EAM Diagrams - A Framework to Systematically Describe AI Systems for Effective AI Risk Assessment

Ronald Schnitzer
Techincal University of Munich, Germany
Siemens AG, Munich, Germany

Andreas Hapfelmeier
Siemens AG, Munich, Germany

Sonja Zillner
Techincal University of Munich, Germany
Siemens AG, Munich, Germany

Abstract

Artificial Intelligence (AI) is a transformative technology that offers new opportunities across various applications. However, the capabilities of AI systems introduce new risks, which require the adaptation of established risk assessment procedures. A prerequisite for any effective risk assessment is a systematic description of the system under consideration, including its inner workings and application environment. Existing system description methodologies are only partially applicable to complex AI systems, as they either address only parts of the AI system, such as datasets or models, or do not consider AI-specific characteristics at all. In this paper, we present a novel framework called EAM Diagrams for the systematic description of AI systems, gathering all relevant information along the AI life cycle required to support a comprehensive risk assessment. The framework introduces diagrams on three levels, covering the AI system's environment, functional inner workings, and the learning process of integrated Machine Learning (ML) models.

2012 ACM Subject Classification Software and its engineering → System description languages

Keywords and phrases AI system description, AI risk assessment, AI auditability

Digital Object Identifier 10.4230/OASIcs.SAIA.2024.3

Category Academic Track

1 Introduction

The integration of Machine Learning (ML) models into complex systems is essential for a wide range of applications, including autonomous vehicles, medical decision support systems, and many more. However, the integration of such ML technologies also introduces new risks which have to be considered, especially in critical applications.

Consequently, novel methods to audit and assess the risks of such systems are crucial. Existing approaches typically analyze ML-based risks by focusing on the models themselves. However, a comprehensive risk assessment should consider the entire system, because although many risks originate from the ML model, their impact and potential mitigation strategies need to be evaluated within the context of the whole system.

To support a meaningful risk assessment, the assessor consequently needs to have an overall understanding of the system under evaluation. For that, a comprehensive description of the entire system, including its application context, functional architecture, and development process, is needed. This AI system description should be easily understandable while providing all relevant information to identify potential risk sources and enable a systematic estimation and evaluation of these risks.

In this paper, we investigate the requirements for an AI system description to facilitate effective risk assessment. The goal of the paper is to answer the following research question:

- What are the essential components and characteristics of an AI system description that support a comprehensive and effective AI risk assessment?

Based on this analysis, we present the contribution of this paper: A framework that defines the structure and essential elements of a comprehensive description of AI systems.

The paper is structured as follows: In Section 2, we define the scope of the paper, provide theoretical background, and position our paper in the research landscape of related work. In Section 3, we present the results of a requirements analysis for systematically describing AI systems to support effective risk assessment. Subsequently, we introduce a framework to guide practitioners in developing suitable AI system descriptions. In Section 5, we discuss how the in Section 3 identified requirements are met by our framework, including a comparative analysis of related work, and outline the limitations of our approach. Finally, the paper is concluded in Section 6.

2 Theoretical Background and Related Work

This paper studies the requirements for and presents a derived framework to develop an AI system description that supports efficient risk assessment. To enable this investigation, it is crucial to define the following terms: AI system and AI risk assessment. This section provides definitions and important background information on these terms. Furthermore, it highlights significant work related to ours and positions our contribution within the existing research landscape by emphasizing our differentiation from previous work.

2.1 AI Systems

First, a clear definition of the object of consideration is needed. Artificial Intelligence and its subfield, Machine Learning, lack universally accepted definitions in the literature. Recently, the European Union published the EU AI Act [4], setting the rules for developing and operating AI systems within the European Union and potentially having worldwide effects [18]. Due to the significance of the EU AI Act to the AI landscape worldwide, we adhere to the definition provided in the regulation: AI systems are defined as "machine-based systems that are designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment, and that, for explicit or implicit objectives, infer from the input they receive how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments" [4].

Based on this definition, we make two important remarks: First, AI systems are software systems that might be integrated into larger systems potentially consisting of other components, including hardware and sensors. In the following, we refer to anything outside the boundaries of the AI system as its environment. Second, AI systems themselves can be composed of several ML models that are connected in a way to solve a particular objective.

For instance, the perception module of an autonomous vehicle might be an AI system that contains several AI models, such as a computer vision-based pedestrian detector, and a detector to detect danger zones in the environment. Additionally, this AI system might include non-AI-based components, such as the deterministic fusion of several sensors, which relies on conventional software and does not fall under the scope of AI.

The framework for describing the architecture and context of AI systems proposed in Section 4 is designed for AI systems built upon Machine Learning techniques, representing the majority of today's deployed AI systems. The framework might be generalized to other technologies in the scope of AI, but since this is not the focus of this work we postpone such an analysis to future work.

2.2 Risk Assessment

To understand the term risk assessment, it is essential to consider the broader concept of risk management, independent of the technology. ISO 31000 defines *risk management* as "coordinated activities to direct and control an organization with regard to risk" [7]. Within this framework, risk assessment is a part of risk management that involves "a structured process that identifies how objectives may be affected, and analyses the risk in term of consequences and their probabilities before deciding on whether further treatment is required." [6].

IEC 31010, specifically, provides a variety of methods for practical risk assessment applications [6]. The choice of an appropriate method depends on several factors, including the type, domain, complexity, and available resources for developing and operating the system under assessment. For instance, *failure mode and effect analysis (FMEA)* is a method that systematically analyzes potential failures of a system and is considered best practice in safety-relevant applications. Notably, FMEA is also part of the end-to-end framework for algorithmic auditing introduced in [14].

Generally, all methods mentioned in [6] rely on some form of system description as input for risk assessment. What these methods have in common is the requirement for a systematic description of the object under assessment.

2.3 Related Work

The significant importance of a structured AI system description is highlighted in the literature on AI assessments and auditing [3, 13, 21]. For example, the AI assessment catalog [13] provides a high-level description of the AI system under assessment, although it primarily focuses on analyzing relevant dimensions of trustworthiness in AI systems rather than on the AI system description itself.

Several established methods systematically describe and document parts of AI systems, such as datasheets for datasets [5] and model cards [11], which document relevant aspects of the data and the model itself. While these works focus on specific elements of AI systems, they do not address the holistic representation of AI systems. However, some works are moving in this direction. For instance, [16] introduce AI Fact Sheets, which address a whole system/service perspective but do not focus on the particular requirements from a risk assessment perspective.

Another important source of related work involves system descriptions not specifically for AI systems but for complex software systems in general. Many of these approaches build upon standardized notations for describing complex software systems, such as UML (Unified Modeling Language) [1] and its evolution, SysML [12]. The integrated system nature is particularly well reflected by the C4 model introduced by Simon Brown [2], which also builds upon the UML notation.

The C4 model introduces diagrams showing different levels of detail, enabling the assessor to understand the broader picture while also allowing detailed inspection where needed. First, the system context diagram depicts the high-level perspective on the system and the environment it interacts with. Then, the container diagram shows the main functional blocks of the system and their interconnections. Third, the component diagram provides abstractions of components within individual containers that should map to real abstractions in the codebase. Finally, the code diagram illustrates relations at the code level, such as relations between implemented methods and functions.

Since the C4 model is designed for software systems in general, it does not account for the specifics introduced by AI systems. The framework we present in this paper is inspired by the structure of the C4 model but is adapted to appropriately reflect AI-specific properties in the context of AI systems.

3 Methodology

By assessing the risks of AI use cases in the form of unstructured interviews with data scientists, we identified five requirements a structured AI system description needs to fulfill to effectively support the assessor in conducting a risk assessment. In the following, we describe these five requirements summarizing our insights from the interviews.

3.1 R1: Flexibility Across Diverse AI Systems

AI systems are utilized in a broad spectrum of applications, each with its unique characteristics and requirements. Therefore, a framework to describe AI systems must be sufficiently flexible to reflect this diversity. A one-size-fits-all template for describing AI systems is impractical, as noted by [16], highlighting the challenge of creating a universally applicable template due to the varying needs of different stakeholders.

However, with focusing on risk assessments and the specific needs of the persons preparing and executing them, it is essential that the AI system description can be tailored to the specific context and requirements of the assessment process. A flexible yet standardized format ensures that the AI system description is applicable to various AI systems, enabling assessors to adapt it to the specific needs of the system being evaluated.

Moreover, this flexibility must not compromise the consistency and comparability of the AI system descriptions. By allowing tailored AI system descriptions within a standardized framework, assessors can achieve a balance between adaptability and the uniformity needed for efficient and effective risk assessment.

3.2 R2: Ensuring Reproducibility

A core feature of an AI system description is reproducibility. An AI system description created by several persons or by the same person several times should be approximately identical to guarantee reproducibility and to enable reproducibility on a risk assessment level. Providing strict and clear guidelines on how to create the specific AI System description is crucial. Reproducibility in AI System descriptions also enhances the efficiency of risk assessors. Variations in AI System descriptions increase the time assessors need to understand the system. Standardized AI System descriptions allow them to quickly grasp the system's details, especially when multiple systems are assessed by the same individual, thereby reducing the time needed for comprehension.

3.3 R3: Integration with Risk Assessment

The goals of a risk assessment process are to identify risk sources, estimate their impact and likelihood, evaluate their significance, and propose mitigation measures. The AI system description must enable the allocation of risk sources to relevant parts of the system.

For instance, 24 AI-specific risk sources that might cause harm have been identified in [17]. Furthermore, it was found that some risk sources manifest at the system level, while others manifest in particular system components such as datasets or the AI models themselves.

Additionally, information about the environment is crucial to estimate and evaluate the severity and probability of certain risks. The AI system description should recognize these factors. For instance, the presence of platform edge doors at railway stations drastically decreases the risk of people being harmed by trains, and are therefore important context information when performing a risk assessment for autonomous trains.

Moreover, the part of a system where the risk source manifests and where a mitigation measure is applied might differ. To enable systematic risk management, the AI system description must cover all these factors comprehensively.

3.4 R4: Grounded in Best Practices

While AI technology introduces many new aspects and risks to consider in risk assessments, the approach to assess these risks should adhere as closely as possible to established procedures. Assessors are already familiar with state-of-the-art methodologies, and unnecessary changes would require additional resources to train new assessors.

Therefore, concepts for describing AI systems should not contradict established procedures. Wherever possible, standardized methods that have proven to be effective, such as using UML or SysML, should be incorporated into the new solution. Additionally, best practices from AI documentation, such as datasheets for datasets [5] or model cards [11], should be recognized.

3.5 R5: Reflecting AI-Specific Characteristics

When describing an AI system, it is important to address the aspects ML solutions introduce in comparison to conventional software. It has been intensively discussed in the literature that the use of ML techniques introduces a new set of risks [17, 19, 23]. These ML-specific characteristics include the opaqueness, unpredictability, and complexity of such systems. It is crucial that the framework for describing AI systems reflects AI-specific characteristics to effectively enable AI risk assessment. Another main difference between ML-based solutions and traditional software is how ML models are developed [20]. Especially, the dependence on data sources for the development of ML models is a unique characteristic that must be included in the AI system description. These AI-specific aspects are essential for a comprehensive and accurate risk assessment.

4 EAM Diagrams

Based on the conducted interviews and the requirement analysis, we developed EAM Diagrams, our solution to create an AI system description that is flexible yet comprehensive enough to support efficient risk assessment.

Our concept follows a holistic approach, capturing the interplay between the ML model(s), the AI system and its environment by defining three levels of detail, each represented by its own diagram:

1. The Environment Diagram sets the AI system into context and describes relevant external factors impacting the risk assessment.
2. The Architecture Diagram illustrates the functional inner workings of the AI system, enabling the allocation of different risk sources to the relevant components and representing the operational state of the AI system.
3. The Modeling Diagram relates to the development setup of the ML models integrated into the AI system and covers all aspects of the AI life cycle stages prior to deployment.

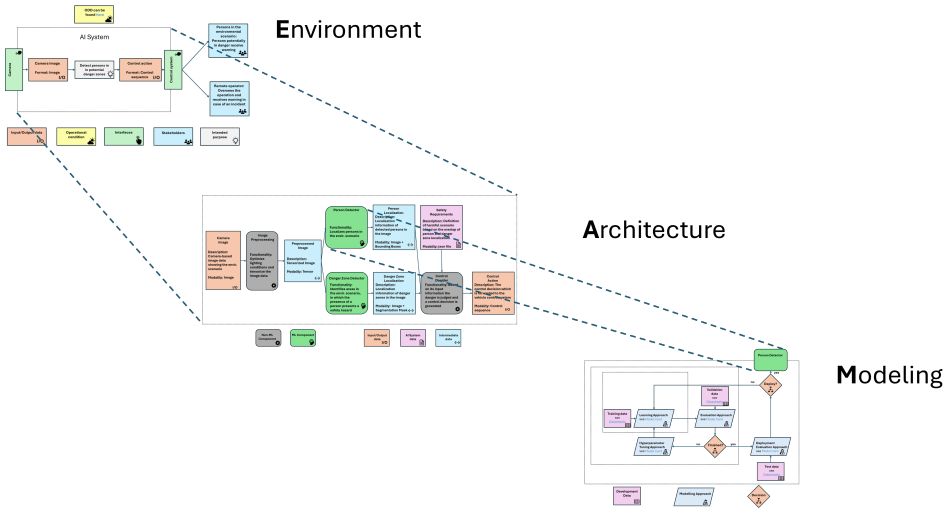

Figure 1 Overview of the relation between the different levels of EAM diagram. The three individual diagrams are shown in more detail in Figure 2, Figure 3, and Figure 4.

In the following sections, we describe these three levels in detail as shown in Figure 1, explaining all elements comprising the diagrams. Furthermore, to illustrate the application of EAM Diagrams each level of detail is motivated with a use case as a running example. Figure 2, Figure 3, and Figure 4 show the respective diagrams.

4.1 Running Example

The use case considers a driverless vehicle that operates in an open world and perceives its surroundings via camera input. The person detection system within this vehicle is the AI system under assessment. It aims to detect persons in danger zones, e.g., an area in front of the vehicle and sends respective control sequences to the vehicle's control system to initialize countermeasures in case of danger.

4.2 Environment Diagram

As with any risk assessment process, the first step involves setting the context of the assessment and therefore, the system under consideration [7]. Setting the context for an AI system involves concretely specifying the operating conditions and the system's interfaces to its environment. These aspects form the elements of the Environment Diagram, as depicted in Figure 2 for the running example of a driverless vehicle. In the following, we describe each element, why it is relevant for the risk assessment, and how the right information is included in the diagram.

4.2.1 Intended Purpose

Specifying the intended purpose of an AI system is essential as it not only helps assessors understand the scope of the system but also provides an initial insight into potential risks posed by the system.

The description of the intended purpose should include a textual explanation of the intended use of the AI system by its developer. It is particularly important to describe the intended purpose as concretely as possible, especially if the AI system is based on foundation models, which can be utilized for various downstream tasks. These downstream tasks significantly influence the risks posed by the system.

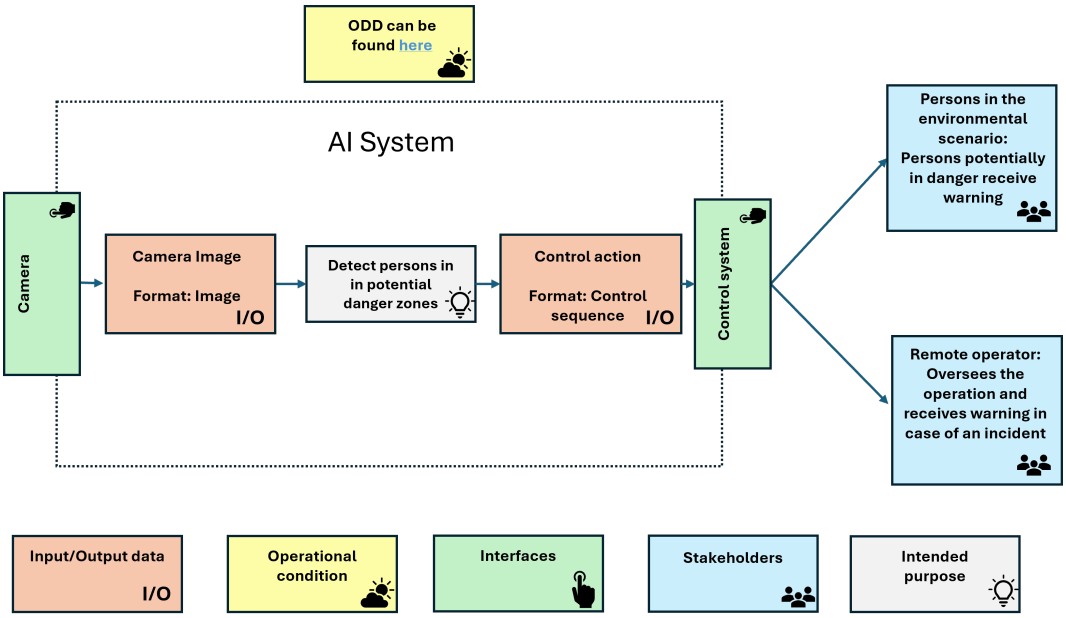

Figure 2 A schematic Environment Diagram of the driverless vehicle use case.

In case the intended purpose description is too complex for its integration into the diagram, a reference to the document containing the information can be added to ensure accessibility for the assessor.

4.2.2 Operational Conditions

Specifying the operational conditions provides valuable information about the limitations of the system. The level of detail required in documenting these conditions varies depending on the application context. For instance, in a Q&A system, the operational conditions might only state that the system receives user questions of any type, potentially limited to a certain number of input characters. On the other hand, in high-stakes applications, where AI systems operate in complex environments, a more detailed description is necessary. For instance, in the automotive and railway sectors, autonomous vehicles function as high-risk applications in complicated environments. In these sectors, the operational domain is modeled using a so-called operational design domain (ODD), which systematically describes the operating conditions of autonomous systems [8, 10]. This comprehensive description of operational environments is also reflected in the ODD being a crucial input for safety argumentation [22].

In case the description of the operational conditions is too complex to be included in the diagram (e.g., ODDs are typically large documents on their own), the diagram should contain references to the right documents.

4.2.3 Interfaces

AI systems are often part of larger systems. For example, the perception module (an AI system) may be embedded in an autonomous train (the whole system). Issues in the interfaces between the system and its environment might present sources of risk. In such cases, it is crucial to understand the interfaces between the AI system and its environment. If, for example, the AI system receives input from sensors, information about the type of sensor can

be valuable for risk assessment. Another example is a chatbot based on generative AI models, where the interface between the AI system and the human user is typically a graphical user interface (GUI). By making appropriate design choices at the GUI level, such as informing the user that they are interacting with an AI system, risks can be mitigated. Each interface in which the AI system receives input from its environment should be mentioned in the diagram.

4.2.4 Input and Output Data

Understanding the input and output data, as well as their modality, is essential for identifying or excluding certain risks. Specific risk sources may only exist for particular modalities.

For instance, the risk of an AI system hallucinating is associated with AI systems that generate content, such as text, audio, image, or video data [15]. However, this risk is not relevant for an AI system performing a binary classification task. Therefore, understanding the input and output data during operation is crucial information for risk assessment.

Consequently, the modality of the data, as well as a short description, should be denoted in the diagram.

4.2.5 Stakeholder groups

Stakeholders refer to all human persons involved in the operation of the AI system who can be the cause or affected by risks. One potential stakeholder group is the user and its profile, since these are important pieces of information for risk assessment. For instance, if the user group is restricted to specifically trained personnel, the potential for misuse (affecting the risk level of the system) is treated differently compared to a situation where the system is accessible to the general public.

From a risk assessment perspective, understanding the user group helps in identifying reasonably foreseeable misuse of the AI system. This is a crucial part of any risk analysis following ISO/IEC Guide 51 [9] and is required for high-risk systems according to the EU AI Act, Article 9 [4].

For many systems, the operation is overseen by a human person. For systems to be classified as high-risk by the EU AI Act [4], this is even a mandatory requirement in accordance to Article 14. The presence and level of human oversight provide an indication of the AI system's level of autonomy. Human oversight significantly impacts the risk level and related aspects such as liability and accountability.

Human oversight consists of two key aspects. First, the individuals overseeing the AI system must be aware of the AI system's capabilities and limitations, requiring specific competencies. Second, suitable human-machine interfaces are necessary to enable effective human oversight.

Another important stakeholder group consists of the affected persons. Understanding which groups of persons and how many are affected by the output of the AI system is a crucial factor in evaluating its risk. Analyzing the impact a system failure could have on affected persons is an important part of established risk analysis procedures.

Besides the above-mentioned stakeholder groups, there might be more depending on the specific use case. Of course, if so, they can be added to the diagram as well.

For each stakeholder group, a short description of their role and characteristics relevant to the risk assessment should be provided in the diagram.

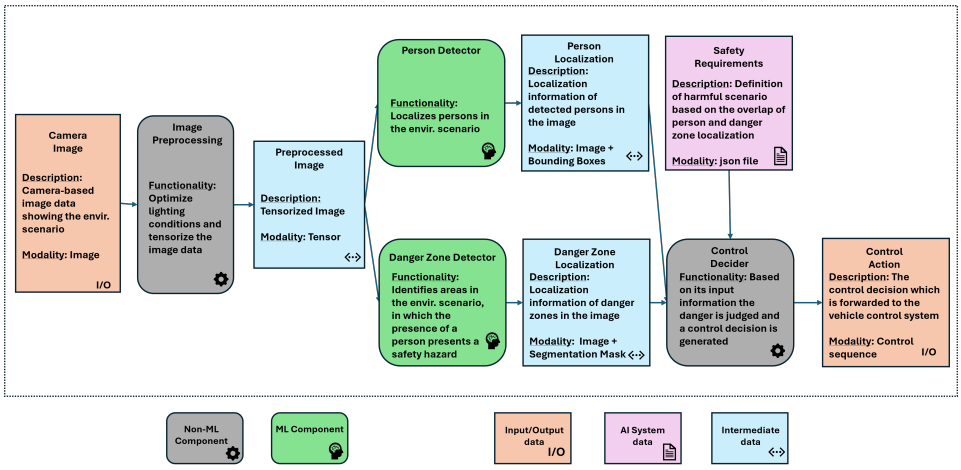

Figure 3 A schematic Architecture Diagram of the driverless vehicle use case.

4.3 Architecture Diagram

The purpose of the Architecture Diagram is to describe the inner workings of the AI system. This data flow diagram illustrates the flow of data from input to output, covering all data transformations that occur during the operation of the AI system. It is important to note that the Architecture Diagram does not include the training of the ML models; it only represents the system in its operational state. We introduce two different groups of elements: data elements representing the data itself and component elements representing components that perform data transformations of any type. Together with the data flow (represented as arrows), they form the Architecture Diagram. An application of the Architecture Diagram to the driverless vehicle use case is presented in Figure 3.

4.3.1 Component Elements

Within the component elements of the Architecture Diagram, we distinguish between two types of components. Every component in the data flow diagram should include at least a description of the component's functionality.

ML Model Components. ML Model Components refer to any ML model created using Machine Learning techniques that, during operation, receives input data to infer output data. This output data is either an output of the AI system or used for further processing within the AI system by other components.

Non-ML Components. In contrast to ML Model Components, non-ML components refer to all components developed without the use of Machine Learning techniques. This includes deterministic pre-processing steps of data, data storage units, and API interfaces. The definition of non-ML components is broad to accommodate a wide range of applications. Necessary information about these components should be detailed in their descriptions.

4.3.2 Data Elements

There are three different types of data elements that are considered. Each data element includes at least a textual description of the underlying data as well as its modalities.

Intermediate Data. Intermediate data refers to any data that has been processed by a component within the AI system's data flow but is not the final output. Intermediate data is the output of one component and the input of another. To avoid inconsistencies, intermediate data is defined as individual data elements.

Input/Output Data. The input and output elements are the same as in the Environment Diagram. However, since they mark the start and end of the operational data flow, they should also be represented in the operational data flow diagram.

AI System Data. AI System data refers to any type of static data that is used by the AI system during inference and is not introduced by the environment. Examples of AI system data include documents that are retrieved by a Retrieval Augmented Generation based Q&A system or (parameter) specifications.

4.3.3 Rules for Constructing the Architecture Diagram

The elements described above form the foundation for the operational data flow diagram. To ensure reproducible results and avoid ambiguity, specific rules must be applied to describe how the different elements are connected. The rules can be summarized as follows:

1. Every component element must receive at least one data element as input and produce at least one data element as output.
2. Every intermediate data element must serve as the input to a component element and be the output to another component element.
3. Input data and AI system data elements can not be the output of a component element
4. Output data elements can not be the input to component elements
5. Data elements are only connected with component elements and vice versa.

The Architecture Diagram can also be mathematically represented by a directed graph $G = (V, E)$, where V is a set of nodes and $E \subset V \times V$ is a set of edges, fulfilling the following conditions.

1. $V = V_c \dot{\cup} V_d$, where V_c, V_d represent the set of component and data elements, respectively. $\dot{\cup}$ denotes the disjoint union.
2. $V_d = V_{in} \dot{\cup} V_i \dot{\cup} V_s \dot{\cup} V_{out}$, where $V_{in}, V_i, V_s, V_{out}$ represent input, indermediate, system, and output data, respectively.
3. $e = (v_1, v_2) \in E \Rightarrow e \in (V_d \times V_c) \cup (V_c \times V_d)$
4. $v_c \in V_c \Rightarrow Out(v_c) > 0$, where $Out(v)$ refers to the number of output nodes of the node v.
5. $v_d \in V_{in} \cup V_s \Rightarrow In(v_d) = 0$, where $In(v)$ refers to the number of input nodes to the node v.
6. $v_d \in V_{out} \Rightarrow Out(v_d) = 0$,
7. $v_d \in V_i \Rightarrow In(v_d) > 0 \wedge Out(v_d) > 0$

4.4 Modeling Diagram

One main distinguishing factor between conventional software and ML models is the way they are developed. Conventional software relies on fixed rules specified by the developer for inference, whereas the rules in ML models' are obtained by a learning process using datasets. While ML approaches can solve tasks that conventional software cannot, the mentioned learning process introduces new risks. To address these risks, it is important to understand how the learning process was executed. The ML Modeling Diagram depicts the most important aspects of developing ML models. Figure 4 shows the Modeling Diagram applied to the driverless vehicle use case.

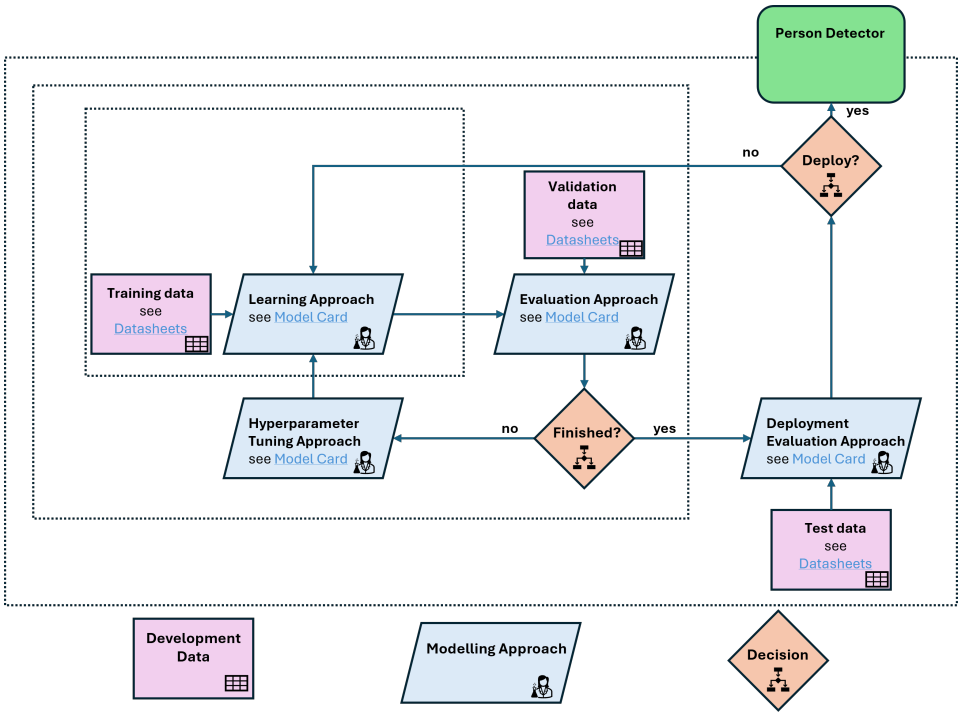

Figure 4 A schematic Modeling Diagram of the driverless vehicle use case showing an exemplary development setup of the Person Detector ML model.

In the following, we detail the relevant elements to create the Modeling Diagram. Note, that arrows in the Modeling Diagram depicts the sequential order in which steps are performed during modeling.

4.4.1 Development Data Elements

There are typically three types of data in the training setup of an ML model: training data, validation data, and test data. Training data is used to adapt the model parameters using the learning algorithm. Validation data is used to evaluate the performance of the trained model, with hyperparameters being tuned based on this evaluation. Once the hyperparameters are fixed, the model undergoes a final evaluation using the test dataset.

Various information about these datasets is relevant for assessing quality and related risk sources, such as quantity and statistical distribution. Use case specific risks may also arise, such as the distribution of ethical characteristics being important in use cases where discrimination is a valid risk. How to comprehensively document datasets in Machine Learning is suggested in datasheets for datasets [5]. If applied, references to the filled datasheets can be added to the ML Modeling Diagram.

4.4.2 Modeling Approach Elements

The Modeling Approach refers to all aspects relating to the actual training process of the ML model. This process includes several design choices by the developer, such as model type, model architecture, hyperparameters, and the learning approach, which may include a loss function and an optimizer.

Another crucial part of any ML model development process is the model evaluation step. Suitable quantification metrics need to be selected to meaningfully describe the relevant performance of the AI system. This also includes specific analyses for certain properties of the AI system, such as accuracy or robustness. Since the choice of quantification metrics and the results can provide valuable insights into how certain risks are controlled, these are relevant information for the risk assessor and are, therefore, reflected in the ML Modeling Diagram.

Note that the aspects of the ML model and ML model evaluation are also addressed by ML model cards [11]. Similar to datasheets, references to these model cards can be included in the ML Modeling Diagram.

4.4.3 Decision Elements

Finally, it is important to highlight that the process of developing an ML model is highly iterative. At various points in the development process, it may be necessary to return to a previous step. This often occurs when evaluation results are not deemed satisfactory. The iterative nature of the development process can be indicated by decisions where a return to a previous step is possible based on specific insights or outcomes.

4.4.4 Rules for Constructing the ML-Modeling Diagram

Since ML developments vary greatly, for instance, based on the type of ML (e.g., supervised, unsupervised, or reinforcement learning) and often involve several iterations, we do not enforce strict rules as we do for the more straightforward Architecture Diagram. However, comprehensive documentation describing the aforementioned aspects should be provided. This documentation is crucial to help assessors identify and evaluate potential risk sources in the ML development setup.

5 Discussion

In this section, we discuss how EAM Diagrams meet the requirements outlined in Section 3 and compare this with related work. Additionally, we address the limitations of our approach.

5.1 Requirement Analysis

In the following we perform a comparative analysis by considering the most related approaches identified in literature. These are the C4 model [2], which introduces also a multi-level diagram concept, and AI Fact Sheets [16]. Although AI Fact Sheets do not introduce a diagram concept, they still introduce a methodology to document various aspects of AI systems, which we consider valid for comparison. An overview of the comparative requirement analysis between the approaches is provided in Table 1.

5.1.1 R1: Flexibility Across Diverse AI Systems

EAM Diagrams are designed with a focus on risk assessments of diverse AI systems. This diversity is considered in two ways: firstly, by defining diagram core elements that are necessary and applicable to various AI systems, and secondly, by ensuring that the properties of these elements are flexible enough to be adjusted to the specific needs of different AI systems while still conveying the necessary information. Consequently, flexibility across diverse AI systems is guaranteed.

The C4 model also provides sufficient flexibility to model any kind of complex software system, including AI systems, using similar arguments. AI Fact Sheets explicitly highlight the diversity of AI systems and their various stakeholders. By providing a general seven-step process for creating AI Fact Sheets for a given use case, they are applicable to a wide range of AI systems.

5.1.2 R2: Ensuring Reproducibility

EAM Diagrams guarantee reproducibility by providing diagram core elements for the AI system description and establishing diagram construction rules where necessary to ensure consistent results.

Similarly, the C4 model introduces specific elements that must be included in the diagrams. AI Fact Sheets also ensure reproducibility by providing explicit guiding questions within each step of the Fact Sheet creation process.

5.1.3 R3: Integration with Risk Assessment

EAM Diagrams are designed to cover the entire AI life cycle: the Modeling Diagram for pre-deployment and the Architecture Diagram for post-deployment. By decomposing complexity into building blocks across different diagrams, while still providing an overview and context of the AI system, assessors can identify and match risk sources, estimate their impact and likelihood, evaluate their significance, and allocate mitigation measures. The visual representation of the AI system provided by EAM Diagrams shows all components and their coordination. This supports a better understanding of the interaction of several risk sources and potential mitigation measures and, therefore, helps in assessing corresponding risks.

The purpose of C4 models is to provide different perspectives on a software system to various stakeholders. However, they do not offer insights into the development process of these systems, which is crucial information for risk assessment due to potential risk sources.

AI Fact Sheets aim to provide holistic documentation of all relevant aspects of an AI system. While they offer general guidelines applicable to any type of stakeholder, they do not focus on specific needs, such as those of risk assessors.

5.1.4 R4: Grounded in Best Practices

EAM Diagrams are grounded in best practices. Their structure is inspired by the C4 model [2] and is compatible with UML and SysML notation, which are established notations in the domain of system architecture descriptions. Furthermore, EAM Diagrams explicitly encourage the integration of best practices in documenting certain aspects of AI systems, such as datasheets for datasets [5] and model cards for model reporting [11].

The C4 model is explicitly compatible with UML. Similarly, AI Fact Sheets build upon state-of-the-art practices in AI documentation and are therefore grounded in best practices.

5.1.5 R5: Reflecting AI-Specific Characteristics

EAM Diagrams are designed to cover the complexity of AI systems and specifically reflect AI-specific characteristics, such as the AI life cycle and the learning process of ML models. To address the learning process, we introduced the Modeling Diagram that explicitly covers these aspects. Additionally, the deployment phase of the AI system is covered by the Environment and Architecture Diagrams. These AI-specific aspects are essential for a comprehensive risk assessment.

Table 1 Overview of the requirement comparison analysis.

Requirement (Section 3)	EAM Diagrams	C4 Model [2]	AI Fact Sheets [16]
R1(Flexibility Across Diverse AI Systems)	✓	✓	✓
R2(Ensuring Reproducibility)	✓	✓	✓
R3(Integration with Risk Assessment)	✓	×	×
R4(Grounded in Best Practices)	✓	✓	✓
R5(Reflecting AI-Specific Characteristics)	✓	×	✓

While the C4 model is well-suited for describing complex software systems, it is not designed to reflect the unique characteristics of AI systems. Consequently, the AI life cycle and the learning process of ML models are not considered in the C4 model. AI Fact Sheets, on the other hand, are specifically designed for documenting relevant aspects of AI systems and therefore reflect AI-specific characteristics.

5.2 Limitations

The EAM Diagrams provide a framework for systematically gathering all relevant information for the risk assessment of AI systems in a diagram format. While the EAM Diagrams claim to be applicable to a wide range of AI systems, they might not be universally applicable to every AI system.

This is due to an implicit assumption about AI systems that was necessary for the structured representation of EAM diagrams. We assumed that the AI life cycle can be strictly divided into development and operation. AI systems that continuously learn after deployment may not be adequately represented by EAM Diagrams. A potential solution would be to integrate a fourth level for such systems depicting the continuous learning plan for such systems. However, since this was not the focus of this research we leave this aspect for future work.

Additionally, the approach assumes that the information required for constructing the AI system diagram is always available. This might not always be the case. When ML models developed or already deployed by third parties are integrated into the AI system, not all necessary information for creating a complete EAM Diagram may be accessible. For instance, if an AI system integrates a deployed version of a foundation model, such as GPT-4 or BART, the integrator might not have access to the information of the training setup or the datasets used in the development process. Notably, this issue is a general problem for all kinds of AI documentation.

6 Conclusion

We presented a framework for systematic creation of an AI system description that collects relevant information to enable efficient risk assessment. If applied, the framework yields three structured diagrams introducing three levels of detail. These represent the AI system's environment, its functional inner workings, and the development process of the integrated Machine Learning model(s). We then demonstrated the effectiveness of the approach by discussing how determined requirements are fulfilled by our approach compared to related work. In future work, we will investigate how EAM Diagrams support compliance with regulations, such as the EU AI Act.

References

1 Grady Booch, James Rumbaugh, and Ivar Jacobson. *The unified modeling language user guide*. The Addison-Wesley object technology series. Addison Wesley, 1999.

2 Simon Brown. The C4 Model for Software Architecture, June 2018. URL: https://www.infoq.com/articles/C4-architecture-model/.

3 J Eichler and D Angermeier. Modular Risk Assessment for the Development of Secure Automotive Systems, January 2015.

4 European Parliament and Council of the European Union. Artificial intelligence Act, July 2024. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

5 Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for Datasets, December 2021. doi:10.48550/arXiv.1803.09010.

6 IEC. 31010:2012 - Risk Management - Risk assessment techniques.

7 ISO. 31000: Risk Management — Guidelines, 2009.

8 ISO. TR 4804:2020 Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation, 2020.

9 ISO/IEC. GUIDE 51: Safety aspects: Guidelines for their inclusion in standards, 2014.

10 ISO/IEC. 21448:2022 Road vehicles — Safety of the intended functionality, 2022.

11 Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229, January 2019. doi:10.1145/3287560.3287596.

12 Object Management Group (OMG). OMG Systems Modeling Language (OMG SysML™), V1.0, 2007.

13 Maximilian Poretschkin, Anna Schmitz, Maram Akila, Linara Adilova, Daniel Becker, Armin B Cremers, Dirk Hecker, Sebastian Houben, Julia Rosenzweig, Joachim Sicking, Elena Schulz, Angelika Voss, and Stefan Wrobel. Guideline for Trustworthy Artificial Intelligence – AI Assessment Catalog, 2023. doi:10.48550/arXiv.2307.03681.

14 Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing, 2020. arXiv:2001.00973, doi:10.48550/arXiv.2001.00973.

15 Vipula Rawte, Amit Sheth, and Amitava Das. A Survey of Hallucination in Large Foundation Models, September 2023. doi:10.48550/arXiv.2309.05922.

16 John Richards, David Piorkowski, Michael Hind, Stephanie Houde, and Aleksandra Mojsilović. A Methodology for Creating AI FactSheets, June 2020. arXiv:2006.13796, doi:10.48550/arXiv.2006.13796.

17 Ronald Schnitzer, Andreas Hapfelmeier, Sven Gaube, and Sonja Zillner. AI hazard management: a framework for the systematic management of root causes for AI risks. In Mina Farmanbar, Maria Tzamtzi, Ajit Kumar Verma, and Antorweep Chakravorty, editors, *Frontiers of artificial intelligence, ethics, and multidisciplinary applications*, pages 359–375, Singapore, 2024. Springer Nature Singapore. doi:10.1007/978-981-99-9836-4_27.

18 Charlotte Siegmann and Markus Anderljung. The Brussels Effect and Artificial Intelligence. preprint, Politics and International Relations, October 2022. doi:10.33774/apsa-2022-vxtsl.

19 André Steimers and Moritz Schneider. Sources of Risk of AI Systems. *International Journal of Environmental Research and Public Health*, 19(6):3641, March 2022. doi:10.3390/ijerph19063641.

20 Stefan Studer, Thanh Binh Bui, Christian Drescher, Alexander Hanuschkin, Ludwig Winkler, Steven Peters, and Klaus-Robert Müller. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Machine Learning and Knowledge Extraction*, 3(2):392–413, April 2021. doi:10.3390/make3020020.

21 Laura Waltersdorfer, Fajar J. Ekaputra, Tomasz Miksa, and Marta Sabou. AuditMAI: Towards An Infrastructure for Continuous AI Auditing, June 2024. doi:10.48550/arXiv.2406.14243.

22 Gereon Weiss, Marc Zeller, Hannes Schönhaar, Chrstian Drabek, and Andreas 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. doi:10.1145/3644815.3644944.

23 Oliver Willers, Sebastian Sudholt, Shervin Raafatnia, and Stephanie Abrecht. Safety Concerns and Mitigation Approaches Regarding the Use of Deep Learning in Safety-Critical Perception Tasks, January 2020. arXiv:2001.08001, doi:10.48550/arXiv.2001.08001.