Ensuring the Trustworthy Development of AI-Based Applications Compatible to Future EASA Regulations

Master thesis at Ulm
University of Applied Sciences
Department of Computer Science
Degree Program Intelligent Systems

Presented by
Friedrich Werner

January 2025

1. Reviewer: Professor Dr. Christian Schlegel
2. Reviewer: Professor Dr.-Ing. Philipp Graf

Declaration of Authenticity

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated.
I have only used the resources given in the list of references.

Ulm, January 28, 2025 Friedrich Werner

Abstract

Due to the increased capabilities of artificial intelligence (AI) in recent years to solve a wide variety of tasks, the aviation industry sees the introduction of AI into systems as a viable and important solution for current and upcoming challenges. However, due to the paradigm change to a data-driven development, the previous certification framework relying on a requirement-driven development is unsuitable for certifying AI-based systems. To solve this issue, the European Union Aviation Safety Agency is currently developing guidelines for the development of certifiable AI-based systems. In its guidelines, EASA describes different objectives that developers must achieve to provide evidence that the AI system is trustworthy and can be certified. One central part of this trustworthiness argument is the learning assurance framework. The learning assurance is EASAs equivalent to the development assurance for traditional software development. Central to the learning assurance are the definitions of a Concept of Operations (ConOps), an Operational Domain (OD), and an AI/ML constituent Operational Design Domain (ODD). The latter is defined to describe the operating conditions for the AI/ML constituent and to be used for the data management. While EASA specifies certain properties concerning these concepts, no guidance for developers on the format or methodology to define these concepts is given. Therefore, this thesis introduces a methodology to define these concepts, satisfying the requirements outlined by EASA. Concerning the concepts of the OD of the overall system and the AI/ML constituent ODD, a tabular definition language is introduced based on ISO 34503:2023. Furthermore, processes were introduced to define the different necessary artifacts. For the specification process of the AI/ML constituent ODD, different preexisting steps were identified to incorporate the data-specific considerations, that must be included in the AI/ML constituent ODD. These steps include the projection of system-level attributes to AI/ML constituent attributes based on the perception of the system and the identification of domain-specific concepts for the AI/ML constituent. To validate the methodology, it was applied to the PYCASX system, an open-source implementation of an airborne collision avoidance system, where the goal is to introduce machine learning to reduce the memory footprint. Using the proposed methodology, a ConOps and an OD for the PYCASX system were defined, and individual AI/ML constituent ODDs for the two subsystems, VCAS and HCAS. The methodology proved it was able to produce an AI/ML constituent ODD of finer detail compared to other ODDs defined for the same airborne collision avoidance use case. Furthermore, it was shown that most of the requirements for the considered objectives of EASA can be achieved. Thus, the proposed novel framework is an important step toward a holistic framework following EASA's guidelines contributing significantly to current research.

Acknowledgments

I want to express my gratitude to Prof. Dr. Christian Schlegel, for his support as my supervisor. His guidance and feedback helped to shape the research progress of my thesis. I am also grateful to Prof. Dr.-Ing. Philipp Graf, for taking over the role as my second supervisor.

I am also indebted to M.Sc. Johann Maximilian Christensen and M.Eng. Thomas Stefani, my supervisors at the DLR, for their constant support and openness to share their expertise. Furthermore, many thanks to all colleagues at the DLR Institute for AI Safety and Security for all the insightful conversations. Additionally, I would like to thank the DLR very much for the opportunity to gain experience in a state-of-the-art field of research.

Lastly, I'd like to thank my family for their continuous support throughout this thesis and my studies.

Contents

List of Figures

List of Tables

Acronyms

ACAS	Airbone Collision Avoidance System.
ADS-B	Automatic Dependent Surveillance - Broadcast.
AI	Artificial Intelligence.
AIR	Aerospace Information Report.
ARP	Aerospace Recommended Practice.
BER	Berlin Brandenburg Airport.
ConOps	Concept of Operations.
CPA	Closest Point of Approach.
DO	Document.
EASA	European Union Aviation Safety Agency.
ED	EUROCAE Document.
EUROCAE	European Organization for Civil Aviation Equipment.
FAA	Federal Aviation Administration.
FDH	Friedrichshafen Airport.
GNSS	Global Navigation Satellite System.
HCAS	Horizontal Collision Avoidance System.
ICAO	International Civil Aviation Organization.
IFR	Instrument Flight Rules.
LUT	Lookup Table.
MDP	Markov Decision Process.
ML	Machine Learning.
MOSP	Minimum Operational Performance Standards.
MTOW	Maximum Take-off Weight.
MUC	Munich Airport.
OD	Operational Domain.
ODD	Operational Design Domain.
OP	Operating Parameter.
OSED	Operational Services and Environment Description.
RTCA	Radio Technical Commission for Aeronautics.
SAE	SAE International.
sd	Standard Deviation.
SysML	Systems Modeling Language.
TCAS	Traffic Alert and Collision Avoidance System.
VCAS	Vertical Collision Avoidance System.
VFR	Visual Flight Rules.
WiP	Work in Progress.

1 Introduction

Artificial Intelligence (AI) and Machine Learning (ML) have become increasingly capable of solving a wide range of tasks and are therefore being adopted by an increasing number of industries in recent years. One of those industries is aviation, an industry governed by one of the strictest certification processes. Still, there are no certification processes available for AI-based systems in aviation. To solve this issue and create a certification process for those systems stakeholders from the industry, the scientific community, and the regulatory authorities are working together [1].

In the European Union, the European Union Aviation Safety Agency (EASA) is responsible for the regulation and certification of the equipment used in aircraft [2]. The hurdle for the certification of such equipment is stringent, as ensuring the safety of the equipment is one of the highest priorities in the aviation industry [3]. For the development of traditional software systems, standards such as ARP4754A [4], DO-178C [5], or DO-200B [6] are used to ensure a system is compliant to the regulatory requirements. However, it has been identified that these existing standards are insufficient for the development of AI-based systems [7]. These insufficiencies in the existing standard are caused by the paradigm change from a requirements-driven development to a data-driven development used to develop AI-based systems [8]. Therefore, to allow the application of AI-based systems in aviation, EASA in cooperation with other stakeholders is currently developing a guidance framework to ensure the trustworthiness of AI-based systems [1], [8].

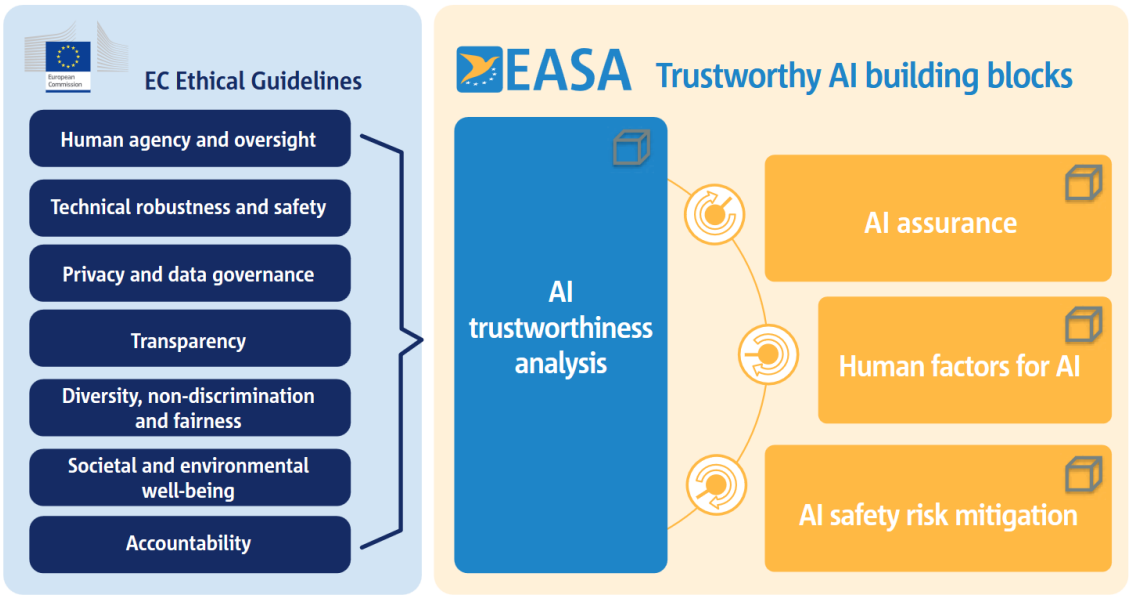


Figure 1.1: The trustworthy AI building blocks according to EASA. Taken from [1].

As shown in Figure 1.1, EASA identified four building blocks that are necessary to achieve the European Commission ethical guidelines' definition of AI trustworthiness and to enable the use of AI-based systems in aviation. The *AI trustworthiness analysis* framework, is a set of assessments

that characterizes the AI application. The framework of the *human factors for AI* considers the human factors that are connected to the usage of AI, and the *AI safety risk mitigation* framework introduces steps to deal with the uncertainty of AI [8]. Ultimately, the *AI assurance* block addresses the required guidance for the development and post-ops, specific to AI-based systems. The most important component of an AI-based system that distinguishes it from traditional software systems is the AI/ML constituent. This constituent contains the machine learning inference model. For this AI/ML constituent, an ODD must be defined according to EASA, capturing the operating conditions under which the AI/ML constituent must function as expected [8]. Furthermore, this ODD must also provide "a framework for the selection, collection, preparation of the data" [8, p. 17], that is used when developing the ML inference model. This ODD is therefore crucial, as it is a key component in the developer's argumentation in the learning assurance process, which is part of the AI assurance. This concept is introduced, as EASA acknowledges that for the development of AI/ML components, current development assurance methods are not sufficient [8]. The main reason for this insufficiency of current development assurance methods is "the paradigm shift from programming to learning" [8, p. 9]. Therefore, EASA developed concepts such as the AI/ML constituent ODD, to allow the Safety-by-Design development of AI-based systems. To achieve the goal of safety by design, EASA outlines different objectives that must be achieved by the developer when developing an AI-based system. However, there are currently no regulatory guidelines on the tools and methods which should be used to achieve the various objectives that are connected to these concepts.

1.1 Research Question

One of the many remaining challenges in developing safe AI-based systems using EASA's framework, especially the learning assurance process, is how to define the OD, the AI/ML constituent ODD, and subsequently, the neural networks that are part of the AI/ML constituent. Thus, the goal of this thesis is to research and apply different Safety-by-Design methodologies to help developers of AI-based systems achieve the objectives of the learning assurance, focusing on the AI/ML constituent ODD [8]. This is currently an open question, as EASA does not specify the tools and processes one must use to achieve the different outlined objectives of the AI trustworthiness analysis and AI assurance [8]. Therefore, the main research question this thesis aims to address, is *how to ensure a trustworthy development of AI-based applications in the scope of EASA trustworthiness guidelines?* This thesis will contribute a set of methodical approaches that allow developers to achieve the objectives of EASA's framework [8]. EASA's AI trustworthiness framework contains more than 100 objectives that developers have to consider; the thesis only considers a subset of these which are closely connected to the aforementioned concepts of the OD and ODD. The considered objectives are listed in Table 1.1. Based on these considered objectives, the research question can be split into three distinct problems, namely:

1. How can developers define a Concept of Operations, an Operational Domain, and a functional decomposition for the AI-based system? This is necessary to fulfill the requirements of the objectives CO-01, CO-02, CO-04, and CO-06.

2. Based on the artifacts produced by the first research problem, how can developers define an AI/ML constituent Operational Design Domain? This second research problem is of interest as it is required for the fulfillment of the requirements of objective DA-03.

3. How can the AI/ML constituent architecture be designed based on the defined AI/ML constituent ODD and the requirements allocated to this AI/ML constituent?

Table 1.1: The objectives of EASA's trustworthiness frameworks [8] that are considered for the methodology of this thesis.

Objective	Description
CO-01	"The applicant should identify the list of end users that are intended to interact with the AI-based system, together with their roles, their responsibilities [...] and expected expertise [...]."
CO-02	"For each end user, the applicant should identify which goals and associated high-level tasks are intended to be performed in interaction with the AI-based system."
CO-04	"The applicant should define and document the ConOps for the AI-based system, including the task allocation pattern between the end user(s) and the AI-based system. A focus should be put on the definition of the OD and on the capture of specific operational limitations and assumptions."
CO-06	"The applicant should perform a functional analysis of the system, as well as a functional decomposition and allocation down to the lowest level."
DA-03	"The applicant should define the set of parameters pertaining to the AI/ML constituent ODD, and trace them to the corresponding parameters pertaining to the OD when applicable."
DA-06	"The applicant should describe a preliminary AI/ML constituent architecture [...]"
LM-01	"The applicant should describe the ML model architecture."

By answering these three problems, the goal is to introduce a set of processes and tools that developers can use in the development of an AI-based system to conform to the regulatory guidelines of EASA, answering the proposed research question.

1.2 Thesis Structure

This thesis is composed of seven chapters. The current chapter provided an introduction to the topic and the research question this thesis will answer. In chapter 2 all necessary background information required to understand the thesis is introduced. The subsequent chapter, chapter 3, outlines the current state of the regulatory and scientific approaches to the issue of safe and certifiable AI-based systems in aviation. The main focus of this chapter is on the research, that surrounds the *EASA Concept Paper: Guidance for Level 1 & 2 machine learning applications* [8]. The developed methodology to answer the research question is introduced in chapter 4. On the one hand, this includes a specification of the formalism to define the OD and the AI/ML constituent ODD. On the other hand, processes are introduced that can be used by developers to define the OD and AI/ML constituent ODD in a structured approach for their individual use cases. The focus is on the definition of the AI/ML constituent ODD. Chapter 5 introduces the validation of the methodology by applying it to the airborne collision avoidance use case. In the validation, it is measured if the defined processes and formalisms adhere to the requirements that are stated in the objectives of EASA. In chapter 6 the findings of the previous two chapters are discussed. This includes the advantages and limitations of the introduced methodology, as well as a discussion about the applicability of some of the considered objectives. The last chapter, chapter 7, summarizes the findings and gives an outlook for future research.

2 Theoretical Background

This chapter introduces additional background information, that is useful for the understanding of the thesis. This includes the certification framework which is currently used for the development of systems in aviation, with a focus on software systems. In addition, an overview of the current EASA guidance paper for the development of AI-based systems, the *EASA Concept Paper: Guidance for Level 1 and 2 machine learning applications* [8], is given in section 2.2. In this section, the main focus is on the novel learning assurance concept, as it is the primary foundation on which this work is built. The last section introduces the airborne collision avoidance use case. This includes the systems of TCAS II and the new standard ACAS X.

2.1 Certification in Aviation

In aviation, safety-critical systems and components must be certified, before they can be employed. In Europe, the European Union Aviation Safety Agency has the responsibility to certify and approve aviation systems [3]. For the certification, EASA defines regulations that have to be fulfilled by the developer of a system. Depending on the system, different regulations apply, for example, regulation CS-25 [9] applies for the certification of large airplanes. However, while EASA specifies the regulations that have to be followed by the developers it does not specify how to achieve those. Rather, EASA publishes advisories on acceptable means of compliance and guidance material with which a developer can provide evidence that the developed system or component adheres to the relevant regulation. The development of acceptable means of compliance and guidance material is supported by the development of industry standards by agencies such as the European Organization for Civil Aviation Equipment (EUROCAE) [10]. As shown in Figure 2.1, one key framework used in aviation is the ARP4754A/ED-79A standard which defines the system development process following the V-lifecycle. For this development process, one important input is the safety assessment which has to be conducted according to ARP4761A [11]. Depending on the severity of a failure for system function, a development assurance level is assigned. This assigned criticality level is then used in the subsequent hardware or software item development. The higher the criticality, the higher the assigned development assurance level, which means that a developer has to achieve a larger set of required objectives. For traditional software development, one important guidance standard is the DO-178/ED-12 [5], and for hardware development, the DO-254/ED-80 [12] is an accepted means of compliance.

2.2 EASA Concept Paper: Guidance for Level 1 and 2 Machine Learning Applications

As introduced in chapter 1, AI is seen by many stakeholders in the aviation industry as an important technology in the future. Due to the change from a requirements-driven to a data-driven development, and the capability to achieve human-like performance in tasks of AI-based systems, EASA acknowledges the potentials and risks of AI-based systems in aviation [14]. As a result of these changes that AI-based systems bring with them compared to traditional software, processes such as certification,

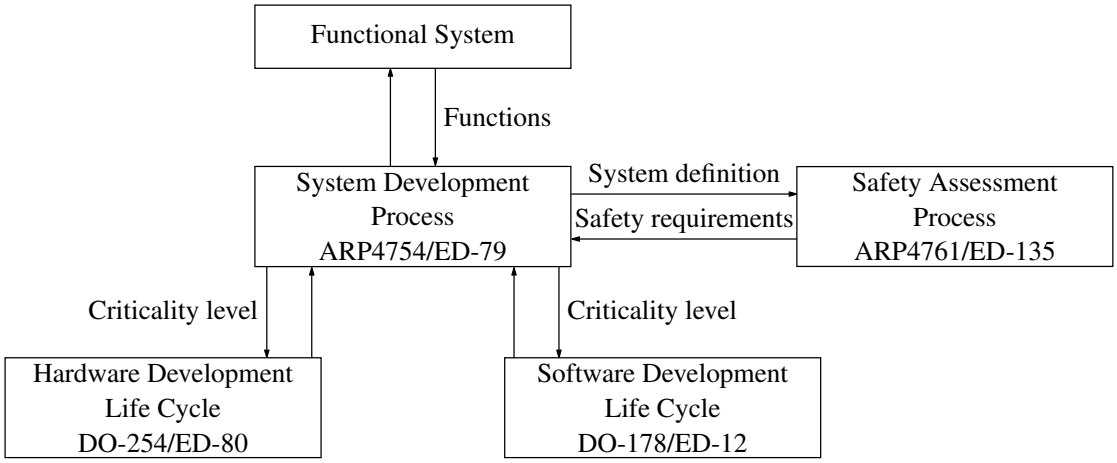


Figure 2.1: Important standards used in the development of an aviation system to provide evidence to meet the regulatory requirements. Adapted from [13].

rulemaking, and standardization are affected and have to be adapted [14]. Therefore, as early as 2018 EASA started to develop an approach to how AI can be introduced into the aviation sector [14]. One recent key publication by EASA is the concept paper "Guidance for Level 1 & 2 ML applications"[8], which was published in March 2024. This concept paper is a revision of the initial concept paper "First usabele guidance for Level 1 ML applications"[8] and EASA expects that this guidance for Level 1 & 2 AI/ML applications will be finalized in 2026 [14]. With these concept papers, EASA introduces technical objectives and organization provisions that it sees necessary for the approval of Level 1 and 2 applications. In EASA's terminology, Level 1 applications are AI/ML-based systems that provide "assistance to humans", Level 2 systems provide "human-AI teaming", and Level 3 are systems of "advanced automation". Importantly, EASA also introduces a terminology for the AI-based system and its components. As shown in Figure 2.2, an AI-based system can contain *traditional subsystems* and *AI-based subsystems*. The former describes subsystems, which are only based on traditional software or hardware items. The latter describes subsystems, which can also contain traditional software or hardware items, but importantly contain at least one AI/ML constituent to fulfill its allocated function. This AI/ML constituent is composed of one or multiple ML inference models. In addition, the constituent also contains all pre- and post-processing items that are necessary to support the ML inference models.

As previously explained, the introduced guidance contains four building blocks for the development of an AI-based system, see Figure 1.1. The following sections introduce these different building blocks with the main focus on the AI assurance.

2.2.1 AI Trustworthiness Analysis

The AI trustworthiness analysis contains a characterization and different assessments of the AI-based system. In the AI-based system characterization, the goal is to characterize the AI application by considering the whole system. This includes the definition of the high-level task(s) and the AI-based system definition, which determines what the AI-based system does and with whom it interacts. Furthermore, this also includes the description of a Concept of Operations (ConOps) for the system. This ConOps provides a refined description of the AI-based system and a description of the operating environment in which the AI-based system will operate. Also included in the characterization is the functional analysis of the system, with the specialty of highlighting the functions that are

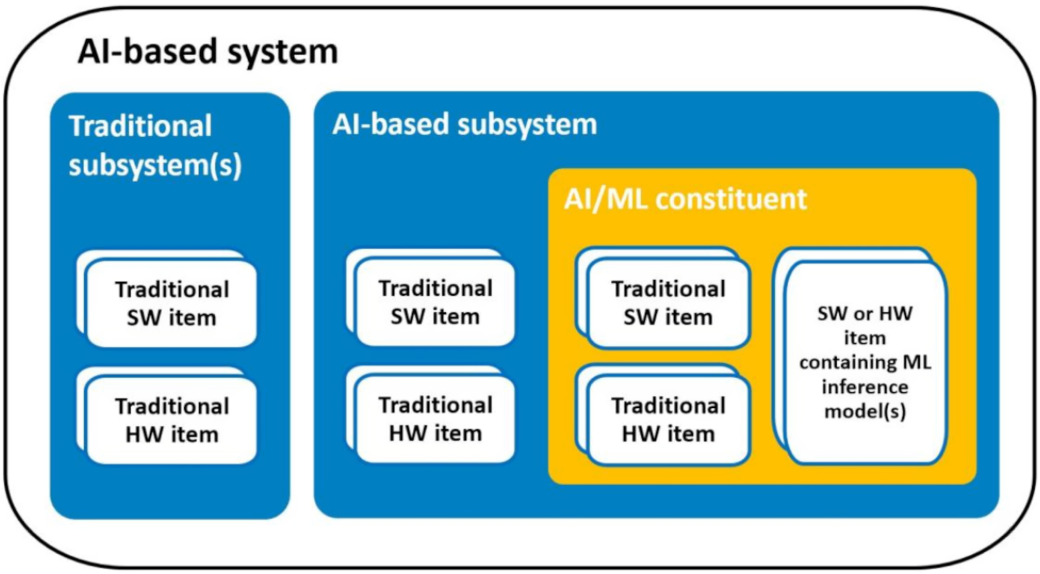


Figure 2.2: The different components of an AI-based system according to EASA. Taken from [8].

implemented using AI or machine learning. The last important part of this characterization is the classification of the AI application. In this classification, an AI level is assigned to the AI-based system depending on the allocated function to the system and the authority the end-user has over the AI-based system. The possible range of classification ranges from level 1A, "Automation support to information acquisition" [8, p. 27], to level 3B, "Non-supervised automatic decision and action implementation" [8, p. 27]. This classification of the system determines, the set of objectives that have to be fulfilled by a developer for the certification of an AI-based system.

In addition to the AI-based system classification, an ethics-based assessment of the system must be conducted. In this assessment, the different gears of the Assessment List for Trustworthy AI [15] must be taken into account. Additionally, a safety assessment and a security assessment of the AI-based system must be conducted by the developer.

2.2.2 AI Assurance

With the AI assurance, EASA addresses the specific guidance required for the development of an AI-based system. This consists on the one hand of the development and post-ops AI explainability, and on the other hand of the learning assurance.

The development and post-ops AI explainability is required to include all needs of the stakeholders in the development and post-operational phase [8]. In EASA's terms, explainability can either describe, the "information required to make an ML model interpretable for the users" [8, p. 14] or the "information for the end user on how the system came to its results" [8, p. 14]. This explainability is required to allow a greater insight into the inner workings of the AI-based system. Such an explainability might be required by stakeholders such as the certification authorities or the safety investigators [8].

The learning assurance is a new concept introduced by EASA, with the goal to define a development assurance method that can be applied to the development of an AI/ML constituent. EASA defines the learning assurance as all actions required to ensure, with an adequate level of confidence, that errors have been identified and corrected in the learning process to satisfy the required level of performance

for the AI/ML constituent. The level of performance also includes the robustness and generalization capability of the AI/ML constituent [8]. The purpose of this concept is to open the AI black box as much as possible, to allow confidence that the AI/ML constituent fulfills its functionality. EASA proposes the W-shaped process for the implementation of the learning assurance. This process consists of nine steps as shown in Figure 2.3. The step of the *AI/ML constituent requirements management* contains all preparatory measures to define the requirements necessary for the AI/ML constituent design phase. The *data management* contains all the steps needed to create the datasets and the *learning process management* includes preparatory steps for the training of the ML model, such as the definition of the metrics and the model selection strategy. In the steps of the *model training* and the *learning process verification*, ML models are trained, and the best-performing model is evaluated on the test dataset. The following *model implementation* is then concerned with the implementation of the ML model into the deployment software and hardware platform. The last three steps of the assurance process are in place to perform a verification of the steps of the model integration, the data management, the learning process verification, and the AI/ML constituent requirements.

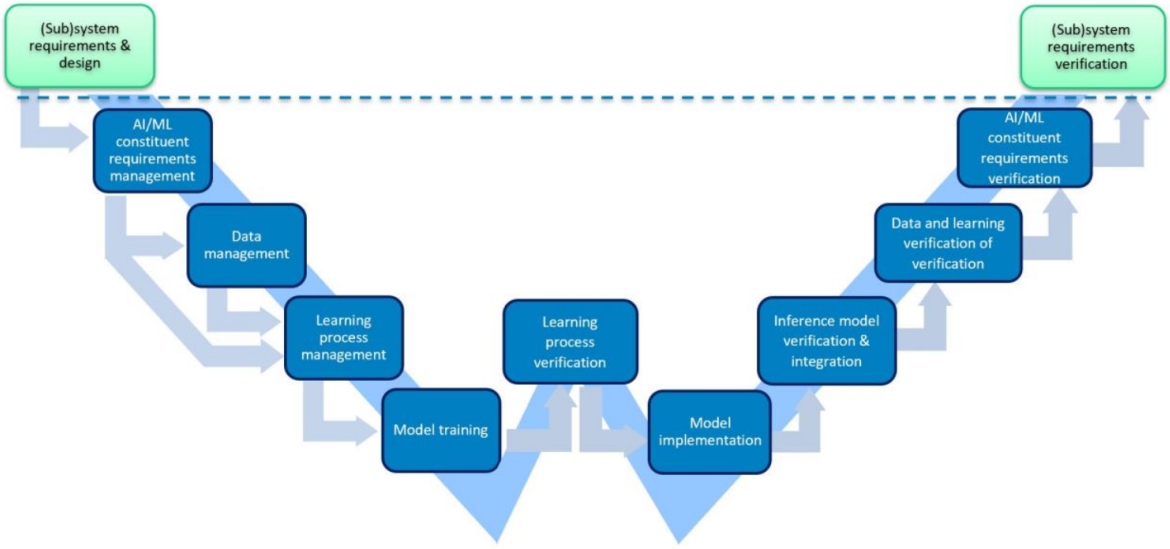


Figure 2.3: W-shaped learning assurance process [8]. All steps below the dashed line are part of the learning assurance process. Above the dashed line, traditional development assurance processes are applied. Taken from [8].

In this assurance process, one important concept introduced by EASA is the AI/ML constituent Operational Design Domain (ODD). With the AI/ML constituent ODD, a developer must define the operating conditions under which the AI/ML constituent is expected to function as intended. However, the AI/ML constituent ODD must also be "a framework for the selection, collection, preparation of the data during the learning phase, as well as the monitoring of the data in operations" [8, p. 17]. Therefore, this ODD has to include a finer level of detail compared to the OD that is defined at the system level, to allow the use of the ODD in the data and learning management. Importantly, for the AI/ML constituent ODD, EASA defines certain properties the ODD must be able to depict. As shown in Figure 2.4, one important distinction of the OOD compared to the OD, is the definition of additional operating parameters, which are only relevant for the ODD. This includes operating parameters that are linked to sensors, for example, such as the noise level. Also, the ranges of the operating parameters in the AI/ML constituent ODD, which exist in the ODD and OD, can either be a subset or superset

of the ranges of the same parameters in the OD. If the ODD parameter range is a subset, this means that the AI/ML constituent is restricted to only operate in this reduced range in the OD. The OOD parameter range is also allowed to be a superset, see *OP #2.1* in Figure 2.4, to improve the model performance by being trained on a wider range of available data [8]. Also, the individual ranges of

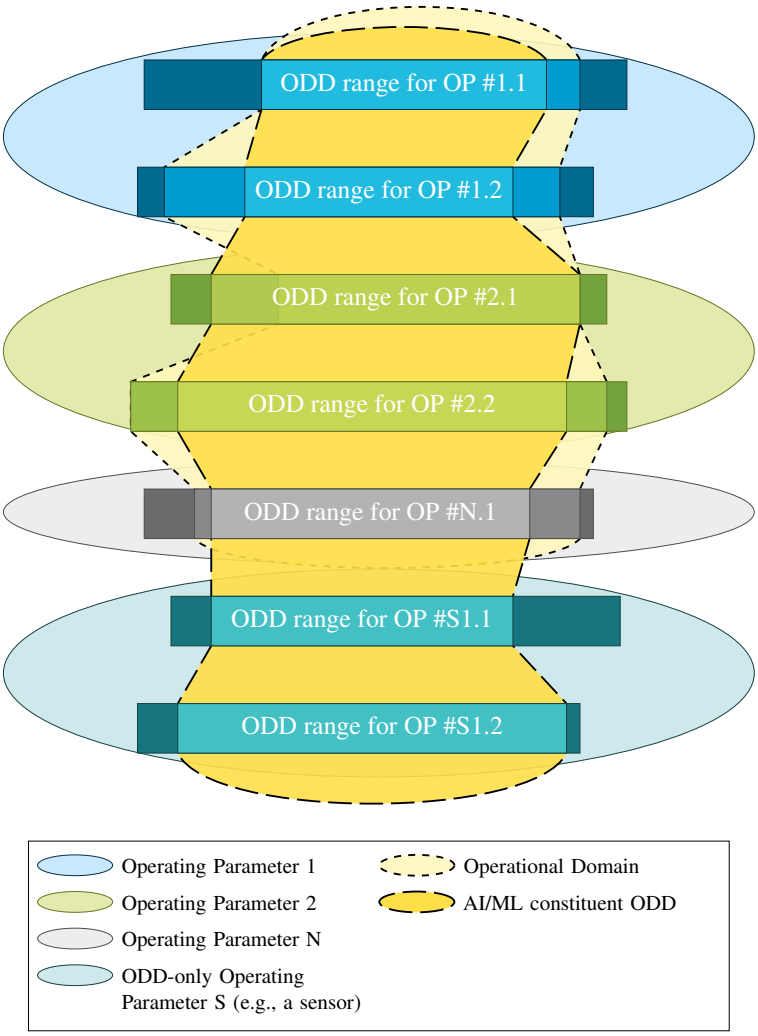


Figure 2.4: The AI/ML constituent ODD and its relation to the OD. Adapted from [8].

the different operating parameters can depend upon the ranges of other parameters. The specification of an ODD for an AI/ML constituent is important, as the defined attributes and ranges are used to construct data quality requirements and to verify if the collected dataset for the development of the ML inference model is complete and representative of the expected later operating conditions of the AI/ML constituent. Furthermore, this AI/ML constituent ODD is also used to estimate the generalization capabilities of the ML inference model and to define the behavior of the AI/ML constituent when exposed to out-of-distribution data.

2.2.3 Human Factors for AI

With the *human factors for AI* EASA introduces guidance to allow the usage of the AI-based system by the end user and that the end user builds trust into the AI-based system [8]. This is important, to enable the end user to share its authority with the AI-based system. To allow this, EASA emphasizes the need to include the AI operational explainability with which the user can understand the decisions and the reasoning behind them by the AI-based system. Such an explainability is also important due to the novel concepts of human-AI teaming, where the system and the user cooperate or collaborate to achieve goals [8]. A further consideration of human factors for AI is the error management, as AI can introduce additional errors. For example, the end user can build an over-reliance on the AI-based system, without considering their limitation. To prevent such errors, EASA outlines different objectives that a developer of an AI-based system has to consider.

2.2.4 AI Safety Risk Mitigation

EASA acknowledges that while the previous frameworks can reduce the risk of AI, they may not be able to completely reduce all identified risks for an AI-based system to an acceptable level. Therefore, the *AI safety risk mitigation* framework, should enable a developer to define and implement mitigation strategies for these identified risks which were not addressed adequately by the previous frameworks for a specific AI-based system. A mitigation strategy could be the use of a backup system that is used as a recovery when a failure is detected in the AI-based system [8]. Importantly, EASA only allows the use of such mitigation strategies for risks stemming from insufficient coverage of explainability or learning assurance objectives, not as a general approach to managing the risks of a system [8].

2.3 Airborne Collision Avoidance System

Following a fatal mid-air collision over the Grand Canyon in 1956 [16], the development of airborne collision avoidance systems was started to prevent such accidents in the future [16]–[22]. The current state-of-the-art solution is the Airborne Collision Avoidance System (ACAS) II and its reference implementation, the Traffic alert and Collision Avoidance System (TCAS) II [21]. Under current rules, TCAS II is required for all aircraft with a maximum take-off mass (MTOW) over 5700 kg or approved to transport more than 19 passengers under International Civil Aviation Organization (ICAO) and European Union Safety Agency (EASA) regulations [18], [19], or more than 30 seats under Federal Aviation Administration (FAA) regulations [20]. To prevent mid-air collisions, TCAS II is designed to give on the one hand traffic advisories to alert pilots of an intruder aircraft, and on the other hand resolution advisories with different vertical rates for the pilot to follow if the intruder aircraft becomes a threat [16], [17]. These advisories are chosen based on a simple rule/heuristics-based logic [17]. While the introduction of TCAS II has decreased the risk of mid-air collisions and near mid-air collisions significantly [23], the current implementation has some drawbacks, such as generating false-positive alerts, even when the aircraft were safely separated [17]. Furthermore, the ACAS II standard is not designed for the upcoming increase of air traffic [22] and the challenges of autonomous and urban air mobility.

To resolve these drawbacks of the ACAS II standard, the new ACAS X standard is proposed and developed [22]. This new standard introduces a set of different collision avoidance standards that can be applied to different types of aircraft. As shown in Figure 2.5, these aircraft range from commercial jetliners to unmanned aircraft and helicopters. Most notable are the ACAS X_A standard [24], which is intended as a replacement for the ACAS II standard, and the ACAS X_U standard [25], which is intended

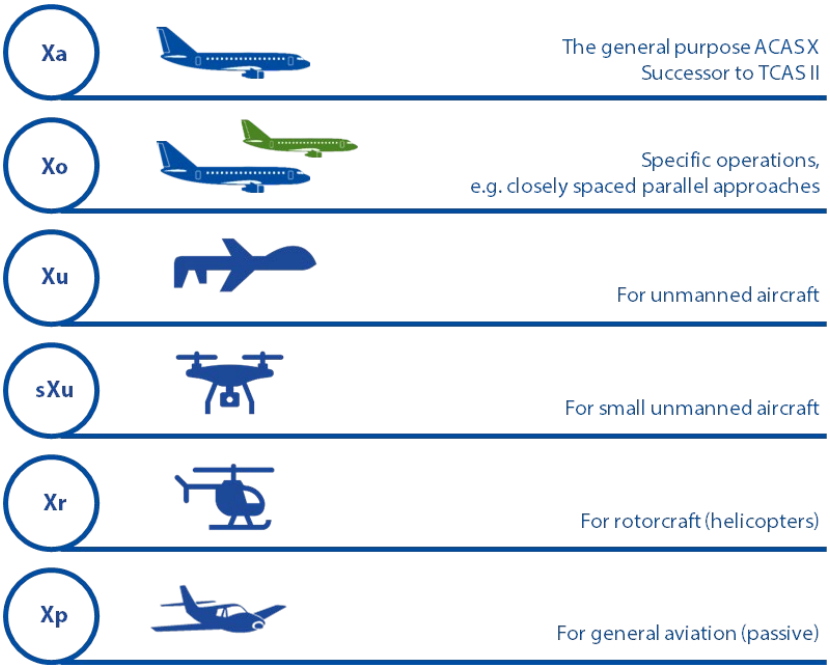


Figure 2.5: The different ACAS X variants. Taken from [17].

for the use in unmanned aircraft. One key difference of the ACAS X standard is, that the advisories are now "based upon a numeric lookup table optimized with respect to a probabilistic model of the airspace and a set of safety and operational considerations" [17, p. 23]. These numeric lookup tables are generated based on a Markov Decision Process (MDP) during the system design phase [17]. The MDPs in the design process approximate the encounter dynamics between the ownship and the intruder aircraft. For the ownship aircraft in these MDPs, different actions are available, which the ownship can choose with the goal of preventing a mid-air collision. Using dynamic programming, these defined MDPs are solved, to calculate for each state of these MDPs the cost for each available action. The action that has the lowest cost in a state is the best action for that state. All costs for all states are saved in a lookup table. In-flight for a single intruder a state distribution is determined, and for this distribution using the lookup table, the overall action cost is determined for this intruder. In addition, to these calculated action costs, an additional cost may be added for each action based on additional online costs. An additional online cost can be, for example, the action costs that originate from the coordinate between the intruder and the ownship. In the end, the action with the lowest action cost is provided to the pilots as a resolution advisory. While this new approach has multiple advantages compared to the rule-based approach of the TCAS II system [26]–[28], one drawback is the large memory footprint of the lookup tables. Depending on the ACAS X variant, the lookup tables require *at least* 4 GB of memory, therefore it is currently not feasible to store these lookup tables on current avionics hardware [26].

3 Related Work

This chapter introduces the current advances and research topics relevant to the research questions that are considered in this thesis. This includes the current developments of regulatory bodies in aviation on the certification of AI-based systems, as well as the advances of the scientific community to support these developments. The main focus is on the topics of the AI assurance and the Operational Design Domain (ODD). Furthermore, as the topic of the ODD is not limited to the aviation sector, related work from other domains is introduced as well.

3.1 Current Regulatory and Standardization Efforts For AI-Based Systems in Aviation

As outlined in chapter 1, current regulatory frameworks are insufficient for the certification of AI-based systems due to their data-driven nature. Therefore, different regulatory agencies launched projects to update or introduce frameworks, that allow the certification of AI-based systems in aviation. Two important projects are the FAA "Roadmap for Artificial Intelligence Safety Assurance" [29], and the EASA "Artificial Intelligence Roadmap" [1]. Both projects acknowledge, that the current software and hardware development guidance is insufficient for AI-based systems [1], [29]. To solve this issue, EASA is currently developing guidance material on how to address these discovered insufficiencies in its guidance and allow the certification of AI-based systems in aviation. The current state of EASA guidance is the *EASA Concept Paper: Guidance for Level 1 & 2 machine learning applications: Issue 2* [8] released in March 2024. With this newest release, EASA extended their first guidance [30] with level 2 (human/machine teaming) applications and further concretized the introduced concepts. EASA is further expanding its guidance to include level 3 (advanced automation) applications in the future. The guidance for level 1 and 2 AI/ML applications is expected to be finalized in 2026. Importantly, in their work, EASA identified four building blocks to ensure the trustworthiness of AI and its application in aviation [8]. These building blocks include the AI trustworthiness analysis, the AI assurance, the human factors for AI, and the AI safety risk mitigation. A detailed introduction to these building blocks is provided in section 2.2. In their guidance, one key concept is the W-shaped learning assurance process of the AI assurance block. This process should ensure the correctness and completeness of the data and the scenarios used in the development, and provide a sufficient level of confidence in the generalization capabilities of the developed ML models [8]. To support their guidance development, EASA also launched different accompanying research projects [31]–[33]. These projects, developed and tested methodologies for the data management, learning process management, as well as the verification of the learning process and the inference model.

In addition to the guidance of regulatory agencies, industry standards are developed, that provide processes that enable developers of systems to provide evidence, that their developed system is in accordance with the regulatory requirements. Similar to the regulatory agencies, it has been found that current existing standards, such as ARP4754A [4] or DO-178C [5], contain gaps when these are used in the development of AI or ML-based systems [7]. One identified key issue is the traceability of the requirements, and the verification of the neural networks and the training data sets. To solve

this gap standardization organizations have launched different working groups, such as WG114 from EUROCAE [34] and G34 from SAE [35], to develop new standards for AI-based systems in aviation. Most notably, ED-324 and ARP6983(WIP) [36] will introduce a process for the certification and approval of AI in safety-related products. Both standards are expected to be published in 2025.

3.2 Certification of AI-Based Systems in Aviation

To support the efforts of the regulatory and standardization organization, the research community also develops methodologies, tools, and processes to allow certification of AI in aviation. For each of the previously introduced building blocks in EASA guidelines, a wide variety of approaches and methodologies were developed to ensure that AI-based systems are trustworthy, safe, and secure and adhere to ethical guidelines. For each of these research topics, the authors of [37] and [13] provide an overview of the current advances and challenges for the certification of AI in aviation. Key challenges are the verification of neural networks, the determination of generalization capabilities, and methods to measure the data completeness and representativeness. Additionally, process models and frameworks are developed that should allow the certification of AI-based systems in the future. For example, in [10] a framework is developed for the certification of ML systems, that covers all steps from the raw data collection to the evaluation and failure probability estimation. Their developed framework is built upon a modified V-model of [38], which was adapted to fit AI-based system development. Importantly, the development of process models for AI-based systems is not isolated to the aviation domain [39]–[41]. While these do not include specific requirements from aviation regulators, challenges such as the data management and requirements specification for data sets and ML components are shared across domains.

3.3 Operational Design Domain

Due to the *black-box* nature of AI-based systems, especially neural network-based systems, new approaches are needed to ensure the safety and correctness of these systems. This is especially necessary for systems that provide high-level automation features [8]. One approach, developed to solve this issue in the automotive domain for advanced driving systems, is the concept of the Operational Design Domain [42]. An ODD is defined to specify "the specific conditions under which a given driving automation system or feature thereof is designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and the requisite presence or absence of certain traffic or roadway characteristics" [42]. Therefore, the ODD for a driving automation system is seen as a specification to which the designed system must comply [43]. The ODD is employed in the design process, the verification and validation, and the real-time monitoring of such a driving automation system [44]. Two important applications of the ODD are simulation-based testing, where testing approaches are developed to ensure, that the system will function in its defined ODD [45]–[47], and ODD monitoring, where techniques are developed to ensure, that the system is only activated and used under the specified conditions it was designed for [48]. Importantly, in the automotive sector, the ODD is defined as a hierarchical structure of different attributes, where the individual attributes describe an operating condition and the range under which the system has to function. For this hierarchical structure of the ODD, different taxonomies were developed [49]–[53]. A common structuring of these taxonomies is the subdivision into the groups of scenery, environmental, and dynamic elements. The need for such taxonomies has been identified, as only when an ODD specification is complete in regards to its underlying use case, an argument of safety can be made using the ODD [54]. In addition

to these taxonomies, strategies were developed to identify attributes when developing an ODD [43], [47], [55].

As the ODD concept is applied in other domains, such as maritime [56] and railway [57], a generalized ODD definition process for autonomous systems was introduced by [58]. Their process consists of multiple sub-processes, that define the initialization and refinement of an ODD. In the initialization, the ODD attributes of scenery, dynamic elements, and environmental conditions are identified based on the customer's expectations and already existing solutions. In the refinement, through the assessment of relevant situations, new influence factors, i.e., attributes, are identified for the ODD specification. While this approach is designed to be used in any domain, one drawback of this approach is, that it does not consider the specifics required for an ODD by individual certification agencies, such as EASA.

The idea of the ODD to specify the operation conditions, under which an AI-based system is designed to function, has also been transferred to the aviation sector. Different works have applied the ODD concept in aviation domains such as air traffic management [44], airborne collision avoidance [59]–[62], and unmanned aircraft [63]. A challenge, that was identified by these works, is the necessity to adapt and identify the taxonomies used to define the ODD for the individual use case. This is an ongoing challenge, as the aviation sector contains multiple different areas of applications, such as predictive maintenance, flight operation, and air traffic management, whereby a separate taxonomy must be defined for each area [13].

In addition, a challenge in the application of the concept of the ODD in the aviation sector is, that it has to conform to the requirements given by EASA's guidelines. For example, one requirement is, that the ConOps and system-level ODD have to be consistent with each other. An approach to achieve this requirement has been explored by [61]. Furthermore, EASA defined the concept of the ODD and how it is used differently, compared to other domains. Especially, the decomposition of the ODD into a system-level OD and an AI/ML constituent ODD requires the further adaption and development of methodologies that incorporate this distinction. Compared to the ODD definition from the automotive domain, the specialty of the AI/ML constituent ODD is the use of the ODD as a framework for the data management. Therefore, the ODD at the AI/ML constituent level is now data-centric, compared to the scenario-based ODD at the system level [64]. For this data-centric approach to the ODD, different methodologies were introduced [57], [65], [66], to enable a definition of an ODD that includes data-specific considerations. Work, such as [65], introduced a methodology of how system-level parameters can be translated into parameters of the AI/ML constituent ODD. In their approach, they developed a methodology to define an image-based AI/ML constituent ODD. Importantly, they focused on how geometric parameters of the system-level ODD can be translated into AI/ML constituent ODD parameters and how image-level features can be identified for the AI/ML constituent ODD. To represent the data distributions for attributes in the ODD, [66] developed an approach for describing these distributions in an ODD. Most notably, they applied vine copula dependence structures to describe data distributions for attributes in the ODD, where dependencies between these attributes exist. The idea to include data-specific consideration in the ODD is also explored outside the aviation domain [57], [67]. The work of [67], for example, introduced the use of ontology-based models to identify ODD attributes such as materials and sensor characteristics. This extension of the ODD to include materials and sensor characteristics is introduced by [67], as these elements are important to describe the operation conditions for a perception-based ML system. Similarly to the ODD, approaches, such as data quality requirements engineering, are used to specify requirements for the data of an AI-base system [39], [41], [68]. These approaches also recognize the need to precisely describe the requirements for the data to ensure that they are representative of the intended use case. However, compared to the ODD approach, these do not rely on the taxonomy-based

definition of the operating conditions of the AI-based system.

To the best of the knowledge of the authors of this thesis, currently, no approach exists, that unifies these different approaches into a single process for the definition of the AI/ML constituent ODD that adheres to the requirements of EASA's guidelines.

4 Methodology

With the publication of the concept paper for guidance for level 1 & 2 machine learning applications, EASA further detailed a path towards the use of AI-based systems in aviation [8]. As shown in section 2.2, EASA introduced four different frameworks, each with its unique objectives that must be considered for the development of an AI-based system. For these frameworks, EASA adapted already existing concepts for the use of AI-based systems, such as the ConOps and OD description, but also introduced novel concepts, such as the AI/ML constituent ODD, see chapter 2. Due to the novelty of the AI/ML constituent ODD and the envisioned relation to other concepts introduced by EASA, there is a lack of methodologies that could be used in the development of AI-based systems when following EASA's framework. Therefore, the goal of this section is to develop a methodology that builds upon the introduced concepts of EASA and can be used as a part of the development of an AI-based system. For the development of the methodology, the different sections introduce the considered approaches and the rationale for why an approach is chosen. At the end of this chapter, a structured approach is identified, that can be used by developers to define the ConOps and OD, a way to derive the AI/ML constituent ODD from the OD, and guidance on the development of the AI/ML constituent architecture based on the AI/ML constituent ODD. To achieve this, section 4.1 gives an overview of the considered objectives and concepts. Furthermore, section 4.1 structures the considered concepts and objectives into a step-by-step approach, to highlight the dependencies between these concepts and their corresponding objectives. Section 4.2 to section 4.6 then introduce the individual approaches for the different steps.

4.1 Overview

For the development of the novel methodology, the considered objectives and concepts from EASA's concept paper [8] are grouped into five steps. These steps are shown in Figure 4.1. The goal of these steps is to fulfill the objectives from two blocks of the EASA trustworthiness guidelines, namely from the AI trustworthiness analysis and the AI assurance. The purpose of the AI trustworthiness analysis framework is to characterize the AI application and to assess the AI application in terms of safety, security, and ethical considerations [8]. Therefore, the steps are the ConOps and OD definitions as well as a functional decomposition of the system. Completing these steps should give the stakeholders of the system a clear understanding of the capabilities and limitations of the system [8]. The steps of the learning assurance that are part of the AI assurance framework should contain approaches for the development of the AI/ML constituent and ML inference model [8]. Following these steps should help the developers build an assurance case that the AI-based system complies with a defined level of performance and the defined requirements [8]. Importantly, the AI assurance is based upon the artifacts produced by the AI trustworthiness analysis.

The ordering of these steps in Figure 4.1 is based on two factors, firstly on the objective(s) a step fulfills, and secondly on the framework these objective(s) are a part of. As steps (1) to (3) are fulfilling the objectives of the AI trustworthy analysis, these steps have to be before steps (4) to (5), as these are part of the AI assurance [8]. The ordering of steps, (1) to (3), was chosen, as the definition of

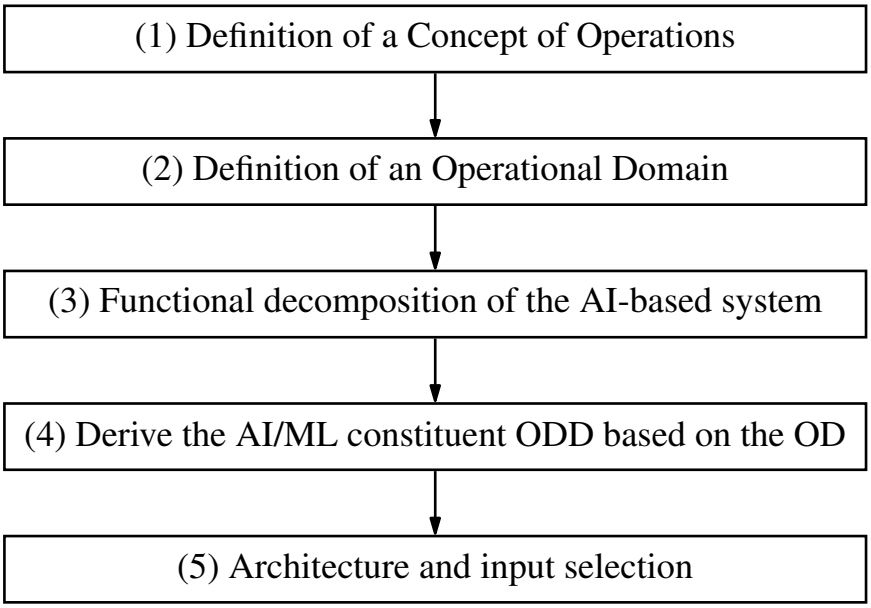


Figure 4.1: The proposed methodology with its steps and their sequence.

the concept of operations is the initial artifact that describes an envisioned system. The step of the definition of the operational domain, is after the ConOps definition, as the description provided in the ConOps builds the basis to define the operational domain. While, these two steps are in sequential order, due to their close relationship [8], an iterative definition of the ConOps and OD may be necessary. Similar to the OD, the definition of the functional decomposition is step three, as this decomposition depends on the defined high-level functions described in the ConOps. The ordering of steps (4) and (5) are predetermined based on the W-shaped learning assurance process, as the objectives of step (4) are before the objectives of step (5) in the W-shaped learning assurance process [8].

In the first step of the methodology shown in Figure 4.1, an approach is required to define the Concept of Operations for an AI-based system. This definition of the ConOps is required by EASA, see objective CO-04 [8]. For the definition of the ConOps, EASA requires the developers to take into account the potential users of the system and the task allocation pattern between the users and the AI-based system [8]. The developers must also capture these elements for the fulfillment of objectives CO-01 and C0-02. Therefore, this step finds an approach, that can fulfill these objectives. Additionally to the ConOps, objective CO-04 requires the definition of the OD. The OD must capture the operating limitations and conditions of the full AI-based system. The capture of operation limitations and conditions is not new in aviation [8], but EASA argues that for AI-based systems the formalization of the notion has to be improved. For this reason, a formal representation of an OD is created in this step by examining approaches from other industries, such as the automotive sector. In general, common elements the method needs to cover for the definition of an OD, are the clear definition of an OD taxonomy, an OD definition language, and steps to specify an OD for an AI-based system. At the end of the second step, the developed methodology should enable developers to capture an OD at the system level. Step three of the proposed methodology, shown in Figure 4.1, contains the functional decomposition of the AI-based system, that is designed to achieve the objective CO-06 [8]. This step identifies the different functions of the AI-based system and defines a preliminary AI-based system architecture for the different subsystems. Furthermore, each of the identified functions must be allocated to the different defined subsystems that are contained in the AI-based system. Additionally,

each subsystem must be classified if it is an AI/ML-based subsystem or if it is not, based on the allocated functions. Therefore, if the first three steps are completed, all objectives from the AI trustworthiness analysis considered in this thesis are fulfilled.

The next step outlines an approach for the definition of the AI/ML constituent ODD. This AI/ML constituent ODD must contain the ranges and distributions of the operating parameters for which the AI/ML constituent is designed to operate. This step fulfills part of the objective DA-03 [8], see Table 1.1, that states that a developer must "define the set of parameters pertaining to the AI/ML constituent ODD" [8, p. 53]. To achieve this, EASA outlines, that the parameters of the AI/ML constituent ODD must be traced to the parameters of the OD where possible. Furthermore, EASA outlines, in an anticipated extension of the objective DA-03, that the parameters of the AI/ML constituent ODD must be chosen based on relevant industry standards [8]. As no further guidelines are given to developers to achieve this objective, the purpose of the developed methodology is to close this gap. To achieve this, the proposed method will consider a definition of an AI/ML constituent ODD taxonomy, an AI/ML constituent ODD definition language, and the specification of a structured approach to define the AI/ML constituent ODD based on the findings of the AI trustworthiness analysis. For the definition of the AI/ML constituent ODD, the artifacts of the previous steps such as the defined ConOps and OD, build an important basis. This structured approach for defining the AI/ML constituent ODD is one of the main challenges, due to the novelty of the AI/ML constituent ODD in EASA's concept paper [8]. Therefore, for this structured approach, the intention is to extract guidelines from both literature and also other domains that use the concepts imposed by EASA or try to solve similar tasks.

The last step of the methodology, shown in Figure 4.1, introduces guidance on how the specified AI/ML constituent ODD can influence the AI/ML constituent architecture with a focus on the input feature selection for the constituent. While these steps do not fulfill a certain objective, they supply a necessary basis for other objectives that are part of the data preparation of the data management, such as DM-04 or DM-05, or are part of the learning process management, such as LM-01 or LM-02 [8].

In summary, the main objective is the introduction of a methodology that bridges the gap between the AI trustworthiness analysis and the AI/ML constituent ODD that is part of the AI assurance. This methodology should enable a developer to fulfill the mentioned EASA objectives [8], which must be met by the developer when building an assurance case to allow a possible certification of the AI-based system by EASA. Importantly, using the proposed methodology, only a subset of all objectives of EASA's guidelines are fulfilled. Therefore, the proposed methodology will not be sufficient to build a complete assurance case for an AI-based system. Nevertheless, fulfilling the considered objectives is a mandatory prerequisite when building an assurance case according to EASA [8].

4.2 Definition of a Concept of Operation

The first step in the development of any AI-based system in aviation is the definition of said AI-based system and the description of the Concept of Operations [8]. This step aims to identify all users of the system and to describe the capabilities and limitations of the system. Based on EASA's suggested objectives CO-01 and CO-02, for the system, all end users must be identified and documented that interact with the AI-based system [8]. For each of the identified end users the role, responsibility, and expertise must be recorded [8]. Furthermore, this documentation also includes the goals and high-level tasks a user intends to perform when interacting with the AI-based system [8]. Following the identification of the users of the AI-based system, the next step is the definition of the Concept of Operations for the AI-based system. The ConOps is documenting the characteristics of the AI-based system from the users' operational viewpoint [8]. Therefore, this methodology proposes the five steps

for the definition of the ConOps based on the ISO/IEC/IEEE 29148:2018 [69]. These steps consist of a description of the current system, the justification for and nature of the changes, the description of the proposed AI-based system, the definition of the task allocation pattern, and lastly the description of operational scenarios. These steps of the ISO/IEC/IEEE 29148:2018 [69] are chosen as these are common across different standards, such as ANSI/AIAA G-043B-2018 [70], to define a concept of operations for a system [71]. One step commonly included in the definition of the ConOps, excluded in the proposed approach, is the definition of the operational environment. This step is excluded, as the operational environment, is replaced by the operational domain, which is defined in the following section.

As the first step, a description of the current system must be created [69]. This description must include an overview of the provided system functionality, an explanation of the underlying technology, and a list of aviation standards that apply to the system. On the one hand, this gives other stakeholders an understanding of the current state of the problem domain [69], and, on the other hand, it also introduces the initial scope for the definition of the intended use of the system. The inclusion of relevant standards is recommended, as safety-critical systems in aviation are regularly subject to a multitude of regulations and standards. The step of the description of the current system is also an important prerequisite for the next step of the justification and nature of the changes. As recommended in ISO/IEC/IEEE 29148:2018 [69], the justification of changes must highlight the shortcomings of the current system or situation. Furthermore, the nature of the proposed changes has to be stated [69]. This is required, as it highlights the changes that will be incorporated into the updated AI-based system. Especially, the changes that are introduced with the AI component have to be stated, to communicate the expected effect on the use of the AI components.

Based on these two steps, a description of the AI-based system must be created. This description of the proposed AI-based system must contain "[t]he operational environment and its characteristics" [69, p. 82]. Furthermore, this description must also include the capabilities and functions that will be provided by the proposed system [69]. This must also include the major system components that are needed for those capabilities and functions [69]. Lastly, the description also provides the task allocation between the end users(s) and the AI-based system [8]. The task allocation must also include the interactions between the end user(s) and the AI-based system [8]. The description must enable all stakeholders to have a clear understanding of the functionality and responsibilities of the AI-based system. Furthermore, the description must be extended by the description of operational scenarios. As defined by EASA, "a scenario is: in a given context/environment, a sequence of actions in response to a triggering event that aims at fulfilling a (high-level) task" [8, p. 23]. Therefore, the first part of an operational scenario is to state the context in which the AI-based system operates. If the AI-based system is part of a higher-level system, e.g., a component of an aircraft, the context of the overall system should be introduced with the necessary depth. Additionally, relevant environmental conditions must be stated to introduce the expected conditions in which the AI-based system can function. The second part of the scenario is the introduction of the trigger event. The trigger event marks the starting point for the sequence of actions that the user and AI-based system should execute to accomplish the higher-level task or goal. Following the trigger event, the individual steps performed by the AI-based system and the human should be stated. These steps must be described to highlight the individual system functions and the task allocation between the end user and the AI-based system, that are needed to achieve the higher-level task or goal [8]. Furthermore, these steps must be described in such a way that each stakeholder, regardless of technical background, can understand the system's functionalities, responsibilities, and limitations. This is important, as this understanding of the AI-based system, especially from the end user's perspective, will be the foundation for the trust in the system [8]. In addition, these scenario descriptions must also include scenarios where the AI-based system is outside

its designed operation conditions [8], to highlight the fallback measures and limitations of the AI-based system.

It must be noted, that the methodology of the ConOps definition is only proposed at a scale to satisfy the specified objectives in EASA's AI trustworthiness analysis and to be sufficient for the following steps of said methodology. If an AI-based system is part of a larger system and a ConOps of larger scale is needed, methodologies such as ISO/IEC/IEEE 29148:2018 [69] or ANSI/AIAA G-043B-2018 [70] can be used to fulfill the requirements of other regulatory guidelines that may apply, such as CS-25 [9].

4.3 Definition of an Operational Domain

The operational domain (OD) of an AI-based system describes the operating conditions, under which it is designed to function as expected [8]. Furthermore, the OD must be in accordance with the defined ConOps for the AI-based system [8]. As introduced in section 4.1, EASA states that the capturing of the operational conditions for a system is already a practice in the aviation sector, but that this is not formalized enough for systems that will be AI-based [8]. While EASA formalized properties for the OD concept, it did not introduce guidelines on how to specify an OD in any formalized way. The automotive sector has already advanced this topic of capturing the operational conditions by standardizing different aspects, for example, the OD[1] concept is already standardized for the industry [49], [50]. Furthermore, this concept has been already used successfully for the development of a level 3 automated driving system [72]. Therefore, this approach aims to use ideas and concepts from other domains to build the OD concept for the aviation domain. This approach is selected to ease the future transfer of concepts, for example, the OD testing of an automated function between different domains. This can be of use as the aviation sector also envisions AI-based automation systems with high levels of autonomy, for instance, autonomous urban air mobility [73].

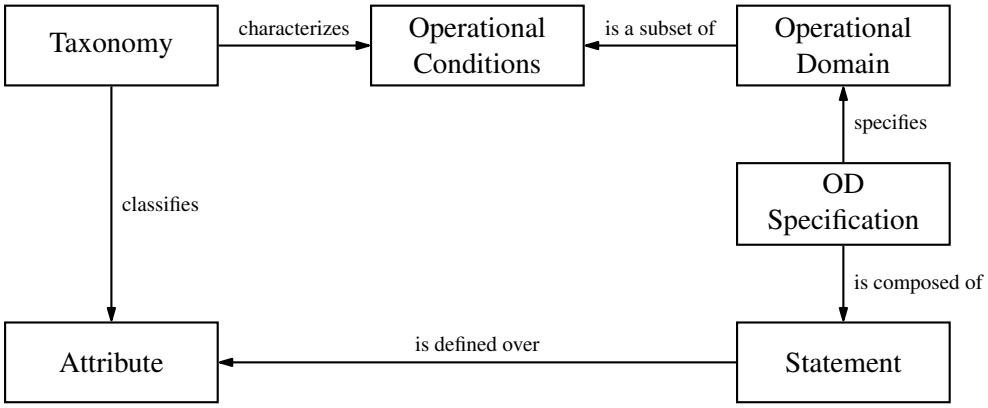


Figure 4.2: Relation between the different concepts that are connected to an operational domain. The figure is adapted from [74], which defined the relations and components for the automotive concept of an OD.

To specify the OD for an AI-based system, different concepts, and their relations are introduced in Figure 4.2. The Figure 4.2 is based on the concepts and relationships of an OD in the automotive domain [74] but is adapted in such a way, that the terminology does fit the concepts and definitions of EASA. The concept of the taxonomy characterizes the operational conditions by classifying these

[1] In the automotive domain the OD concept from EASA is known as the ODD. For consistency, EASA's terminology is used.

through a set of attributes [50]. These attributes are used in the statements to define the individual ranges of the operational conditions, as shown at the bottom right of Figure 4.2. A statement describes an operational condition that is either excluded or included in the OD specification. The collection of statements composes the OD specification for the AI-based system, i.e., the operating conditions under which the AI-based system is expected to function. Importantly, the OD is normally a subset of all possible operational conditions, as systems can be restricted to certain operational conditions due to their function and design [74]. For the statements that define the OD, a specification language is needed as this allows the communication and understanding of the OD by different stakeholders [50]. Therefore, to specify an OD for an AI-based system, both an OD taxonomy and OD specification language have to be defined [74]. The definition of an OD taxonomy is specific to the use case for which the AI-based system is built while the OD specification language should be usable for any use case. These two components are used to define a concrete OD for an AI-based system.

The following sections will first introduce a hierarchical structure for the OD taxonomy and a definition language for the OD. Secondly, an approach on how to specify a concrete OD for AI-based systems use case is outlined. An important assumption for this approach is, that the AI-based system fits the notion of an artificial agent, where the agent, i.e., the AI-based system, percepts an environment through sensors[2] and can interact with the environment with some sort of actuators[3] [75]. This assumption is important as without a defined environment for the AI-based system, capturing operational conditions will be challenging.

4.3.1 Operational Domain Taxonomy

An OD taxonomy defines the attributes that can make up the operating environment of the AI-based system and organizes them in a hierarchical structure [50]. Thus, the taxonomy must cover all possible attributes that are necessary to define the elements and conditions of the operating environment that affect the AI-based system. This is important, as an OD can only be considered complete if no safety-relevant attributes are missing [54]. Therefore, completeness is an important prerequisite to argue for a systems safety based on the OD [54]. However, due to the diversity of use cases in the aviation sector for AI-based systems [1], no single universal taxonomy can be developed that fits all use cases. For example, the domain of air traffic management, and the domain of aircraft production and maintenance, only share minimal similarities. Therefore, while the individual domains may share similar taxonomies for different use cases, across domains a single universal OD taxonomy is not feasible. Thus, the goal of the proposed methodology is to define a generic approach that builds a foundation for the OD taxonomy definition and can be applied to various use cases. For the basic structure of the OD taxonomy, three top-level attributes are proposed, namely *scenery elements*, *environmental elements*, and *dynamic elements*. These three elements are commonly used to describe the top-level attributes of an OD across different domains [50], [58]. Other approaches, especially in the automotive domain, introduce additional top-level attributes. For example, in other works [45] six different top-level attributes are proposed for the definition of an OD taxonomy for an ADS. This is suitable if a highly specialized OD taxonomy definition for a specific use case is created. However, as this methodology is designed to cover as many aviation domains and use cases as possible, the top-level structure consisting of three main groups is adopted to allow broad applicability. In addition, many of the additional introduced top-level attributes can be classified as a sub-attribute of the outlined three

[2] A sensor in this context is seen as the part of the AI-based system that can extract information from the environment, and therefore, must not be a physical sensor.

[3] Similar to the sensor, an actuator is a way with which the AI-based system can communicate with the environment. This can include anything from a physical actuator to a segmented image that is displayed to a human operator.

top-level attributes. Furthermore, initial results from the use cases of air traffic management [44] and air operations [60] show the applicability of the proposed top-level attribute structure.

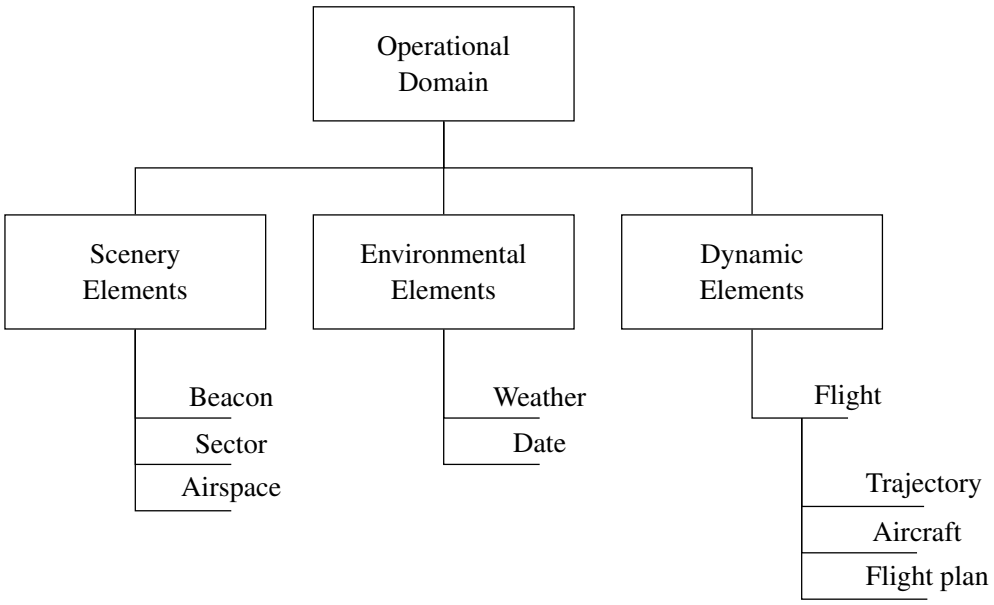


Figure 4.3: An excerpt of a hierarchical OD taxonomy using the three top-level attributes for an air traffic control use case [44].

The first top-level attribute, the scenery elements, should include all attributes that define the elements that can be considered spatially fixed in the operating environment of the AI-based system [50]. While these elements are fixed, it does not mean that they have to be in a static state. For example, as shown in Figure 4.3, airport beacons that signal the position of an airport with a flashing light [76] would be considered an attribute part of the scenery elements. Next, the environmental elements should include all attributes that describe the weather, atmospheric conditions, and all other attributes that are considered to be non-scenery elements [50]. A non-scenery element can be for example the connectivity of the system with an external infrastructure [50]. Finally, the dynamic elements describe all attributes that describe other participants in the environment and the AI-based system itself [50]. The other participants can be seen as agents that can move or change in the operating environment of the AI-based system. The term agent is used to highlight that these participants can control their movement or actions in the operating environment. As shown in the example Figure 4.3, a *Flight*, consisting of an *Aircraft*, a *Trajectory*, and a *Flight plan*, is a dynamic element in the air traffic control use case. The *Flight* is classified as a dynamic element because it acts in the defined environment and interacts with elements in it, such as other aircraft. Additionally, the dynamic elements should include attributes that may restrict the performance or capabilities of the AI-based system [50]. For example, if the AI-based system is part of an aircraft, the maximum speed of the aircraft must be defined as an attribute when the AI-based system is limited in that regard. This outlined taxonomy classification only gives a high-level guideline on the different attributes the developer has to define for its use case. This high-level view is chosen as these guidelines must apply to a vast variety of use cases in the aviation domain such as air traffic management or maintenance. Therefore, depending on the use case for some top-level attributes, no attributes might be identified.

Currently, no standard format exists for the definition of the OD taxonomy. For instance, ISO 34503:2023 [50] and BSI PAS 1883:2020 [49] define their taxonomy in a textual format where the

different attributes and possible values are listed. Other OD taxonomies that were created for a specific use case, such as [44] and [60] used a block definition diagram written in SysML. While both the textual and visual approach can have their disadvantages and advantages [77], the main objective of structuring the elements of the operational domain hierarchically can be achieved by both approaches. Therefore, both approaches can be used for the definition format of the OD taxonomy.

4.3.2 Operational Domain Definition Language

The OD definition language is an important part of an OD definition as it enables developers to capture the operational conditions consistently and accurately and also to be understandable by different stakeholders. The latter is required, as EASA sees the OD as a part of the ConOps which has to be understandable from a user's perspective [8]. In the automotive domain, the OD definition language is already standardized with the ISO 34503:2023 [50]. Therefore, for the definition language, a syntax based on the tabular approach of the ISO 34503:2023 [50] is proposed. This format was introduced by ISO 34503:2023 [50], as it allows the OD to be readable by all stakeholders connected to the system. The proposed format is shown by Table 4.1. The first column indicates the top-level attribute, i.e., in

Table 4.1: Tabular format for the specification of an OD. The taxonomy and values are based on an example from the domain of air traffic management [44].

	Top level attribute	Sub-attribute	Sub-sub-attribute	Qualifier	Attribute	Attribute value	Unit
1	Scenery	Airspace	Class C	Include	Minimum flight level	100	FL
2		Sector	Celle	Include	Frequency	[130, 140]	Mhz
3	Dynamic elements	Flight	Aircraft	Include	Type	{A320, A310}	
4				Exclude	Type	Rotorcraft	

which of the three top-level groups the attribute is classified. Based on the hierarchical taxonomy, see Figure 4.3, the following columns then describe each lower-level attribute in the taxonomy leading up to the lowest level attribute. This lowest level attribute is described in the column "Attribute". The "Qualifier" column before the attribute column describes if an attribute and its values are either excluded ("Exclude") or included ("Include") in the OD. If an attribute and its value are included then this specifies that the system can function under the stated condition. This also includes the combination of all possible variations of other included attributes and their attribute values. If the qualifier is "Exclude", then the attribute and its value are a condition in which the system is specified to not function. Therefore, in these situations, the system is not allowed to be operated or must be deactivated. Attributes that are not listed in the specification of the OD are assumed to not affect the AI-based system [50]. Therefore, not included attributes can be seen as a part of the OD, but do not have to be monitored during the operation of the system [50]. The advantage of this approach is that only the relevant attributes must be defined in the OD, all other non-relevant attributes for the operation of the system must not be defined for the OD. However, it must be ensured that each attribute that is not defined does not affect the operation of the AI-based system. These assumptions must be later validated to ensure their correctness. The value of the attribute is described in the column "Attribute value". The actual value of the attribute can be described in a vast variety of styles. As shown in Table 4.1, these can include simple numerical values (row 1), intervals (row 2), sets of categorical values (row 3), and abbreviations (row 4). For the different symbols, terms, and abbreviations used in the definition of the different attributes, a clarifying appendix should be included in the OD specification, if necessary. This can be necessary to prevent ambiguities in the understanding of the values and to clarify the meaning

behind each abbreviation. For example, the term of a rotorcraft can be defined as “aircraft that use rotating blades to generate lift, such as helicopters or gyroplanes” [78]. The last column describes the unit for each attribute.

The advantage of this approach is that this tabular format is readable for humans [50] on the one hand, but can also be easily translated into a machine-readable format on the other. Other approaches, such as the ASAM OpenODD [79] initiative, try to introduce definition formats that fit into larger frameworks that can be used for the testing and deployment of automatic driving system features [50]. While such a framework would also be desirable for AI-based systems in aviation, especially for systems with high levels of automation and authority, building such frameworks is only feasible for these use cases, where simulation-based testing strategies can be employed. Due to the variety of applications in aviation, simulation-based testing approaches cannot be realized for every application, e.g., for applications such as image-based damage detection on aircraft parts. Therefore, to allow wide applicability, the tabular format based on ISO 34503:2023 [50] is chosen for the methodology.

4.3.3 OD Specification

For the specification of an OD the *domain-agnostic and risk-based OD definition approach* [58] is used as a basis and is adapted to better fit the framework proposed by EASA [8]. The approach was chosen as a basis, as it defined the OD definition process domain independently. This is an advantage compared to other processes [43], [80], as these mainly focus on the automotive domain. As EASA specified certain aspects around the OD, these aspects are incorporated into the existing process. Furthermore, the existing process is extended with additional guidance on the identification and selection of attributes and their value ranges. As shown in Figure 4.4, the first step is to initialize the OD. The starting point in the initialization is the definition of the taxonomy for the use case, following the outlined structure from subsection 4.3.1. This initialization of the taxonomy is based on the ConOps for the AI-based system, that was developed in the previous section 4.2. The ConOps on the one hand provides a generic description of the system, and on the other hand, gives concrete descriptions of scenarios the system will be operated in. Both of these descriptions can be used to identify the attributes of the taxonomy. Furthermore, the ConOps also provides a description of the system, that will be replaced or the current situation the new system should improve. Based on these, the operational conditions of these previous systems can be identified or are already available. These operational conditions can then also be used to extend the taxonomy if they still apply to the new AI-based system. For the identification of the different elements in the operational conditions for the individual scenarios, approaches similar to the 6-layer model [55], which was developed for the automotive domain, can be employed. In the approach of the 6-layer model, the idea is to split the operating environment into the spatial layers of the road network, roadside structures, and temporary modifications, and the temporal layers of the dynamic objects, environmental conditions, and digital information [55]. The idea is, that the decomposition of the operating environment into the different layers allows a structural approach for the system developer to identify all relevant elements in the operation environment. What each individual layer describes, has to be adapted for the individual use cases, the model is applied to. For the airborne collision avoidance use case, these layers can be the aerial structures, permanent objects in the airspace, temporary modifications to the airspace, dynamic elements in the airspace, environmental conditions in the airspace, and communication infrastructure of the airspace. In addition, depending on the use case, standardization efforts may already provide an initial set of attributes for an OD taxonomy. In the automotive sector, for example, standardization efforts such as the ISO 34503:2023 [50] or the BSI PAS 1883:2020 [49] try to define an OD taxonomy, that can be used for the design of automated driving systems. As of today, we are not aware of any similar efforts in the aviation domain. However,

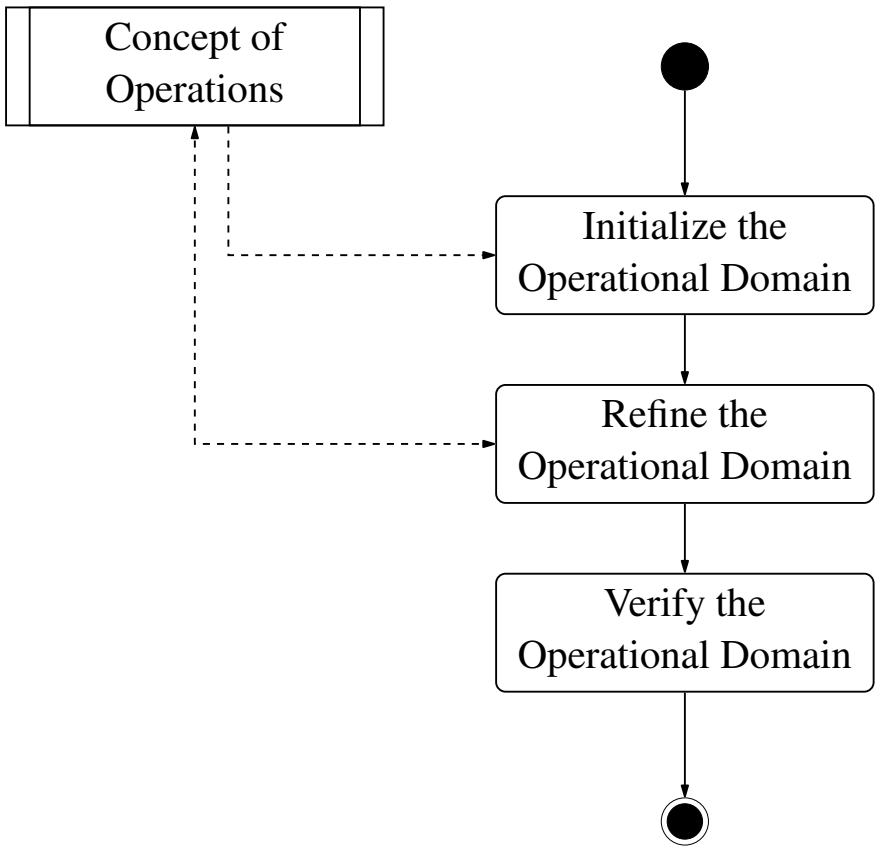


Figure 4.4: OD specification process, based on [58].

with the increased autonomy of AI-based systems and the upcoming challenges such as urban air mobility, similar efforts may be started in the aviation domain. When an initial taxonomy is defined, it can be used to define an OD specification for the system. If an initial OD taxonomy is defined, the next step is to assign values to all relevant attributes. For this initial assignment of values, the ConOps for the AI-based system should be used. An important part of the determination of an attribute range is the identification of the qualifier. This allows to determine whether an attribute from the taxonomy is relevant for the operation of the AI-based system and whether it is a range that is included or excluded from the operating conditions of the AI-based system.

If an initial OD is defined, it builds the basis for the refinement. This refinement consists of multiple possible tasks, similar to prior works [44], [58]. Firstly, based on the individual attributes, that were identified based on scenarios from the ConOps, it should be determined if these attributes require clarifications in the scenarios. This is required due to the close connection between the scenarios of the ConOps and the OD, and the requirement that the ConOps and OD must be consistent [8]. For example, an attribute that is consistent in one scenario might be inconsistent with another. These ambiguities must be eliminated to achieve a consistent ConOps and OD. Furthermore, if such inconsistencies are found, the scenarios in the ConOps must be adapted, and it must be analyzed if these adaptions in the ConOps require changes in the OD. Secondly, standards that are connected to the use case are analyzed. Depending on the proposed AI-based system, standards may already have defined operational services and environment descriptions, or minimum design and performance requirements. For example, the ED-313, which specifies the operational services and environment definition for detect and avoid in

Class A-G airspace under IFR, already limits certain characteristics of the other traffic participants. Additionally, as noted by EASA [8], the capturing of the operational conditions for a system is already a practice in the aviation sector. For example, the certification specifications of large airplanes [9], already require some of the attributes that have to be collected [8]. Thirdly, subject matter experts can be used to refine the defined OD taxonomy or the OD specification. These experts can be either from the domain of the use case or from the domain of the AI-based system. The latter is recommended to identify operational conditions that are currently not feasible with the current state of the art in AI-based systems.

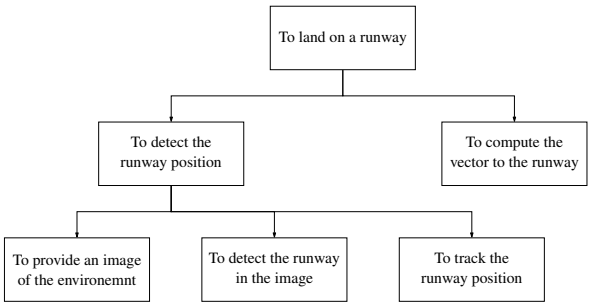
Importantly, as the last step of this refinement, again, the consistency between the OD and ConOps must be checked, as newly identified attributes in the refinement introduce additional operational conditions for the system. These newly identified operational conditions might have to be reflected in the operational scenario descriptions of the ConOps. This can lead to an adaption of the already defined scenarios or to the introduction of new scenario descriptions that include the newly identified operational conditions. The need for adaption is a bidirectional relation between the ConOps and OD specifications. Therefore, if the ConOps is adapted this induces adaptions required in OD taxonomy or specification. Adaptions such as these, are necessary as otherwise inconsistencies would persist between the OD and ConOps for the AI-based system.

If the refinement is complete, a verification and validation of the specified OD is necessary. This verification and validation should ensure, that the OD describes all necessary operating conditions to meet the requirements of the system and that the OD and ConOps are consistent.

4.4 Functional Decomposition of the AI-Based System

To achieve a functional decomposition of the AI-based system a two-step approach is proposed. The first step is a functional analysis of the system, the second step is the definition of the preliminary system architecture. At the end of the functional decomposition, an allocation of the functions to the AI/ML constituent should be achieved. This is a prerequisite needed for the development of the AI/ML constituent ODD [8].

The goal of the functional analysis is to identify the different functions that are implemented by the AI-based system. As described by the ARP4754 [4], a function should capture the behavior of a system regardless of the chosen implementation. The functions of a system can be captured by a functional tree [81]. The functional tree decomposes the system into its basic functions. Here, the analysis starts with the high-level function of the system. This high-level function is then subdivided into lower-level functions [81]. These lower-level functions are necessary for the implementation of the higher-level function. The subdivision of the individual functions is continued until a basic function is reached that cannot be split further. These basic functions can be then used to create the functional requirements of the system. As shown in Figure 4.5, the resulting hierarchical function decomposition can be either visualized by a hierarchical tree diagram [81] or by a structural list [8]. For the creation of the individual functions, a set of different rules is recommended [81]. Firstly, the function should be described using a verb and a noun [81]. Furthermore, each function should be described as general as possible to not restrict the variety of solutions [81]. In addition, a lower-level function should be a part of a higher-level function and should describe how a higher-level function can be achieved [81]. What constitutes a basic function depends on the level of detail that should be achieved by the analysis [81]. Importantly, a functional tree is not unambiguous. Depending on the expertise and choices made during the analysis, different-looking functional trees might be developed [81]. Following these steps and rules, based on the ConOps, the individual basic functions can be identified for the system.



(a) Functional tree diagram

F1 To land on a runway

F1.1 To detect the runway position

F1.1.1 To provide an image of the environment

F1.1.2 To detect the runway in the image

F1.1.3 To track the runway position

F1.2 To compute the vector to the runway

(b) Textual description

Figure 4.5: Exemplary visualization of a functional decomposition using the example of a visual landing guidance [31].

In the second step, a preliminary system architecture is introduced. This includes an overview of all components in the system and shows the information flow between these components [8]. Due to the diversity of use cases in the aviation sector, and the subsequent diversity of architectures, standardization, and regulations that can apply to an AI-based system in the different use cases [1], no single methodology can be developed for the creation of a system architecture. One specialty required by EASA when creating an AI-based system architecture, is the labeling of the different components, functions, and items, whether they are AI/ML based or not [8]. A function is AI/ML-based if it is implemented by an item that contains an AI/ML constituent. This classification is important to specify and highlight which parts of the system the AI assurance has to be applied to. In addition, processes such as ARP4754 [4] or ARP4761 [82] can be used to identify additional functional, safety, and security requirements. Based on the functional decomposition of the previous step, and the additional identified requirements, a necessary foundation for the creation of the AI/ML constituent requirements is created. However, as noted in AIR6988 [7], the ARP4754 [4] guideline has gaps when assessing AI/ML-based systems. Therefore, when dealing with an AI-based system, ARP4754 [4] might not be sufficient, and additional processes such as ARP6983(WiP) [36] have to be applied.

4.5 Definition of the AI/ML Constituent Operational Design Domain

The previous sections introduced the steps of the proposed methodology that can be used to fulfill the required objectives at the system level. For the AI assurance, EASA [8] requires further refinement and allocation of the system-level requirements to the AI/ML constituent. As introduced in subsection 2.2.2, an important concept at the level of the AI/ML constituent is the definition of the ODD. Furthermore, the AI/ML constituent ODD must also define constraints and requirements for the ML inference model and the data used to build and implement this model [8]. Lastly, the AI/ML constituent ODD must also incorporate constraints and requirements on the data that the ML inference model will be exposed to during the inference operations [8]. As the OD and the ODD are conceptually similar, the proposed methodology of this chapter is built upon the OD framework introduced in section 4.3. Importantly, due to some key differences between the OD and ODD, several adaptions must be incorporated into the methodology for the ODD definition. Firstly, the AI/ML constituent ODD parameters must be traced to the OD parameters if possible [8]. Secondly, different relations can be possible between the parameters

of the AI/ML constituent ODD and the OD [8]. These relations must include the possibility, that the AI/ML constituent ODD attributes and their ranges can be a subset or superset of the OD attributes. Thirdly, the distribution of the parameters in addition to the parameter ranges has to be recorded [8]. This is necessary, as the performance of the ML inference model of the AI/ML constituent, depends on these statistical properties [8]. Lastly, the AI/ML constituent ODD must include considerations required for the data and learning management. This last adaption is important, as the ODD must provide "a framework for the selection, collection, and preparation of the data during the learning phase" [8, p. 17]. The necessary changes to the definition approach of the OD, which are necessary due to these conceptual differences of the ODD, are presented in the following subsections.

4.5.1 AI/ML Constituent ODD Taxonomy

Similar to the OD taxonomy at the system level, the AI/ML constituent ODD taxonomy defines the attributes of the operating environment for the AI/ML constituent [8]. Therefore, a similar approach to define the attributes for the AI/ML constituent ODD was chosen as was used for the attribute definition of the OD. This includes the structuring of the operating conditions into the top-level attributes of *scenery elements*, *environmental elements*, and *dynamic elements*. As the AI/ML constituent ODD has to include specifics about the data and is specified at the subsystem level, the ODD must include aspects that the system itself can have on the data that is provided to the constituent [8], [67]. Such aspects could be, for example, sensor characteristics that are reflected in the data [8]. Therefore, the top-level attributes are extended with the top-level group of *operating parameters* [8]. This group of attributes must contain all the additional operating parameters introduced by other system components the AI/ML constituent interacts with. Importantly, while the idea of the taxonomy and its hierarchical structure for the ODD is the same as for the OD as these two taxonomies are defined at the different system levels the concretely defined taxonomies can be vastly different. This can be caused by the allocated function and system architecture that affects the AI/ML constituent ODD. This point will be further discussed in subsection 4.5.3, where the approach on how to identify attributes for the AI/ML constituent ODD is introduced.

4.5.2 AI/ML Constituent ODD Definition Language

The definition language of the OD is used as a basis for the definition language of the AI/ML constituent ODD. Therefore, the same tabular structure as in subsection 4.3.2, is adjusted with two additional columns. As shown in Table 4.2, for each attribute the distribution of the data must be defined [8] and where possible, the system component, e.g., sensor, from which the data originates should be recorded. The "Distribution" column describes the distribution of the data under which the AI/ML constituent

Table 4.2: Tabular format for the specification of an AI/ML constituent ODD. Based on the example of a drone landing scenario [66]. For better readability, the sub-attribute columns were removed.

ID	Top level attribute	...	Qualifier	Attribute	Attribute value	Unit	Distribution	Source
1	Scenery		Include	Time	[5, 12)	h	Uniform in [5, 12)	-
2			Include	Altitude	[5, 50]	m	Gaussian, mean 23, sd 7	-
3	Environment		Include	Fog	[0, 1]	-	Gaussian, mean 0.3, sd 0.08	-
4			Include	Rain	[0, 1]	-	Uniform in [0, 1]	-
5			Include	Snow	[0, 1]	-	Uniform in [0, 1]	-
6	Operating parameters		Include	Blur	[0, 2]	-	Uniform in [0, 2]	Camera

is expected to function [8]. The distribution of each parameter describes the assumed underlying distribution, from which the data were sampled independently [8]. These distributions are important to record, as the generalization capability of an ML model is approximated based upon the assumption that the out-of-sample data is sampled from the same distribution as the in-sample data [8]. The in-sample data, in this context, is the data that was used to develop the ML model, and the out-of-sample data is the data the ML model will see during the inference operation that it has not seen before [8]. This is important because the generalization capability of an ML model is a probabilistic statement that is only valid if the assumption of the same underlying distribution is met [31]. In addition, it has been shown, that if a ML model is exposed to data from a different distribution compared to the distribution used to develop the ML model, the performance of a ML model can see a significant reduction [83]. In experiments, it was demonstrated, that if an ML model is exposed to out-of-distribution data, the performance of the ML model may only be 60 % to 90 % of the performance of the ML model achieved on the test set during development [83]. Such a reduction in performance may not be acceptable for an ML model that needs to achieve a certain level of performance, especially when this performance level is required to fulfill a safety-critical function. Therefore, this distribution information is important to record in the defined ODD, as it then can be used in the ODD monitoring, to ensure that the ML model is only exposed to data distributions it was trained on and that therefore the AI/ML constituent will provide its intended behavior [8].

In this thesis, the addition of the sensor in the column “Source” is proposed to record the source of the data in the system. First, this information can be used in the data management to ensure the correctness of the data, as the knowledge about the sensor allows for the identification of potential errors that might be introduced with the chosen sensor setup during the data collection. For example, different cameras can have different properties and therefore can introduce different characteristics, such as bias, into the dataset [8]. Secondly, the inclusion of the sensors in the ODD allows for an assessment of the sensor setup [43]. This assessment of the sensor setup and its characteristics allows to determine if the sensor setup at the system level is sufficient for the defined AI/ML constituent ODD. Otherwise, an iteration and modification of the sensor setup at the system level might be required [43]. Importantly, it must be noted that not every ODD parameter must be directly measured by a sensor. For example, if a camera is used as a sensor, many ODD parameters, such as the weather, might only be indirectly measured. Therefore, these parameters might only have to be collected in the data acquisition phase to ensure that the specified conditions were captured. During operation, depending on the ML model, these parameters might not be directly used as input for the ML model in the training and in the inference operation and are only indirectly captured, for example, the weather conditions in an image. However, these attributes, still have to be included in the AI/ML constituent ODD and in the collected dataset to allow a successful training of the ML model in those different specified conditions. In addition, these attributes are important to record, as these can require special monitoring capabilities, such as out-of-distribution discriminators [31]. If these attributes are not included in ODD, the ODD monitoring would not detect that the AI/ML constituent is exposed to situations for which it was not trained. Lastly, based on the sensor information and the safety assessment at the system level through processes such as ARP 4761A [11], it is possible to additionally identify potential invalid or erroneous input data regions. Such an assessment can be used to identify scenarios and data that can be used to test the ML model behavior under these degraded conditions, and this information can be used for the definition of the ODD monitoring capability of the system [8]. Furthermore, these identified input data regions can be classified based on their relation to the ODD, and based on this classification different mitigation strategies can be used in the development of the AI/ML constituent [64].

4.5.3 AI/ML Constituent ODD Specification

The previous two sections, subsection 4.5.1 and subsection 4.5.2, defined the components for the ODD specification. As with the OD, additionally, a process is needed for a methodical development of an ODD specification for a use case or AI-based system. The proposed process shown in Figure 4.6, is based on the *domain-agnostic and risk-based ODD definition approach* [58], that again was adapted to fit to the concepts of EASA [8]. The first step in the process is to initialize the AI/ML constituent ODD. This initialization is based on the functional decomposition, the OD, and the requirements allocated to the AI/ML constituent. It defines an ODD based on the system-level information. When the initial definition of the ODD is completed, the next step is to refine the ODD to incorporate AI/ML constituent-specific information and concepts. This refinement includes specifics about the sensor setup and the AI/ML constituent architecture which has be considered for the definition of the ODD. These considerations are focused on the incorporation of specifics that are necessary for the data management and learning process management of the AI assurance process [8]. This is required, as the AI/ML constituent ODD must provide "a framework for the selection, collection, preparation of the data" [8, p. 17] that is used for the development of the ML inference model [8].

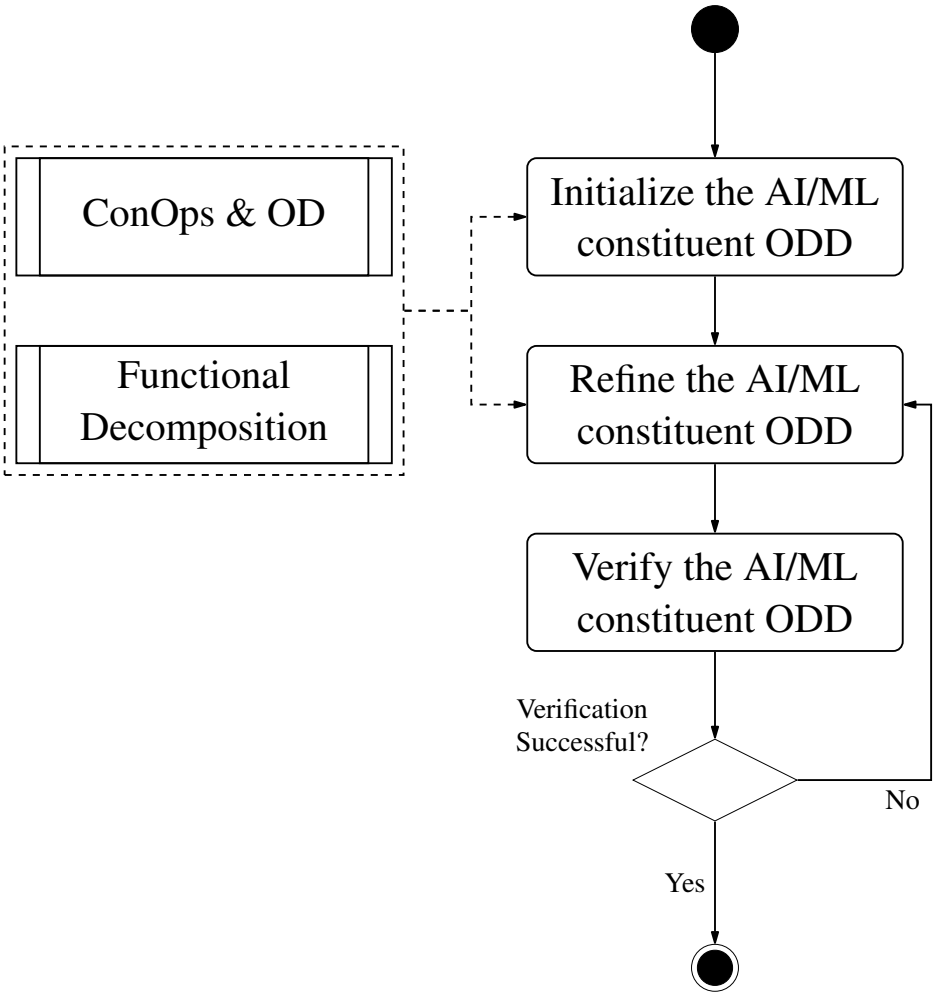


Figure 4.6: AI/ML constituent ODD specification process, based on [58].

The last step in the process depicted in Figure 4.6 is the verification and validation of the AI/ML constituent ODD. This verification and validation is outlined in the anticipated MOC DA-07 [8], where the verification and validation consists of subject matter experts reviewing the operating parameters regarding correctness and completeness. In this verification and validation, it should be ensured that the defined ODD is consistent and aligned with the OD and the functional requirements that were allocated to the AI/ML constituent [8]. If the verification and validation are successful, the specification of the AI/ML constituent ODD is completed. However, if the verification and validation are not successful, a further refinement of the AI/ML constituent based on the identified gaps is necessary.

In the following, the steps of the AI/ML constituent ODD initialization and refinement are introduced further with a higher level of detail.

4.5.3.1 AI/ML Constituent ODD Initialization

The initialization, in the domain-agnostic and risk-based ODD definition approach, is based on the identification of attributes for the ODD based on the expectations and requirements of the user(s) [58]. As the AI/ML constituent is part of a subsystem, an end user does not have direct expectations and requirements for this AI/ML constituent. Rather, at this level, the ODD has to be defined based on the ConOps, OD, functional decomposition, and requirements allocated to the AI/ML constituent. Therefore, the different artifacts from the AI trustworthiness analysis have to be used instead for the initialization of the AI/ML constituent ODD [8]. Using the functions and requirements that are allocated to the AI/ML constituent, for each OD attribute it should be determined whether it is a relevant operating condition for the AI/ML constituent. For example, if a system is designed to function at a vast variety of altitudes, but the AI/ML constituent is only designed to be used in lower altitudes near the ground, all attributes that are specific to higher altitudes can be disregarded for the definition of the AI/ML constituent ODD. To guide this identification, each parameter should be classified into a group of either a physical range or a parameter that affects the distribution of the data, such as the behavior of dynamic elements. If an attribute is classified as a physical range, based on the allocated function and requirements to the AI/ML constituent, this range can then be adapted accordingly. Using the previous example, the altitude range in the OD is restricted to 0 m to 20 000 m, but as the AI/ML constituent allocated function should only operate below 1000 m, the range of the altitude attribute in the AI/ML constituent ODD is restricted to 0 m to 1000 m. The second possibility is to classify an attribute as one that influences the distribution of the data. These attributes are important to recognize, as they will have a large impact on how the individual samples are distributed in the dataset used for the development of the ML inference model. Furthermore, in the data collection, these attributes have to be monitored to ensure that the conditions that these attributes imply will be included in the dataset. For example, if an ML system is developed to manage air traffic in a sector [44], i.e., resolving conflicts between aircraft, one important consideration is how the flight paths can be planned in the airspace structure. On the one hand, a sector can have an air traffic service route network, where flights have to be planned according to this route network by the airspace users [84]. In such an organization of an airspace sector, the paths the airplanes take in the sector and therefore the likeliness of conflicts between two aircraft are often predetermined by the route network of the sector. On the other hand, instead of a route network, the air traffic service can allow airspace users to plan routes in the airspace freely with minimal restrictions in the sector [85]. In such a free route airspace the way conflicts between two aircraft arise and their likeliness will change compared to a sector that relies on a route network [85]. Therefore, such a behavior-determining parameter must be included in the ODD for the AI/ML constituent, as depending on this parameter the distribution of the conflicts between aircraft and how they can be resolved are influenced. Using this classification for each attribute, it can later be

determined if it is relevant for the AI/ML constituent and the data management. It must be noted, that not every attribute will be distinguishable in one of these two groups and therefore, this classification will depend on the subjective judgment of the developers.

At the end of the initialization of the AI/ML constituent ODD, an ODD is created. The created ODD is a subset of the OD as the initialization only relies on the attributes and their ranges that are introduced in the OD. This will likely change with the refinement of the ODD in the next stage of the process.

4.5.3.2 AI/ML Constituent ODD Refinement

This section outlines the different steps that are required for the refinement of the AI/ML constituent ODD. The steps are based on approaches from aviation and other sectors and are combined and adapted into the context of EASA's concept paper [8].

In the previous section, an initialized version of the AI/ML constituent ODD was introduced. But as this ODD is purely based on the OD, and the OD is not sufficient for the data management and learning process management [8], the next step is the refinement of the initialized ODD into a form that is useable for the data management and learning process management. To achieve this, the projection of attributes, the determination of sensor characteristics, the identification of domain-specific concepts, and the analysis of existing data sources are introduced. These steps aim to introduce a level of detail that will be sufficient for the design and development of the ML inference model contained in the AI/ML constituent.

Projection of OD Attributes As discussed prior, the attributes of the initial ODD are based on the system-level OD. However, these attributes can be perceived differently by the AI/ML constituent than they are defined at the system level, which is due to the sensors selected in the system architecture. Therefore, the attributes and ranges from the initial ODD might have to be projected into the dimensional space which can be perceived by the AI/ML constituent. One necessary projection of attributes might be the transformation of physical attributes. These transformations can be simple unit conversions, but also more complex operations, if, for example, semantically defined attributes of the OD have to be described in concrete physical sensor values. Another possible necessary transformation can be the transformation of geometric attributes into data characteristic properties, where these properties depend on the available sensor [65]. One example of such a transformation can be the transformation of geometric OD attributes into image features when a camera-based system is used [65]. In a visual-based landing system, the OD that is described at the system level, for example, can describe the landing approach in geometric properties, such as the approach angle [65]. But as the system is based on a camera at the constituent level, these geometric properties have to be transformed into image-level features [65]. One such feature is the position of the runway in the image. This feature is necessary to ensure that only valid approach angles are in the image dataset and that the image data covers a vast variety of runway positions [65]. Also, it is possible that these transformations have to be applied in reverse, to construct geometric properties based on image-level features. Applying these different transformations to the different attributes of the initial ODD should yield an ODD that fits the perception of the AI/ML constituent and is useable in the data management.

Sensor Characteristics An important aspect of the data management is the collected data, which is used for the development of the ML inference model. For this data, it must be ensured that the characteristics of the data match the data encountered by the ML inference model during its application [39], [67]. One important aspect are properties of the data that are caused by the

characteristics of the sensors and their setups, which are used in the system. These characteristics can be the orientation, type, and position of the sensor or specific properties that are determined based on the sensor. In general, these properties can include the resolution or signal-to-noise ratio of the sensor [86]. But this can also include specific properties that depend on the type of sensor that is used in the system. For example, cameras can introduce or influence effects such as blur, contrast, or brightness in an image. Such characteristics are important to describe, as it was shown that a high level of blur or noise present in an image can negatively impact the performance of ML inference models [87]. The identification of these sensor characteristics can be based on the sensor specifications and how the sensor is installed in the overall system.

Identification of Domain-Specific Concepts As previously explained, the OD describes the operating conditions of the AI-based system, while the AI/ML constituent ODD describes the operating conditions of the AI/ML constituent. This difference in definition implies additional or different conditions compared to the defined OD, as the AI/ML constituent can be influenced differently by those different conditions. For example, if an AI-based system is designed to function in all kinds of weather conditions, and if this system uses an AI/ML constituent based on image data, additional concepts have to be identified from the weather conditions that can affect the data that the AI/ML constituent receives. In this example, these additional concepts can include all attributes that affect the image data, such as fog or precipitation, that the AI/ML constituent receives as input. If the AI/ML constituent will not be able to function under these conditions due to its design, these must be excluded from the AI/ML constituent ODD. The additional conditions will largely depend on the perception, e.g., sensor setup, of the AI/ML constituent. Furthermore, as the AI/ML constituent ODD must provide "a framework for the selection, collection, preparation of the data" [8, p. 17], the ODD should also incorporate specific concepts that impact the ability of the ML inference model to learn its designated function and can affect the performance of the ML inference model in operation [65]. For example, if an AI/ML constituent is tasked to detect a runway in an image but is only trained on data that does not include similar structures such as highways, these other similar concepts can pose a risk for the AI/ML constituent. When the AI/ML constituent is exposed to such structures during operation it can lead to possible misclassification of these structures as runways [65]. As there is no simple way to ensure the validity of the output of an ML inference model during operation, especially in case of a systematic design fault, such erroneous outputs and their cause will be challenging to discover. Therefore, the identification of such concepts is important in the design of the AI/ML constituent ODD. To identify such attributes for the AI/ML constituent ODD, other works have proposed the identification and classification of necessary, supportive, irrelevant, and false-positive concepts [65]. Necessary concepts include all attributes that are necessary for the ML model to learn its designated function, while supportive concepts are all attributes whose presence can support the ML model in learning its designated function, while their absence has no large negative impact [65]. Irrelevant concepts are all attributes that do not have an impact on the ML model [65]. The last group of false-positive concepts includes all attributes or values that negatively influence the performance of the ML model [65]. These may be objects that have been misclassified due to their similarity in an image, or an incorrect regression value due to noise or distortion in a value entered by a sensor. A challenge for these concepts is their identification in the domain of the AI/ML constituent. As ML models, especially deep learning architectures, are designed to extract patterns and features by themselves, it is unclear to the developer, which of the known or unknown patterns and features in the data have the highest significance on the ability of the ML model to learn its designated function. Nevertheless, such an identification is important to ensure that the dataset that is collected accurately reflects the later operating conditions of

the AI/ML constituent. For the identification of the attributes based on these different concepts, an approach similar to the 6-layer model can be used, which was introduced in subsection 4.3.3. Therefore, based on the same idea of the 6-layer model [55] that was used at the system level, a model of the operating conditions for the AI/ML constituent can be built. Importantly, the defined model at the system level can be used as a basis but has to be extended based on the perception of the AI/ML constituent. Furthermore, for the identification of all relevant attributes in the model that may affect the signal recorded by the sensors, the path of the signal should be mapped in this model [57]. Through thorough mapping, the different objects that interact with the signal can be discovered and described as an AI/ML constituent ODD attribute. Additionally, to accurately describe and build a model for a use case, an ontology-based domain model can be built [88]. Such a domain model represents the natural language domain knowledge in a graphical form [88]. The resulting graph consists of elements that can be classified into entities, relations, attributes, and values that describe the corresponding concepts of the real world [88]. Based on the entities and attributes of the domain model, attributes in the AI/ML constituent ODD can be defined. The values that are in relation to these entities and attributes in the domain model then can be used to define the value range of the corresponding AI/ML constituent ODD.

Analysis of Available Data Sources Depending on the use case, available data sources can be used to identify relevant aspects for the AI/ML constituent ODD [67]. Such data sources can include standards that apply to the use case, available specifications for sensors, or available data sets from the same or a similar use case. Similar to the definition of the OD, standards can already provide a list or collection of operation conditions under which the AI/ML constituent might have to operate. In the aviation sector, standards exist that specify operational services and environment definitions, or minimum aviation system performance for systems that are used to provide certain functionalities. The information or requirements that are defined in these documents can be transferred into attributes for the AI/ML constituent ODD. In addition to these standards, sensor characteristics can be identified based on the specifications that are provided for the sensors that are used in the AI-based system. As explained by the identification of the sensor characteristics section, these factors are important, as a change in the sensor characteristics during operation can affect the performance of the ML inference model. Furthermore, this identification is also important to spot inconsistencies between the sensor setup and the expected input for the ML inference model. Inconsistencies can occur due to the usage of sensors from different manufacturers. The final source of ODD attributes considered are available data sets of the same or similar use cases [67]. These discovered data sets can be explored to find additional attributes for the ODD. The exploration of a dataset consists of the identification of properties and their relation in the dataset [89]. For this exploration, different tools and methods can be employed, for example, principal component analysis to reduce the dimensionality of a dataset [90] or data visualization tools [91] to allow a better understanding of the underlying data. Furthermore, the exploration of different data sets allows to determine, whether the value ranges of already identified attributes of the ODD are representative of the data that can be encountered by the system. Using preexisting data sets alone for the definition of an ODD will likely not be sufficient, as these will only describe the characteristics of the available data sets. Moreover, it can be unclear if these match the conditions that the AI/ML constituent will encounter. Therefore, an ODD that is solely defined based on the available data sets, will not contain specifics that are introduced by the requirements from the system level where these requirements depend on the defined ConOps, OD, and overall system architecture. Nevertheless, an analysis of existing data sets or other data sources can yield information previously unknown to the developer.

Outline of the Refinement To summarize, the steps in this section are introduced for a developer as they allow a refinement of an AI/ML constituent ODD suitable for the intended use for the selection, collection, and preparation of the dataset. These steps are necessary as only if the ODD is representative of the operating conditions, an argument can be built for the safety of the system [54]. Each step can provide information about an attribute or range of an attribute that has to be part of the AI/ML constituent ODD. Different steps might be needed to identify an attribute and the accompanying range. Not part of the introduced refinement is the determination of the distribution that must be defined for each attribute. This is excluded in the refinement, as the distribution must be determined based on data samples that were collected for the dataset that is used in the development [8].

4.5.3.3 AI/ML Constituent ODD Verification and Validation

The last step in the AI/ML constituent specification process is an independent AI/ML constituent ODD verification and validation. As previously outlined, an insufficient AI/ML constituent ODD can pose a safety risk for the overall system [54]. The verification and validation of the correctness and completeness of the ODD must be done by subject matter experts [8]. In addition to the expert judgment, this verification should ensure that all relevant domain standards that apply to the system are considered [54]. Furthermore, additional data sets should be used in this verification and validation. Similar to the discovery of new attributes using data sources in the paragraph 4.5.3.2, these additional data sets can be used to ensure that the attributes and their values considered all relevant conditions [54].

4.6 Model Architecture and Input-Feature Selection

Following the definition of the AI/ML constituent ODD, the next step in the process, see Figure 4.1, is the definition of the AI/ML constituent architecture and input selection based on this defined ODD. The AI/ML constituent architecture includes all components ranging from the preprocessing, the ML inference model, and the post-processing necessary to provide the function allocated to the AI/ML constituent [8]. As the AI/ML constituent ODD is not the only influencing factor for the architecture, a mapping of the different influence factors is depicted in Figure 4.7. These influence factors were identified based on the different objectives of EASA's concept paper [8]. As shown in Figure 4.7, the *Inputs* for the AI/ML constituent that are used in the ML model are influenced by the AI/ML constituent ODD, the type of data that is collected, and the architecture. The AI/ML constituent ODD determines the different characteristics of the data to be captured based on the different conditions required based on the different ODD attributes and their ranges. Also, the ODD determines which attributes are collected based on the defined system architecture. However, the ODD does not define the data type or format that is collected. Therefore, the data can be structured or unstructured, which will then largely determine what type of inputs are feasible for an ML model. For example, if an AI/ML constituent in an airplane uses weather information, this information can be provided via satellite communication as structured data [92], or as unstructured data from weather radars [92] or cameras [93]. While all these can provide the necessary information that might be described in the AI/ML constituent ODD, due to the different types of data that is available for the same ODD different types of input features are possible. One important step for the definition of the inputs can be the application of feature engineering methodologies to extract the most useful input features [91]. All necessary feature engineering steps must be described in the AI/ML constituent architecture. However, these feature engineering steps might not be necessary if an ML model architecture is selected that incorporates feature-extracting components. The architecture for the ML model is not directly dependent on the

defined AI/ML constituent ODD, it is only influenced indirectly through the collected data types. In addition, the requirements [39] and allocated functions to the AI/ML constituent have a major influence on the architecture. The requirements determine the performance metrics the AI/ML constituent and ML model have to meet [39]. The selection of a suitable ML model architecture depends on the achieved performance that an individual ML model architecture has achieved in the different experiments [94]. These experiments include the training and evaluation of ML models with different architecture variations and hyperparameters [94]. Most importantly, the different components and the ML model have to adhere to the allocated requirements and derived performance metrics.

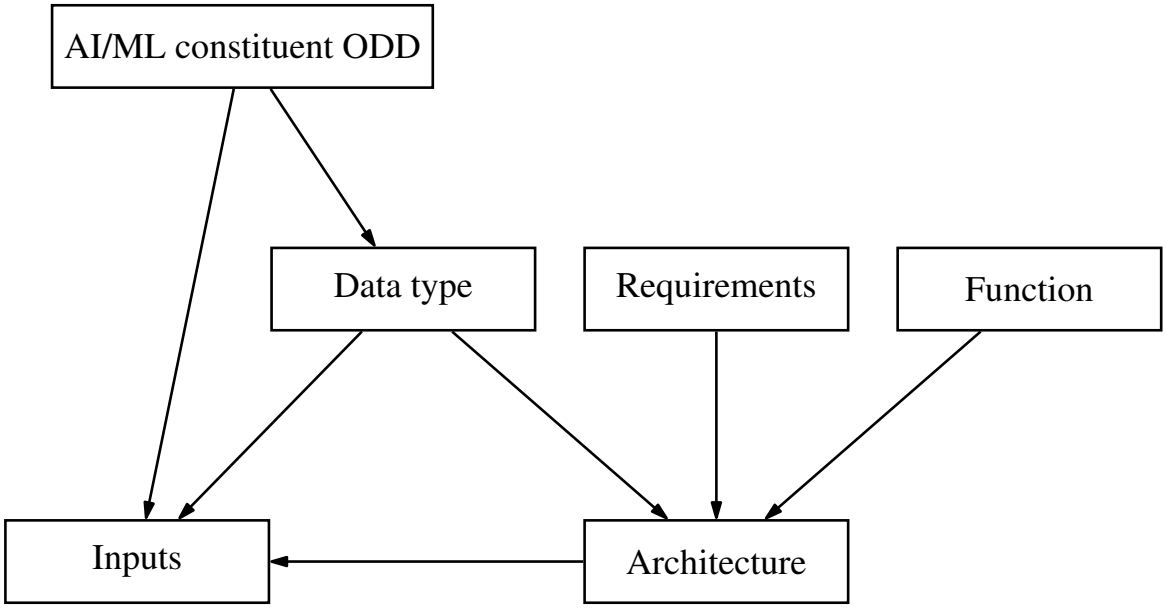


Figure 4.7: A mapping of the influence factors when designing a ML model.

5 Methodology Validation

This section introduces the application of the proposed methodology for the selected airborne collision avoidance system use case. This application of the methodology aims at validating whether the application of the methodology, conforms to the considered EASA objectives, see Table 1.1. For the use case, the vertical collision avoidance systems (VCAS) [95] and horizontal collision avoidance systems (HCAS) [96] were selected, which are part of PYCASX [28], [97]. The PYCASX system is a Python-based implementation of an airborne collision avoidance system, inspired by the ACAS X standard [24], [25]. The reasoning of why the PYCASX, VCAS, and HCAS use case was selected are introduced in section 5.1. Furthermore, this also introduces the issue with the current implementation of VCAS and HCAS. Based on this issue, the goal is to replace the neural networks with the Safety Net concept. The replacement of the neural networks should conform to the considered objectives of the EASA concept paper [8]. Therefore, the methodology that was introduced in the previous chapter 4 is applied. The application of the methodology to the use case is introduced from section 5.2 to section 5.6. This covers the definition of the ConOps, see section 5.2, the OD, see section 5.3, and the functional decomposition of PYCASX, see section 5.4. In section 5.5, the step from the system level to the AI/ML constituent level with the definition of the AI/ML constituent ODD for VCAS and HCAS is presented. Section 5.6 then introduces the architecture and input selection for the VCAS and HCAS systems. In each section, the result is verified whether it conforms to the considered objectives, see Table 1.1. For the verification of each objective, the relevant requirements are identified and for each, it is determined if the results conform to the identified requirements. With this verification, it is measured if the introduced methodology of chapter 4 will enable a developer to fulfill the objectives of EASA [8], therefore validating the methodology. This validation of the whole framework is discussed in the section 5.7.

5.1 Vertical and Horizontal Airborne Collision Avoidance System

As introduced in section 2.3, one of the changes of the ACAS X standard is the use of LUTs for the advisory selection. A challenge with this approach is the large size of the LUTs that have to be stored on certified avionics hardware [26], [27]. This is especially challenging, as ACAS X should also be deployable to already existing hardware. Therefore, it is necessary to compress the LUTs further to reduce the memory footprint. One approach to compress data is to use the generalization and memorization capabilities of neural networks [98]. This was implemented in the Vertical and Horizontal Collision Avoidance System, where neural networks are used to approximate the LUTs that store the information for the advisory selection [26]. The LUTs that store the action costs for the vertical collision resolution are approximated by VCAS, and the LUTs for the action costs for the horizontal collision resolution are approximated by HCAS. Importantly, the primary objective of the neural networks is to provide at all times the same advisory as the original LUTs. Therefore, the compression of the same advisory costs is considered an optional behavior of the neural networks [99]. Using this approach, VCAS and HCAS [26] were able to reduce the memory footprint of the LUTs to a few hundred kilobytes from the original sizes of gigabytes and megabytes. But while this approach reduced

the memory footprint with great effect, the compression is lossy. For VCAS the correct advisory is only guaranteed in 94.9 % of the cases, and for HCAS it is only guaranteed in 97.9 % of the cases [26]. This poses a potential safety risk as wrong advisories could fail to prevent collisions between two aircraft, which could have been avoided if the original LUTs were used. Therefore, a solution is needed that can compress the LUTs with 100 % accuracy, i.e., being a lossless compression, while still reducing the memory footprint as much as possible. One approach would be to simply increase the neural networks with additional layers or neurons to a point where the network can completely represent the LUTs. It was shown by previous works [100] that when using large enough neural networks, it is possible to train neural networks to remember all the training data with 100 % accuracy. As in the selected use case, the training and test data are the same dataset, this approach is feasible but contains two potential drawbacks. Firstly, if the neural networks have to be increased in size, this results in a larger memory footprint of those networks and an increase in the inference computing cost. The latter point can be the larger issue, as safety-critical avionics hardware still often relies on single-core systems [101]. Therefore, keeping the computational requirements as low as possible is an important goal to fulfill when developing the neural network. For this reason, this can introduce the issue that no neural network can be designed and trained, that can compress the LUTs with 100 % accuracy while, at the same time, the computational effort is low enough to be computable in reasonable time on avionics hardware. Secondly, it is not guaranteed that the training process of neural networks will lead to a neural network that is able to compress the LUTs with 100 % accuracy each time the network is trained. Since it is not initially known if a neural network structure can achieve a lossless compression, large trial, and error experimentation efforts might be needed to find out if any given neural network structure has this capability. While systematic approaches such as hyper-parameter optimization can be employed, such as Optuna [102], to find neural network structures that can compress the LUTs lossless, due to the stochastic nature of the training of neural networks, reproducibility of the lossless compression is not guaranteed. Therefore, this can lead to the issue that even when only one entry is compressed wrongly, the neural network is unusable as it poses a potential safety hazard. To summarize, while it is feasible to just rely on neural networks for lossless compression, the constraints of the avionics hardware capabilities and the stochastic nature of machine learning make this approach unpractical.

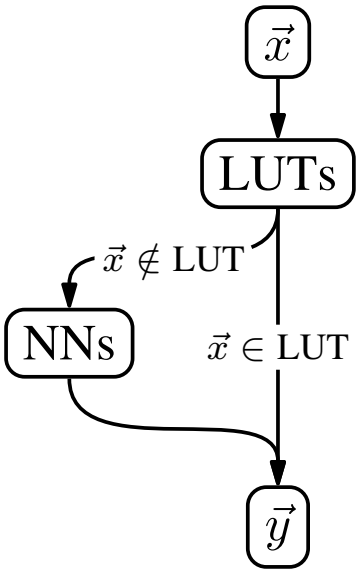


Figure 5.1: Building blocks of the Safety Net concept. Based on [27], [62], [103].

Considering these drawbacks of an approach that solely relies on neural networks, extending this approach using a fallback mechanism, that can compensate the issues of the neural networks will be of advantage. Therefore, one suggestion is to apply the Safety Net concept as a hybrid architecture for the compression of the LUTs [27], [62], [103]. As shown in Figure 5.1, the Safety Net concept combines the neural network with a LUT, where the LUT contains all entries of the original LUT that the Safety Net compressed wrongly [27]. During inference, if an entry is looked up in the Safety Net, first it is checked if it is contained in the LUT. If the LUT contains the entry, then the value from the LUT is used, otherwise, the neural network is able to reproduce the original LUT value. The goal of this approach is, that the neural network should compress most of the data and that only a small amount of entries have to be saved in the LUT in the Safety Net. With this approach, it should be possible to reduce the memory footprint with the neural network while maintaining the correctness of the advisory [27], [62], [103]. The advantage compared to the neural network approach is, that no trial and error phase is needed to find a neural network that can compress the data with 100 % accuracy. Furthermore, the issue of the hardware capability requirements, now becomes an optimization issue, where depending on the available hardware the size of the neural network and the LUT can be optimized. While the Safety Net approach is more suitable for the compression task, it shares one crucial precondition with the neural network approach. Both approaches rely on the fact, that the data can be approximated with a neural network. If this precondition does not hold, the required effort to develop and certify the neural network might be too large compared to other approaches that do not rely on machine learning.

Additionally, the VCAS and HCAS systems were selected, as they are an important part of PYCASX [28], [97]. The goal of PYCASX is to introduce a Python-based ACAS that implements similar functionalities as the new ACAS X standards [28], [97]. Importantly, the goal of the PYCASX system is to provide a basis for the development of methodologies, that can be used to build AI-based systems in aviation in the future. A framework that will be important for the development of AI-based systems, is the EASA concept paper that was introduced in section section 2.2. As the VCAS and HCAS systems were not designed and built on the EASA framework, the goal of the validation is to re-engineer these components using the developed methodology. Applying the methodology should provide a basis for the further development of AI assurance approaches using the PYCASX system.

The developed methodology includes steps at the system level. As PYCASX contains multiple subsystems next to the VCAS and HCAS subsystems, these subsystems will be included in the following sections. But as these subsystems are not AI-based, the assumption is made that these subsystems will be built according to the relevant aviation guidelines and standards, such as DO-178C [5]. Therefore, in the following sections, these subsystems will only be introduced to the extent necessary for understanding the VCAS and HCAS subsystems and the overall functionality of the system.

5.2 Definition of a ConOps for PYCASX

The task of PYCASX is to provide the ACAS capability to the pilots of an aircraft. Therefore, the following subsections will introduce the Concept of Operations for the use case of PYCASX. For the definition of the ConOps, the following section introduces the individual steps, that were defined in the methodology in section 4.2.

5.2.1 Description of the Current System

The first step according to the methodology is to provide a description of the current system. In the ACAS use case this can include two systems, TCAS II and ACAS X_U, and the components, VCAS and HCAS. Each of these systems or components might potentially be replaced by PYCASX.

Under current rules, TCAS II is required for all aircraft with a maximum take-off mass (MTOW) over 5700 kg or approved to transport more than 19 passengers under International Civil Aviation Organization (ICAO) and European Union Safety Agency (EASA) regulations [18], [19], or more than 30 seats under Federal Aviation Administration (FAA) regulations [20]. Furthermore, the TCAS II is standardized in the DO-185B [21]. TCAS II determines the advisories to prevent a collision based on a rule-based heuristic logic [16]. The shortcomings of the TCAS II system are the generation of false positive resolution advisories, and the impracticability to adapt it to the increase of air travel and new types of aircraft [22].

The ACAS X_U is the next generation ACAS that is designed for unmanned aircraft [25]. This system relies on a probabilistic approach to determine the resolution advisories for the pilot. Furthermore, this system also provides resolution advisories, in the vertical and horizontal plane. ACAS X_U is standardized in the DO-386 [25]. A drawback of this approach is the large memory footprint of the system [98].

VCAS and HCAS are a simplified implementation of the resolution advisory selection of the ACAS X_U system. The main focus of the system is to compress the required data for the resolution advisory selection with neural networks. The neural network compression does not achieve a lossless compression of the data.

In addition to these systems, further standards were identified that may apply to the ACAS use case. The ED-271 describes the "Minimum Aviation System Performance Standard for Detect and Avoid (Traffic) in Class A-C airspaces" [104], and the ED-313 describes the "OSED for Detect and Avoid (Traffic) in Class A to G Airspaces under IFR" [105]. These additional standards were identified, as both describe requirements for systems providing detect and avoid capabilities.

5.2.2 Justification For and Nature of the Changes

The justification of the changes is based on the shortcomings of ACAS X_U, VCAS, and HCAS. Firstly, the justification for the change based on ACAS X_U is the necessity to reduce the memory footprint of the LUTs that are used in the system. This reduction is necessary as the current memory footprint of these LUTs is too large for current avionics hardware [27]. The nature of change is the use of a neural network to compress the data of the LUTs. The application of neural networks does transform PYCASX to an AI-based system. Secondly, the justification for the change of VCAS and HCAS is the lossy compression that is provided by the neural networks. This lossy compression is an issue as a wrong advisory for the pilots can fail to prevent a potential collision between two aircraft. Therefore, the hybrid approach, combining neural networks and LUTs, is introduced, to ensure that the compressed LUTs and the uncompressed LUTs result in the same advisories. The replacement of the neural networks in PYCASX with the hybrid approach should only affect the VCAS and HCAS subsystems, all other subsystems or functionalities will not be affected by the proposed change.

5.2.3 Description of the Proposed AI-Based System

The PYCASX system is designed to provide the pilots of an airplane with last-resort measures to prevent a collision with other airplanes. The system is designed to operate in European airspace *type C*, where airplanes typically operate in the IFR or VFR mode. The airspace *type C* also entails that no

geographic features or structures are present in the airspace. This is important because the system is not designed to consider these aspects in its advisory selection. For the system, the airplane in which it is installed is the ownship and all other airplanes in the vicinity are the intruders. Each of the intruders and the ownship must be equipped with an ADS-B system. This ADS-B system allows each airplane to determine its position and broadcast it to all other airspace participants. Based on the information about the position and dynamics of the aircraft, the system can determine if the ownship airplane is on a collision course with an intruder airplane. If an intruder airplane is on a collision course, the system shall generate a collision advisory for the pilot to prevent a potential collision. The collision advisories can either be on a vertical or horizontal plane. The possible collision advisories from which the system can choose to prevent a collision are shown in Table 5.1. If an advisory is announced to the pilot(s), the pilot(s) have to execute the suggested advisory within 5 s unless it would endanger the safe operation of the airplane. The system can generate advisories simultaneously in the vertical and horizontal plane if necessary. If multiple intruders are on a collision course the system announces the strongest advisory to the pilot. If the system generates conflicting advisories, the system will announce or display the conflicting intruders but will not determine an action for the pilot. Furthermore, no coordination between the ownship and intruder aircraft takes place.

Table 5.1: Available advisories for the PYCASX system [26].

(a) Advisories in the vertical plane.

Advisory	Description
COC	Clear Of Conflict
DND	Do Not Descend
DNC	Do Not Climb
DES 1500	Descend $\geq 1500\,\mathrm{ft\,min^{-1}}$
CL 1500	Climb $\geq 1500\,\mathrm{ft\,min^{-1}}$
SDES 1500	Strengthen Descend to $\geq 1500\,\mathrm{ft\,min^{-1}}$
SCL 1500	Strengthen Climb to $\geq 1500\,\mathrm{ft\,min^{-1}}$
SDES 2500	Strengthen Descend to $\geq 2500\,\mathrm{ft\,min^{-1}}$
SCL 2500	Strengthen Climb to $\geq 2500\,\mathrm{ft\,min^{-1}}$

(b) Advisories in the horizontal plane.

Advisory	Description
COC	Clear Of Conflict
WL	Weak Left $\leq -1.5\,^\circ\,\mathrm{s^{-1}}$
WR	Weak Right $\geq 1.5\,^\circ\,\mathrm{s^{-1}}$
SL	Strong Left $\leq -3.0\,^\circ\,\mathrm{s^{-1}}$
SR	Strong Right $\geq 3.0\,^\circ\,\mathrm{s^{-1}}$

5.2.4 Definition of a Task Allocation Pattern

Based on the previous section's description of the system, the tasks allocated to the system are the detection of intruder aircraft, the determination of a collision advisory, and the announcement to the pilot(s). The tasks for the pilot(s) are to decide that the collision advisory can be executed safely and to execute the given collision advisory, as it is specified in Table 5.1. Furthermore, the pilot(s) also need to perform collision avoidance if the system is not able to determine a suitable collision advisory. Lastly, the pilot(s) are also responsible for informing the air traffic control of the incident, when PYCASX detected and adverted a potential collision.

5.2.5 Description of the Operational Scenarios

Following the introduction of the PYCASX system in a generic way, as explained in section 4.2, the ConOps must also include the description of operational scenarios. As explained, this should give

the end user(s) a better understanding of the system and its capabilities. For the PYCASX system, a description of an operational scenario can be the following:

The scenario takes place in an onwship flight from FDH to BER. After takeoff and climb to cruising altitude, the airplane is configured by the pilot(s) into en-route operation under IFR in airspace type C. The airspeed of the aircraft is in the typical cruising speed range and is surrounded by non-conflicting traffic. The scenario starts when two traffic airplanes change trajectory setting them on to a collision course. The surveillance of the ownship detects both trajectory changes of the threat intruders, based on the ADS-B broadcast messages. Based on this information, the PYCASX *system calculates the expected point where the two airplanes potentially collide. Based on this estimation, and the information about the ownship and intruders, the system generates two advisories. For the first intruder, the system determines that an action in the vertical plane, e.g., DES 1500, is necessary and for the second intruder, the system determines that an action in the horizontal plane, e.g., WL, is necessary. These actions are announced to the pilot(s) and they determine that the actions can be safely executed. After the execution of these actions for 20 s, the system determines that the sufficient separation between the intruders and the ownship is maintained, and therefore announcing to the pilots a clear of conflict. The pilot(s) level of the airplane, and inform the responsible air traffic controller. The scenario ends, as no further conflicts arise for the rest of the flight.*

This outlined description of an operational scenario describes a situation that is expected to be the system's normal operation. Additional scenario descriptions are provided in Appendix A. These include scenarios with normal and degraded system operation.

5.2.6 Verification

This section aims at fulfilling the outline objectives CO-01, CO-02, and CO-04 by applying the methodology of section 4.2. For these objectives, the relevant requirements are shown in Table 5.2. Comparing the defined ConOps for the PYCASX system with the requirements shows that the applied methodology meets two requirements partially and three completely. The two partially fulfilled requirements are the *Identification of end users* and the *Goals of the end users*. These are only fulfilled partially, as the applied methodology only introduces the users to the generic description of the system and the operational scenarios. However, no explicit definition of the level of expertise or goals for each end user is provided. The requirements of the *High-level tasks of the end user*, the *Operational scenarios*, and the *Task allocation in the operation scenarios* are fulfilled completely, as the ConOps, based on ISO/IEC/IEEE 29148:2018 [69], includes these requirements.

Table 5.2: Requirements that have to be achieved by the ConOps definition for the PYCASX use case.

Objective	Requirement	Fulfilled
CO-01	Identification of end users	Partially
CO-02	Goals of the end users	Partially
	High-level tasks of the end users	Completely
CO-04	Operational scenarios	Completely
	Task allocation in the operational scenarios	Completely

5.3 Definition of an OD for PYCASX

Following the introduction of the system capabilities and the operational description in section 5.2, the next step is to define the Operational Domain for the PXCASX system. As explained in the process of subsection 4.3.3, this consists of the OD taxonomy definition and the OD specification. For the identification of the elements of the OD, the recommended deconstruction of the environment using a layer model was applied [55]. The model consists of six layers as shown in Figure 5.2, and in this use case based on the ConOps, the model has to describe all elements that make up the airspace for en-route operations. Layer 1 describes the aerial structure necessary for the airspace that is relevant for en-route operations. This includes the attributes such as the airspace *type* and the possible *flight rules* of the airspace. Furthermore, the airspace *type* attribute introduces additional attributes in this layer, such as *altitude*, as the airspace type is normally restricted to certain altitudes. Layer 2 includes all attributes that belong to optional permanent objects in the airspace. These include attributes such as geography or structures. The next layer, layer 3, describes temporary modifications of the layers 1 and 2. Temporary flight restrictions [106], that temporarily restrict areas in the airspace, can be such temporary modifications. But as PYCASX is designed as a last resort measure to prevent a collision, such temporary flight restrictions are not relevant for the determination of the collision advisories and therefore not included in the OD. Layer 4 describes all dynamic elements, i.e., the elements that are not stationary in the airspace. For the use case of airborne collision avoidance, dynamic elements are the traffic/intruder aircraft and the ownship aircraft itself. These dynamic elements have a multitude of sub-attributes that govern the performance characteristics of the aircraft. One such attribute is the aircraft *type*. This attribute is important as the ConOps limits the provided collision avoidance capabilities to only airplanes. This is important as other types of aircraft have different types of behavior that have to be accounted for in the collision avoidance logic. Layer 5 describes all elements that describe environmental conditions. Environmental conditions contain all weather and atmospheric conditions [55]. As PYCASX is designed as a last-resort-measure, it should function in all weather and atmospheric conditions. The only exceptions are adverse weather and atmospheric conditions that would prohibit an airplane from achieving the required performance expected by PYCASX. The last layer, layer 6, is the data and communication layer [55]. In the use case, this layer contains all necessary attributes that allow the traffic aircraft to communicate their position and that allow each airplane to determine its position. As described in the ConOps, for PYCASX the traffic position is communicated using ADS-B. The position of all airplanes is determined based on GPS data.

For these defined attributes, in the initialization some values were already defined based on the ConOps, for example, the airspace *type* which was defined to be type C [107]. But other attribute values, especially the intruder and ownship airplane performance characteristics, were not defined in the ConOps. These attributes were defined in the refinement step based on the identified standards that apply to the PYCASX system. For example, the identified ED-313 [105] specifies that the maximum allowed horizontal velocity of an intruder is 600 kn and the maximum allowed horizontal acceleration is 1.5 g. Following the proposed steps of subsection 4.3.3, the Table 5.3 is the resulting OD for the PYCASX system.

5.3.1 Verification

The definition of an OD is a requirement introduced as a part of objective CO-04. The main requirement for the OD is that it must capture the limitations and assumptions of the operating conditions of the system. As shown in this section, the operating conditions for PYCASX are successfully captured by the introduced tabular structure. Therefore, the defined PYCASX OD fulfills this requirement.

Table 5.3: The specified OD for PYCASX.

Top-level attribute	Sub-attribute	Qualifier	Attribute	Attribute value	Unit
Scenery	Airspace	Include	Type	C	-
	Airspace	Include	Flight Rule	IFR, VFR	-
	Airspace	Include	Altitude	[10 000, 66 000]	ft
	Airspace	Include	Latitude	[−90, 90]	°
	Airspace	Include	Longitude	[−180, 180]	°
	Airspace	Include	Route Type	Free Route Airspace	-
	Airspace	Exclude	Geography	Any	-
	Airspace	Exclude	Structures	Any	-
Environment	Weather	Exclude	Adverse Conditions	Any	-
	Connectivity	Include	Satellite Positioning	GPS	-
	Connectivity	Include	Communication Type	ADS-B	-
	Connectivity	Include	Communication Range	[20,)	NM
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-
	Intruder	Include	Maximum Agent Density	0.06	NM^{-2}
	Intruder	Include	Latitude	[−90, 90]	°
	Intruder	Include	Longitude	[−180, 180]	°
	Intruder	Include	Altitude	[10 000, 66 000]	ft
	Intruder	Include	Horizontal Airspeed	[0, 600]	kn
	Intruder	Include	Horizontal Acceleration	[−48.261, 48.261]	$ft\,s^{-2}$
	Intruder	Include	Vertical Rate	[−5000, 5000]	$ft\,min^{-1}$
	Intruder	Include	Vertical Rate Acceleration	[−10.725, 10.725]	$ft\,s^{-2}$
	Intruder	Include	Heading	[−180, 180]	°
	Intruder	Include	Communication Type	ADS-B	-
	Ownship	Include	Agent Type	Airplane	-
	Ownship	Include	Latitude	[−90, 90]	°
	Ownship	Include	Longitude	[−180, 180]	°
	Ownship	Include	Altitude	[10 000, 66 000]	ft
	Ownship	Include	Horizontal Airspeed	[0, 600]	kn
	Ownship	Include	Horizontal Acceleration	[−48.261, 48.261]	$ft\,s^{-2}$
	Ownship	Include	Vertical Rate	[−5000, 5000]	$ft\,min^{-1}$
	Ownship	Include	Vertical Rate Acceleration	[−10.725, 10.725]	$ft\,s^{-2}$
	Ownship	Include	Heading	[−180, 180]	°
	Ownship	Include	Vertical Rate Capability	$\geq$2000	$ft\,min^{-1}$
	Ownship	Include	Vertical Rate Acceleration Capability	$\geq$10.725	$ft\,s^{-2}$
	Ownship	Include	Turn Rate Capability	$\geq$3	$°\,s^{-1}$
	Ownship	Include	Turn Rate Acceleration Capability	$\geq$1	$°\,s^{-2}$
	Ownship	Include	Pilot Type	Pilot, Remote Pilot	-

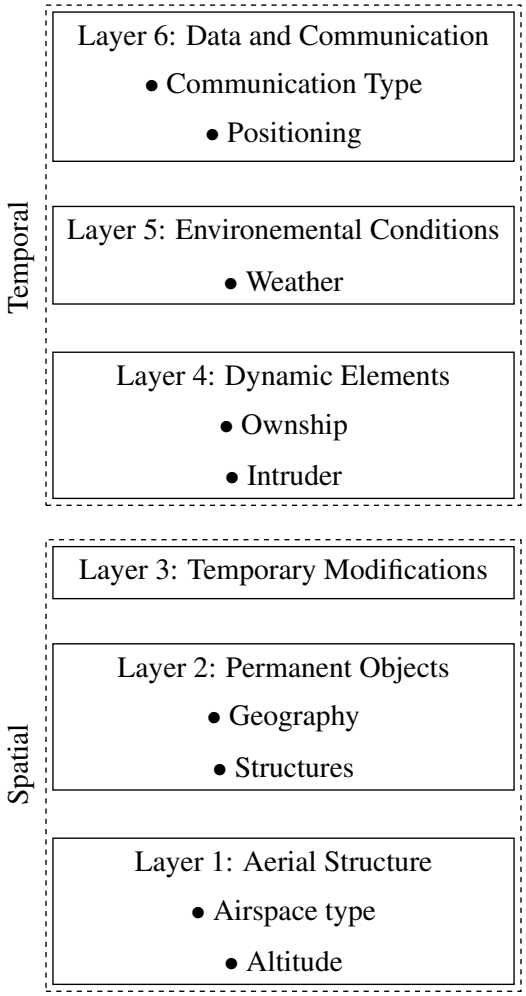


Figure 5.2: The 6-layer model applied to the PYCASX use case.

5.4 Functional Decomposition of PYCASX

Based on the introduced ConOps a functional decomposition of the PYCASX was conducted. In the ConOps description, three main functions were identified.

The first main function that was identified was the detection of other airplanes and their position. This function is split into three subfunctions as shown in Figure 5.3. The first two subfunctions are necessary to detect and to determine all information about the intruder. The third subfunction is necessary to determine all necessary information about the ownship itself. Importantly, all these functions will be traditional software or hardware items.

The second main function is the determination of a collision advisory for an intruder. As shown in Figure 5.4, this function is split into height subfunctions. The first four subfunctions are necessary for the determination if a collision between an intruder and the ownship is possible. The first subfunction is necessary to pre-process the data from the ADS-B sensor. Subfunctions *F2.2* and *F2.3* are necessary to extrapolate the movement of the ownship and intruder into the future. Based on this extrapolation, the closest point of approach must be calculated to determine whether a possible near mid-air collision would occur between the two airplanes. If the closest point of approach is classified as a near mid-air

F1 : To detect the position of the intruder and ownship aircraft.

F1.1 : To detect the intruder aircraft in the surroundings.

F1.2 : To determine an intruder's position, height, and velocity.

F1.3 : To determine the position, height, and velocity of the ownship.

Figure 5.3: Decomposition of the first identified main function.

F2 : To provide collision avoidance advisories.

F2.1 : To pre-process GNSS data.

F2.2 : To interpolate the ownship movement.

F2.3 : To interpolate the intruder movement.

F2.4 : To calculate the CPA between each intruder and ownship.

F2.5 : To determine the vertical advisory for each intruder.

F2.6 : To determine the horizontal advisory for each intruder.

F2.7 : To detect conflicts between chosen advisories.

F2.8 : To select a final advisory.

Figure 5.4: Decomposition of the second identified main function.

collision, vertical or horizontal advisories must be calculated by the system. This functionality is implemented by the functions *F2.5*, *F2.6*, *F2.7*, and *F2.8*. The functions *F2.5* and *F2.6* are AI/ML-based, while all other functions are traditional software and hardware items.

The third main function is the interface to the aircraft avionics. This main function was further divided into three subfunctions, as shown in Figure 5.5. Functions *F3.1* and *F3.2* provide the ability to display the information of the PYCASX system to the pilot(s). The function *F3.3* is required to additionally announce the chosen advisory to the pilot via a speaker or similar. Such an announcement is standard for ACAS [21]. All functions are based on traditional software or hardware items.

Additionally, a preliminary system architecture is defined for the PYCASX system to realize the specified functions. As shown in Figure 5.6 the system consists of seven components. The first component is the *ADS-B sensor* that receives the broadcast messages from intruders nearby. The second necessary component is the *GPS*, which provides the location and the velocity information about the ownship. The information provided by these two components is used in the *CPA calculation* component. This component uses the position and velocity of the intruder and ownship to determine

F3 : To provide collision avoidance advisories.

F3.1 : To display the intruder.

F3.2 : To display the vertical and horizontal advisory.

F3.3 : To announce the vertical and horizontal advisory.

Figure 5.5: Decomposition of the third identified main function.

if a potential mid-air collision between these two aircraft will occur. The calculated CPA and the information of the intruder and ownship are used then in the *VCAS* and *HCAS* components. The *VCAS* component is tasked with the calculation of the vertical advisories and the *HCAS* component is tasked with the calculation of the horizontal advisories. Therefore, to the *VCAS* component, the function *F2.5* is allocated and to the *HCAS* component, the function *F2.6* is allocated. Importantly, both the *VCAS* and *HCAS* components are the AI/ML constituent in the PYCASX system, as these are based on neural network compression. Additionally, to the allocated functions, to the *VCAS* and *HCAS* components, two safety and one performance requirement were allocated. The safety requirements are on the one hand that these AI-based subsystems shall compress the data with no errors, i.e., produce the same advisory as the LUT, and on the other hand these AI-based subsystems shall not perform worse than the LUT in any situation. The performance requirement is that the AI-based subsystems shall be stored on avionics hardware. Following the determined action for each intruder, the next component, the *Multi-intruder advisory selection*, determines which advisory to announce to the pilot. This is necessary as for different intruders different advisories might be necessary. If for two intruders, *VCAS* or *HCAS* determines that two different advisories are necessary in the same vertical or horizontal plane, then this component, requires a logic to determine which of the generated advisories should be chosen. The last component necessary is the *Display* and *Speaker* that announces the chosen advisory to the pilot(s).

5.4.1 Verification

For the functional decomposition in the relevant objective, CO-06, three requirements were identified as shown in Table 5.4. The requirement of the *Functional decomposition of the system* is fulfilled completely, as a function analysis and decomposition was done for the PYCASX system. As for the PYCASX system, these functions were allocated in the preliminary system architecture, and it was identified that only the *VCAS* and *HCAS* components were AI/ML items, the requirements of the *Function allocation* and the *Classification of AI/ML items* are seen as fulfilled.

Table 5.4: Requirements that have to be achieved by the functional decomposition for the PYCASX use case.

Objective	Requirement	Fulfilled
CO-06	Functional decomposition of the system	Completely
	Function allocation in the system architecture	Completely
	Classification of AI/ML items	Completely

5.5 Definition of an AI/ML Constituent ODD for VCAS and HCAS

Following the previous steps for the PYCASX system, the next step for the development of the AI/ML constituents is to define an ODD for each AI/ML constituent, i.e., an ODD for VCAS and HCAS separately. This is necessary, as VCAS only provides advisories in the vertical plane and HCAS only in the horizontal plane. Therefore, these AI/ML constituents are created based on different data sets and each data set requires an individual-defined ODD.

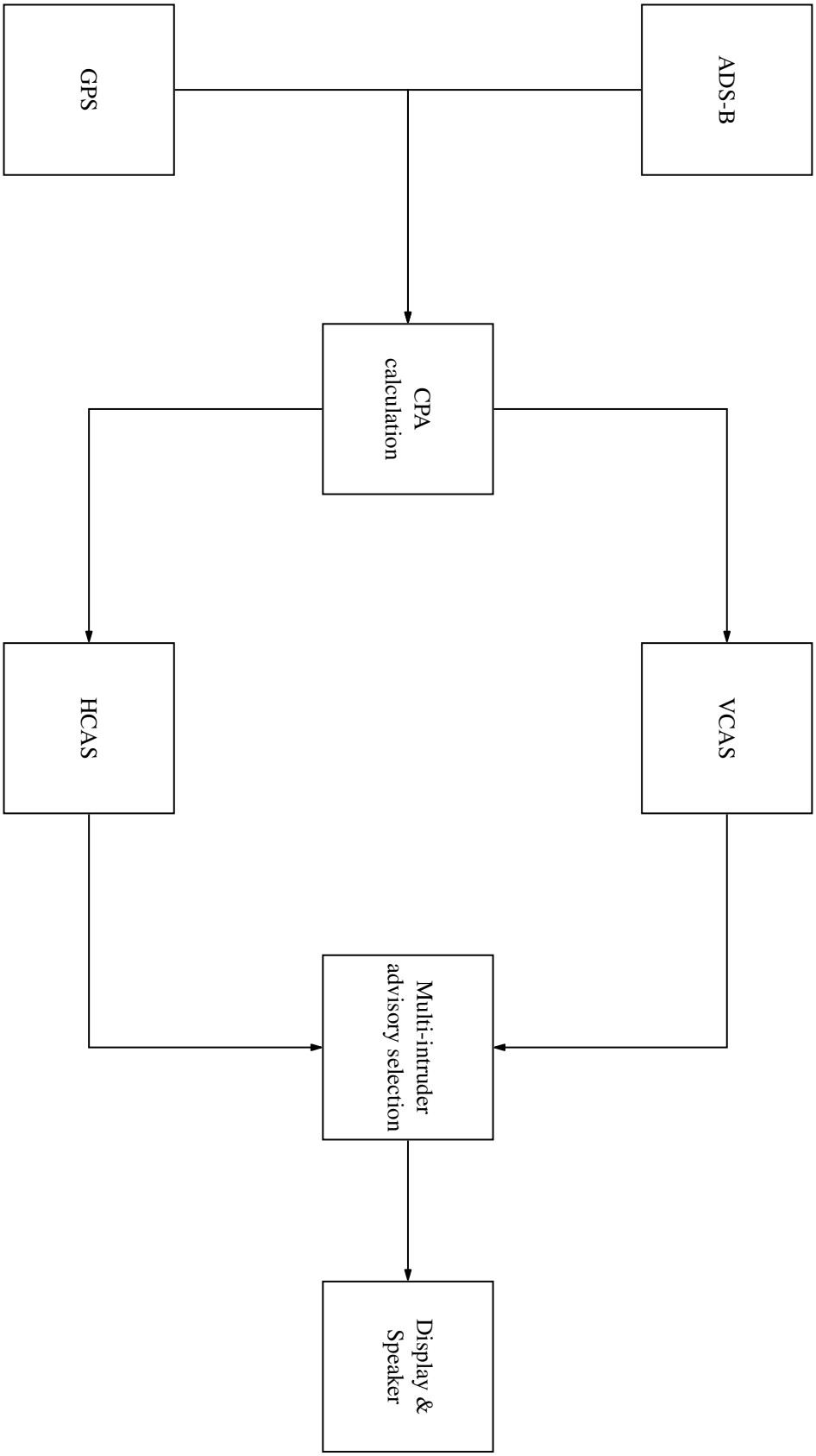


Figure 5.6: The preliminary architecture of the PYCASX system [28], [97].

5.5.1 The AI/ML Constituent ODD for VCAS

Applying the methodology of subsection 4.5.3, the previously defined ConOps, OD, and functional decomposition of PYCASX are used in the definition of the AI/ML constituent ODD. As VCAS only provides advisories in the vertical plane, the AI/ML constituent ODD is only defined for the data in this vertical plane.

Initialization Using the defined OD from section 5.3, the first step is the identification of the individual parameters of the OD, that must be part of the AI/ML constituent ODD. For this, as described in subsection 4.5.3, all attributes were classified if they were attributes of a physical range or if they were behavior-determining attributes. The result of this classification for the scenery attributes is shown in Table 5.5 in the *Classification* column. Using this classification and the allocated function to VCAS for each attribute, it is determined if it is relevant for the AI/ML constituent ODD. This determination for the *scenery* attributes is shown in Table 5.5 in the *Relevant* column. The *airspace type* was determined to be a behavior attribute, as the airspace type determines what type of flight rules are possible and if the airspace is controlled or uncontrolled [108]. This can affect the data distribution and therefore is seen as relevant for the VCAS ODD. The attributes of the airspace, *altitude* and *coordinates* are both classified as physical ranges. As VCAS shall be able to resolve a conflict regardless of the airspace dimension, these are seen as not relevant for the VCAS ODD. Similar to the *type*, the *flight rule* and *route type* are both elements that determine the behavior of aircraft, and therefore were also seen as relevant for the VCAS ODD. For the attributes of *geography* and *structures*, no classification was possible as these are high-level attributes that if necessary for the system can be further subdivided into a larger subset of attributes. Each of these sub-attributes could be either a behavior or physical range attribute. These attributes of *geography* and *structures* were also determined to be not relevant for the VCAS, as VCAS and PYCASX are only designed to detect and resolve conflicts between two aircraft. The detection of conflicts with *geography* or *structures* is handled by other systems, for example, the ground proximity warning system [109]. The same steps were done for the attributes of *environment* and *dynamic elements*. The results of these attributes are included in Appendix B. Based on these results an initial ODD for VCAS was defined, which was further refined in the following steps.

Projection of Attributes for VCAS Following the definition of the initial ODD for VCAS, the next step is the projection of attributes into the perception of the VCAS system. In this case, this only affects the *altitude* attribute. With VCAS, it is not the absolute altitude of the ownship and the intruder that matters, but the difference in altitude. Therefore, for the data management, it will be important to collect data that contains sufficient samples for all possible relative altitudes between the ownship and the intruder. The maximum relative altitude ($\Delta h_{\max}$) between ownship and intruder is given by the maximum vertical rate ($\dot{h}_{\max}$) and the maximum time to the closest point of approach (τ). Importantly, as both the ownship and intruder can climb or descend with the same maximum vertical rate, just in opposite directions, the maximum vertical rate ($\dot{h}_{\max}$) has to be multiplied by 2. Following Equation 5.1 to Equation 5.3, results in a maximum relative altitude ($\Delta h_{\max}$) of 10 000 ft.

$$\Delta h_{\max} = \tau \cdot 2 \cdot \dot{h}_{\max} \tag{5.1}$$

$$\Delta h_{\max} = 60\,\mathrm{s} \cdot 2 \cdot \frac{5000\,\mathrm{ft\,min^{-1}}}{60\,\mathrm{s}} \tag{5.2}$$

$$\Delta h_{\max} = 10\,000\,\mathrm{ft} \tag{5.3}$$

Table 5.5: For each attribute of the OD for PYCASX, it was classified if the attribute is a physical range or a behavior-determining attribute. Based on this and the allocated function to VCAS, for each attribute it is identified if it is relevant for the VCAS component, i.e., should be included in the ODD.

Top-level attribute	Sub-attribute	Attribute	Classification	Relevant
Scenery	Airspace	Type	Behavior	Yes
Scenery	Airspace	Altitude	Physical Range	No
Scenery	Airspace	Longitude	Physical Range	No
Scenery	Airspace	Latitude	Physical Range	No
Scenery	Airspace	Flight Rule	Behavior	Yes
Scenery	Airspace	Route Type	Behavior	Yes
Scenery	Geography		-	No
Scenery	Structures		-	No

Sensor Characteristics for VCAS The next step for the VCAS ODD is the identification of sensor characteristics. As shown in Figure 5.6, the only sensors used in PYCASX are GPS and ADS-B. ADS-B is only a broadcast of information about the aircraft's position, speed, and altitude that was determined based on a global navigation satellite system [110]. In the PYCASX use case, the global navigation satellite system is assumed to be GPS for both the intruder and the ownship. Therefore, the data of the ownship and intruder are affected by the same sensor characteristics. As GPS directly provides the required information, the only additional sensor characteristic introduced is the inaccuracy of the GPS measurements. However, as GPS has a high accuracy [111] and at large heights, only uncommon causes, such as solar storms [112], cause larger inaccuracies, for the VCAS ODD these inaccuracies are assumed to be zero. This GPS accuracy was added as an additional attribute under the top-level attribute for the *operating parameters*.

Identification of Domain-Specific Concepts for VCAS To identify additional domain-specific concepts for VCAS an ontology-based domain model was created for an encounter between an ownship and a single intruder. As shown in Figure 5.7, the model consists of the entity's intruder and ownship. Both of these entities have the three attributes of the *Velocity*, the *GNSS*, and the *Position*. As previously described for the *GNSS* it is assumed that both the intruder and the ownship use (GPS). The *GPS* is, therefore, a value in the domain model. The *Velocity* is further split into the attributes of a *Vertical* and *Horizontal* velocity. The values of the *Vertical* and *Horizontal* velocity in the model are determined by the values identified for the PYCASX OD. Both, the *Vertical* and *Horizontal* velocity also have the additional attribute of the *Acceleration* and *Turn Rate* with their corresponding values. The *Position* attribute is further split into the attributes of *Altitude*, *Latitude*, *Longitude*, and *Heading*. Each attribute has value, again determined by the PYCASX OD. For VCAS, the attributes of *Horizontal Velocity*, *Horizontal Velocity Acceleration*, *Altitude*, and *GNSS* are relevant. While these are important attributes for VCAS, these were already identified in a previous step. For the ownship one special attribute was identified, the *Pilot* that has an attribute *Response Time*. This attribute describes the time the pilot needs from the announcement of the advisory to the start of the execution of the advisory. This is important as when the system has to factor in such a delay, the system can generate advisories earlier compared to a system that does not factor in such a delay [113]. For the sake of simplicity, this paper assumes that the reaction time is 0 s.

Analysis of Available Data Sources for VCAS For VCAS a dataset is available that was used to build the original version of VCAS [26]. For their dataset, the variables and ranges shown in Table 5.6 were assumed by [26]. Next to the advisories for the VCAS system, for the encounter data, their dataset consisted of four different attributes. Each of these attributes were already identified in previous steps and therefore this dataset did not yield any additional attributes for the VCAS ODD.

Table 5.6: Variable of the VCAS dataset. Adapted from [26].

Attribute	Value range	Unit
Relative intruder altitude	[−8000, 8000]	ft
Ownship vertical rate	[−6000, 6000]	$\mathrm{ft\,min^{-1}}$
Intruder vertical rate	[−6000, 6000]	$\mathrm{ft\,min^{-1}}$
Time loss of horizontal separation	[0, 40]	s

Resulting VCAS ODD The final AI/ML constituent ODD that was designed for VCAS, is shown in Table 5.7. In total 16 attributes were identified for the VCAS ODD. Three attributes are under the top-level attribute *Scenery*, eleven under *Dynamic Elements*, and two under *Operating Parameters*. For the *Environment* top-level attribute, no relevant attributes were discovered. Compared to the PYCASX OD, the defined AI/ML constituent ODD for VCAS contains fewer attributes. These fewer attributes result from the allocated function and requirements to the AI/ML constituent, as the AI/ML constituent has to only cover a smaller subset of conditions compared to the conditions at the system level. For example, the VCAS component does not have to consider the attributes that define the operating conditions in the horizontal plane as it is only designed to resolve conflicts in the vertical plane. The three attributes for the *Scenery*, are all attributes that influence the behavior of the ownship and intruder. These are important, to ensure that these different behaviors are included in the dataset. Under the dynamic elements, most of the identified attributes are physical ranges that determine the allowed distances and rates of the intruder and ownship. A specialty for the ownship is the inclusion of attributes that require certain performance criteria for the ownship itself. These attributes are already included in OD, but are also relevant for the ODD, to ensure that the ownship conforms to the requirements needed to execute the advisories as expected. Were possible, for each attribute it was recorded if it is measured by a sensor in the system architecture. In the VCAS use case, this is either the *ADS-B* or *GPS* sensor. The *Distribution* was only determined for the attributes that are a constant, i.e., their range includes only a single value. For all other attributes, no distribution was determined, as their distribution will depend on the collected data. Therefore, this column can only be completed when the data was collected, that was used to build the final AI/ML constituent.

5.5.2 The AI/ML Constituent ODD for HCAS

As HCAS is similar to VCAS, in the following section the AI/ML constituent ODD for HCAS is introduced, but only in a shortened form that highlights the differences to the VCAS ODD definition. The initialization of the ODD for HCAS uses a similar approach to VCAS with the only difference, that for the *dynamic elements* all attributes in the vertical plane are discarded and only the attributes in the horizontal plane are kept, see Appendix C. For the projection of attributes, the maximum distance between an onwship and intruder aircraft must be calculated based on the maximum closing speed and the maximum time to the closest point of approach. Calculating this for the values in the PYCASX OD

Table 5.7: AI/ML constituent ODD for VCAS.

Top-level attribute	Sub-attribute	Qualifier	Attribute	Attribute value	Unit	Distribution	Source
Scenery	Airspace	Include	Type	C	-	Constant	-
	Airspace	Include	Flight Rule	IFR, VFR	-	-	-
	Airspace	Include	Route Type	Free Route Airspace	-	Constant	-
Environment	-	-	-	-	-	-	-
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-	Constant	ADS-B
	Intruder	Include	Vertical Rate	[−5000, 5000]	$\mathrm{ft\,min^{-1}}$	-	ADS-B
	Intruder	Include	Vertical Rate Acceleration	[−10.725, 10.725]	$\mathrm{ft\,s^{-2}}$	-	-
	Intruder	Include	Relative Altitude to Ownship	[−10 000, 10 000]	ft	-	-
	Intruder	Include	Time until loss of separation	[0, 60]	s	-	-
	Ownship	Include	Agent Type	Airplane	-	Constant	-
	Ownship	Include	Vertical Rate	[−5000, 5000]	$\mathrm{ft\,min^{-1}}$	-	GPS
	Ownship	Include	Vertical Rate Acceleration	[−10.725, 10.725]	$\mathrm{ft\,s^{-2}}$	-	GPS
	Ownship	Include	Vertical Rate Capability	$\geq$2000	$\mathrm{ft\,min^{-1}}$	-	-
	Ownship	Include	Vertical Rate Acceleration Capability	$\geq$10.725	$\mathrm{ft\,s^{-2}}$	-	-
	Ownship	Include	Pilot reaction time	[0]	s	Constant	-
Operating Parameters	Ownship	Include	GPS Inaccuracy	None	-	Constant	GPS
	Intruder	Include	GPS Inaccuracy	None	-	Constant	GPS

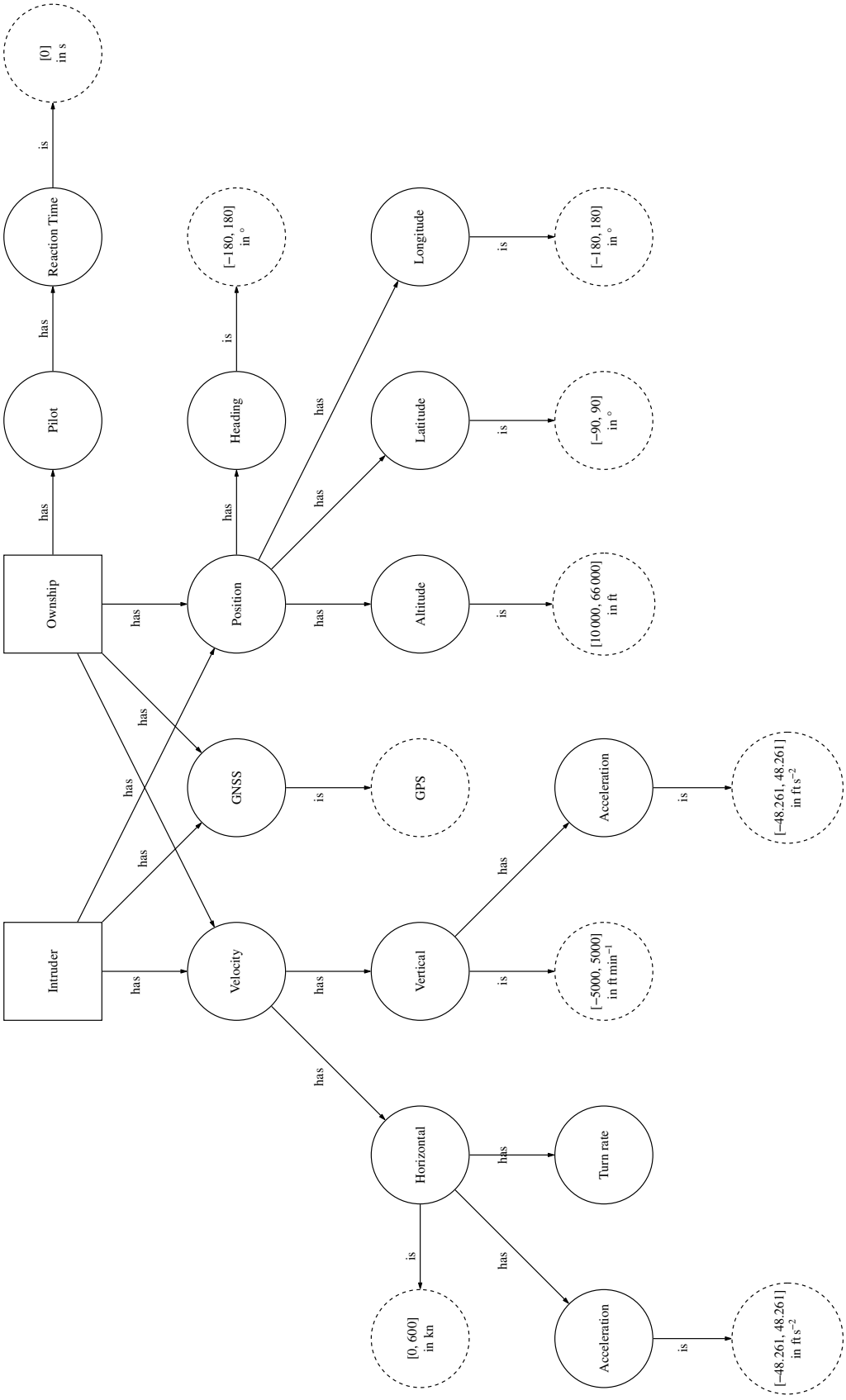


Figure 5.7: Ontology-based domain model for an encounter between the ownship and an intruder. An entity is depicted with a square, an attribute with a circle, a value with a dashed circle, and a relation with an arrow.

results in a maximum distance between the ownship and intruder of 122 000 ft. As HCAS also relies on the same sensors as VCAS the same additional attributes of the GPS accuracy were identified for the *operating parameters*. The identification of domain-specific concepts for HCAS was conducted by using the same ontology-based domain model of Figure 5.7. Similar to VCAS, this model only provided the *Response Time* of the pilot as an additional attribute. For HCAS, two available data sources were identified. On the one hand, the dataset that was used for the original HCAS implementation, and on the other hand a defined ODD for the ACAS X_U system. The original HCAS dataset is shown in Table 5.8a and the ACAS X_U ODD is shown in Table 5.8b. As shown in Table 5.8, while the attributes of each dataset have different descriptions, each of the six attributes has an equivalent in the other dataset. Importantly, based on these attributes it was identified that the heading of each the ownship and intruder can be described as a relative heading based on the ownship. Therefore in the HCAS ODD, the attributes of the *heading* are replaced with the relative attributes of the *Angle to intruder* and *Relative intruder heading*. Otherwise, all other attributes are already covered by the defined HCAS ODD, and no additional attributes were added.

Table 5.8: Identified available HCAS data sets.

(a) Attributes of the HCAS dataset. Taken from [26].

Attribute	Value range	Unit
Range to intruder	[0, 50 000]	ft
Bearing angle to intruder	[−180, 180]	°
Relative heading angle of intruder	[−180, 180]	°
Ownship speed	200	$\mathrm{ft\,s^{-1}}$
Intruder speed	185	$\mathrm{ft\,s^{-1}}$
Time to loss of vert. separation	[0, 80]	s

(b) Attributes of the ACAS X_U ODD. Taken from [33].

Attribute	Value range	Unit
Distance from ownship to intruder	[499, 185 318]	ft
Angle to intruder relative to ownship heading	[−3.14, 3.14]	
Heading angle of intruder relative to ownship heading direction	[−3.14, 3.14]	
Speed of ownship	[50, 1200]	$\mathrm{ft\,s^{-1}}$
Speed of intruder	[0, 1200]	$\mathrm{ft\,s^{-1}}$
Time until loss of vertical separation	[0, 101]	s

The finalized AI/ML constituent ODD for HCAS is shown in Table 5.9. As already described, the major difference between the VCAS ODD and the defined HCAS ODD is that the encounters are on a horizontal plane. Therefore, the attributes of the *Scenery*, the *Environment*, and the *Operating Parameters* are the same as in the defined VCAS ODD. Only for the top-level attribute of *Dynamic Elements*, new attributes were introduced and attributes of the vertical plane were discarded. In the horizontal plane, the encounter geometry between the ownship and intruder is described differently with the horizontal *Distance to Intruder* and the angles between the intruder and the ownship. Also for HCAS the advisory maneuvers are executed in the horizontal plane, by implementing a left or right turn with the airplane. To ensure that the ownship executes these advisories in the dataset as specified, the attributes of the *Turn Rate Capability* and the *Turn Rate Acceleration Capability* are included in the HCAS ODD. The *Pilot reaction time* is included in the HCAS ODD and is again assumed to be 0 s.

5.5.3 Verification

The definition of AI/ML constituent ODD for VCAS and HCAS must adhere to the objective DA-03 of the learning assurance [8]. The two extracted requirements from this objective are shown in Table 5.10. The first requirement is the definition of a *set of parameters for the AI/ML constituent* that describe the operation conditions of this AI/ML constituent. For both VCAS and HCAS, these operating conditions are described with an attribute, a range, and a distribution in the ODD. But as the distribution could not be determined for each attribute, as this is only possible for some attributes based on the collected data, the requirement of the *set of parameter for the AI/ML constituent* is only seen to be fulfilled *partially*. The second identified requirement is the tracing of the parameters to the OD. Through the application of the steps introduced in chapter 4, it is argued, that a traceability is given for the parameters in the AI/ML constituent ODDs for VCAS and HCAS. The individual steps allow the traceability of how the ODD for both components was constructed based on the OD. Therefore, the requirement *Traced parameters to the OD*, is seen as fulfilled completely.

Table 5.10: Requirements that have to be achieved by the AI/ML constituent ODD of VCAS and HCAS.

Objective	Requirement	Fulfilled
DA-03	Set of parameters for the AI/ML constituent	Partially
	Traced parameters to the OD	Completely

5.6 Architecture Design and Input-Feature Selection for VCAS and HCAS

Following the definition of the AI/ML constituent ODD, the next step is the definition of the AI/ML constituent architecture. For both VCAS and HCAS, the same AI/ML constituent architecture was designed, shown in Figure 5.8. The input into the AI/ML constituent is the GPS data for each intruder and ownship, as well as the closest point of approach between these. This input is used in the first step to calculate the inputs for the ML inference model. For the ML inference model, these inputs that have to be calculated are given by the LUTs that are compressed by the ML inference model. The LUTs for VCAS and HCAS are indexed by different parameters that describe the encounter between the ownship and the intruder. Following the calculation of the inputs for the ML inference model, the next step is the discretization of these inputs. This is necessary as the LUTs are discrete, and the ML inference model only approximates these discrete states in the LUT and is not designed to extrapolate these discrete states to contiguous states. For the discretization, a simple nearest-neighbor approach is chosen. Following the discretization, the next component is the Safety Net for the ML inference model. As stated in a requirement for VCAS and HCAS, the LUTs must be compressed in such a way that for all states of the LUTs, VCAS and HCAS outputs the same advisory. Therefore, no difference in advisory is allowed between the LUTs, and VCAS or HCAS. To achieve this, as described in section 5.1, all states the neural network's compressed wrongly are stored in a smaller sparse LUT. This sparse LUT is saved as a hash table, using the LUT indexes as keys. The hash table saves both the advisory cost and the accompanying LUT index. A hash table is chosen as it has in the best case a look-up complexity of $\mathcal{O}(1)$ [114]. But this optimal complexity is only achieved when the collisions in the hash table are kept to a minimum [114]. The minimization of the collision can be achieved, as the sparse LUTs do not change after the neural networks are trained and when the AI/ML constituent is in use. Therefore,

the hash table can be optimized to minimize these collisions by for example the design of the hashing function. To select between the neural network and the sparse LUT, it is simply checked if, at the calculated position in the hash table, the LUT indexes match. If these do not match, the neural network is used to predict the advisory, otherwise the entry in the sparse LUT is used. For the neural network architecture, a feedforward architecture based on multi-layer perceptrons is chosen. The output of the neural network and sparse LUT is the cost of each possible advisory, in post-processing, this advisory cost is transformed into a single advisory by selecting the lowest cost advisory. This final selected advisory is then passed to *Multi-intruder advisory selection* component, see Figure 5.6.

5.6.1 Verification

Objective DA-06 requires the description of the AI/ML constituent architecture. Additionally, objective LM-01 requires the description of the description of the ML model architecture. However, for both objectives, no specific requirements regarding the description of the architectures are given, therefore these objectives are seen as fulfilled completely by the above-provided description.

5.7 Framework Validation

The goal of the introduced methodology is to enable the developers of an AI-based system to fulfill the objectives introduced by EASA. For the methodology, a subset of all EASA's objectives was selected, and these selected objectives imposed the conditions, that have to be fulfilled by the individual steps in the methodology. These imposed conditions were verified in the previous sections by the application to the PYCASX use case. The methodology was able to fulfill 76.9 % of the considered objectives requirements completely. No requirements were not fulfilled and 23.1 % of considered objective requirements were fulfilled only partially. These requirements were not fulfilled completely as either, the methodology only achieved results in the use case that covered the requirements indirectly, or the results depended on later steps of the W-shaped learning assurance process and therefore can only be fulfilled at the end of the development. Nevertheless, as most of the requirements are fulfilled, the verification of the methodology, using the PYCASX use case, is seen as successful. Furthermore, as shown in the previous sections, the methodology extended and concretized different concepts of the EASA concept paper. Firstly, a specification language was introduced for the OD. In the presented use case, it was demonstrated, that this introduced language allowed the definition of an OD for a system designed for the airborne collision avoidance use case. Secondly, as demonstrated in section 5.5, the possibility of defining an AI/ML constituent ODD by a developer using the introduced specification format was presented. To the best of the knowledge of the authors, so far no such specification language existed before regarding the AI/ML constituent ODD concept of EASA. Furthermore, the application of the introduced processes demonstrated that these allow a developer to meet the required traceability requirements by EASA. Importantly, as shown in the steps of defining the ODD for VCAS and HCAS, an approach was demonstrated to get from the system level to the AI/ML constituent level. Thus, the application of the methodology to the PYCASX use case, confirmed the benefits of the introduced formalisms and processes required by a developer, therefore completing the necessary validation of the methodology.

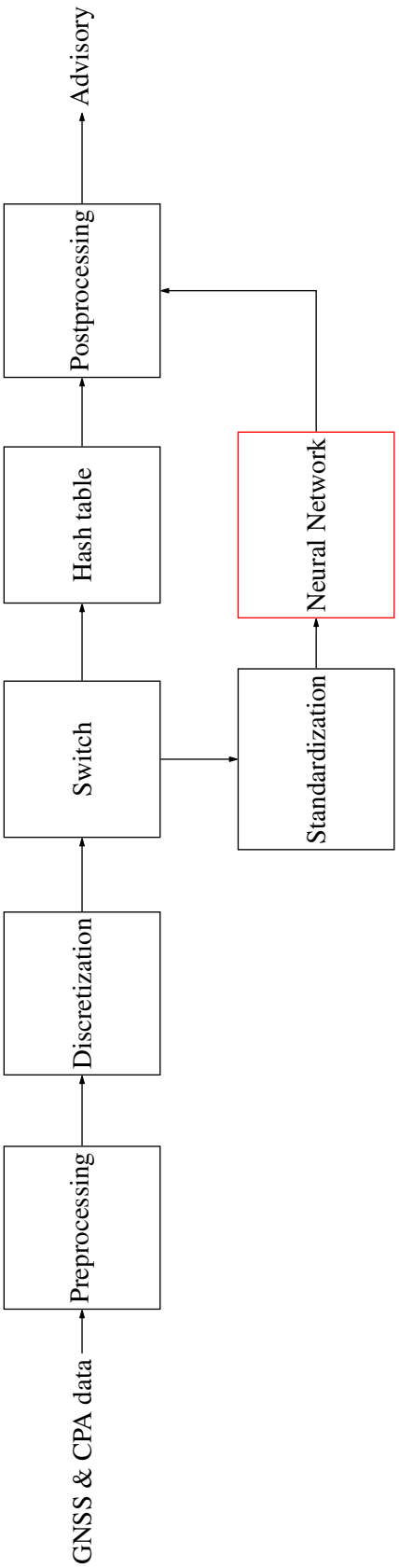


Figure 5.8: Schematic of the AI/ML constituent architecture. Each arrow represents the flow of information from one item to the next. Each box represents a software item in the architecture. The ML inference model item is highlighted with a red outline.

Table 5.9: AI/ML constituent ODD for HCAS.

Top-level attribute	Sub-attribute	Qualifier	Attribute	Attribute value	Unit	Distribution	Source
Scenery	Airspace	Include	Type	C	-	Constant	-
	Airspace	Include	Flight Rule	IFR, VFR	-	-	-
	Airspace	Include	Route Type	Free Route Airspace	-	Constant	-
Environment	-	-	-	-	-	-	-
Dynamic Elements	Intruder	Include	Agent Type	Airplane	-	Constant	ADS-B
	Intruder	Include	Horizontal Airspeed	[0, 600]	kn	-	ADS-B
	Intruder	Include	Horizontal Acceleration	[−48.261, 48.261]	$\mathrm{ft\,s^{-2}}$	-	-
	Intruder	Include	Relative Angle to Ownship	[−180, 180]	°	-	-
	Intruder	Include	Time until loss of separation	[0, 60]	s	-	-
	Ownship	Include	Agent Type	Airplane	-	Constant	-
	Ownship	Include	Horizontal Airspeed	[0, 600]	kn	-	GPS
	Ownship	Include	Horizontal Acceleration	[−48.261, 48.261]	$\mathrm{ft\,s^{-2}}$	-	GPS
	Ownship	Include	Distance to Intruder	[0, 122 000]	ft	-	-
	Ownship	Include	Angle to Intruder	[−180, 180]	°	-	-
	Ownship	Include	Turn Rate Capability	$\geq$3	$^\circ\,\mathrm{s^{-1}}$	-	-
	Ownship	Include	Turn Rate Acceleration Capability	$\geq$1	$^\circ\,\mathrm{s^{-2}}$	-	-
	Ownship	Include	Pilot reaction time	[0]	s	Constant	-
Operating Parameters	Ownship	Include	GPS Inaccuracy	None	-	Constant	GPS
	Intruder	Include	GPS Inaccuracy	None	-	Constant	GPS

6 Discussion

After the introduction of the methodology and the validation, this section discusses the findings of both sections. In section 6.1, the methodology is analyzed regarding its advantages and gaps. Following the analysis, a comparison of the defined AI/ML constituent ODD is conducted using the example of the HCAS ODD. Lastly, the chapter concludes by discussing the applicability of the guidelines given by EASA's concept paper and how these guidelines can be further refined.

6.1 Analysis of the Methodology

In the previous chapter, the feasibility of the proposed methodology to fulfill the different requirements based on the considered objectives, see Table 1.1, was shown. With this novel methodology, it was possible to fulfill most of the objectives for the selected use case of PYCASX and the corresponding AI/ML constituents VCAS and HCAS. The fulfillment of not all requirements of the objectives was due to the fact, that the proposed methodology only covers a subset of all steps in the development of an AI-based system. For example, for the AI/ML constituent ODD not for every attribute a distribution was determined as these distributions depend on the collected data sets. These data sets can only be determined after the AI/ML constituent is fully developed according to EASA's guidelines. Nevertheless, despite not completely meeting all requirements in the methodology's application of the use case, the methodology still showed its advantage of consolidating different approaches and concepts into one unified approach. As introduced in chapter 3, different approaches deal with defining concepts, such as the OD and the ODD. However, each of these is considered only a part of the steps necessary for the definition of the concepts. Furthermore, they also do not consider the dependencies between the different concepts, as is required by EASA [8]. Therefore, the novelty of the introduced methodology is the combination and extension of these approaches and the tailored adherence to EASA's requirements.

In the methodology for the specification of the OD and AI/ML constituent ODD, a tabular format was chosen. The chosen format was based on ISO 34503:2023 [50], which formalized an ODD specification format for automated driving systems using a hierarchical taxonomy. The advantage of the tabular format is that it is readable by the different stakeholders connected to the system [50]. However, one disadvantage of the tabular format is the missing of an abstract syntax that formalizes the metamodel of the OD and ODD. The syntax of the specification is only bound by the given textual description and by the reference to ISO 34503:2023 [50]. Therefore, a syntactically wrongly defined OD or ODD would only be discovered in the verification. Defining an abstract syntax for the OD and ODD, and a concrete syntax using a modeling language, can solve this issue. Furthermore, using a modeling language would allow the use of model-to-text transformations. Such transformations can be useful in the context of scenario-based testing, as these transformations could automatically generate the necessary testing configurations.

Applying the introduced methodology in chapter 5, showed its usability to identify and specify individual operation conditions for the OD or ODD. However, one type of relation that is currently not included in the methodology is the dependencies between operating parameters. Such dependencies

can be important to represent parameters that entail different requirements or restrictions for other parameters. For example, the parameter *Airspace Type* of the PYCASX OD, see Table 5.3, imposes restrictions on other parameters such as the *Altitude*. For the chosen attribute value *C*, only altitudes of 10 000 ft to 66 000 ft are allowed. In its current form, this dependency of the *Altitude* on the *Type* is not depicted by the OD. Depiction of these dependencies is also important for the AI/ML constituent ODD, to accurately depict changes in the data distribution based on these dependencies.

In the validation, the methodology was applied to the airborne collision avoidance use case with a focus on the PYCASX system. For the AI/ML constituents of VCAS and HCAS, an ODD was defined for each by following the introduced steps in the methodology. Two key steps in the methodology are the identification of sensor characteristics and the identification of domain-specific concepts. However, in the selected use case these steps only yielded two additional attributes, the *Response Time* and the *GPS accuracy*. Therefore, these steps seem to only provide little additional information for the specification of the AI/ML constituent ODD. While this is true for the selected use case, the steps were selected based on methodologies that were applied to other use cases [65], [67]. In those use cases, these methodologies identified numerous additional attributes. Therefore, if a different use case had been selected for the validation, e.g., a camera-based system, these steps would have identified additional attributes. Furthermore, these steps are still important for the specification of the AI/ML constituent ODD, because only when an ODD is complete, can an argument be made for the safety of the system [54].

As shown in the validation, the methodology allows a specification of an ODD that fulfills the requirements outlined in the objectives of EASA [8], and a specification of an ODD that provides a high level of detail necessary for the data management and ODD monitoring. However, the latter argument only is based on the application of the methodology to the selected use case. As the methodology describes only a process, the results of this process are not guaranteed. For example, in Figure 5.7 an ontology-based domain model was defined. This model was defined based on the knowledge of the authors of this thesis. A different set of expert knowledge may produce a different model. Therefore, while the proposed methodology combined and introduced steps to reduce the reliance on expert knowledge in the process of the methodology, it is not able to eliminate the reliance on expert knowledge. Therefore, the proposed methodology is only a process and the quality of the results is not guaranteed by the process. Nevertheless, the introduced methodology made the steps required in the process independent of expert knowledge. Furthermore, the proposed methodology relies on the assumption of the development assurance, that a more stringent process delivers a system with fewer errors than a less stringent process [115]. Therefore, as the proposed methodology is more stringent and works in the framework of the AI assurance process, it is assumed that the development assurance assumption applies to the proposed methodology. If such an assumption is valid for the AI assurance of EASA, on which the introduced methodology is based on, is not seen as the scope of this thesis.

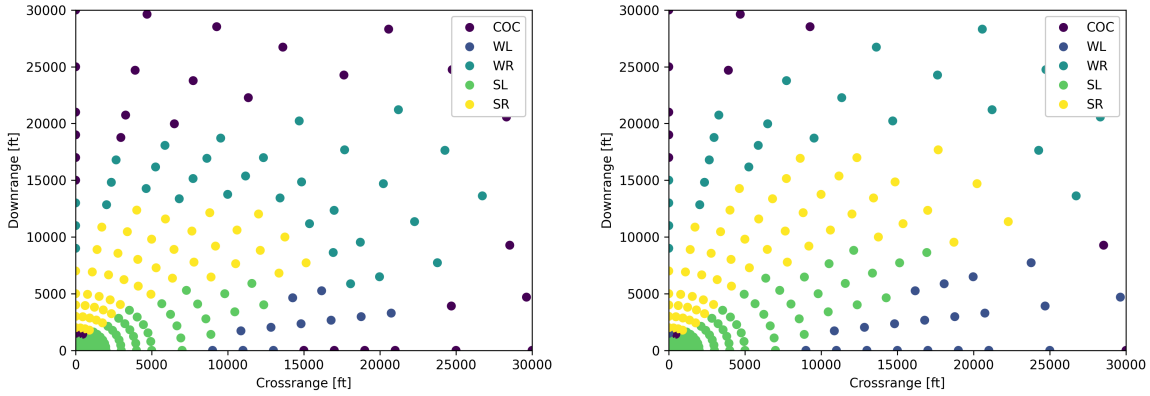
6.2 Comparison of the Defined HCAS ODD

The advantage of the introduced methodology can be exemplified when comparing the results of the defined AI/ML constituent ODD with other ODDs that were specified for the same use case. One example of an ODD, defined for the same use case of the airborne collision avoidance system, is shown in Figure 6.1. The specified ACAS X_U ODD only contains six attributes and their ranges. The HCAS ODD of this work is the equivalent to the shown ACAS X_U ODD. Compared to the ODD specified in this work, the ACAS X_U ODD lacks the attribute classifications, the units, the qualifier, the attribute sources, and the distributions for the attributes. The latter is significant, as this is a requirement for an

$$ODD_{\text{ACAS X}_\text{U}} = \begin{cases} \tau \in [0, 101] \\ \theta \in [-3.14, 3.14] \\ v_{int} \in [0, 1200] \\ v_{own} \in [50, 1200] \\ \psi \in [-3.14, 3.14] \\ \rho \in [499, 185318] \end{cases}$$

Figure 6.1: The ACAS X_U ODD defined in the MLEAP report [33]. From top to bottom, the individual attributes describe the time to loss of separation in seconds, the relative angle to the intruder, the speed of the intruder in feet per minute, the speed of the ownship in feet per minute, the intruder heading relative to the onwship, and the distance of the intruder to the onwship in feet. Taken from [33].

AI/ML constituent ODD according to EASA [8]. Therefore, the specified ACAS X_U ODD, as shown in Figure 6.1, does not fulfill the requirements of EASA. Furthermore, while the specified HCAS ODD contains 16 attributes, the MLEAP's ACAS X_U ODD only contains six attributes. The six attributes are all contained in the HCAS ODD, and the major difference in attributes are the attributes that determine the behavior of the ownship and intruder, such as the *Turn Rate Capability*, see Table 5.9. However, it is argued that these attributes have to be included in the ODD, as they determine different behavior characteristics that can change the optimal advisories in different scenarios. As the advisories in the LUTs are based on the result of a simulation process, different assumptions for this simulation can be compared [96]. In Figure 6.2, an encounter between an ownship and intruder and the best advisories for each state of the LUTs is shown. The only difference between these two figures is that in Figure 6.2a, the *Turn Rate Capability* of the HCAS ODD, and in the Figure 6.2b a *Turn Rate Capability* of $2\,^\circ\,s^{-1}$ is used. Therefore, it is assumed that the maximum *Turn Rate Capability* is limited to $2\,^\circ\,s^{-1}$ and the response for each advisory is only half the expected turn rate, in Figure 6.2b. In the depicted case, an intruder approaches the ownship by flying parallel to the *Downrange* axis, while the ownship flies along the *Crossrange* axis. Comparing the advisories for each point in the LUT, one can observe that in the case where the airplane has the lower maximum turn rate, the HCAS changes its advisories for intruders further away to a *strong right (SR)* compared to the *weak right (WR)* for the airplane with the higher maximum turn rate. In the ACAS X_U case, both depicted LUTs conform to the ACAS X_U ODD, while for the specified HCAS ODD, only the left LUT is permitted, as the right LUT does not satisfy the *Turn Rate Capability*. The right LUT is unsuitable as it does not conform to the performance requirements of the DO-386 [25]. The missing parameters in the MLEAP's ACAS X_U ODD are also a problem for the ODD monitoring, as they are not included in the ODD and therefore these will not be monitored for an AI/ML constituent. This becomes an issue because it cannot be adequately determined if the AI/ML constituent was designed to operate in these scenarios. Comparing the two ODDs shows the advantage of the introduced methodology, as it can produce an ODD with a higher level of detail. Such a level of detail is important, as only if the ODD accurately describes the conditions the AI/ML constituent can encounter, a safety argument can be built upon the ODD for the system [54]. Furthermore, as the AI/ML constituent ODD is also the framework for the selection, collection, and preparation of the data used to develop the ML inference model, the completeness of the data also depends on the defined ODD [8]. For the ACAS X_U ODD, the issue with these missing attributes is, that even when a dataset is collected, which is shown to completely represent the defined ACAS X_U ODD, it is not guaranteed that this dataset covers all operational conditions to which the ML inference model can be exposed. Only if all relevant operational conditions are described with an attribute in the ODD, a collected dataset



(a) Advisories where the ownship can achieve the expected turn rate.

(b) Advisories where the ownship can only achieve half the expected turn rate.

Figure 6.2: An excerpt of the advisories for an intruder which approaches the ownship with a speed of 200 ft s^{-1} from an −90° angle. The ownship speed is 200 ft s^{-1}, and the previous advisory is clear of conflict (COC). Each dot represents an entry in the LUT for which an advisory was calculated. The parameter configurations for the advisories in the simulation are listed in Appendix D.

can be accurately determined to cover all these conditions. For example, the parameter of the *Agent Type* is included in the HCAS ODD but is missing in the ACAS X_U ODD. Using the ACAS X_U ODD allows for all types of agents, such as rotorcraft, to be included in the dataset, while the HCAS ODD is limited to airplanes. In the worst case for the ACAS X_U ODD, all the data collected can be from agent types for which the system is not designed to generate collision advisories, while the collected data still conforms to the defined ODD. It must be noted, that for the chosen use case the data source, a simulation process, is known. As the assumptions and properties of the simulation process are clearly defined and are driven by a set of stated requirements [25], the included attributes of the HCAS ODD are already covered in the LUTs. Therefore, it is possible to argue in this use case, that the inclusion of these attributes in the ODD for the AI/ML constituent is redundant. While this argument can apply to this use case, there are two arguments against relying on it. Firstly, it is only valid for this use case but not in general, and secondly, the required purpose of the ODD to be "a framework for the selection, collection, preparation of the data" [8, p. 17] is not fulfilled. Therefore, relying solely on the definition of important data characteristics outside the ODD is seen as inadequate for the definition of the AI/ML constituent ODD.

6.3 Application of EASA's Guidelines

As introduced in chapter 3, the guidance for level 1 and 2 machine learning applications by EASA was only released in 2024, and the guidance for level 3 is only expected in the upcoming years. Therefore, some objectives and their subsequent derived requirements may change when this new guidance is released. Such changes may then have to be incorporated into an updated form of the proposed methodology. But, as these guidelines are currently still in development, the findings of this thesis can be seen, as feedback for this development. For example, other works have identified issues or needs for improvements in some objectives of the AI assurance [116]. The main issue that arose in the development of the methodology of this work was the ambiguity that existed in some of the

objectives considered. One example, where ambiguities exist, is the definition of the OD by EASA. The OD is defined similarly to the ODD, in that it should describe the operation conditions at the system level [8]. The OD is defined to describe the operation conditions at the system level, which is similar to the ODD [8]. However, in their definition, it is not clear if the idea of an OD is solely based on already existing practices in the aviation sector, or if the OD is also based on practices of other industries, such as the automotive sector. In this work, the latter interpretation was chosen. Also for the AI/ML constituent ODD, it is unclear if EASA permits the specification of the AI/ML constituent ODD based on collected data or if the AI/ML constituent ODD only relies on a top-down allocation approach of requirements and functions. In the extreme case, an ODD could be solely built based on the extracted characteristics of an already collected dataset, and the system is then adapted according to the defined ODD. This would defeat the purpose of the AI/ML constituent ODD to be "a framework for the selection, collection, preparation of the data" [8, p. 17]. However, as EASA allows an iterative learning assurance process, such an ODD specification approach would be possible. In this work, the AI/ML constituent ODD was interpreted to be required to be the result of a top-down allocation approach of requirements and functions.

7 Summary

The following section provides a summary of the thesis. This includes a conclusion of the findings of the previous sections. Furthermore, the contributions of this thesis to the research questions that were outlined in chapter 1 are given. Lastly, an outlook for future research directions is given.

7.1 Conclusion

This work addressed the issue of specifying the operating conditions of an AI-based system to comply with potential future EASA regulations. Therefore, a methodology was introduced that is built according to the objectives outlined in the objectives of EASA's current guidelines [8]. The methodology includes formats to define the OD, functional decomposition, and the AI/ML constituent ODD. The formats of the OD and AI/ML constituent ODD are based on developed formats from the automotive domain but are extended to fit the requirements of EASA guidelines. Especially, for the AI/ML constituent ODD, additions to this format were made to incorporate the data-specific conditions, which EASA requires. In addition to the formats required for the concepts, processes are introduced for the application of these concepts to concrete use cases. Importantly, a process was introduced that derives the AI/ML constituent ODD based on the OD and the description of the system. This process consists of steps, such as the projection of OD attributes or the development of models that describe and deconstruct the operating environment, to successfully identify the individual operating conditions for the AI/ML constituent. Furthermore, a mapping of the individual factors that influence the AI/ML constituent and ML model input and architecture was created.

To validate its capabilities, the methodology was applied to an airborne collision avoidance use case, where the goal is to introduce a neural network-based compression. Based on the use case, a ConOps, OD, and functional decomposition of the AI-based system was conducted according to the defined methodology. The usability of decomposing the use case environment using a six-layer modeling approach to identify attributes that describe operational conditions was demonstrated. Furthermore, the applicability of the process to define the AI/ML constituent ODD was demonstrated. For the use case, two different AI/ML constituent ODDs had to be defined as the system determines advisories in the vertical and horizontal plane separately. The first step in their definition was to project system-level attributes of encounters onto encounters as perceived by the AI/ML constituent. Additionally, sensor characteristics and domain-specific concepts such as the response time of the pilot were identified. Lastly, an AI/ML constituent architecture was defined that incorporated the specifics of the defined ODD and implemented the Safety Net.

In the validation, it was verified that most of the requirements of EASA's objective were satisfied. The methodology also showed the traceability of the defined AI/ML constituent ODD attributes to the defined OD and ConOps through the application of the individual steps of the process. A comparison of the resulting HCAS ODD also showed that with the application of the proposed methodology, an ODD with a higher level of detail was constructed for the use case. Importantly, the methodology was able to identify patterns that a developer can use to specify an AI/ML constituent ODD. However, while the methodology introduced a process independent of expert knowledge, the individual steps and

their result largely depend on expert knowledge. For example, the definition of the ontology-based domain model for the identification of domain-specific concepts largely depends on the developer's knowledge. Lastly, based on the experience of the application of EASA's concepts, a further need to clarify aspects of EASA guidelines was identified.

7.2 Contributions

In chapter 1 three research questions were introduced, that are addressed in this work. The contribution to each research question can be listed as the following:

1. **How can developers define a Concept of Operations, an Operational Domain, and a functional decomposition for the AI-based system?**
   For the functional decomposition and the Concept of Operations, existing approaches such as the ISO/IEC/IEEE 29148:2018 [69] were chosen, which only required small adaptions to allow their application to fulfill EASA's requirements. The specification of the operational domain, a tabular format was introduced based on the ISO 34503:2023 [50]. The most notable change required is the definition of a taxonomy that has to be adapted for individual use cases in aviation. Therefore an identification approach was introduced, which is based on the decomposition and structuring of the operational environment described in the ConOps using a layered approach. Using the introduced 6-layer approach, the individual attributes for the *scenery elements*, *environmental elements*, and *dynamic elements* of the taxonomy can be identified by a developer. Additionally, a process was introduced that can be used to specify an OD for an AI-based system.

2. **Based on the artifacts produced by the first research problem, how can developers define an AI/ML constituent Operational Design Domain?**
   For the definition of the AI/ML constituent ODD, a specification language and process was introduced. The introduced specification language is based on the specification language that was introduced for the OD. However, due to the focus of the AI/ML constituent ODD being a framework for the data management, additional aspects such as the distributions were introduced into the specification language. The same requirement to include data-specific considerations for the AI/ML constituent ODD was also added to the process for the definition of the AI/ML constituent ODD. Especially in the refinement step of the process, additional steps, such as the projection of attributes, the identification of sensor characteristics, and the identification of domain-specific concepts were added. These steps are important, as only when the data characteristics that are described with the AI/ML constituent ODD match the characteristics of the data in the deployment of the ML inference model, the performance of the ML inference model can be guaranteed.

3. **How can the AI/ML constituent architecture be designed based on the defined AI/ML constituent ODD and the requirements allocated to this AI/ML constituent?**
   The AI/ML constituent architecture depends on many different aspects that are allocated to the AI/ML constituent. These aspects include the AI/ML constituent ODD, the data type, the requirements, and the allocated function. Importantly, the AI/ML constituent ODD has a direct influence on the inputs, as the ODD defines the information that must be collected. Furthermore, the AI/ML constituent ODD has a direct influence on the data type, i.e., what type of data has to be collected for the AI/ML constituent. This data type then determines the possible inputs and architecture that can be chosen for the available data.

7.3 Outlook

As discussed in chapter 6, this thesis provided a baseline for the definition of the ConOps, OD, and AI/ML constituent ODD, but there are also gaps and areas of possible improvement. Firstly, the currently provided processes and guidance are mostly conceptual, which have to be further refined and concretized. This concretization can be in the form of the development of a tool and an abstract syntax with which the OD and ODD can be defined. Such tools are valuable, as they remove the necessity to validate whether an OD or ODD conforms to the defined syntax, and as they allow model-to-text transformations, that can be used in the OD or ODD-based testing. Secondly, the introduced methodology was only applied to a single use case, therefore an application to a greater variety of use cases can identify further possible steps to refine the methodology. Especially, use cases that employ complex sensor setups are of interest, as these would further validate and challenge the proposed methodology to identify sensor characteristics or domain-specific concepts. Thirdly, the current processes and the conducted validation only considered the specification of the operating conditions for a system, therefore the next step required is the definition of processes for the data management based on the defined AI/ML constituent ODD. One important step in this process will be the verification that a collected dataset conforms to the defined AI/ML constituent ODD. Furthermore, the collected data can also provide additional operating conditions that have to be included in the AI/ML constituent ODD. Therefore, the adaption of the proposed processes to allow an iterative specification process may be needed. Lastly, as the introduced processes currently only use the available EASA guidance for level 1 and 2 machine learning applications, adaptions of the processes might be necessary based on the upcoming guidance for level 3. As already suggested, the proposed processes should be applied to further use cases for additional validation. Here a use case of level 3 could be chosen, to find gaps in the current processes. But importantly, this application to level 3 use cases can also be used to provide feedback to EASA in their development of the guidance for level 3 machine learning applications.

Bibliography

[1] European Union Aviation Safety Agency (EASA), "Artifical Intelligence Roadmap 2.0, Human-centric approach to ai in aviation," European Union Aviation Safety Agency (EASA), Postfach 10 12 53, 50452 Cologne, Germany, Tech. Rep., version 2.0, 2023-05.

[2] Council of European Union, *Regulation (EC) No 1592/2002 of the European Parliament and of the Council of 15 July 2002 on common rules in the field of civil aviation and establishing a European Aviation Safety Agency (Text with EEA relevance)*, European Parliament, Ed., Document 32002R1592, European Union, 2002-09-07. [Online]. Available: http://data.europa.eu/eli/reg/2002/1592/oj (visited on 2024-12-18).

[3] European Union Aviation Safety Agency. "The Agency." European Union Aviation Safety Agency, Ed., European Union Aviation Safety Agency. (2024), [Online]. Available: https://www.easa.europa.eu/en/the-agency/the-agency (visited on 2024-08-15).

[4] S-18 Aircraft and Sys Dev and Safety Assessment Committee, *Guidelines for Development of Civil Aircraft and Systems*, 2010-12. DOI: 10.4271/ARP4754A.

[5] RTCA, Inc., "DO-178C, Software considerations in airborne systems and equipment certification," GlobalSpec, Washington, DC, USA, Tech. Rep., 2012-01. [Online]. Available: https://my.rtca.org/productdetails?id=a1B36000001IcmqEAC.

[6] RTCA, Inc., "DO-200B, Standards for processing aeronautical data," GlobalSpec, Washington, DC, USA, Tech. Rep., 2015-06-18. [Online]. Available: https://my.rtca.org/productdetails?id=a1B36000001IclyEAC.

[7] G-34 Artificial Intelligence in Aviation, *Artificial Intelligence in Aeronautical Systems: Statement of Concerns*, 2021-04. DOI: 10.4271/AIR6988.

[8] European Union Aviation Safety Agency (EASA), "EASA Concept Paper: Guidance for Level 1 & 2 machine learning applications, A deliverable of the EASA AI Roadmap," European Union Aviation Safety Agency (EASA), Postfach 10 12 53, 50452 Cologne, Germany, Tech. Rep., version Issue 02, 2024-03-06. [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-concept-paper-issue-2.

[9] European Aviation Safety Agency, "Certification Specifications for Large Aeroplanes CS-25," Tech. Rep., 2003-10-17.

[10] J.-G. Durand, A. Dubois, and R. J. Moss, "Formal and Practical Elements for the Certification of Machine Learning Systems," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, 2023, pp. 1–10. DOI: 10.1109/DASC58513.2023.10311201.

[11] S-18 Aircraft and Sys Dev and Safety Assessment Committee, *Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment*, 2023-12. DOI: 10.4271/ARP4761A.

[12] RTCA, Inc., "DO-254, Design Assurance Guidance for Airborne Electronic Hardware)," GlobalSpec, Washington, DC, USA, Tech. Rep., 2000-04-19. [Online]. Available: https://products.rtca.org/21dis94/.

[13] B. Luettig, Y. Akhiat, and Z. Daw, "ML meets aerospace: challenges of certifying airborne AI," *Frontiers in Aerospace Engineering*, vol. 3, 2024, ISSN: 2813-2831. DOI: 10.3389/fpace.2024.1475139.

[14] European Union Aviation Safety Agency (EASA), "Artificial Intelligence Roadmap 1.0, A human-centric approach to ai in aviation," European Union Aviation Safety Agency (EASA), Postfach 10 12 53, 50452 Cologne, Germany, Tech. Rep., version 1.0, 2020-02-07, pp. 1–33, 33 pp. [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-roadmap-10.

[15] High-Level Expert Group on AI, *The Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment*. European Commission, 2020-07-17.

[16] Federal Aviation Administration, *Introduction to TCAS II: Version 7.1*, FAA TCAS Program Office, AJP-67, 800 Independence Avenue, S.W., Washington, D.C. 20591, 2011-02-28. [Online]. Available: https://www.faa.gov/documentlibrary/media/advisory_circular/tcas%20ii%20v7.1%20intro%20booklet.pdf (visited on 2024-08-12).

[17] The European Organisation for the Safety of Air Navigation (EUROCONTROL), "ACAS Guide, Airborne collision avoidance systems," EUROCONTROL, Tech. Rep., version 4.1, 2022-03-25. [Online]. Available: https://www.eurocontrol.int/publication/airborne-collision-avoidance-system-acas-guide (visited on 2025-01-02).

[18] International Civil Aviation Organization (ICAO), "Airborne Collision Avoidance System (ACAS) Manual," International Civil Aviation Organization (ICAO), Tech. Rep., version 1.0, 2006, Doc 9863 AN/461.

[19] Council of European Union, "Commission Regulation (EU) No 1332/2011, Laying down common airspace usage requirements and operating procedures for airborne collision avoidance," Tech. Rep., 2011-12-16. [Online]. Available: http://data.europa.eu/eli/reg/2011/1332/2016-08-25.

[20] Office of the Federal Register National Archives and Records Administration, "Code of Federal Regulations: Title 14, Aeronautics and space," Office of the Federal Register National Archives and Records Administration, Tech. Rep., 2023-01-01.

[21] RTCA, Inc., "DO-185B, Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II) Airborne Equipment," GlobalSpec, Washington, DC, USA, Tech. Rep., 2008-06-19. [Online]. Available: https://my.rtca.org/productdetails?id=a1B36000001IcmYEAS.

[22] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, "Next-Generation Airborne Collision Avoidance System," *Lincoln Laboratory Journal*, vol. 19, no. 1, pp. 17–33, 2012. [Online]. Available: https://www.ll.mit.edu/sites/default/files/publication/doc/next-generation-airborne-collision-avoidance-system-kochenderfer-ja-20264.pdf.

[23] A. Gjersvik, "Enhanced Parallel Simulation for ACAS X Development," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1–7. DOI: 10.1109/HPEC43674.2020.9286197.

[24] RTCA, Inc., "DO-385, Minimum Operational Performance Standards for Airborne Collision Avoidance System X (ACAS X) (ACAS Xa and ACAS Xo)," GlobalSpec, Washington, DC, USA, Tech. Rep., 2018-10-02. [Online]. Available: https://standards.globalspec.com/std/14222352/RTCA%20DO-385.

[25] RTCA, Inc., "DO-386 Volume 1 & 2, DO-386 Vol I Minimum Operational Performance Standards for Airborne Collision Avoidance System Xu (ACAS Xu) (Vol I), and DO-386 Vol II Minimum Operational Performance Standards for Airborne Collision Avoidance System Xu (ACAS Xu) (Vol II: Algorithm Design Description)," GlobalSpec, Washington, DC, USA, Tech. Rep., 2020-12-17. [Online]. Available: https://standards.globalspec.com/std/14360425/rtca-do-386-volume-1-2.

[26] K. D. Julian and M. J. Kochenderfer, "Guaranteeing Safety for Neural Network-Based Aircraft Collision Avoidance Systems," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, IEEE, 2019-09. DOI: 10.1109/dasc43569.2019.9081748.

[27] J. M. Christensen, W. Zaeske, A. Anilkumar Girija, *et al.*, "Towards Certifiable AI in Aviation: A Framework for Neural Network Assurance Using Advanced Visualization and Safety Nets," in *2024 IEEE/AIAA 43st Digital Avionics Systems Conference (DASC)*, 2024-09.

[28] J. M. Christensen, A. Anilkumar Girija, T. Stefani, *et al.*, "Advancing the AI-Based Realization of ACAS X Towards Real-World Application," in *36th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2024-10.

[29] Federal Aviation Administration, "Roadmap for Artificial Intelligence Safety Assurance," Federal Aviation Administration, 800 Independence Avenue, SW, Washington, DC 20591, Tech. Rep., 2024-07-23. [Online]. Available: https://www.faa.gov/media/82891 (visited on 2025-01-20).

[30] European Union Aviation Safety Agency (EASA), "EASA Concept Paper: First usable guidance for Level 1 machine learning applications, A deliverable of the EASA AI Roadmap," European Union Aviation Safety Agency (EASA), Postfach 10 12 53, 50452 Cologne, Germany, Tech. Rep., version Issue 01, 2021-12-20. [Online]. Available: https://www.easa.europa.eu/en/downloads/134357/en.

[31] EASA and Daedalean, "Concepts of Design Assurance for Neural Networks (CoDANN)," Public Report Extract, Tech. Rep., version 1.0, 2020-03-31. [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/concepts-design-assurance-neural-networks-codann.

[32] EASA and Daedalean, "Concepts of Design Assurance for Neural Networks (CoDANN) II with appendix B," Public Extract, European Union Aviation Safety Agency (EASA), research rep., version 1.1, 2024-01-24, pp. 1–118, 118 pp. [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/concepts-design-assurance-neural-networks-codann-ii.

[33] MLEAP Consortium, "EASA Research – Machine Learning Application Approval (MLEAP) Final Report," European Union Aviation Safety Agency (EASA), Tech. Rep. 1, 2024-05, pp. 1–522, 522 pp. [Online]. Available: https://www.easa.europa.eu/en/research-projects/machine-learning-application-approval (visited on 2024-09-22).

[34] A. Gu/egan, *WG-114 AI Standards in Aviation*. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2024-04/20240429-flyai-forum-session-2-guegan.pdf (visited on 2025-01-10).

[35] "G-34 Artificial Intelligence in Aviation," SAE International. (), [Online]. Available: https://standardsworks.sae.org/standards-committees/g-34-artificial-intelligence-aviation# (visited on 2025-01-10).

[36] G-34 Artifical Intelligence In Aviation Committee. "Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI," SAE International. (), [Online]. Available: https://www.sae.org/standards/content/arp6983/ (visited on 2024-11-16).

[37] H. Bello, D. Geißler, L. Ray, *et al.*, *Towards certifiable ai in aviation: Landscape, challenges, and opportunities*, 2024. [Online]. Available: https://arxiv.org/abs/2409.08666.

[38] M. Gariel, B. Shimanuki, R. Timpe, and E. Wilson, *Framework for certification of ai-based systems*, 2023. [Online]. Available: https://arxiv.org/abs/2302.11049.

[39] C. Hasterok, J. Stompe, J. Pfrommer, *et al.*, "PAISE®. Das Vorgehensmodell für KI-Engineering," German, Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung (IOSB), Tech. Rep., 2021. [Online]. Available: https://www.ki-engineering.eu/content/dam/iosb/ki-engineering/downloads/PAISE(R)_Whitepaper_CC-KING.pdf (visited on 2024-12-18).

[40] F. Mamalet, E. Jenn, G. Flandin, *et al.*, "White Paper Machine Learning in Certified Systems," IRT Saint Exupéry ; ANITI, Research Report, 2021-03. [Online]. Available: https://hal.science/hal-03176080.

[41] R. Zhang, A. Albrecht, J. Kausch, H. Putzer J, T. Geipel, and P. Halady, "DDE process: A requirements engineering approach for machine learning in automated driving," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*, IEEE, 2021, pp. 269–279.

[42] On-Road Automated Driving (ORAD) Committee, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 2021. DOI: 10.4271/J3016_202104.

[43] F. Eichenseer, S. Sarkar, and A. Shakeri, "A Systematic Methodology for Specifying the Operational Design Domain of Automated Vehicles," in *35th IEEE International Symposium on Software Reliability Engineering, ISSRE 2024*, IEEE, 2024.

[44] T. Stefani, M. Jameel, R. Hunger, *et al.*, "Towards an Operational Design Domain for Safe Human-AI Teaming in the Field of AI-Based Air Traffic Controller Operations," in *2024 IEEE/AIAA 43st Digital Avionics Systems Conference (DASC)*, IEEE, 2024-09.

[45] E. Thorn, S. C. Kimmel, M. Chaka, B. A. Hamilton, *et al.*, "A Framework for Automated Driving System Testable Cases and Scenarios," United States. Department of Transportation. National Highway Traffic Safety Administration, Washington, DC, USA, Tech. Rep., 2018-09-01. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/38824.

[46] S. Hallerbach, "Simulation-based testing of cooperative and automated vehicles," Ph.D. dissertation, Universität Oldenburg, 2020.

[47] PEGASUS RESEARCH PROJECT, "The PEGASUS Method," Tech. Rep., 2019-02-14. [Online]. Available: https://www.pegasusprojekt.de/en/pegasus-method (visited on 2024-12-09).

[48] T. Charmet, V. Cherfaoui, J. Ibanez-Guzman, and A. Armand, "Overview of the operational design domain monitoring for safe intelligent vehicle navigation," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 5363–5370. DOI: 10.1109/ITSC57777.2023.10421823.

[49] The British Standards Institution, *PAS 1883:2020 - Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification*. BSI Standards Limited, 2020-08, ISBN: 9780539067354. [Online]. Available: https://www.bsigroup.com/en-GB/insights-and-media/insights/brochures/pas-1883-operational-design-domain-odd-taxonomy-for-ads-specification/.

[50] ISO, *ISO 34503: Road Vehicles – Test scenarios for automated driving systems – Specification for operational design domain*, en. Berlin: Beuth Verlag, 2023-08.

[51] K. Czarnecki, "Operational world model ontology for automated driving systems–part 1: Road structure," *Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo*, 2018. DOI: http://dx.doi.org/10.13140/RG.2.2.15521.30568.

[52] K. Czarnecki, "Operational world model ontology for automated driving systems–part 2: Road users, animals, other obstacles, and environmental conditions,"" *Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo*, 2018. DOI: http://dx.doi.org/10.13140/RG.2.2.11327.00165.

[53] L. Mendiboure, M. L. Benzagouta, D. Gruyer, T. Sylla, M. Adedjouma, and A. Hedhli, "Operational design domain for automated driving systems: Taxonomy definition and application," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–6. DOI: 10.1109/IV55152.2023.10186765.

[54] G. Weiss, M. Zeller, H. Schoenhaar, C. D. Fraunhofer, and A. Kreutz, "Approach for Argumenting Safety on Basis of an Operational Design Domain," in *2024 IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, 2024, pp. 184–193.

[55] M. Scholtes, L. Westhofen, L. R. Turner, *et al.*, "6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment," *IEEE Access*, vol. 9, pp. 59 131–59 147, 2021. DOI: 10.1109/ACCESS.2021.3072739.

[56] K. Fjørtoft and Ø. Rødseth, "Using the operational envelope to make autonomous ships safer," in *The 30th European Safety and Reliability Conference, Venice, Italy*, 2020.

[57] S. Picard, C. Chapdelaine, C. Cappi, *et al.*, "Ensuring Dataset Quality for Machine Learning Certification," in *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2020, pp. 275–282. DOI: 10.1109/ISSREW51248.2020.00085.

[58] M. Adedjouma, B. Botella, J. Ibanez-Guzman, K. Mantissa, C.-M. Proum, and A. Smaoui, "Defining operational design domain for autonomous systems: A domain-agnostic and risk-based approach," in *SOSE 2024 - 19th Annual System of Systems Engineering Conference*, Tacoma, United States, 2024-06. [Online]. Available: https://hal.science/hal-04613329 (visited on 2024-08-21).

[59] T. Stefani, J. M. Christensen, E. Hoemann, *et al.*, "Applying Model-Based System Engineering and DEVOPS on the implementation of an AI-based collision avoidance system," in *34rd Congress of the International Council of the Aeronautical Sciences, ICAS 2024*, DGLR, 2024-09. [Online]. Available: https://elib.dlr.de/207884/.

[60] A. Anilkumar Girija, J. M. Christensen, T. Stefani, *et al.*, "Towards the Monitoring of Operational Design Domains using Temporal Scene Analysis in the realm of Artificial Intelligence in Aviation," in *2024 IEEE/AIAA 43st Digital Avionics Systems Conference (DASC)*, IEEE, 2024-09.

[61] T. Stefani, A. Anilkumar Girija, R. Mut, S. Hallerbach, and T. Krüger, "From the Concept of Operations towards an Operational Design Domain for safe AI in Aviation," in *DLRK 2023*, Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V., 2023-09. [Online]. Available: https://elib.dlr.de/197957/.

[62] C. Gabreau, A. Gauffriau, F. D. Grancey, J.-B. Ginestet, and C. Pagetti, "Toward the certification of safety-related systems using ML techniques: the ACAS-Xu experience," in *11th European Congress on Embedded Real Time Software and Systems (ERTS 2022)*, Toulouse, France, 2022-06. [Online]. Available: https://hal.science/hal-03761946.

[63] C. Torens, F. Juenger, S. Schirmer, S. Schopferer, D. Zhukov, and J. C. Dauer, "Ensuring safety of machine learning components using operational design domain," in *AIAA SciTech 2023 Forum*, 2023, p. 1124.

[64] F. Kaakai, S. Adibhatla, G. Pai, and E. Escorihuela, "Data-centric Operational Design Domain Characterization for Machine Learning-based Aeronautical Products," in *International Conference on Computer Safety, Reliability, and Security*, Springer, 2023-09-11, pp. 227–242.

[65] C. Cappi, N. Cohen, M. Ducoffe, *et al.*, "How to design a dataset compliant with an ML-based system ODD?" In *12th European Congress on Embedded Real Time Software and Systems*, Toulouse, France, 2024-06. [Online]. Available: https://hal.science/hal-04614554.

[66] L. Höhndorf, K. Dmitriev, J. K. Vasudevan, S. Subedi, N. Klarmann, and F. Holzapfel, "Artificial Intelligence Verification Based on Operational Design Domain (ODD) Characterizations Utilizing Subset Simulation," in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, 2024, pp. 1–10. DOI: 10.1109/DASC62030.2024.10749586.

[67] M. Herrmann, C. Witt, L. Lake, *et al.*, "Using ontologies for dataset engineering in automotive AI applications," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 526–531. DOI: 10.23919/DATE54114.2022.9774675.

[68] P. Heck, "What about the data? a mapping study on data engineering for ai systems," in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, 2024, pp. 43–52.

[69] "ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering," *ISO/IEC/IEEE 29148:2018(E)*, pp. 1–104, 2018. DOI: 10.1109/IEEESTD.2018.8559686.

[70] A. I. of Aeronautics and Astronautics, *Guide to the Preparation of Operational Concept Documents (ANSI/AIAA G-043B-2018)*, 2018.

[71] A. Väätänen, J. Laarni, and M. Höyhtyä, "Development of a concept of operations for autonomous systems," in *Advances in Human Factors in Robots and Unmanned Systems: Proceedings of the AHFE 2019 International Conference on Human Factors in Robots and Unmanned Systems, July 24-28, 2019, Washington DC, USA 10*, Springer, 2020, pp. 208–216.

[72] Mercedes-Benz Research & Development North America, Inc. "Introducing DRIVE PILOT: An Automated Driving System for the Highway." (2019), [Online]. Available: https://group.mercedes-benz.com/dokumente/innovation/sonstiges/2023-03-06-vssa-mercedes-benz-drive-pilot.pdf (visited on 2024-11-07).

[73] P. Kopardekar, "Enabling Autonomous Flight and Operations in the National Airspace System," in *Enabling Autonomous Flight and Operations in the National Airspace Workshop 2*, 2019. [Online]. Available: https://www.nari.arc.nasa.gov/sites/default/files/attachments/Enabling%20Autonomous%20Flight%20and%20Operations%20in%20the%20NAS%20Report%20-%20Urban%20Air%20Mobility.pdf (visited on 2024-11-27).

[74] A. Shakeri, "Formalization of Operational Domain and Operational Design Domain for Automated Vehicles," in *2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, 2024-04, pp. 990–997. DOI: 10.1109/QRS-C63300.2024.00131.

[75] S. Russell and P. Norvig, *Artificial Intelligence, Global Edition A Modern Approach*. Pearson Deutschland, p. 1168, ISBN: 9781292401133. [Online]. Available: https://elibrary.pearson.de/book/99.150005/9781292401171.

[76] Pilot Institute. "Airport Beacons Explained," Pilot Institute. (2023), [Online]. Available: https://pilotinstitute.com/airport-beacons-explained/ (visited on 2024-10-23).

[77] P. Graf, "Model-Driven Development: 4. Concrete Syntax: Textual and Graphical," Lecture at Ulm University of Applied Sciences, [PowerPoint slides], 2023.

[78] Pilot Institute. "The Difference Between Category, Class, and Type of Aircraft," Pilot Institute. (2023), [Online]. Available: https://pilotinstitute.com/category-class-and-type-of-aircraft/ (visited on 2024-11-08).

[79] ASAM e.V., "ASAM OpenODD: Concept Paper," ASAM e.V., Tech. Rep., version 1.0, 2021-10-01. [Online]. Available: https://www.asam.net/index.php?eID=dumpFile%5C&t=f%5C&f=4544%5C&token=1260ce1c4f0afdbe18261f7137c689b1d9c27576 (visited on 2024-11-20).

[80] C. W. Lee, N. Nayeer, D. E. Garcia, A. Agrawal, and B. Liu, "Identifying the operational design domain for an automated driving system through assessed risk," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1317–1322. DOI: 10.1109/IV47402.2020.9304552.

[81] N. Viola, S. Corpino, M. Fioriti, F. Stesina, *et al.*, "Functional analysis in systems engineering: Methodology and applications," in *Systems engineering-practice and theory*, InTech, 2012, pp. 71–96.

[82] S-18 Aircraft and Sys Dev and Safety Assessment Committee, *GUIDELINES AND METHODS FOR CONDUCTING THE SAFETY ASSESSMENT PROCESS ON CIVIL AIRBORNE SYSTEMS AND EQUIPMENT*, 1996-12. DOI: 10.4271/ARP4761.

[83] P. W. Koh, S. Sagawa, H. Marklund, *et al.*, "WILDS: A Benchmark of in-the-Wild Distribution Shifts," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021-07, pp. 5637–5664. [Online]. Available: https://proceedings.mlr.press/v139/koh21a.html.

[84] AJV-P Policy-Mission Support Services, "Order JO 7400.2P Procedures for Handling Airspace Matters," Federal Aviation Administration, Tech. Rep., 2023-03-17.

[85] "Free route airspace," EUROCONTROL. (), [Online]. Available: https://www.eurocontrol.int/concept/free-route-airspace (visited on 2024-12-10).

[86] "General Properties and Characteristics of Sensors," Monolithic Power Systems. (), [Online]. Available: https://www.monolithicpower.com/en/learning/mpscholar/sensors/intro-to-sensors/general-properties-characteristics?srsltid=AfmBOooPonoNMaH1XZkUdaUvY50wL1EeRCkqxWEdjWBG8xttduX4OQ1V (visited on 2024-12-10).

[87] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, 2016, pp. 1–6. DOI: 10.1109/QoMEX.2016.7498955.

[88] L. Jiang and X. Wang, "Dataset Constrution through Ontology-Based Data Requirements Analysis," *Applied Sciences*, vol. 14, no. 6, 2024, ISSN: 2076-3417. DOI: 10.3390/app14062237.

[89] N. W. Paton, J. Chen, and Z. Wu, "Dataset Discovery and Exploration: A Survey," *ACM Computing Surveys*, vol. 56, no. 4, 2023-11-09, ISSN: 0360-0300. DOI: 10.1145/3626521.

[90] F. L. Gewers, G. R. Ferreira, H. F. D. Arruda, *et al.*, "Principal Component Analysis: A Natural Approach to Data Exploration," *ACM Computing Surveys*, vol. 54, no. 4, 2021-05-24, ISSN: 0360-0300. DOI: 10.1145/3447755.

[91] J. C. da Cunha Davison, P. I. Tostes, and C. A. Guerra Carneiro, "Framework Architecture for AI/ML Data Management for Safety-Critical Applications," in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, 2024, pp. 1–9. DOI: 10.1109/DASC62030.2024.10748782.

[92] EASA, "Weather Information to Pilots Strategy Paper, An Outcome of the All Weather Operations Project," European Union Aviation Safety Agency (EASA), research rep., 2018-01-19, pp. 1–45. [Online]. Available: https://www.easa.europa.eu/en/downloads/45593/en.

[93] R. Rathnakumar and Y. Liu, "Towards safer general aviation operations using a vision-based decision support system for weather threat avoidance," *Journal of Air Transport Management*, vol. 123, p. 102 709, 2025, ISSN: 0969-6997. DOI: https://doi.org/10.1016/j.jairtraman.2024.102709.

[94] D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (mlops): Overview, definition, and architecture," *IEEE Access*, vol. 11, pp. 31 866–31 879, 2023. DOI: 10.1109/ACCESS.2023.3262138.

[95] Stanford Intelligent Systems Laboratory. "VerticalCAS Repository." (2020-09-01), [Online]. Available: https://github.com/sisl/VerticalCAS (visited on 2024-09-12).

[96] Stanford Intelligent Systems Laboratory. "HorizontalCAS Repository." (2020-07-30), [Online]. Available: https://github.com/sisl/HorizontalCAS (visited on 2024-09-12).

[97] J. M. Christensen, A. Anilkumar Girija, T. Stefani, *et al.*, *Advancing the ai-based realization of acas x towards real-world application*, version 1.0.0, 2024-09. DOI: 10.5281/zenodo.13815668.

[98] K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen, and M. J. Kochenderfer, "Policy compression for aircraft collision avoidance systems," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–10. DOI: 10.1109/DASC.2016.7778091.

[99] K. D. Julian, M. J. Kochenderfer, and M. P. Owen, "Deep Neural Network Compression for Aircraft Collision Avoidance Systems," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 3, pp. 598–608, 2019-03, ISSN: 1533-3884. DOI: 10/gr4v7d.

[100] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Commun. ACM*, vol. 64, no. 3, pp. 107–115, 2021-02, ISSN: 0001-0782. DOI: 10.1145/3446776.

[101] A. Christmann, A. Kostrzewa, R. Ernst, *et al.*, "Integrating Multi-/Many-Cores in Avionics: Open Issues and Future Concepts," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021, pp. 1–8. DOI: 10.1109/DASC52595.2021.9594458.

[102] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[103] M. Damour, F. De Grancey, C. Gabreau, *et al.*, "Towards Certification of a Reduced Footprint ACAS-Xu System: A Hybrid ML-Based Solution," in *Computer Safety, Reliability, and Security*. Springer International Publishing, 2021, pp. 34–48, ISBN: 9783030839031. DOI: 10/gtd4pf.

[104] EUROCAE, "ED-271, Minimum Aviation System Performance Standard for Detect and Avoid (Traffic) in Class A-C airspaces - with Corrigendum 1," EUROCAE, Saint-Denis, France, Tech. Rep., 2022-05.

[105] EUROCAE, "ED-313, OSED for Detect and Avoid (Traffic) in Class A to G Airspaces under IFR," EUROCAE, Saint-Denis, France, Tech. Rep., 2023-08.

[106] "Temporary Flight Restrictions (TFRs)," Federal Aviation Administration. (), [Online]. Available: https://www.faa.gov/uas/getting_started/temporary_flight_restrictions (visited on 2024-12-20).

[107] "Regulations in German Airspace," Deutscher Gleitschirmverband und Drachenflugverband e.V. (), [Online]. Available: https://www.dhv.de/en/sites-nature/airspace-germany/ (visited on 2024-12-20).

[108] "Classification of Airspace," SKYbrary Aviation Safety. (), [Online]. Available: https://skybrary.aero/articles/classification-airspace (visited on 2025-01-01).

[109] "GPWS," SKYbrary Aviation Safety. (), [Online]. Available: https://skybrary.aero/gpws (visited on 2024-12-28).

[110] C. A. A. of New Zealand, *ADS-B in New Zealand*, 2020-01. [Online]. Available: https://www.nss.govt.nz/assets/nss/resources/2018-07-31-ADSB-FAQ-Document-V0.3.docx.pdf (visited on 2025-01-10).

[111] "GPWS," United States Space Force. (2020-10), [Online]. Available: https://www.spaceforce.mil/About-Us/Fact-Sheets/Article/2197765/global-positioning-system/ (visited on 2024-12-29).

[112] "GPS Accuracy," National Oceanic and Atmospheric Administration. (2022-03-22), [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/ (visited on 2024-12-29).

[113] J. P. Chryssanthacopoulos and M. J. Kochenderfer, "Collision avoidance system optimization with probabilistic pilot response models," in *Proceedings of the 2011 American Control Conference*, 2011, pp. 2765–2770. DOI: 10.1109/ACC.2011.5990776.

[114] G. Saake and K.-U. Sattler, *Algorithmen und Datenstrukturen: Eine Einführung mit Java*. Dpunkt. Verlag, 2020-10.

[115] L. Rierson, *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance*. CRC Press, 2017-12, ISBN: 9781439813690. DOI: https://doi.org/10.1201/9781315218168.

[116] F. de Grancey, S. Gerchinovitz, L. Alecu, *et al.*, "On the Feasibility of EASA Learning Assurance Objectives for Machine Learning Components," in *ERTS 2024*, ser. ERTS2024, Best paper award at ERTS 2024., Toulouse, France, 2024-06. [Online]. Available: https://hal.science/hal-04575318.

Appendices

A Concept of Operations

A.1 Operational Scenario of a Single Intruder Encounter

The scenario takes place in an onwship flight from MUC to BER. After takeoff and climb to cruising altitude, the airplane is configured by the pilot(s) into en-route operation under IFR in airspace type C. The airspeed of the aircraft is in the typical cruising speed range and is surrounded by non-conflicting traffic. The scenario starts when a traffic airplane changes trajectory setting it on to a collision course. The surveillance of the ownship detects the trajectory change of the threat intruder, based on the ADS-B broadcast messages. Based on this information, the PYCASX *system calculates the expected point where the two airplanes potentially collide. Based on this estimation, and the information about the ownship and intruder, the system generates an advisory. For the intruder, the system determines that an action in the vertical plane, e.g., CL 1500, is necessary. The action is announced to the pilot(s) and they determine that the action can be safely executed. After the execution of the action for 10 s, the system determines that the sufficient separation between the intruder and the ownship is maintained, and therefore announcing to the pilots a clear of conflict. The pilot(s) level of the airplane, and inform the responsible air traffic controller. The scenario ends, as no further conflicts arise for the rest of the flight.*

A.2 Degraded Operational Scenario of a Multi-Intruder Encounter

The scenario takes place in an onwship flight from MUC to BER. After takeoff and climb to cruising altitude, the airplane is configured by the pilots into en-route operation under IFR in airspace type C. The airspeed of the aircraft is in the typical cruising speed range and is surrounded by non-conflicting traffic. The scenario starts when two traffic airplanes change trajectory setting them on to a collision course. The surveillance of the ownship detects both trajectory changes of the threat intruders, based on the ADS-B broadcast messages. Based on this information, the PYCASX *system calculates the expected point where the two airplanes potentially collide. Based on this estimation, and the information about the ownship and intruders, the system generates two advisories. For the first intruder, the system determines that an action in the vertical plane, e.g., DES 1500, is necessary, and for the second intruder, the system determines that an action in the vertical plane, e.g., CL 1500, is necessary. As both actions are in opposite directions, the system is unable to determine a final action. Therefore, the system announces this issue to the pilot(s). The system still displays all available information for both intruders, but the action selection responsibility rests with the pilot(s). Based on the displayed information, the pilots chose an adequate action to prevent the collision. Afterward, the pilots inform the responsible air traffic controller. The scenario ends, as no further conflicts arise for the rest of the flight.*

B Vertical CAS

B.1 Additional VCAS Tables

Table B.1: For each environment attribute of the OD for PYCASX, it was classified if the attribute is a physical range or a behavior-determining attribute. Based on this and the allocated function to VCAS, for each attribute it is identified if it is relevant for the VCAS component, i.e., should be included in the ODD.

Top-level attribute	Sub-attribute	Attribute	Classification	Relevant
Environment	Weather	Adverse Conditions	Behavior	No
Environment	Connectivity	Satellite Positioning	Physical Range	No
Environment	Connectivity	Communication Type	Physical Range	No
Environment	Connectivity	Communication Range	Physical Range	No

Table B.2: For each dynamic elements attribute of the OD for PYCASX, it was classified if the attribute is a physical range or a behavior-determining attribute. Based on this and the allocated function to VCAS, for each attribute it is identified if it is relevant for the VCAS component, i.e., should be included in the ODD.

Top-level attribute	Sub-attribute	Attribute	Classification	Relevant
Dynamic Elements	Intruder	Agent Type	Behavior	Yes
Dynamic Elements	Intruder	Maximum Agent Density	Physical Range	No
Dynamic Elements	Intruder	Latitude	Physical Range	No
Dynamic Elements	Intruder	Longitude	Physical Range	No
Dynamic Elements	Intruder	Altitude	Physical Range	Yes
Dynamic Elements	Intruder	Horizontal Airspeed	Physical Range	No
Dynamic Elements	Intruder	Horizontal Acceleration	Physical Range	No
Dynamic Elements	Intruder	Vertical Rate	Physical Range	Yes
Dynamic Elements	Intruder	Vertical Rate Acceleration	Physical Range	Yes
Dynamic Elements	Intruder	Heading	Physical Range	No
Dynamic Elements	Intruder	Communication Type	Physical Range	No
Dynamic Elements	Ownship	Agent Type	Behavior	Yes
Dynamic Elements	Ownship	Latitude	Physical Range	No
Dynamic Elements	Ownship	Longitude	Physical Range	No
Dynamic Elements	Ownship	Altitude	Physical Range	Yes

Table B.3: Continuation of Table B.2.

Top-level attribute	Sub-attribute	Attribute	Classification	Relevant
Dynamic Elements	Ownship	Horizontal Airspeed	Physical Range	No
Dynamic Elements	Ownship	Horizontal Acceleration	Physical Range	No
Dynamic Elements	Ownship	Vertical Rate	Physical Range	Yes
Dynamic Elements	Ownship	Vertical Rate Acceleration	Physical Range	Yes
Dynamic Elements	Ownship	Heading	Physical Range	No
Dynamic Elements	Ownship	Vertical Rate Capability	Physical Range	Yes
Dynamic Elements	Ownship	Vertical Rate Acceleration Capability	Physical Range	Yes
Dynamic Elements	Ownship	Turn Rate Capability	Physical Range	No
Dynamic Elements	Ownship	Turn Rate Acceleration Capability	Physical Range	No
Dynamic Elements	Ownship	Pilot Type	Behavior	No

C Horizontal CAS

C.1 Additional HCAS Tables

Table C.1: For each scenery attribute of the OD for PYCASX, it was classified if the attribute is a physical range or a behavior-determining attribute. Based on this and the allocated function to HCAS, for each attribute it is identified if it is relevant for the HCAS component, i.e., should be included in the ODD.

Top-level attribute	Sub-attribute	Attribute	Classification	Relevant
Scenery	Airspace	Type	Behavior	Yes
Scenery	Airspace	Altitude	Physical Range	No
Scenery	Airspace	Longitude	Physical Range	No
Scenery	Airspace	Latitude	Physical Range	No
Scenery	Airspace	Flight Rule	Behavior	Yes
Scenery	Airspace	Route Type	Behavior	Yes
Scenery	Geography		-	No
Scenery	Structures		-	No

Table C.2: For each environment attribute of the OD for PYCASX, it was classified if the attribute is a physical range or a behavior-determining attribute. Based on this and the allocated function to HCAS, for each attribute it is identified if it is relevant for the HCAS component, i.e., should be included in the ODD.

Top-level attribute	Sub-attribute	Attribute	Classification	Relevant
Environment	Weather	Adverse Conditions	Behavior	No
Environment	Connectivity	Satellite Positioning	Physical Range	No
Environment	Connectivity	Communication Type	Physical Range	No
Environment	Connectivity	Communication Range	Physical Range	No

Table C.3: For each dynamic elements attribute of the OD for PYCASX, it was classified if the attribute is a physical range or a behavior-determining attribute. Based on this and the allocated function to HCAS, for each attribute it is identified if it is relevant for the HCAS component, i.e., should be included in the ODD.

Top-level attribute	Sub-attribute	Attribute	Classification	Relevant
Dynamic Elements	Intruder	Agent Type	Behavior	Yes
Dynamic Elements	Intruder	Maximum Agent Density	Physical Range	No
Dynamic Elements	Intruder	Latitude	Physical Range	Yes
Dynamic Elements	Intruder	Longitude	Physical Range	Yes
Dynamic Elements	Intruder	Altitude	Physical Range	No
Dynamic Elements	Intruder	Horizontal Airspeed	Physical Range	Yes
Dynamic Elements	Intruder	Horizontal Acceleration	Physical Range	Yes
Dynamic Elements	Intruder	Vertical Rate	Physical Range	No
Dynamic Elements	Intruder	Vertical Rate Acceleration	Physical Range	No
Dynamic Elements	Intruder	Heading	Physical Range	Yes
Dynamic Elements	Intruder	Communication Type	Physical Range	No
Dynamic Elements	Ownship	Agent Type	Behavior	Yes
Dynamic Elements	Ownship	Latitude	Physical Range	Yes
Dynamic Elements	Ownship	Longitude	Physical Range	Yes
Dynamic Elements	Ownship	Altitude	Physical Range	No

Table C.4: Continuation of Table C.3.

Top-level attribute	Sub-attribute	Attribute	Classification	Relevant
Dynamic Elements	Ownship	Horizontal Airspeed	Physical Range	Yes
Dynamic Elements	Ownship	Horizontal Acceleration	Physical Range	Yes
Dynamic Elements	Ownship	Vertical Rate	Physical Range	No
Dynamic Elements	Ownship	Vertical Rate Acceleration	Physical Range	No
Dynamic Elements	Ownship	Heading	Physical Range	Yes
Dynamic Elements	Ownship	Vertical Rate Capability	Physical Range	No
Dynamic Elements	Ownship	Vertical Rate Acceleration Capability	Physical Range	No
Dynamic Elements	Ownship	Turn Rate Capability	Physical Range	Yes
Dynamic Elements	Ownship	Turn Rate Acceleration Capability	Physical Range	Yes
Dynamic Elements	Ownship	Pilot Type	Behavior	No

D Software

D.1 Software Versions

Table D.1: List of used software versions for the different systems, which were used in the thesis.

	Available	Version
PYCASX	[28]	V1.0.0
Vertical CAS	[95]	bc56a28cfe70b0486b6875deacfda947500bddcb (Commit-SHA)
Horizontal CAS	[96]	7edae51b7cf348e8341b4a4d01897d7513c8e64c (Commit-SHA)

D.2 Parameter Configurations

Table D.2: MDP parameter configuration for the response of the ownship with the expected turn rate.

Advisory	Minimum Response	Average Response	Maximum Response
WL	1.25	1.5	2.0
WR	−1.25	−1.5	−2.0
SL	2.0	3.0	4.0
SR	2.0	−3.0	−4.0

Table D.3: MDP parameter configuration for the response of the ownship with half the expected turn rate.

Advisory	Minimum Response	Average Response	Maximum Response
WL	0.625	0.75	1.0
WR	−0.625	−0.75	−1.0
SL	1.0	1.5	2.0
SR	1.0	−1.5	−2.0