran R. A method for identification of automation potential through modelling of engineering processes and quantification of information waste. Advanced Engineering Informatics. Elsevier Ltd; 1 August 2015; 29(3): 307–321. Available at: DOI:10.1016/j.aei.2015.03.003

20. Li R., Verhagen WJC., Curran R. Stakeholder-oriented systematic design methodology for prognostic and health management system: Stakeholder expectation definition. Advanced Engineering Informatics. Elsevier Ltd; 1 January 2020; 43. Available at: DOI:10.1016/j.aei.2020.101041

21. Boland A., Cherry G., Dickson R. Doing a Systematic Review: A Student's Guide. SAGE Publications; 2017.

22. Okoli C. The view from giants' shoulders: Developing theory with theory-mining systematic literature reviews. Available at SSRN 2699362. 2015;

23. Okoli C. A guide to conducting a standalone systematic literature review. Communications of the Association for Information Systems. 2015; 37(1): 43.

24. Fernández-López M., Gómez-Pérez A., Juristo N. Methontology: from ontological art towards ontological engineering. American Asociation for Artificial Intelligence; 1997;

25. Yousefzadeh Aghdam M., Kamel Tabbakh SR., Mahdavi Chabok SJ., kheyrabadi M. Ontology generation for flight safety messages in air traffic management. Journal of Big Data. Springer Science and

Business Media Deutschland GmbH; 1 December 2021; 8(1). Available at: DOI:10.1186/s40537-021-00449-3

26. Egami S., Lu X., Koga T., Sumiya Y. Ontology-based data integration for semantic interoperability in air traffic management. Proceedings - 14th IEEE International Conference on Semantic Computing, ICSC 2020. Institute of Electrical and Electronics Engineers Inc.; 2020. pp. 295–302. Available at: DOI:10.1109/ICSC.2020.00059

27. Stefanidis D., Christodoulou C., Symeonidis M., Pallis G., Dikaiakos M., Pouis L., et al. The ICARUS Ontology: A general aviation ontology developed using a multi-layer approach. PervasiveHealth: Pervasive Computing Technologies for Healthcare. ICST; 2020. pp. 21–32. Available at: DOI:10.1145/3405962.3405983

28. Ledvinka M., Kremen P., Lališ A., Křemen P. Towards Data-Driven Safety: An Ontology-Based Information System Digital integration of information/data/processes in CAE/FM industry leading to apply the full potential of ICT View project MONDIS View project Towards Data-Driven Safety: An Ontology-Based Information System. Available at: https://www.researchgate.net/publication/332171513

29. Keller RM., Ranjan S., Wei MY., Eshow MM. Semantic representation and scale-up of integrated air traffic management data. Proceedings of the ACM SIGMOD International Conference on Management of Data. Association for Computing Machinery; 2016. Available at: DOI:10.1145/2928294.2928296

30. Koornneef H., Verhagen WJC., Curran R. Automating contextualized maintenance documentation. Advances in Transdisciplinary Engineering. IOS Press BV; 2016. pp. 861–870. Available at: DOI:10.3233/978-1-61499-703-0-861

31. Sanya IO., Shehab EM. An ontology framework for developing platform-independent knowledge-based engineering systems in the aerospace industry. International Journal of Production Research.

Taylor and Francis Ltd.; 2014; 52(20): 6192–6215. Available at: DOI:10.1080/00207543.2014.919422

32. Qiu H., Wang B., Yu J. Aircraft Maintenance Process Knowledge Modeling Method. 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT. 2020. pp. 717–723. Available at: DOI:10.1109/ICCASIT50869.2020.9368785

33. Sheng Y., Chen X., Mo H., Chen X., Zhang Y. An Ontology for Decision-Making Support in Air Traffic Management. Lecture Notes in Electrical Engineering. Springer; 2020. pp. 458–466. Available at: DOI:10.1007/978-981-15-0187-6_55

34. Insaurralde CC., G. Costa PC., Blasch E., Sampigethaya K. Uncertainty Considerations for Ontological Decision-Making Support in Avionics Analytics. 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). 2018. pp. 1–9. Available at: DOI:10.1109/DASC.2018.8569816

35. Insaurralde CC., Blasch E. Veracity metrics for ontologica! Decision-making support in avionics analytics. 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC). 2017. pp. 1–9. Available at: DOI:10.1109/DASC.2017.8101999

36. Blasch E. Ontologies for nextgen avionics systems. 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC). 2015. pp. 3B5-1-3B5-13. Available at: DOI:10.1109/DASC.2015.7311395

37. Ruiz A., Habli I., Espinoza H. LNCS 7613 - Towards a Case-Based Reasoning Approach for Safety Assurance Reuse. LNCS. 2012.

38. Zhu H., Gao J., Li D., Tang D. A Web-based product service system for aerospace maintenance, repair and overhaul services. Computers in Industry. May 2012; 63(4): 338–348. Available at: DOI:10.1016/j.compind.2012.02.016

39. Wang M., Qin G., Hu N. Research on knowledge representation for health management of aerospace based on ontology. 2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering. 2012. pp. 587–591. Available at: DOI:10.1109/ICQR2MSE.2012.6246302

40. Gringinger E., Eier D., Merkl D. NextGen and SESAR moving towards ontology-based software development. 2011 Integrated Communications, Navigation, and Surveillance Conference Proceedings. 2011. pp. H3-1-H3-10. Available at: DOI:10.1109/ICNSURV.2011.5935286

41. Zayas DS., Monceaux A., Ait-Ameur Y. Knowledge models to reduce the gap between heterogeneous models: Application to aircraft systems engineering. Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems, ICECCS. 2010. pp. 355–360. Available at: DOI:10.1109/ICECCS.2010.56

42. Zhou Y., Li Q., Zuo Y. Fault knowledge management in aircraft maintenance. 2009 8th International Conference on Reliability, Maintainability and Safety. 2009. pp. 645–649. Available at: DOI:10.1109/ICRMS.2009.5270111

43. Liu J., Tang SY., Li Q., Wang WP. An ontology for Aircraft route planning. 1st International Conference on Advances in System Simulation, SIMUL 2009. 2009. pp. 68–72. Available at: DOI:10.1109/SIMUL.2009.25

44. Keller RM. Ontologies for aviation data management. AIAA/IEEE Digital Avionics Systems Conference - Proceedings. Institute of Electrical and Electronics Engineers Inc.; 2016. Available at: DOI:10.1109/DASC.2016.7777971

45. Zhou Y., Li Q., Zuo Y. Fault knowledge management in aircraft maintenance. 2009 8th International Conference on Reliability,

Maintainability and Safety. 2009. pp. 645–649. Available at: DOI:10.1109/ICRMS.2009.5270111

46. Koornneef H., Verhagen WJC., Curran R. Contextualising aircraft maintenance documentation. International Journal of Agile Systems and Management. Inderscience Publishers; 2017; 10(2): 160–179. Available at: DOI:10.1504/IJASM.2017.086159

47. Verhagen WJC., Curran R. An ontology-based approach for aircraft maintenance task support. 20th ISPE International Conference on Concurrent Engineering, CE 2013 - Proceedings. 2013. pp. 494–506. Available at: DOI:10.3233/978-1-61499-302-5-494

48. Insaurralde CC., Blasch E. Veracity metrics for ontologica! Decision-making support in avionics analytics. 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC). 2017. pp. 1–9. Available at: DOI:10.1109/DASC.2017.8101999

49. Sheng Y., Chen X., Mo H., Chen X., Zhang Y. An Ontology for Decision-Making Support in Air Traffic Management. Lecture Notes in Electrical Engineering. Springer; 2020. pp. 458–466. Available at: DOI:10.1007/978-981-15-0187-6_55

50. Gómez-Pérez A., Fernández-López M., Corcho O. Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer Science & Business Media; 2006.

51. Suárez-Figueroa MC., Gómez-Pérez A., Fernández-López M. The NeOn methodology for ontology engineering. Ontology engineering in a networked world. Springer; 2012. pp. 9–34.

52. Peroni S. A simplified agile methodology for ontology development. OWL: Experiences and Directions–Reasoner Evaluation. Springer; 2016. pp. 55–69.

53. Machado JB., Isotani S., Barbosa A., Bandeira J., Alcantara W., Bittencourt I., et al. OntoSoft Process: Towards an agile process for ontology-based software. 2016 49th Hawaii International Conference on System Sciences (HICSS). 2016. pp. 5813–5822.

54. Sure Y., Staab S., Studer R. On-to-knowledge methodology (OTKM). Handbook on ontologies. Springer; 2004. pp. 117–132.

55. Abdelghany AS., Darwish NR., Hefni HA. An agile methodology for ontology development. International Journal of Intelligent Engineering and Systems. 2019; 12(2): 170–181. Available at: DOI:10.22266/IJIES2019.0430.17

56. Esposito A. The semantic web. Advances in Information Technologies for Electromagnetics. JSTOR; 2006. pp. 29–44. Available at: DOI:10.1007/978-1-4020-4749-5_3

57. Lenzerini M. Data integration: A theoretical perspective. Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. 2002. pp. 233–246.

58. Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R., Ruberti GA. Ontology-Based Data Access and Integration. 2018.

59. Cruz IF., Xiao H., others. The role of ontologies in data integration. Engineering intelligent systems for electrical engineering and communications. Citeseer; 2005; 13(4): 245.

60. Egami S., Lu X., Koga T., Sumiya Y. Ontology-based data integration for semantic interoperability in air traffic management. Proceedings - 14th IEEE International Conference on Semantic Computing, ICSC 2020. Institute of Electrical and Electronics Engineers Inc.; 2020. pp. 295–302. Available at: DOI:10.1109/ICSC.2020.00059

61. Berners-Lee T., Hendler J., Lassila O. The semantic web. Scientific american. JSTOR; 2001; 284(5): 34–43.

62. Staab S., Studer R. Handbook on ontologies. Springer Science \& Business Media; 2013.

63. Gómez-Pérez A., Fernández-López M., Corcho O. Theoretical foundations of ontologies. Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. Springer; 2004; : 1–45.

64. Hepp M. Possible ontologies: How reality constrains the development of relevant ontologies. IEEE Internet Computing. IEEE; 2007; 11(1): 90–96.

65. Gruber T. Shared Reusable Knowledge Bases (SRKB) mailing list. 1994.

66. Gruber TR. A translation approach to portable ontology specifications. Knowledge acquisition. Elsevier; 1993; 5(2): 199–220.

67. Smith B. Logic and Formal Ontology. In: Mohanty JN, McKenna W (eds.) Logic and formal ontology. In J. N. Mohanty & W. McKenna (eds.), Husserl's Phenomenology: A Textbook. University Press of America; 1989. pp. 29–67.

68. Borst WN. Construction of engineering ontologies for knowledge sharing and reuse. University Of Twente; 1999.

69. Guarino N. Understanding, building and using ontologies. International journal of human-computer studies. Elsevier; 1997; 46(2–3): 293–310.

70. Van Heijst G., Schreiber AT., Wielinga BJ. Using explicit ontologies in KBS development. International journal of human-computer studies. Elsevier; 1997; 46(2–3): 183–292.

71. Alberts LK. YMIR: An ontology for engineering design. 1995;

72. Wang M., Qin G., Hu N. Research on knowledge representation for health management of aerospace based on ontology. 2012

International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering. 2012. pp. 587–591. Available at: DOI:10.1109/ICQR2MSE.2012.6246302

73. Keller RM., Ranjan S., Wei MY., Eshow MM. Semantic representation and scale-up of integrated air traffic management data. Proceedings of the ACM SIGMOD International Conference on Management of Data. Association for Computing Machinery; 2016. Available at: DOI:10.1145/2928294.2928296

74. Fernández-López M., Gómez-Pérez A. Overview and analysis of methodologies for building ontologies. The knowledge engineering review. Cambridge University Press; 2002; 17(2): 129–156.

75. Ibrahim F. Knowledge Management Methodology: Developing a Phenomenalogical Middle-range Thinking Approach for Knowledge Management Research. Journal of Management Research. 2017; 9(4): 110.

76. Machado JB., Isotani S., Barbosa A., Bandeira J., Alcantara W., Bittencourt I., et al. OntoSoft Process: Towards an agile process for ontology-based software. 2016 49th Hawaii International Conference on System Sciences (HICSS). 2016. pp. 5813–5822.

77. Peroni S. A simplified agile methodology for ontology development. OWL: Experiences and Directions–Reasoner Evaluation. Springer; 2016. pp. 55–69.

78. McMahon C., Ball A. Information systems challenges for through-life engineering. Procedia CIRP. Elsevier; 2013; 11: 1–7.

79. Pinto HS., Martins JP. Reusing ontologies. AAAI 2000 Spring Symposium on Bringing Knowledge to Business Processes. 2000. p. 7.

80. Abdallah AA., Fan I-S. Emerging Challenges of Digital Aircraft Operations and Maintenance: A Knowledge Management Perspective. TESConf; 2020;

81. Breznik K. Knowledge management–from its inception to the innovation linkage. Procedia-Social and Behavioral Sciences. Elsevier; 2018; 238: 141–148.

82. Sylvester A., Tate M., Johnstone D. Beyond synthesis: Re-presenting heterogeneous research literature. Behaviour & Information Technology. Taylor & Francis; 2013; 32(12): 1199–1215.

83. Brereton P., Kitchenham BA., Budgen D., Turner M., Khalil M. Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software. 2007; 80(4): 571–583. Available at: DOI:https://doi.org/10.1016/j.jss.2006.07.009

84. Ridley D. The literature review: A step-by-step guide for students. Sage; 2012;

85. Metaxiotis K., Ergazakis K., Psarras J. Exploring the world of knowledge management: agreements and disagreements in the academic/practitioner community. Journal of knowledge management. Emerald Group Publishing Limited; 2005;

86. Davenport TH., Prusak L., others. Working knowledge: How organizations manage what they know. Harvard Business Press; 1998.

87. Wiig KM. Knowledge Management Methods: Practical Approaches to Managing Knowledge. Schema Press; 1995; 3.

88. Alavi M., Leidner D. Knowledge management systems: issues, challenges, and benefits. Communications of the Association for Information systems. 1999; 1(1): 7.

89. Baskerville R., Dulipovici A. The theoretical foundations of knowledge management. Knowledge Management Research & Practice. Springer; 2006; 4(2): 83–105.

90. Insaurralde CC., G. Costa PC., Blasch E., Sampigethaya K. Uncertainty Considerations for Ontological Decision-Making Support in Avionics Analytics. 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). 2018. pp. 1–9. Available at: DOI:10.1109/DASC.2018.8569816

91. Zayas DS., Monceaux A., Ait-Ameur Y. Knowledge models to reduce the gap between heterogeneous models: Application to aircraft systems engineering. Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems, ICECCS. 2010. pp. 355–360. Available at: DOI:10.1109/ICECCS.2010.56

92. Qiu H., Wang B., Yu J. Aircraft Maintenance Process Knowledge Modeling Method. 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT. 2020. pp. 717–723. Available at: DOI:10.1109/ICCASIT50869.2020.9368785

93. Zhou Y., Li C., Huang G., Guo Q., Li H., Wei X. A Short-Text Similarity Model Combining Semantic and Syntactic Information. Electronics. MDPI; 2023; 12(14): 3126.

94. Zhou Y., Li C., Huang G., Guo Q., Li H., Wei X. A Short-Text Similarity Model Combining Semantic and Syntactic Information. Electronics. MDPI; 2023; 12(14): 3126.

95. Wu Y., Ebrahimipour V., Yacout S. Ontology-Based Modeling of Aircraft to support Maintenance Management System. 2014.

96. Guizzardi G. Theoretical foundations and engineering tools for building ontologies as reference conceptual models. Semantic Web. IOS Press; 2010; 1(1, 2): 3–10.

97. Ruiz A., Habli I., Espinoza H. LNCS 7613 - Towards a Case-Based Reasoning Approach for Safety Assurance Reuse. LNCS. 2012.

98. Resnik P. Semantic Similarity in a Taxonomy. Journal of Artificial Intelligence Research. 1999; (11): 95–130.

99. Resnik P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. http://arxiv.org/abs/cmp-lg/9511007. 1995; 1. Available at: http://arxiv.org/abs/cmp-lg/9511007

100. Zhu H., Gao J., Li D., Tang D. A Web-based product service system for aerospace maintenance, repair and overhaul services. Computers in Industry. May 2012; 63(4): 338–348. Available at: DOI:10.1016/j.compind.2012.02.016

101. Berners-Lee T., Hendler J., Lassila O. The semantic web. Scientific american. JSTOR; 2001; 284(5): 34–43.

102. Baliyan N., Kumar S. Adaptation of software engineering to semantic web based system development. 2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA). 2013. pp. 1–5.

103. Gómez-Pérez A., Fernández-López M., Corcho O. Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer Science & Business Media; 2006.

104. Davidson SB., Overton C., Buneman P. Challenges in integrating biological data sources. Journal of Computational Biology. 1995; 2(4): 557–572.

105. Batini C., Lenzerini M., Navathe SB. A comparative analysis of methodologies for database schema integration. ACM computing surveys (CSUR). ACM New York, NY, USA; 1986; 18(4): 323–364.

106. Ledvinka M., Kremen P., Lališ A., Křemen P. Towards Data-Driven Safety: An Ontology-Based Information System Digital integration of information/data/processes in CAE/FM industry leading to apply the full potential of ICT View project MONDIS View project Towards Data-Driven Safety: An Ontology-Based Information System. Available at: https://www.researchgate.net/publication/332171513

107. Wache H., Voegele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., et al. Ontology-based integration of information-a survey of existing approaches. Ois@ ijcai. 2001.

108. Yousefzadeh Aghdam M., Kamel Tabbakh SR., Mahdavi Chabok SJ., kheyrabadi M. Ontology generation for flight safety messages in air traffic management. Journal of Big Data. Springer Science and Business Media Deutschland GmbH; 1 December 2021; 8(1). Available at: DOI:10.1186/s40537-021-00449-3

109. Virginija U., Rimantas B. Ontology-based Foundations for Data Integration. The First International Conference on Business Intelligence and Technology Copyright (c) IARIA. 2011.

110. Adams T., Dullea J., Clark P., Sripada S., Barrett T. Semantic integration of heterogeneous information sources using a knowledge-based system. Proc 5th Int Conf on CS and Informatics (CS&I'2000). 2000.

111. IATA. Guidance Material for the implementation of Paperless Aircraft Operations in Technical Operations. 2017.

112. Ullah MR., Chowdhury NS., Tawsif FM. Impact of Combining Syntactic and Semantic Similarities on Patch Prioritization while using the Insertion Mutation Operators. arXiv preprint arXiv:2302.03730. 2023;

113. Al-Emran M., Mezhuyev V., Kamaludin A., Shaalan K. The impact of knowledge management processes on information systems: A

systematic review. International Journal of Information Management. Elsevier; 2018; 43: 173–187.

114. Kim W., Seo J. Classifying schematic and data heterogeneity in multidatabase systems. Computer. IEEE; 1991; 24(12): 12–18.

115. Park J., Ram S. Information systems interoperability: What lies beneath? ACM Transactions on Information Systems (TOIS). ACM New York, NY, USA; 2004; 22(4): 595–632.

116. Ram S., Ramesh V. Schema integration: Past, present, and future. Management of Heterogeneous and Autonomous Database Systems. 1998. pp. 119–155.

117. Jirkovsky V., Obitko M., Mavrik V. Understanding data heterogeneity in the context of cyber-physical systems integration. IEEE Transactions on Industrial Informatics. IEEE; 2016; 13(2): 660–667.

118. Pinto HS., Martins JP. Reusing ontologies. AAAI 2000 Spring Symposium on Bringing Knowledge to Business Processes. 2000. p. 7.

119. Euzenat J., Shvaiko P., others. Ontology matching. Springer; 2007.

120. Sheth AP. Changing focus on interoperability in information systems: from system, syntax, structure to semantics. Interoperating geographic information systems. Springer; 1999. pp. 5–29.

121. Kostov B., Ahmad J., Křemen P. Towards ontology-based safety information management in the aviation industry. OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". 2016. pp. 242–251.

122. Cheatham M., Pesquita C. Semantic data integration. Handbook of big data technologies. Springer; 2017; : 263–305.

123. Gill K. On the metaphysical distinction between processes and events. Canadian Journal of Philosophy. Cambridge University Press; 1993; 23(3): 365–384.

124. Partridge C. The role of ontology in integrating semantically heterogeneous databases. Rapport technique. Citeseer; 2002; 5(02).

125. Krzanowski R., Polak P. Ontology and AI Paradigms. MDPI AG; 2022. p. 119. Available at: DOI:10.3390/proceedings2022081119

126. Kau A., He X., Nambissan A., Astudillo A., Yin H., Aryani A. Combining Knowledge Graphs and Large Language Models. 9 July 2024; Available at: http://arxiv.org/abs/2407.06564

127. Velezhancev N., Blanchard D., Stavros W., Vandenberg C., Alighieri P. Towards a Novel Architecture for Semantic Pattern Resolution in Large Language Models. 2024. Available at: DOI:10.22541/au.173083351.19609434/v1

128. Liu B. Comparative Analysis of Encoder-Only, Decoder-Only, and Encoder- Decoder Language Models. INSTICC; 2024. pp. 524–530. Available at: DOI:10.5220/0012829800004547

129. Avramidis E., Manakhimova S., Macketanz V., Möller S. Machine Translation Metrics are better in evaluating Linguistic Errors on LLMs than on Encoder-Decoder Systems.

130. Soliman G., Zaki H., Kilany M. A comparative analysis of encoder only and decoder only models for challenging LLM-generated STEM MCQs using a self-evaluation approach. Natural Language Processing Journal. March 2025; 10: 100131. Available at: DOI:10.1016/j.nlp.2025.100131

131. Lin K. Structured Contexts For Large Language Models. 2024. Available at:

http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-218.html

132. How do LLMs deal with Syntactic Conflicts in In-context-learning ?

133. Rozony FZ., Aktar MNA., Ashrafuzzaman M., Islam A. A SYSTEMATIC REVIEW OF BIG DATA INTEGRATION CHALLENGES AND SOLUTIONS FOR HETEROGENEOUS DATA SOURCES. ACADEMIC JOURNAL ON BUSINESS ADMINISTRATION, INNOVATION & SUSTAINABILITY. 2 October 2024; 4(4): 1–18. Available at: DOI:10.69593/ajbais.v4i04.111

134. Giglou HB., D'Souza J., Engel F., Auer S. LLMs4OM: Matching Ontologies with Large Language Models. 16 April 2024; Available at: http://arxiv.org/abs/2404.10317

135. Yue J., Zhao C., Zhao J., Huang B. Enabling Generalized Zero-shot Learning Towards Unseen Domains by Intrinsic Learning from Redundant LLM Semantics. 21 March 2024; Available at: http://arxiv.org/abs/2403.14362

136. Yang W., Some L., Bain M., Kang B. A Comprehensive Survey on Integrating Large Language Models with Knowledge-Based Methods. 19 January 2025; Available at: http://arxiv.org/abs/2501.13947

137. Sosa RU., Ramamurthy KN., Chang M., Singh M. Reasoning about concepts with LLMs: Inconsistencies abound.

138. Paidisetty MK., Singh S., Sanal Kumar VR., Prakash O., Saladi R. Evaluating the Crucial Influence of Aircraft Maintenance Records and Safety Control on Aviation Safety–A Comprehensive Analysis. AIAA AVIATION FORUM AND ASCEND 2024. 2024. p. 4319.

139. Meng F., Wang CA. Artificial Intelligence and Machine Learning Approaches to Text Recognition: A Research Overview. J Math Techniques Comput Math. 2024; 3(3): 1–05.

140. Tang Q., Lee YS., Jung H. The Industrial Application of Artificial Intelligence-Based Optical Character Recognition in Modern Manufacturing Innovations. Sustainability (Switzerland) . Multidisciplinary Digital Publishing Institute (MDPI); 1 March 2024; 16(5). Available at: DOI:10.3390/su16052161

141. Mehri M. AI-driven methods for pattern segmentation, detection, and recognition. 2024. Available at: https://hal.science/tel-04926736v1

142. Lehmann J., Shamiyeh M., Ziemer S., Fensel A., Daniele L., Aroyo L., et al. Towards Integration and Coverage Assessment of Ontologies for Knowledge Reuse in the Aviation Sector. SEMANTICS Workshops. 2017.

143. Abdallah AA., Fan IS. Towards Building Ontology-Based Applications for Integrating Heterogeneous Aircraft Maintenance Records. IEEE International Conference on Industrial Informatics (INDIN). IEEE; 2022; 2022-July: 293–299. Available at: DOI:10.1109/INDIN51773.2022.9976080

144. Elo S., Kyngäs H. The qualitative content analysis process. Journal of Advanced Nursing. 2008; 62(1): 107–115. Available at: DOI:10.1111/j.1365-2648.2007.04569.x

145. Ferreira R., Cavalcanti GDC., Freitas F., Lins RD., Simske SJ., Riss M. Combining sentence similarities measures to identify paraphrases. Computer Speech and Language. Elsevier Ltd; 2018; 47: 59–73. Available at: DOI:10.1016/j.csl.2017.07.002

146. Doan A., Halevy A., Ives Z. Principles of data integration. Elsevier; 2012.

147. Yin C., Gu J., Hou Z. An Ontology Mapping Approach Based on Classification with Word and Context Similarity. Proceedings - 2016 12th International Conference on Semantics, Knowledge and Grids,

SKG 2016. IEEE; 2017; : 69–75. Available at: DOI:10.1109/SKG.2016.018

148. Obrst L., Ceusters W., Mani I., Ray S., Smith B. The evaluation of ontologies. Semantic web. Springer; 2007. pp. 139–158.

149. Hartmann J., Spyns P., Giboin A., Maynard D., Cuel R., Suárez-Figueroa MC., et al. D1. 2.3 Methods for ontology evaluation. EU-IST Network of Excellence (NoE) IST-2004-507482 KWEB Deliverable D. 2005; 1.

150. Gangemi A., Catenacci C., Ciaramita M., Lehmann J. A theoretical framework for ontology evaluation and validation. SWAP. 2005. p. 16.

151. Hashemi TL., Obrst L., Ray S., Sriram R., Schneider T., Vegetti M., et al. Ontology Summit 2013 Communiqué: Towards Ontology Evaluation across the Life Cycle.

152. McDaniel M., Storey VC. Evaluating domain ontologies: clarification, classification, and challenges. ACM Computing Surveys (CSUR). ACM New York, NY, USA; 2019; 52(4): 1–44.

153. Benjamins VR., Fensel D., Decker S., Perez AG. 2: building ontologies for the internet: a mid-term report. International Journal of Human-Computer Studies. Elsevier; 1999; 51(3): 687–712.

154. Porzel R., Malaka R. A Task-based Approach for Ontology Evaluation. ECAI Workshop on Ontology Learning and Population. 2004;

155. Stapleton J. DSDM, dynamic systems development method: the method in practice. Cambridge University Press; 1997.

156. Bouquet P., Euzenat J., Franconi E., Serafini L., Stamou G., Tessaris S. D2. 2.1 Specification of a common framework for characterizing alignment. University of Trento; 2004;

157. Adams T., Dullea J., Clark P., Sripada S., Barrett T. Semantic integration of heterogeneous information sources using a knowledge-based system. Proc 5th Int Conf on CS and Informatics (CS&I'2000). 2000.

158. Kostov B., Ahmad J., Křemen P. Towards ontology-based safety information management in the aviation industry. OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". 2016. pp. 242–251.

159. Fernández-López M., Gómez-Pérez A. Overview and analysis of methodologies for building ontologies. The knowledge engineering review. Cambridge University Press; 2002; 17(2): 129–156.

160. Obrst L., Ceusters W., Mani I., Ray S., Smith B. The evaluation of ontologies. Semantic web. Springer; 2007. pp. 139–158.

161. Gangemi A., Catenacci C., Ciaramita M., Lehmann J. A theoretical framework for ontology evaluation and validation. SWAP. 2005. p. 16.

162. Hartmann J., Spyns P., Giboin A., Maynard D., Cuel R., Suárez-Figueroa MC., et al. D1. 2.3 Methods for ontology evaluation. EU-IST Network of Excellence (NoE) IST-2004-507482 KWEB Deliverable D. 2005; 1.

163. Suárez-figueroa MC., Upm BV., Yufei Z. NeOn Methodology for Building Contextualized Ontology Networks Deliverable. 2008; : 1–150.

164. Pittet P., Barthélémy J. Exploiting users' feedbacks: Towards a task-based evaluation of application ontologies throughout their lifecycle. IC3K 2015 - Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. 2015; 2: 263–268. Available at: DOI:10.5220/0005608202630268

165. Vennesland A., Keller RM., Schuetz CG., Gringinger E., Neumayr B. Matching ontologies for air traffic management: A comparison and reference alignment of the AIRM and NASA ATM ontologies. CEUR Workshop Proceedings. 2019; 2536: 1–12.

166. Pirró G., Talia D. UFOme: An ontology mapping system with strategy prediction capabilities. Data and Knowledge Engineering. Elsevier B.V.; 2010; 69(5): 444–471. Available at: DOI:10.1016/j.datak.2009.12.002

167. Jung JJ. Reusing ontology mappings for query routing in semantic peer-to-peer environment. Information Sciences. 2010. pp. 3248–3257. Available at: DOI:10.1016/j.ins.2010.04.018

168. Palacios L., Lortal G., Laudy C., Sannino C., Simon L., Fusco G., et al. Avionics maintenance ontology building for failure diagnosis support. IC3K 2016 - Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. 2016; 2(Ic3k): 204–209. Available at: DOI:10.5220/0006092002040209

169. Ridley D. The literature review: A step-by-step guide for students. Sage; 2012;

170. Keele S., others. Guidelines for performing systematic literature reviews in software engineering. 2007.

171. Brereton P., Kitchenham BA., Budgen D., Turner M., Khalil M. Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software. 2007; 80(4): 571–583. Available at: DOI:https://doi.org/10.1016/j.jss.2006.07.009

172. Petticrew M., Roberts H. Systematic reviews in the social sciences: A practical guide. John Wiley & Sons; 2008.

173. Breznik K. Knowledge management–from its inception to the innovation linkage. Procedia-Social and Behavioral Sciences. Elsevier; 2018; 238: 141–148.

174. Al-Emran M., Mezhuyev V., Kamaludin A., Shaalan K. The impact of knowledge management processes on information systems: A systematic review. International Journal of Information Management. Elsevier; 2018; 43: 173–187.

175. Sylvester A., Tate M., Johnstone D. Beyond synthesis: Re-presenting heterogeneous research literature. Behaviour & Information Technology. Taylor & Francis; 2013; 32(12): 1199–1215.

176. Rousseau DM., Manning J., Denyer D. 11 Evidence in management and organizational science: assembling the field's full weight of scientific knowledge through syntheses. Academy of Management Annals. Routledge; 2008; 2(1): 475–515.

177. Moher D., Liberati A., Tetzlaff J., Altman DG., Group P. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. PLoS medicine. Public Library of Science San Francisco, USA; 2009; 6(7): e1000097.

Appendix A Ethical Approval Letter

15 October 2020

Dear Mr Abdallah ,

Reference: CURES/12351/2020
Title: Knowledge Management to Support Digital Operations for Aircraft Through-Life Maintenance and Support

Thank you for your application to the Cranfield University Research Ethics System (CURES).

We are pleased to inform you your CURES application, reference CURES/12351/2020 **has been reviewed. You may now proceed with the research activities you have sought approval for.**

If you have any queries, please contact CURES Support.

We wish you every success with your project.

Regards,

CURES Tea

Appendix B Systematic Review Protocol

B.1 Literature review

Almost 2 decades ago, [6] identified the challenges of information management in the aerospace industry in which he stated that the aerospace industry “poses significant challenges to information management unlike any other industry” and “integrating information from heterogeneous sources continues to be a formidable challenge”. Yet, it can be argued that the status quo is no different today and the need for technological solutions is evident.

Knowledge management is concerned with the creation, storing, and transferring knowledge. It is a framework, a management mind-set that includes building on past experiences including libraries, data-banks, people etc and creating new vehicles for exchanging knowledge [9]. The theoretical foundations of knowledge management [10] are deeply rooted into diverse areas including strategic management, organisational culture, behaviour and structure, information economics, and artificial intelligence. There are numerous factors that have resulted in the popularity of management which include globalization and a drive for sustainable competitive advantage [11]. [12] categorised three elements of a knowledge management system (KMS) as the knowledge source, the knowledge transfer process, and the knowledge recipient. The role of information technology has recently gained the interests of knowledge management researchers and several methods, frameworks, and techniques have been derived [12], [13]. Information technology is, therefore, effective in the process of capturing, storing, and transferring knowledge. As the role of information technology is gaining more prominence in the field, information systems are being regarded as a tool to support KM, and rather than a means of processing data [13]. Moreover, the concept of knowledge representation has led to the application of semantic link network (SLN) [14] in KM which offers a means of data modelling based on semantic rules [15]. SLN is therefore increasingly realised as a means to support and enhance management of information. One such example is a formal ontology which is usually created for a specific domain as a form of highly complex

knowledge representation artefact used for the development of intelligent knowledge management systems.

Several studies that seek to tap on the potential of semantic knowledge representation have been pioneered in aviation research. These include 3 critical areas as follows:

1) Maintenance and Support

2) Operations including Air Traffic Management

3) Aerospace Engineering and Informatics

While KM systems have been theoretically addressed in the literature, there is little known on how it can be realised in practice. Early scoping search that was carried out shows that KM research spans over 3 decades but it's application in the aviation industry was mainly explored over the last decade. Nonetheless, there exists a manifest research gap in the literature regarding the application and practicality of KM in the aviation industry. Although there is an apparent lack of systematic review studies specifically related to KM in the aviation industry, particularly for aircraft operations and maintenance, there are significant number of studies to help in understanding the application and essential elements that form an effective KM system for aircraft operations and maintenance. This is the key motivation to carry out this review.

B.1.1 Protocol and training

This systematic review will involve 2 reviewers in which the early scoping search resulted in 1,491 scholarly articles in just 1 of the identified data sources (Scopus). It is therefore necessary for the reviewers to adhere to the same techniques in order to ensure uniformity of the process, save valuable time by avoiding changes and maintain consistency of the interpretation. The steps involved in the review process have been outlined in Section 2.2 and both reviewers will agree on the establish steps, techniques and procedures in the protocol.

The SQ3R (Survey, Question, Read, Recall, Review) [169] technique will be followed to efficiently read and understand the scholarly documents including the title, keywords, abstract, introduction, conclusion, and the full content of the retrieved documents.

B.1.2 Reliability Measurement

Reliability is an important aspect of every systematic review. [23] noted out that it is meaningful to test a review for reliability. Since this study is limited in terms of number of reviewers, the first reviewer (R1) will screen all retrieved scholarly documents and then conduct a second screen on randomly selected documents. This method was referred to as the "test-retest" approach by [170]. Thereafter, the second reviewer (R2) will check the screening results of R1 (both included and excluded). This approach was found to have no 'major difference' with reviews where each reviewer independently extracted data [171] and that it can be helpful where there are 2 reviewers and large number of papers to review by saving time and effort. Finally, all the article reviews that are of concern to any of the reviewers will be discussed and, in the event, where an agreement is not concluded, the authors will be contacted for further clarity.

B.1.3 Review questions

The most important aspect of every systematic review is the questions it seeks to answer. Therefore, it is critical to carefully frame the questions and understand if a systematic review would be appropriate to answer them. [172] recommended a guideline which considers Population, Intervention, Comparison, Outcomes, and Context (PICOC). This guideline was used in framing the questions from the onset of this review.

Table B-1 Descriptions of the PICOC guidelines

PICOC	Description
Population	Aircraft Operations and Maintenance including MRO
Intervention	KM approaches and Solutions

Comparison	Existing solutions (systems) with KM solutions
Outcomes	Elements that are considered critical in development and application on the KM approaches and solutions
Contexts	Studies on AOM in which the use of KM approaches or solutions have been clearly identified

Thus, this review is set out to answer the following questions:

i. What are the existing KM approaches and solutions for AOM?

ii. How does the KM approaches and solutions differ with regards to development methods?

iii. What are the critical elements that forms KM FOR AOM?

iv. To what extent does the evidence support effectiveness of the different KM solutions?

v. What implications will the research findings have when creating a unified KM solution for AOM?

B.1.4 Apply practical screen

The challenge of going through all retrieved documents is enormous which is why the practical screen is essential. [21] referred to the screening process as ‘explicit elimination of studies’ as it is impractical to search everything. However, it should be comprehensive enough to minimise bias and ensure rigour. The criteria that will be used in this review is described below.

1. **Selection Criteria**

In this review, predefined selection criteria will be applied to identify studies which are relevant and present direct evidence about the review questions. The inclusion and exclusion criteria have been identified to clarify the requirements for eligibility and were derived from the review questions. Thus, allowing the scope of the review to be narrowed to the specified objectives.

Table B-2 Inclusion and exclusion criteria

Inclusion Criteria
This includes studies that involve: ✓ Aviation domain ✓ KM approaches or solutions ✓ Information systems or information modelling – and, ✓ Studies written in English language ✓ Studies that were published between 1990 – 2020
Exclusion Criteria
Studies which do not involve: ✓ Aviation domain ✓ KM approaches or solutions ✓ Information systems or information modelling – and, ✓ Studies not written in English language ✓ Studies that were not published between 1990 – 2020

Firstly, all retrieved scholarly documents will be reviewed by reading through the title, abstract and conclusion. Although it is generally recommended in many SLR guidelines to screen based on title and abstract, this will be extended to include the “conclusion” section of each document except the title and abstract, albeit subjective, shows it is explicitly out of scope of the review. This is because abstracts in information technology and software engineering are “too poor to rely on” [171] when screening primary studies. Nonetheless, all documents (included and excluded) will be clearly documented. Secondly, the documents that have

passed the first step will be reviewed in full strictly based on the inclusion and exclusion criteria of the review (See Table 3).

2. Search for literature

The research articles for this review will be searched for in the data sources identified in Table 2-1. The search process has been tailored to enable retrieval of primary studies relevant to the review questions. Thus, it includes formulating keywords, search strings and identifying data sources.

A scoping search was carried out using the formulated keywords extracted from the research questions which were broken down based on PICOC [170] (See Table B-1). It was then further extended with the keywords from studies identified in the scoping search (

Figure B-1). In addition, the synonyms, and abbreviations, spelling variations of the keywords were identified to examine the number of existing literatures in the research area.



Figure B-1 Search Keywords [4]

Search strings were then formulated using the identified keywords in section 5.1 with Boolean operators ‘AND’ and ‘OR’. It was then tested and revised to ensure inclusion of already known studies pertinent to this research.

Table B-3 Search Strings.

Keywords (from research questions)
("Knowledge Management" OR "Knowledge Discovery") AND ("Aviation" OR "Aerospace") AND ("Maintenance" OR "Operation") AND ("Systems" OR "Digital" OR "Electronic")
Extended Keywords (including keywords from identified studies)
(("Knowledge Management" OR "Knowledge Discovery" OR "Knowledge Management System" OR "Knowledge Representation" OR "Ontologies") AND ("Maintenance" OR "Operation" OR "Aviation" OR "Aerospace" OR "Aircraft") AND ("Systems" OR "Digital" OR "Electronic" OR "Software") AND ("Aviation" OR "Aerospace" OR "Aircraft"))

A comprehensive search of existing studies relevant to this research would be carried on digital libraries, journals, and other sources of grey literature. These include conference proceedings, reports and monographs as well as thesis and dissertations.

The journals were selected based on highest frequency of publications in the area of Knowledge Management in which 16 were identified by [173]. Coincidentally, 12 of the journals are indexed in the digital libraries listed in Table 5 which have been used by several KM researchers [174]. This in addition to the journals that were also identified based on most cited articles in KM. It is worth noting that the list of journals with most cited articles and highest frequency of publications differ as earlier research in KM were published in the former while the latter were more recently established (1999 – 2006). This suggests a substantial increase in KM research and therefore both lists of journals, which are peer-reviewed, would be independently hand searched to ensure that all relevant journal articles are retrieved.

The initial scoping search also shows a tangible number of grey literatures spanning from conference proceedings, research thesis, industrial case studies, and reports which present state-of-the-art research on KM in the aviation industry.

The sources of such literatures, many which are non-peer-reviewed, were identified and form a part of the search.

In this review, data was extracted by reviewing full content of all the identified primary studies following the SQ3R technique. A predefined data extraction form with the following questions will be used to extract data relevant to the research questions. While both reviewers will be trained on the data extraction process, the data will be extracted by R1 and checked by R2 as suggested by [171].

a. What is the context in which the study in carried out? The context of each study will be identified and recorded accordingly. E.g. Air Traffic, Engineering, Operations and Maintenance etc.
b. Which KM approach or solution is being considered in the research and what are its critical elements? The specific approach or solutions will be noted in addition to the critical elements identified by the researchers.
c. Which method was used to develop the KM approach or solution? The development methods used will be explicitly recorded.
d. Is there evidence to support the result?
e. What are the implications of the research towards a unified solution?
f. Any other issues or observations identified.

The purpose quality appraisal or assessment is to build further on the selection criteria in broader details which may result in excluding more studies using a predefined measure or checklist. However, [172] cautioned that this kind of assessment addresses methodological rather than reporting quality. This is particularly true for this review as not all studies may provide access to the developed systems, models or data for proprietary or academic reasons. It can therefore be challenging to verify the reported evidence when this is unclear. Thus, the need to emphasise on methodological quality to mitigate the risk of bias. To appraise the quality of the selected primary studies, this review protocol adapted the quality assessment guidelines by [170] to create a questionnaire with Likert scale and design questions that are relevant for the review. Each question consists of 3 options (Agree, Disagree, Undecided) which have 1, 0 and 0.5

points respectively. Since there are 9 questions, the maximum points that can be accorded to each study is 9 and the studies with less than 7 points will be excluded from the review.

1. Are the research aims clear and explicit?
2. Is the context of the study well defined?
3. Is the research process comprehensibly documented?
4. Are the IS, KM approaches or Solutions (artefacts) explicitly identified?
5. Does the study adequately report the development process of the artefacts?
6. Is the reporting clear and coherent in general?
7. Is there a clear link between the study and the conclusions that were drawn?
8. Are the research findings credible?
9. Do the research findings add to the literature?

B.1.5 Synthesis/Results

This systematic literature review will report information on KM approaches and Solutions for aircraft operations and maintenance record keeping. The studies that will be included in the review are expected to be diverse in terms of study design, method, and outcome assessment. We recognise this diversity and the challenges it presents in adopting a neutral perspective or viewpoint. To mitigate the challenges, Sylvester [175] noted the usefulness of organising the literature based on key concepts to represent diverse viewpoints in IS research.

Furthermore, we have determined that synthesis by aggregation or meta-analysis will not be carried out as it is mainly useful for quantitative studies that are homogeneous and sufficiently similar for the results to be combined [21]. Instead,

we refer to Rousseau et al. [176] synthesis classifications and approach the process through a critical realist approach by means of synthesis by explanation. The data is envisaged to be tabulated based on the review questions in addition to taxonomic maps, groupings, and clusters.

The process of writing the review will be primarily guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [177]. The findings and conclusions resulting from this research will be prepared for submission to a peer-reviewed journal and presented in technical meetings.

Appendix C G-DOCB Documents Indexing

Table C-1 G-DOCB documents

S/N	Box Number	Primary Documents
1	11054000	Special Check
2	11054007	Cabin Defects
3	11053995	Cabin Defects
4	11053998	Airworthiness Control CAA Approval Certificate of Maintenance Review
5	11053988	Airworthiness Control CAA Authorised Release Certificate
6	13010625	Daily, Weekly, monthly checks Authorised Release Certificate
7	11054017	Store Issue Notes Job cards
8	11211839	Tech Logs Cabin Logs
9	11054015	Aircraft Tech Report Acceptable Deferred Defects
10	11053997	Job Cards Acceptable Deferred Defects
11	11053993	Aircraft Tech Report
12	11054004	Job Cards Acceptable Deferred Defects
13	11054033	Cabin Tech Report Cabin Acceptable Deferred Defects Acceptable Deferred Defects Component Change Records
14	11054011	Job Cards
15	11053985	Aircraft Log Book

		Component Repairs Time Limited Checks
16	11053989	EASA Form 1
		Component Change Records Engineering Order
17	11054039	Job card
18	11054003	Aircraft Tech Report
19	13010675	Task cards
		Aircraft Maintenance Cont sheet Aircraft Parts/Materials Release list Task Cards
20	11053987	Job Cards
		Engineering Order
21	11053983	Task Cards
		Aircraft Tech Report
22	11300721	Task Cards
		Airworthiness Certificate EASA Form 1
23	11304773	Task Cards
24	11053981	Job Cards D check
		Cabin Defects Aircraft Tech Report Deferred Defects
25	11054018	Component Change Records
		Job Cards
26	11054002	Deferred Defects
		EASA Form 1
27	11361194	Task Cards
28	11054027	Job Cards Stores Issue Note

		A check
29	11054025	Job Cards A Check
30	11054008	Job Cards Component Change Record B Check
31	11054009	Job Cards Special Check (4C) B Check
32	11054016	Job Cards
33	11054035	Job Cards E check
34	11054012	Task Cards Component Change Record
35	11054038	Aircraft Tech Report Cabin Tech Report
36	11054010	Aircraft Tech Report Deferred Defects Component Change Records
37	11054006	Job Cards
38	11053982	Task cards Job cards Routine Checks
39	11054021	Aircraft Tech Report
40	11054013	Aircraft Tech Report Deferred Defects
41	22144657	Task cards Aircraft Tech Report Cabin Tech Report

		Task cards
42	11117435	Deferred Defects
		Service Bulletin
43	20472758	Task Cards
44	20472670	Task Cards
		Task Cards
45	20233990	Aircraft Tech Report
46	20472762	C Check
		Task Cards
		1A Checks
47	12006721	C Checks
		Task Cards
		1A Checks
48	12006722	C Checks
		Task Cards
49	20341277	Aircraft Tech Report
		A Checks
		C Checks
		Airworthiness Review Certificate
50	12338586	Task Cards
		A Checks
		C Checks
51	12338587	Task Cards
		Aircraft Tech Report T
52	20546317	ask Cards
53	11116502	Aircraft Tech Report
54	20472759	Task cards
55	22195385	Task cards
56	12119842	Aircraft Tech Report

		Task Cards
		2A1 Check
		Aircraft Tech Report
57	20617925	Task Cards
58	20472781	Job Cards
		Aircraft Tech Report
59	20234023	Task Cards
		Job Cards
60	11053994	Deferred Defects
		Job Cards
61	11054024	A Check
		A1 Check
		A2 Check
62	13027078	Task Cards
63	11054001	Aircraft Tech Report
		Aircraft Tech Report
64	12142922	Task Cards
65	11054022	Job Cards
66	20623173	Task cards
67	11054023	Aircraft Tech Report
		Task cards
68	13010587	A Check
		Task cards
69	13010672	A Check
		Task cards
70	13010673	A Check
		Task cards
71	13010674	A Check
72	11053986	EASA Form 1

73	11054019	Job Cards
74	11054014	Job Cards
75	12479024	Aircraft Tech Report Task Cards
76	10939285	Aircraft Tech Report Task Cards Deferred Defects
77	11054028	Job Cards Component Control Log
78	11054166	Aircraft Tech Report
79	11053996	Aircraft Tech Report
80	12816373	Aircraft Tech Report Task Cards
81	11054040	Job Cards D Check
82	11054032	Job Cards Ramp Check
83	11054005	Job Cards
84	11900841	A4 Check Aircraft Tech Report Task Cards
85	11988999	Task cards Aircraft Tech Report
86	11054029	Aircraft Tech Report Component Change Record
87	11054036	Aircraft Tech Report
88	11727380	Cabin Tech Report Aircraft Tech Report
89	11054034	Job Cards E Check

		Cabin Tech Report Aircraft Tech Report
90	11235877	Task Cards
		Aircraft Tech Report Cabin Tech Report
91	11054031	Component Change Records
		Job Cards
92	11054037	Deferred Defects
		Cabin Tech Report
93	11034647	Aircraft Tech Report
		Aircraft Tech Report
94	11607994	Cabin Tech Report
		Aircraft Tech Report
95	11562395	Task Cards
		Aircraft Tech Report
96	11766062	Task Cards
		Aircraft Status Report Airworthiness Flight Test Report
97	11168615	Flight Test Certificate
98	11054030	Job Cards
		Job Cards
99	11053999	Deferred Defects
		Job Cards
100	11054026	A Check
		Task cards
101	11304774	A Check
		Aircraft Tech Report
102	12227452	Task Cards
103	13984634	Aircraft Tech Report

		Task Cards
104	11054020	Job Cards
		Aircraft Tech Report
105	11991771	Task Cards
		4C Check
106	11304775	Task Cards
107	11053984	Aircraft Log Book

Appendix D App Code Snippet for AMRO Ontology

```
    <Class IRI="#3311-FlightCompartment"/>
    <Class IRI="#3315-MasterWarningCaution"/>
    <Class IRI="#3320-PassengerCompartment"/>
    <Class IRI="#3331-CargoServiceCompartment"/>
    <Class IRI="#3340-Exterior"/>
    <Class IRI="#3351-Emergency"/>
  </DisjointClasses>
  <DisjointClasses>
    <Class IRI="#3348-LowRangeRadioAltimeterSys"/>
    <Class IRI="#3411-PitotStaticSystem"/>
    <Class IRI="#3412-AirDataSystem"/>
    <Class IRI="#3415-MachAirspeedWarningSystem"/>
    <Class IRI="#3417-AltitudeAlertingSystem"/>
    <Class IRI="#3422-EFIS"/>
    <Class IRI="#3424-StandbyCompass"/>
    <Class IRI="#3425-StandbyHorizonSystem"/>
    <Class IRI="#3428-InertialReferenceSystem"/>
    <Class IRI="#3429-InstrumentComparatorSystem"/>
    <Class IRI="#3431-VOFIILSNAVSystem"/>
    <Class IRI="#3435-MarkerBeaconSystem"/>
    <Class IRI="#3441-WeatherRadarSystem"/>
    <Class IRI="#3442-GPWS"/>
    <Class IRI="#3445-TCAS"/>
    <Class IRI="#3453-ATCSystem"/>
    <Class IRI="#3455-DMESystem"/>
    <Class IRI="#3457-ADFSystem"/>
    <Class IRI="#3462-FMCSystem"/>
  </DisjointClasses>
```

<DisjointClasses>
<Class IRI="#3512-FlightCrewOxygenSystem"/>
<Class IRI="#3522-PassengerOxygenSystem"/>
<Class IRI="#3531-PortableOxygenEquipment"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#3611-BleedAirDistributionSystem"/>
<Class IRI="#3612-BleedAirPrecoolerSystem"/>
<Class IRI="#3621-BleedPressureIndicatingSystem"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#3811-PotableWaterSystem"/>
<Class IRI="#3831-WasteWaterSystem"/>
<Class IRI="#3832-ToiletSystem"/>
<Class IRI="#3841-WaterTankPressurizationSystem"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#4911-APUPowerPlant"/>
<Class IRI="#4921-APUEngine"/>
<Class IRI="#4931-APUFuelSystem"/>
<Class IRI="#4941-APUIgnitionStalingSystem"/>
<Class IRI="#4951-APUCoolingAirSystem"/>
<Class IRI="#4952-APUBleedAirSystem"/>
<Class IRI="#4961-APUControls"/>
<Class IRI="#4972-APUIndicatingSystem"/>
<Class IRI="#4981-APUExhaustSystem"/>
<Class IRI="#4991-APULubricationSystem"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#5110-RVSM"/>

<Class IRI="#5121-FinishesProtective"/>
<Class IRI="#5131-SealsAndSealing"/>
<Class IRI="#5141-AirframeDrainage"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#5211-ForwardEntryDoor"/>
<Class IRI="#5213-AftEntryDoor"/>
<Class IRI="#5220-EmergencyExits"/>
<Class IRI="#5231-CargoCompartmentDoors"/>
<Class IRI="#5241-GalleyServiceDoors"/>
<Class IRI="#5271-DoorWarningSystem"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#5510-HorizontalStabilizers"/>
<Class IRI="#5520-Elevator"/>
<Class IRI="#5530-VerticalStabilizer"/>
<Class IRI="#5540-Rudder"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#5610-FlightComptWindows"/>
<Class IRI="#5621-PassengerCabinWindows"/>
<Class IRI="#5631-DoorsMountedWindows"/>
<Class IRI="#5641-ViewersObservWindows"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#5710-MainFrame"/>
<Class IRI="#5721-WingTips"/>
<Class IRI="#5730-PlateSkin"/>
<Class IRI="#5740-AttachFilings"/>
<Class IRI="#5751-Ailerons"/>

<Class IRI="#5752-TrailingEdgeFlaps"/>
<Class IRI="#5753-LeadingEdgeFlaps"/>
<Class IRI="#5754-LeadingEdgeSlats"/>
<Class IRI="#5755-Spoilers"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7111-EngineCowling"/>
<Class IRI="#7121-EngineMounts"/>
<Class IRI="#7171-EngineVentsDrains"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7220-FanSectionJTBDOnly"/>
<Class IRI="#7230-CompressorSection"/>
<Class IRI="#7240-CompressorSection"/>
<Class IRI="#7250-HighLowPressureTurbines"/>
<Class IRI="#7260-AccessoryDrivesGearboxes"/>
<Class IRI="#7270-RanExhaustSectionJT8DOnly"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7311-FuelDistributionSystem"/>
<Class IRI="#7321-FuelControlSystem"/>
<Class IRI="#7330-FuelFlowIndicatingWarningSystem"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7411-IgnitionPowerSupply"/>
<Class IRI="#7421-HighTensionDistribution"/>
<Class IRI="#7431-EngineIgnitionControl"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7520-AccessoryCoolingSystem"/>

<Class IRI="#7531-VariableStatorVaneActuationSystem"/>
<Class IRI="#7532-VariableBleedValveSystem"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7611-EngineControlSystem"/>
<Class IRI="#7621-EngineFireEmergencyShutdown"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7712-EngineTachometerSystem"/>
<Class IRI="#7721-EGTIndicationSystem"/>
<Class IRI="#7730-AirborneVibrationMonitoringSystem"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7811-TurbineExhaustSystem"/>
<Class IRI="#7830-ThrustReverserAssembly"/>
<Class IRI="#7834-ThrustReverserControlSystem"/>
<Class IRI="#7836-ThrustReverserIndicatingSystem"/>
</DisjointClasses>
<DisjointClasses>
<Class IRI="#7911-OilStorage"/>
<Class IRI="#7921-OilDistribution"/>
<Class IRI="#7930-OilIndicatingSystem"/>
</DisjointClasses>
<SubObjectPropertyOf>
<ObjectProperty IRI="#hasCheckAction"/>
<ObjectProperty IRI="#hasMaintenanceAction"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
<ObjectProperty IRI="#hasInspectionAction"/>
<ObjectProperty IRI="#hasMaintenanceAction"/>

```
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#hasInvestigationAction"/>
   <ObjectProperty IRI="#hasMaintenanceAction"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#hasRaiseAction"/>
   <ObjectProperty IRI="#hasMaintenanceAction"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#hasRefittingAction"/>
   <ObjectProperty IRI="#hasMaintenanceAction"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#hasReplacementAction"/>
   <ObjectProperty IRI="#hasMaintenanceAction"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#hasStatus"/>
   <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#hasTestAction"/>
   <ObjectProperty IRI="#hasMaintenanceAction"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#isCheckSubjectOf"/>
   <ObjectProperty IRI="#isActioned"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
```

```
    <ObjectProperty IRI="#isConsumableOf"/>
    <ObjectProperty IRI="#isResourceOf"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isDocumentOf"/>
    <ObjectProperty IRI="#isResourceOf"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isInspectionOf"/>
    <ObjectProperty IRI="#isActioned"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isInvestigationOf"/>
    <ObjectProperty IRI="#isActioned"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isIssueRaisedBy"/>
    <ObjectProperty IRI="#isActioned"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isRefittingOf"/>
    <ObjectProperty IRI="#isActioned"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isReplacementOf"/>
    <ObjectProperty IRI="#isActioned"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isTestSubjectOf"/>
    <ObjectProperty IRI="#isActioned"/>
```

```
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#requiresConsumable"/>
   <ObjectProperty IRI="#requiresResource"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#requiresDocument"/>
   <ObjectProperty IRI="#requiresResource"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
   <ObjectProperty IRI="#requiresPersonnel"/>
   <ObjectProperty IRI="#requiresResource"/>
</SubObjectPropertyOf>
<InverseObjectProperties>
   <ObjectProperty IRI="#hasCheckAction"/>
   <ObjectProperty IRI="#isCheckSubjectOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
   <ObjectProperty IRI="#hasInspectionAction"/>
   <ObjectProperty IRI="#isInspectionOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
   <ObjectProperty IRI="#hasRefittingAction"/>
   <ObjectProperty IRI="#isRefittingOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
   <ObjectProperty IRI="#hasReplacementAction"/>
   <ObjectProperty IRI="#isReplacementOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
```

```
    <ObjectProperty IRI="#hasTestAction"/>
    <ObjectProperty IRI="#isTestSubjectOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#isConsumableOf"/>
    <ObjectProperty IRI="#requiresConsumable"/>
</InverseObjectProperties>

<InverseObjectProperties>
    <ObjectProperty IRI="#isDocumentOf"/>
    <ObjectProperty IRI="#requiresDocument"/>
</InverseObjectProperties>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#requiresConsumable"/>
    <Class IRI="#MaintenanceTasks"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#requiresDocument"/>
    <Class IRI="#MaintenanceTasks"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
    <ObjectProperty IRI="#requiresConsumable"/>
    <Class IRI="#MaintenanceConsumables"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
    <ObjectProperty IRI="#requiresDocument"/>
    <Class IRI="#MaintenanceDocuments"/>
</ObjectPropertyRange>

<AnnotationAssertion>
```

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#InspectionTask</IRI>

<Literal>A task specifically aimed at inspecting parts, systems, or components to ensure they meet the required standards.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#InvestigationTask</IRI>

<Literal>A task that has at least 1 investigation action.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#hasCheckAction</IRI>

<Literal>A maintenance task which includes checks or verifications.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#hasInspectionAction</IRI>

<Literal>A maintenance task which involves inspecting elements of the aircraft.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#hasInvestigationAction</IRI>

<Literal>A maintenance task which involves investigating a component, system or issue.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#hasRaiseAction</IRI>

<Literal>A maintenance task which involves raising defects or issues.</Literal>

```
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#hasRefittingAction</IRI>
        <Literal>A maintenance task which includes fitting or installing parts.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#hasReplacementAction</IRI>
        <Literal>A specific maintenance task which involves replacing parts.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#hasTestAction</IRI>
        <Literal>A maintenance task which involves testing components or
systems.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#isCheckSubjectOf</IRI>
        <Literal>Indicates that the component or system is the subject of a check in a
maintenance task.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#isInspectionOf</IRI>
        <Literal>Implies that the component, system, or area is inspected in a
maintenance task.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

<IRI>#isInvestigationOf</IRI>

<Literal>Implies that the component, system, or area is the target of an investigation in a maintenance task.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#isIssueRaisedBy</IRI>

<Literal>Indicates that an issue is linked to and raised by a specific maintenance task.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#isRefittingOf</IRI>

<Literal>Suggests that the part or component is refitted or reinstalled in a maintenance task.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#isReplacementOf</IRI>

<Literal>Indicates that the part or component is used as a replacement in a maintenance task.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#isTestSubjectOf</IRI>

<Literal>Indicates that the component or system is subjected to testing in a maintenance task.</Literal>

</AnnotationAssertion>