Towards Supporting AI System Engineering with an Extended Boxology Notation

Fajar J. Ekaputra[1,2], Alexander Prock[1] and Elmar Kiesling[1]

[1]*Institute of Data, Process, and Knowledge Management, WU, Vienna 1020, Austria*
[2]*Institute of Information System Engineering, TU Wien, Vienna 1040, Austria*

Abstract

As AI systems grow in complexity, understanding their structure and behaviour becomes increasingly challenging. The boxology notation offers a dataflow-oriented abstraction to simplify AI system representation and help address this challenge. In this work, we explore the potential of extending the boxology notation for AI system engineering and introduce the Boxology Extended Annotation Model (BEAM). BEAM enhances boxology through (i) incorporating auxiliary notations to capture engineering-relevant information and (ii) introducing an additional perspective for AI system risk assessment and mitigation. Furthermore, we developed the BEAM ontology as a machine-readable representation of BEAM to support further functionalities. We evaluated the BEAM approach through (a) an initial feasibility evaluation with students in a classroom setting and (b) applying BEAM on use cases as part of a research project. Positive feedback from both evaluations demonstrates its effectiveness in supporting the AI system engineering process.

Keywords

AI System Engineering, AI System Representation, Boxology Notation

1. Introduction

The rapid growth of artificial intelligence (AI) has led to increasingly complex systems that are challenging to understand, analyze, and engineer. Complex AI systems are often seen as ”black boxes” whose internal workings are opaque, even to those who develop and deploy them, as many stakeholders are only responsible for parts of the systems [1, 2], e.g., ethicist may only focus on the ethical, social, and philosophical aspects of AI systems with limited knowledge of the details on system components.

This growing complexity has raised concerns about interpretability, transparency, and reliability, making it crucial to develop systematic approaches to improve the understanding of AI systems and their engineering process. One approach that has gained attention in addressing this challenge is the boxology notation [3], a structured abstraction method that represents AI systems through a visual dataflow perspective. The updated version of this notation [4] provides a simplified, modular visualization of AI systems, allowing researchers and practitioners to describe and decompose complex AI systems into their components and patterns. To the extent of our knowledge, the boxology notation has primarily been used for post-hoc analysis of AI systems [5, 6], facilitating identification and representation of abstract patterns of neuro-symbolic AI systems.

In this work, we investigate how the boxology notation can be leveraged as a supporting tool for AI system engineering, both during design and development phases. To this end, we introduce the Boxology Extended Annotation Model (BEAM), which enriches the boxology notation with further system and annotation elements that capture essential information relevant to the engineering process. Specifically, BEAM extends boxology in two aspects:

(i) *System perspective*: BEAM adds auxiliary elements to capture relevant information for AI system engineering processes, including the notion of abstract system elements, containers, and annotations.

2nd International Workshop on Knowledge Graphs for Responsible AI (KG-STAR) Co-located with the Extended Semantic Web Conference (ESWC 2025)

fajar.ekaputra@wu.ac.at (F. J. Ekaputra); alexander.prock@wu.ac.at (A. Prock); elmar.kiesling@wu.ac.at (E. Kiesling)

(ii) *Risk and Mitigation perspective*: BEAM introduces a specific perspective for representing AI system risks and mitigation strategies within the boxology framework. This perspective enables engineers to systematically assess potential risks and incorporate mitigation strategies in the system design process.

These extensions provide stakeholders with a shared language to describe and communicate the design of AI systems between them. Beyond these extensions, we also developed the BEAM ontology, which provides a machine-readable representation of AI system structures based on the enhanced boxology framework. Our ontology is mapped to existing relevant work, such as EASY-AI [7], SWeMLS ontology [8], and AIRO [9] to ensure interoperability between these approaches. The BEAM ontology enables analysis, integration with other tools, and reasoning capabilities. Furthermore, we are working on two-way transformations between BEAM's formal and visual representations to ensure that stakeholders can seamlessly transition between structured documentation and graphical design views.

To validate the usefulness of BEAM, we demonstrated our approach in both classroom settings and real-world use cases as part of an ongoing research project[1]. Our findings indicate that BEAM improves the clarity and risk awareness in the AI system development process, demonstrating its potential as a valuable tool for AI engineers.

2. Related Work

This section offers a brief overview from two research areas as a basis for our work in this paper, namely (i) AI System Representation, and (ii) AI Risk modeling.

2.1. AI System representation.

A particularly strong need for adequate modeling and representation of AI systems has emerged in the context of hybrid/neurosymbolic AI which combines symbolic and sub-symbolic paradigms, often resulting in complex system architectures. For instance, initial AI system modeling approaches have been developed for the identification and representation of *NeSy-AI systems design patterns*. Early and notable contributions in this area include the boxology framework [3] and its subsequent extension [4] intended to represent design patterns of NeSy-AI systems through visual notations. Recently, the design patterns extended to cater for the emerging combination of Large Language Model (LLM)-based Neurosymbolic AI system [10].

Researchers have proposed several similar approaches for formalizing the boxology notation. For example, Mossakowski [11] has suggested an alternative symbolic representation for the boxology notation through the DOL meta language[2], while Ellis et al. [7] proposed the EASY-AI formalism for the boxology notation with semantically rich axioms for validation and classification purposes. Ellis et al. goes further to develop SNOOP-AI, a support tool for implementation of new design pattern [12]. In our prior work, we also developed a similar representation, focusing on the system perspective [8] as a first-class citizen, in addition to the pattern representation and a set of SHACL constraint validation. This work, however, mainly built on the earlier version of the boxology framework [3], and therefore did not consider processes and actors as part of system representations.

2.2. AI Risk modeling

A large variety of Artificial Intelligence (AI) risk modeling approaches have been developed in recent years, which can be organized into a number of categories:

[1] https://fair-ai.at/

[2] https://dol-omg.org

Regulatory Frameworks at the national and supra-national level – such as the OECD AI principles[3] or the EU AI Act – regulate the use of AI from a risk-driven perspective. Whereas such regulatory frameworks – and the EU AI Act in particular – do not themselves provide specific risk modeling guidance or formalisms, they have been the primary motivation for and heavily influenced many of the AI risk modeling approaches developed in the following categories.

AI Risk Management Frameworks such as the Assessment List for Trustworthy Artificial Intelligence (ALTAI)[4] for self-assessment, the MIT AI Risk Repository[5] [13], the NIST AI Risk Management Framework[6] [14] or the CSIRO Responsible AI Pattern catalogue[7] [15, 16] all provide systematic approaches and normative guidance for AI risk management at a high level. They do not, however, use semantic modeling techniques to link risks to components and design decisions.

Standards AI risk management has also become a very active field in terms of standardization efforts. This includes ISO/IEC standards such as ISO/IEC: 42001:2023 (AI Management system), ISO/IEC 23894:2023 (AI Guidance on risk management), and ISO/IEC 31000 for risk management more broadly. In addition, numerous additional standardization activities, e.g., by organizations such as the European Telecommunications Standards Institute (ETSI) Technical Committee on Securing Artificial Intelligence Framework (SAI)[8] (which primarily approaches AI risks from a security perspective) or Institute of Electrical and Electronics Engineers (IEEE), which for instance has developed IEEE 7000-2021 for addressing Ethical Concerns during System Design more broadly.

Industry Frameworks A number of frameworks for modeling AI risks have also been developed within industry, including for instance IBM Fact Sheets and Risk Atlas[9], Google's Model Cards [17], Microsoft's Responsible AI Standard which provides guidelines for AI developers at Microsoft, or the Algorithmic Impact Assessment (AIA) approach developed by AI Now institute [18].

AI Risk (management) taxonomies and ontologies The National Institute of Standards and Technology (NIST) Taxonomy of AI risk[10], which has developed as part of a larger AI Risk Management Framework, is currently available as a draft. It covers only high-level categories and maps them to relevant policy documents, but does not provide formalizations.

The ETSI SAI AI Threat Ontology [11] aims to describe the AI threat landscape from a security perspective. Although it mentions Resource Description Framework (RDF) and Web Ontology Language (OWL) as options, it only provides illustrative examples of how they could be used to formally specify an SAI ontology, but a full-fledged formal ontology is to our knowledge not available.

More formal ontologies and vocabularies defined in RDF include the AI Risk Ontology (AIRO)[12] [9, 19], Vocabulary of AI Risks (VAIR) [13] and the DPVCG Risk Extension [14]. All of them are primarily motivated by regulatory compliance needs and they consequently provide limited normative guidance and engineering support for risk-aware AI system design.

AIRO in particular is grounded in ISO risk management standards [20] and primarily motivated by regulatory needs imposed by the AI Act; it aims to assist stakeholders in determining "high-risk" AI systems [19] and to this end, it provides the necessary concepts to systematically document identified

[3] https://oecd.ai/en/ai-principles
[4] https://futurium.ec.europa.eu/en/european-ai-alliance/pages/welcome-altai-portal
[5] https://airisk.mit.edu
[6] https://www.nist.gov/itl/ai-risk-management-framework
[7] https://research.csiro.au/ss/science/projects/responsible-ai-pattern-catalogue/
[8] https://www.etsi.org/committee/2312-sai
[9] https://www.ibm.com/docs/en/watsonx/saas?topic=ai-risk-atlas
[10] https://www.nist.gov/system/files/documents/2021/10/15/taxonomy_AI_risks.pdf
[11] https://www.etsi.org/deliver/etsi_ts/104000_104099/104050/01.01.01_60/ts_104050v010101p.pdf
[12] https://delaramglp.github.io/airo/
[13] https://delaramglp.github.io/vair/
[14] https://dev.dpvcg.org/2.1-dev/risk/ (W3C community report)

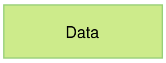
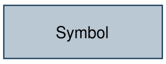
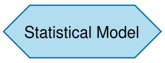
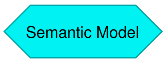
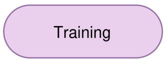
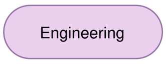
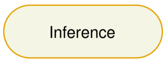
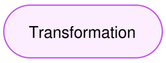
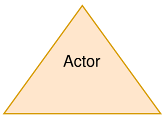
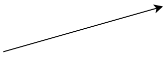
Concept	Hybrid AI Taxonomy	Visualization	Examples
Data	Instance → Data		text, images, tensors
Symbol	Instance → Symbol		knowledge graphs, taxonomies
Statistical Model	Model → Statistical Model		neural networks, large language models
Semantic Model	Model → Semantic Model		rule-based models, ontologies
Training	Processing → Generation ⟶ Training		supervised learning, reinforcement learning
Engineering	Processing → Generation ⟶ Engineering		ontology engineering, rule definition by human expert
Inference	Processing → Inference		classification, natural language generation
Transformation	Processing → Transformation		data imputation, mapping
Actor	Actor		domain expert, user
Workflow Connector	*none*		connecting input/output with processes, usage of a model in an inference process

Table 1
Visual notation elements based on the extended boxology framework and pattern diagrams by van Bekkum et al. [4]. The second column indicates the corresponding concept as well as its superclasses from the hybrid AI taxonomy proposed alongside the extended boxology framework. In the third column, the labels on the visual elements denote the concept represented by them, whereas in a concrete model instance the labels are used to describe what the visual element represents more precisely. For an example of a concrete model instance, see Section 4.3.

risks – primarily at the system level. It does not, however, include concepts for granular system descriptions and does therefore not provide a foundation for AI system engineering activities such as evaluating system designs, automatically suggesting potential risks based on generic risk patterns captured in a knowledge base, reasoning about emergent risks and their interactions, suggest mitigations or help to clarify trade offs involved in design choices. BEAM aims to take first steps to fill this research gap.

3. BEAM: Boxology Extended Annotation Model

BEAM consists of two parts: (i) the BEAM visual notation provides a set of reusable elements to describe AI systems, which can be used as a basis for seamless communication between stakeholders, and (ii) the BEAM ontology, which caters for machine actionability of the BEAM visual notation, facilitating easier analysis and advanced functionalities, such as reasoning and tools integration.

In this section we introduce both the BEAM visual notation and the BEAM ontology, followed by a description of their utilization.

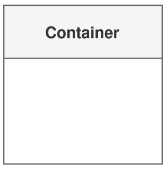
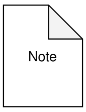
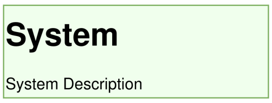
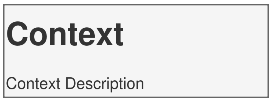
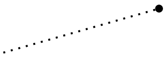
Concept	Visualization	Purpose
Container		grouping of system components, e.g. to represent subsystems or complex processes
Note		free-text documentation
System		descriptive element for the system as a whole
Context		descriptive element for the context the system operates in
Annotation Connector		connection between system components and annotation elements

Table 2
Visual notation of additional annotation elements for capturing engineering-relevant information.

3.1. BEAM Visual Notation

The visual notation of BEAM is based on the hybrid AI taxonomy of the extended boxology framework by van Bekkum et al. [4]. Where applicable, the BEAM visual notation adheres to the shapes used in the pattern diagrams of the extended boxology framework, however, our visual notation introduces different colors for each concept to ease their distinction for the viewer. Table 1 provides an overview of our notation for concepts defined in the extended boxology framework.

Following the notation of the pattern diagrams, data and symbols are represented as rectangles, models as hexagons, and processes as rounded rectangles. The two types of generative processes, i.e. training and engineering, are assigned the same color. Actors are not assigned a shape in the extended boxology framework, hence we chose a triangle to represent them visually. Pattern diagrams include arrows that connect various types of elements to visualize the workflow within the AI system, but these arrows are not explicitly defined in the framework. Our notation defines these arrows as workflow connectors.

The BEAM notation extends the boxology framework in two ways: (i) by introducing annotation elements to capture engineering-relevant information in order to facilitate better communication between various stakeholders already in the design and development phases of AI systems, and (ii) by introducing annotation elements to capture AI system risks and risk controls to enable a systematic assessment of potential risks and risk control strategies in the system design process.

Annotation Elements for Capturing Engineering-Relevant Information. Table 2 provides an overview of the visual notation of the additional annotation elements to capture engineering-relevant information. Container elements can be used for grouping other system components into parts, e.g. to represent subsystems or complex processes. They are flexible in size, s.t. any group of system components can be placed within, and include a title bar. Free-text notes in the style of UML [21] can be used to attach any kind of textual information to system components. System and context annotation elements enable the textual documentation of the system itself, as well as the context in which it operates. To distinguish their connectors from arrows that indicate the workflow within the documented system, a different type of connector is introduced to connect annotations to system

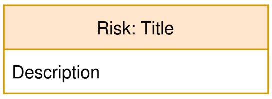
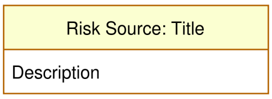
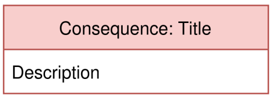
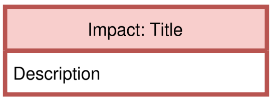
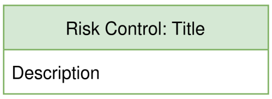
Concept	Visualization	Adapted Definition from AIRO
Risk		the state of uncertainty associated with an AI system, a component of it or a set of such components, that has the potential to cause harms
Risk Source		an element that has the potential to give rise to a risk
Consequence		direct outcome of risk affecting objectives
Impact		the outcome of a consequence on individuals, groups, society, environment, etc.
Risk Control		a measure that maintains and/or modifies one or more risks, risk sources, consequences and impacts

Table 3
Visual notation of additional annotation elements for the AI risk perspective. The definitions are taken from AIRO [20], slightly adapted for our focus on AI system engineering, emphasizing the ability to connect the annotation elements of the risk perspective to individual system elements (or combinations of them) to facilitate fine-grained assessment of risks and risk controls.

components.

Annotation Elements for AI Risk Perspective. Table 3 provides an overview of the visual notation of the AI risk perspective, comprised of five additional annotation elements that represent core concepts of the AI Risk Ontology (AIRO), i.e., risks, risk sources, consequences, impacts, and risk controls. Table 3 additionally provides a definition for each concept, based on the definition in AIRO.

Each of these concepts is visualized as a rectangular shape with a title bar. By convention, the title starts with the type of concept represented, followed by a descriptive part, e.g. *Risk Source: Poor Data Quality*. Below the title bar, a free-text description can be added.

Annotation elements of the risk perspective can be attached to any single or multiple system elements using the annotation connector depicted in Table 2. Connections within the risk perspective, e.g. to connect risks with their consequences or consequences with their impacts, are represented by the workflow connector, depicted in Table 1, which clearly expresses directionality.

3.2. BEAM Ontology

In this section, we will report on the BEAM ontology, a formal representation of the BEAM notation to allow for further reasoning and analysis of the systems. The ontology consists of two parts: (i) BEAM core, which represents elements from the boxology notation [4] and EASY-AI [7], with additional elements for AI system [8], and (ii) the proposed extension on AI system risks and mitigation, inspired from the work of Golpayegani et al. [9]. The BEAM ontology is available online[15].

BEAM Core. The BEAM core ontology is designed to support the engineering process of AI systems through its entire lifecycle. It is developed based on concepts defined by EASY-AI [7] and SWeMLS ontology [8]. Note that we adapt and rename several classes and properties, while keeping the relation

[15] https://w3id.org/beam/

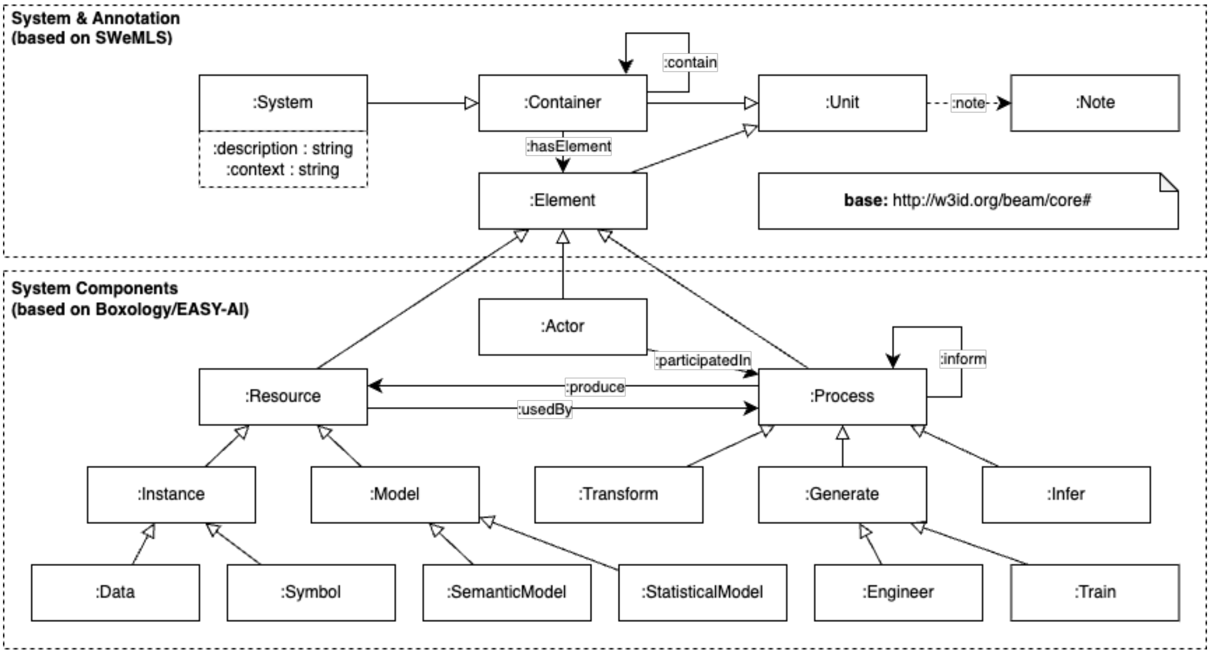


Figure 1: Key concepts and relations within BEAM core ontology

with the original concepts through `rdfs:subClassOf` and `rdfs:subPropertyOf`. Furthermore, we extended the ontology with additional concepts to capture relevant information on AI system engineering process, such as `beam:Resource`, `beam:Container`, `beam:Unit`, and `beam:Note`.

An overview of BEAM's core classes, relations and properties is shown in Figure 1 (note that we omitted sub-properties and inverse properties for clarity). We outline our adaptations and extensions to the existing ontologies in the following.

- Most classes from the boxology notations (cf. bottom part of Figure 1) are direct subclasses of their respective classes from EASY-AI, which includes: `beam:Actor`, `beam:Process` (and its sub-classes), `beam:Model` (including its sub-classes), `beam:Data`, and `beam:Symbol`. Class `beam:Instance`, which revert to its original name from boxology instead of `easy-ai:Artifact`, is an exception.

- Class `beam:Resource` is added as a super-class of both `beam:Instance` and `beam:Model`, representing possible (non-actor) input for `beam:Process`.

- We differentiated object properties from `beam:Resource` and `beam:Process` (i.e., `beam:usedBy`) and between `beam:Resource` and `beam:Process` (i.e., `beam:participatedIn`, for clarity. Furthermore, we added `beam:inform` relation between `Process` instances, inspired from similar concept from PROV-O [22].

- Class `beam:Element` is a super-class of `beam:Resource`, `beam:Process`, and `beam:Actor`. The aim is to represent instances of boxology elements involved in a `beam:System`. Class `beam:Element` is a direct subclass of `swemls:Unit`.

- Class `beam:System` as a direct sub-class of `swemls:System` that contains a set of interconnected `swemls:Element` designed towards addressing specific goal in a given context.

- Class `beam:Container` as a superclass of `beam:System`, representing logical groups of `beam:Element`, that are not necessarily interconnected. A `beam:Container` can contain another `beam:Container` through object property `beam:contain`.

- Class `beam:Unit` is defined as the superclass of `beam:Container` and `beam:Element`, representing all components and logical groups defined over a `beam:System`. Any `beam:Unit` can be linked to `beam:Note` through object property `beam:note`.
- Class `Note` represents various type of annotations that can be attributed to any `beam:Unit`.

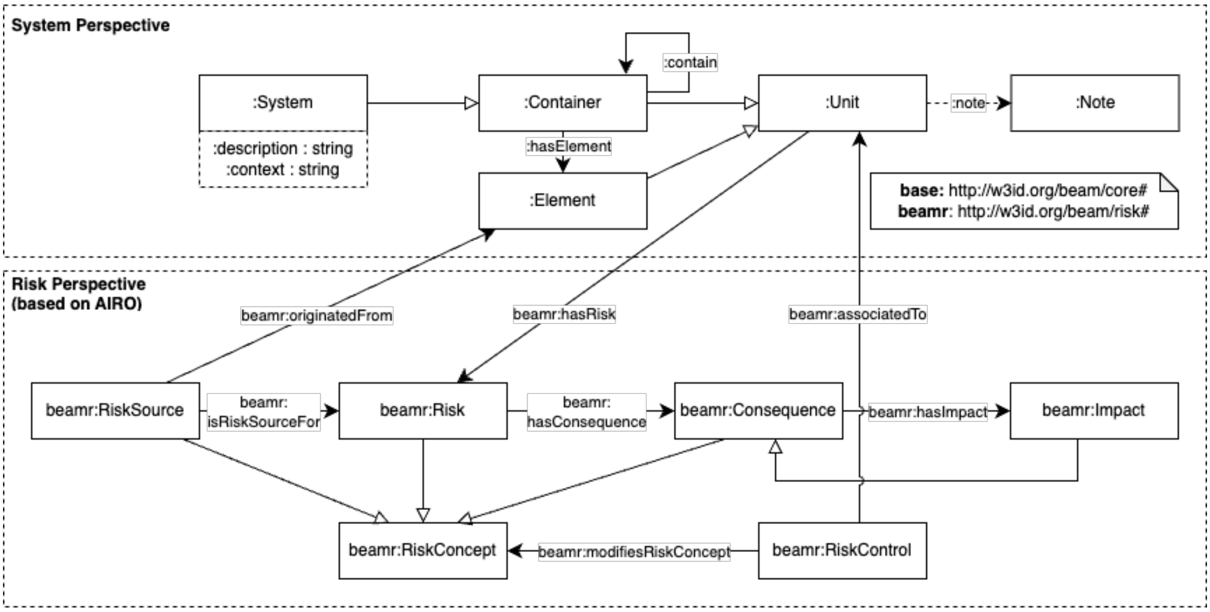


Figure 2: Risk concepts and their relation to core system concepts.

BEAM Risk Perspective. Existing AI risk (management) conceptualizations (see the overview in Section 2) include (i) descriptive taxonomies that provide general definitions of relevant concepts – published for instance by bodies like NIST or ETSI as part of their standardization efforts; (ii) more formally specified ontologies and vocabularies – such as the AIRO or VAIR developed by ADAPT Centre Dublin; and (iii) catalogs that provide best practices – e.g., the responsible AI pattern catalog published by CSIRO[16] [15, 16].

However, whereas taxonomies in category (i) are informative, they are limited to natural language definitions of key terms and do not aim to support modeling or machine-readable knowledge representation. More formal vocabularies and ontologies in category (ii) do provide representations and are useful for high-level descriptions of AI systems and their risks — e.g., for documentation purposes — but they are primarily designed for high-level risk assessment rather than to support automated reasoning about risks and implications in a particular system design context. Therefore, they do not relate risks to granular system elements, do not enable risk-aware AI systems engineering, and do not provide prescriptive guidance. Finally, risk patterns in category (iii) do provide guidance, but they are limited to natural-language descriptions of general patterns on various levels and also do not aim to support AI system engineering through machine-readable descriptions and automated reasoning.

To tackle these limitations and take initial steps towards semantically supported risk-aware AI systems engineering, we base the BEAM Risk Perspective on existing resources – AIRO in particular – and extend them into a more granular conceptualization of AI risks aligned with BEAM to enable detailed risk modeling and risk localization within an architectural design. In the following, we describe how we reuse concepts from AIRO to complement the BEAM core concepts with a *risk perspective.*

Key classes in the risk perspective are subclassed from AIRO to enable reuse and extension:

- Class `beamr:Risk` as a sub-class of `airo:Risk`

[16] https://research.csiro.au/ss/science/projects/responsible-ai-pattern-catalogue/

- Class beamr:RiskConcept as a sub-class of airo:RiskConcept
- Class beamr:RiskSource as a sub-class of airo:RiskSource
- Class beamr:Consequence as a sub-class of airo:Consequence
- Class beamr:Impact as a sub-class of airo:Impact
- Class beamr:RiskControl as a sub-class of airo:RiskControl

Properties in the risk perspectives are partly reused from AIRO as well, but are extended for more granular modeling:

- hasRisk, which unlike the definition of the property in AIRO relates risks and granular system elements (i.e., Resources – including Data, Symbol, and Model; Processes, i.e. Transform, Generate and Infer; and Actors)
- originatesFrom is introduced to associate abstract RiskSources (e.g., *poor_quality_of_input_data*) with concrete Elements in an architecture (e.g., a particular input data set used in the training process).

  These two main properties play distinct roles in modeling risks and localizing the risks in a system architecture — originatesFrom is used to to link risk sources to particular system elements that have the potential to give rise to a risk, whereas hasRisk is used to relate system elements where potential risks materialize to those risks.
- isRiskSourceFor, hasConsequence, hasImpact and modifiesRiskConcept are reused from AIRO.
- associatedTo is introduced to associate RiskControls with concrete Units in an architecture to indicate in which part of the system the RiskControl is implemented or which parts it affects.

3.3. BEAM Utilization

The utilization of BEAM, i.e. the visual notation and the ontology, in practice depends on available tools and the involved stakeholders. The default approach starts with the manual construction of a system model in the visual notation. This does not presuppose any specific technical knowledge by the modeler apart from an understanding of the AI system architecture under consideration, however, we assume that prior experience with modeling, e.g. in UML [21], is helpful. Theoretically, any diagramming or modeling tool capable of depicting the elements of the visual notation can be used, however, a transformation between the visual model and the machine-actionable semantic representation based on the BEAM ontology is only possible if the chosen tool allows the import and export in a machine-readable format. For details on our prototypical implementation, see Section 4.1.

Currently, the transformation from visual notation to the respective semantic representation can only be done manually by a knowledge engineer. We developed an initial prototype for such transformation as part of a master thesis [23]. However, it is preliminary work and does not cater to the current version of the BEAM notation and semantic representation.

The two-way transformation between the visual notation and the semantic representation based on the BEAM ontology is subject to future work, see Section 5. Once a two-way transformation between representations is available, the utilization of BEAM can also start from the semantic representation, e.g. by automated model generation based on textual documentation using an LLM, providing a way to model large-scale AI systems for which manual modeling is not feasible. Our goal is to make BEAM accessible to stakeholders that are not experts in semantic web technologies, thus a direct interaction of stakeholders with the semantic representation is not necessary.

4. Evaluation

This section will report on the initial evaluation of BEAM, consisting of (i) the development of a prototypical implementation of BEAM notations as a draw.io library (cf. Section 4.1), (ii) an initial feasibility evaluation with data science students (cf. Section 4.2), and (iii) a workshop-based evaluation with in an industrial research project (cf. Section 4.3), including an example BEAM instantiation from a simplified use case in the project.

4.1. Prototypical Implementation of BEAM

A prototype of the BEAM visual notation was implemented as a library for the popular diagramming application draw.io.[17] The library is available online on GitHub[18] as an XML file that can be imported into the application. draw.io was chosen as it fulfills all of our requirements, i.e. the definition of elements for the concepts defined in Section 3.1, the ability to predefine these elements in the form of a library or templates, easy manipulation of concrete elements based on the aforementioned library or templates, as well as the ability to save and load the diagrams in a machine-readable format to facilitate the transition between the visual notation and semantic representations based on the BEAM ontology. Furthermore, it is open-source, free, well-known and intuitive to use.

During our feasibility evaluation we also used features that were not strictly necessary but improved the usability of our diagrams, namely layers and collapsible elements. draw.io allows the creation of layers, to which elements can be assigned. The visibility of each layer can be toggled individually, thus presenting a rudimentary way of implementing multi-perspective views. A basic setup can be to add the elements that represent the system components to a base layer that is always visible. Additional layers can be created for engineering-relevant annotations and the AI system risk and mitigation perspective. Using this setup, different stakeholders can take their desired perspective on the documented AI system, optionally including or omitting details. Diagram elements with the collapsible property enabled can be collapsed, including all elements within them. Using the collapsible property on containers that represent subsystems or complex processes allows temporarily hiding details when a high-level overview is desired.

4.2. Initial evaluation in Data Science Lab Student Industry Projects

As an initial step, we invited students working on a capstone project at the end of their Data Science specialization to use and assess BEAM. The setting was a final course where student teams of 3-4 students work on a real-world project for an industry partner for one semester. Each group was assigned a case from a different industry partner, which effectively covered a wide range of sectors and a broad variety of methods and approaches.

Specifically, we provided students with the draw.io-based modeling tool, detailed instructions on how to use it, and invited them to (optional) tutorial sessions. Students were then asked to describe their solution architecture - once as an initial solution architecture during the initial stages of the project as part of a project plan submission and once at the end of the project to document their final solution. So far, this resulted in approximately 25 models of real-world industry use cases collected over the course of two semesters.

We collected initial feedback on the BEAM through a small-scale survey involving selected stakeholders: students, instructors, lecturers, and involved industry partners. We developed the questionnaire based on the TAM model [24], focusing on three aspects: (i) perceived usefulness, (ii) perceived ease of use, and (iii) two open-ended questions about experiences with the notation and feedback on BEAM aspects that could be improved. The participants were asked to provide their scores on a 7-point Likert scale (1-strongly agree, 7-strongly disagree).

[17] https://www.drawio.com/

[18] https://github.com/wu-semsys/beam_tutorial – The repository includes a quick tutorial and further documentation.

The stakeholders generally found the approach easy to use and the resulting models useful (average score: 3/7) and easy to use (average score: 3/7). They found it valuable, particularly for discussing the solution approaches among students, instructors, and industry partners (average score: 2/7). As part of the feedback, a student noted that the BEAM tool helped scope the project, engage with industry partners to elicit requirements, discuss risks and limitations, and develop a joint vision of the solution approach.

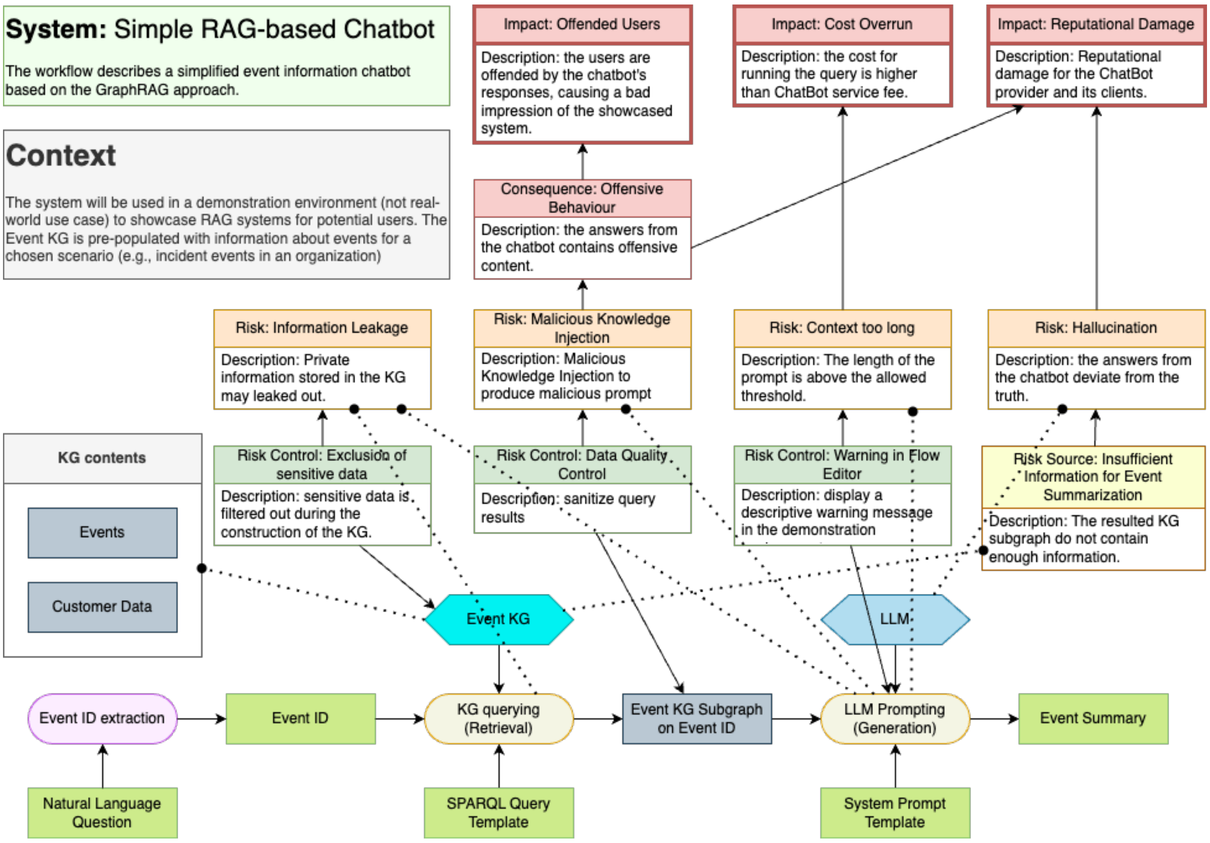

System: Simple RAG-based Chatbot
The workflow describes a simplified event information chatbot based on the GraphRAG approach.
Context
The system will be used in a demonstration environment (not real-world use case) to showcase RAG systems for potential users. The Event KG is pre-populated with information about events for a chosen scenario (e.g., incident events in an organization)
Impact: Offended Users
Description: the users are offended by the chatbot's responses, causing a bad impression of the showcased system.
Impact: Cost Overrun
Description: the cost for running the query is higher than ChatBot service fee.
Impact: Reputational Damage
Description: Reputational damage for the ChatBot provider and its clients.
Consequence: Offensive Behaviour
Description: the answers from the chatbot contains offensive content.
Risk: Information Leakage
Description: Private information stored in the KG may leaked out.
Risk: Malicious Knowledge Injection
Description: Malicious Knowledge Injection to produce malicious prompt
Risk: Context too long
Description: The length of the prompt is above the allowed threshold.
Risk: Hallucination
Description: the answers from the chatbot deviate from the truth.
KG contents
Events
Customer Data
Risk Control: Exclusion of sensitive data
Description: sensitive data is filtered out during the construction of the KG.
Risk Control: Data Quality Control
Description: sanitize query results
Risk Control: Warning in Flow Editor
Description: display a descriptive warning message in the demonstration
Risk Source: Insufficient Information for Event Summarization
Description: The resulted KG subgraph do not contain enough information.
Event KG
LLM
Event ID extraction
Event ID
KG querying (Retrieval)
Event KG Subgraph on Event ID
LLM Prompting (Generation)
Event Summary
Natural Language Question
SPARQL Query Template
System Prompt Template

Figure 3: BEAM visual notation for a simple Graph-RAG based ChatBot system

4.3. Workshop-based evaluation in industrial research projects

As a second evaluation step, we conducted in-depth modeling workshops with seven industry partners as part of an Austrian lighthouse project *FAIR-AI*[19], which aims to contribute towards a principled methodological approach to build trustworthy AI systems. This evaluation was conducted in two phases:

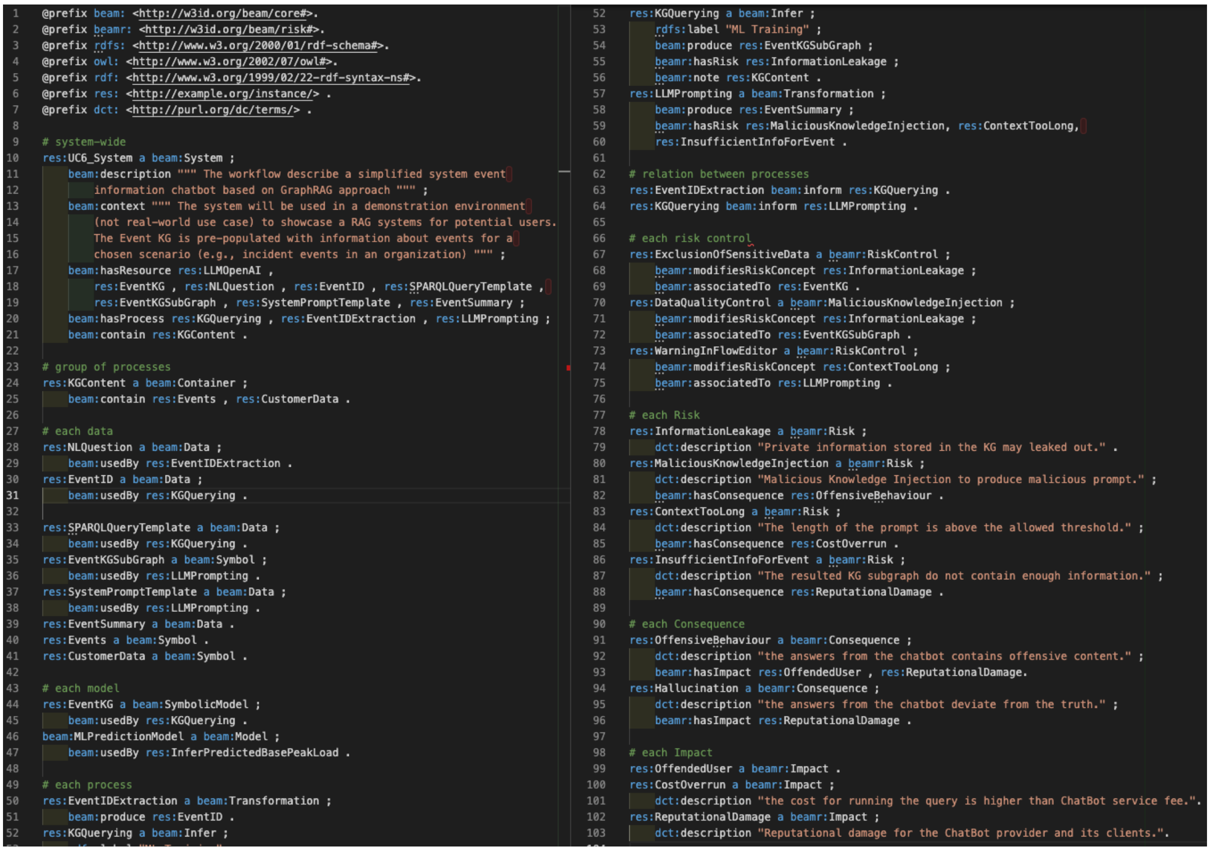
In the *pilot phase*, a core team of six researchers engaged with industrial partners in half-day exploratory workshops to model their use cases and assess the risks involved in these use cases from scratch using the BEAM methodology. Two instances of these workshops were conducted[20].

In the second phase, we conducted guided modeling sessions with five additional project partners from a variety of industries. In this phase, the use cases were pre-modeled by the researchers based on initial use case descriptions and then fleshed in detail in a three-step process (AI system modeling, risk modeling, mitigation modeling). The participants provided both direct qualitative feedback and participated in a survey after the modeling sessions.

[19] https://fair-ai.at/

[20] One of them contributed by a very large-scale industrial manufacturing company focused on microgrid demand predictions and one use case contributed by an SME developing a hybrid AI chatbot frameawork

The overall results were very promising and the general feedback was that BEAM helped to (i) explicate the technical approach and communicate the solution architecture across stakeholders (including – in this setting – researchers, software engineers, and business sponsors), (ii) link technical design decisions to risks and business concerns, and (iii) improve risk-awareness across lifecycle phases (i.e., help to guide risk-aware monitoring after deployment).



```
@prefix beam: <http://w3id.org/beam/core#>.
@prefix beamr: <http://w3id.org/beam/risk#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix res: <http://example.org/instance/> .
@prefix dct: <http://purl.org/dc/terms/> .

# system-wide
res:UC6_System a beam:System ;
    beam:description """ The workflow describe a simplified system event
        information chatbot based on GraphRAG approach """ ;
    beam:context """ The system will be used in a demonstration environment
        (not real-world use case) to showcase a RAG systems for potential users.
        The Event KG is pre-populated with information about events for a
        chosen scenario (e.g., incident events in an organization) """ ;
    beam:hasResource res:LLMOpenAI ,
        res:EventKG , res:NLQuestion , res:EventID , res:SPARQLQueryTemplate ,
        res:EventKGSubGraph , res:SystemPromptTemplate , res:EventSummary ;
    beam:hasProcess res:KGQuerying , res:EventIDExtraction , res:LLMPrompting ;
    beam:contain res:KGContent .

# group of processes
res:KGContent a beam:Container ;
    beam:contain res:Events , res:CustomerData .

# each data
res:NLQuestion a beam:Data ;
    beam:usedBy res:EventIDExtraction .
res:EventID a beam:Data ;
    beam:usedBy res:KGQuerying .

res:SPARQLQueryTemplate a beam:Data ;
    beam:usedBy res:KGQuerying .
res:EventKGSubGraph a beam:Symbol ;
    beam:usedBy res:LLMPrompting .
res:SystemPromptTemplate a beam:Data ;
    beam:usedBy res:LLMPrompting .
res:EventSummary a beam:Data .
res:Events a beam:Symbol .
res:CustomerData a beam:Symbol .

# each model
res:EventKG a beam:SymbolicModel ;
    beam:usedBy res:KGQuerying .
beam:MLPredictionModel a beam:Model ;
    beam:usedBy res:InferPredictedBasePeakLoad .

# each process
res:EventIDExtraction a beam:Transformation ;
    beam:produce res:EventID .
res:KGQuerying a beam:Infer ;
    rdfs:label "ML Training" ;
    beam:produce res:EventKGSubGraph ;
    beamr:hasRisk res:InformationLeakage ;
    beamr:note res:KGContent .
res:LLMPrompting a beam:Transformation ;
    beam:produce res:EventSummary ;
    beamr:hasRisk res:MaliciousKnowledgeInjection, res:ContextTooLong,
    res:InsufficientInfoForEvent .

# relation between processes
res:EventIDExtraction beam:inform res:KGQuerying .
res:KGQuerying beam:inform res:LLMPrompting .

# each risk control
res:ExclusionOfSensitiveData a beamr:RiskControl ;
    beamr:modifiesRiskConcept res:InformationLeakage ;
    beamr:associatedTo res:EventKG .
res:DataQualityControl a beamr:MaliciousKnowledgeInjection ;
    beamr:modifiesRiskConcept res:InformationLeakage ;
    beamr:associatedTo res:EventKGSubGraph .
res:WarningInFlowEditor a beamr:RiskControl ;
    beamr:modifiesRiskConcept res:ContextTooLong ;
    beamr:associatedTo res:LLMPrompting .

# each Risk
res:InformationLeakage a beamr:Risk ;
    dct:description "Private information stored in the KG may leaked out." .
res:MaliciousKnowledgeInjection a beamr:Risk ;
    dct:description "Malicious Knowledge Injection to produce malicious prompt." ;
    beamr:hasConsequence res:OffensiveBehaviour .
res:ContextTooLong a beamr:Risk ;
    dct:description "The length of the prompt is above the allowed threshold." ;
    beamr:hasConsequence res:CostOverrun .
res:InsufficientInfoForEvent a beamr:Risk ;
    dct:description "The resulted KG subgraph do not contain enough information." ;
    beamr:hasConsequence res:ReputationalDamage .

# each Consequence
res:OffensiveBehaviour a beamr:Consequence ;
    dct:description "the answers from the chatbot contains offensive content." ;
    beamr:hasImpact res:OffendedUser , res:ReputationalDamage.
res:Hallucination a beamr:Consequence ;
    dct:description "the answers from the chatbot deviate from the truth." ;
    beamr:hasImpact res:ReputationalDamage .

# each Impact
res:OffendedUser a beamr:Impact .
res:CostOverrun a beamr:Impact ;
    dct:description "the cost for running the query is higher than ChatBot service fee.".
res:ReputationalDamage a beamr:Impact ;
    dct:description "Reputational damage for the ChatBot provider and its clients.".
```

Figure 4: An excerpt of BEAM ontology instances for simple Graph-RAG based ChatBot system

Illustrative Example Instantiation. One of the industry partners involved in the FAIR-AI project provides chatbot solutions for its customers from a variety of industries. They combine Knowledge Graph-based Question Answering with LLMs in their solutions, particularly for answering questions from documents and other necessary tasks for dialog systems. While the real-world implementation of the chatbot is significantly more complex, in the context of our feasibility evaluation, we defined a simplified use case on retrieval of event summaries consisting of the following components:

- *Event Identification*: The process begins by obtaining the identifier of an event from a natural language question. This identification is achieved using an entity extraction algorithms, which can be replaced with a zero- or few-shot prompt using an LLM.
- *KG Querying (Retrieval)*: Once the id is obtained, relevant information is retrieved through Knowledge Graph (KG) querying on a predefined Events KG. The resulting subgraph is then passed to the next component.
- *Answer Generation*: In the final step, an LLM agent generates the response using context retrieved from the knowledge graph.

We described this simplified use case with BEAM notation in Figure 3. Furthermore, we demonstrate the BEAM capability to link risks with examples identified during our workshop, which assists communications between system developers and non-technical stakeholders in the process.

Furthermore, we represent the use case as instances of the BEAM ontology (cf. Figure 4), allowing for further analysis of the system description with a machine-readable representation. The representation helps stakeholders to answer various queries related to the risk of the systems and mitigation strategies that is (or planned to be) implemented to address the aforementioned risk factors, e.g., *Which Elements contains risks that do not have associated risk control/mitigation mechanisms as part of the system?* (cf. Listing 1) or *Which Elements could potentially have multiple (adverse) impacts to the stakeholders?*

```
PREFIX beam: <http://w3id.org/beam/core#>
PREFIX beamr: <http://w3id.org/beam/risk#>

SELECT ?element ?risk
WHERE {
    ?system a beam:System ; beam:hasElement ?element .
    ?element beamr:hasRisk ?risk .
    OPTIONAL { ?mitigation beamr:modifiesRiskConcept ?risk . }
    FILTER NOT EXISTS { ?mitigation beamr:modifiesRiskConcept ?risk }
}
```

Listing 1: An example SPARQL query to detect unmitigated risk in a system

5. Conclusion and Future Work

This paper aimed to investigate how the boxology notation can be leveraged to support AI system engineering. To this end, we proposed the Boxology Extended Annotation Model (BEAM) approach that extends the boxology notations with additional perspectives, namely system and risks. The BEAM approach consists of visual and semantic notations that enable efficient communication to heterogeneous stakeholders, as well as efficient processing, analysis, and querying of AI system representations. For each additional perspective, we developed the extension through literature studies, building on existing approaches such as SWeMLS, EASY-AI, and AIRO towards workable notations and ontology definition.

We evaluated our approach in two qualitative feasibility studies: First, we conducted an evaluation with students in the context of industry Data Science lab projects; Second, we conducted in-depth modeling workshops with industry partners in the context of a national research project. In both evaluations, we received positive feedback from the users. They acknowledged the value of the BEAM notation and found that BEAM is relatively easy to use, which is an encouraging sign for us to incentivize further development of BEAM.

Future Work. We iteratively develop the BEAM notations based on the feedback from each evaluation to improve its quality and utility. We plan to continuously develop BEAM and (re-)evaluate it in both the research project context as well as in the classroom settings. Furthermore, we identified a number of areas where we plan further extension to BEAM, including (i) two-way transformation between visual notations and ontology instances, (ii) automated extraction of semantic representations from scientific publications, (iii) formal definition of the multi-perspective approach for AI system representation, and (iv) development of a knowledge base for risks and system patterns.

Based on the latter, we will also leverage the semantics for reasoning about risks, for instance to facilitate automated risk identification based on risk templates, rule-based risk propagation, or suggestions of design alternatives and suitable risk mitigations.

In addition to this immediate next steps, we also plan to conduct a more structured evaluation and assess the scalability of the approach in complex scenarios. In such settings, BEAM models may consist of numerous data sets and corpora as well as a vast number of components, parameters and hyperparameters, making manual model design inefficient or even infeasible. To support such scenarios (and more generally foster synchronization between BEAM models and implementations), automated BEAM model construction from code[21] and ideally even roundtrip engineering (e.g., generating BEAM models from code and vice versa) would be highly useful.

Acknowledgement

The BEAM visual notation was initially developed as part of a master thesis work of Bernhard Kollmann [23], co-supervised by Fajar J. Ekaputra and Laura Waltersdorfer. This work was supported by the Austrian Science Fund (FWF) Bilateral AI projects (Grant Nr. 10.55776/COE12) and the Austrian Research Promotion Agency (FFG) FAIR-AI project (Grant Nr. FO999904624).

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. Rebstadt, F. Remark, P. Fukas, P. Meier, O. Thomas, Towards personalized explanations for ai systems: designing a role model for explainable ai in auditing, in: Proceedings of the 17th International Conference on Wirtschaftsinformatik, 2022. URL: https://aisel.aisnet.org/wi2022/ai/ai/2.

[2] F. Königstorfer, S. Thalmann, Ai documentation: A path to accountability, Journal of Responsible Technology 11 (2022) 100043.

[3] F. Van Harmelen, A. Ten Teije, A boxology of design patterns for hybrid learning and reasoning systems, Journal of Web Engineering 18 (2019) 97–123. URL: https://doi.org/10.13052/jwe1540-9589.18133.

[4] M. Van Bekkum, M. De Boer, F. Van Harmelen, A. Meyer-Vitali, A. T. Teije, Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases, Applied Intelligence 51 (2021-09) 6528–6546. URL: https://link.springer.com/10.1007/s10489-021-02394-3. doi:10.1007/s10489-021-02394-3.

[5] A. Breit, L. Waltersdorfer, F. J. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, A. ten Teije, F. van Harmelen, Combining machine learning and semantic web: A systematic mapping study, ACM Computing Surveys (2023). URL: https://doi.org/10.1145/3586163, publisher: ACM.

[6] B. P. Allen, F. Ilievski, Standardizing knowledge engineering practices with a reference architecture, TGDK 2 (2024) 5:1–5:23. URL: https://doi.org/10.4230/TGDK.2.1.5. doi:10.4230/TGDK.2.1.5.

[7] A. Ellis, B. Dave, H. Salehi, S. Ganapathy, C. Shimizu, Easy-ai: semantic and composable glyphs for representing ai systems, in: HHAI 2024: Hybrid Human AI Systems for the Social Good, IOS Press, 2024, pp. 105–113.

[8] F. J. Ekaputra, M. Llugiqi, M. Sabou, A. Ekelhart, H. Paulheim, A. Breit, A. Revenko, L. Waltersdorfer, K. E. Farfar, S. Auer, Describing and organizing semantic web and machine learning systems in the SWeMLS-KG, in: Proceedings of the 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28–June 1, 2023, volume 13870 LNCS, 2023, pp. 372–389. URL: https://doi.org/10.1007/978-3-031-33455-9_22.

[9] D. Golpayegani, H. J. Pandit, D. Lewis, AIRO: an ontology for representing AI risks based on the proposed EU AI act and ISO risk management standards, in: A. Dimou, S. Neumaier, T. Pellegrini, S. Vahdati (Eds.), Towards a Knowledge-Aware AI - SEMANTiCS 2022 - Proceedings of the 18th

[21] e.g., for common machine learning libraries like sk-learn, tensorflow and torch

International Conference on Semantic Systems, 13-15 September 2022, Vienna, Austria, volume 55 of *Studies on the Semantic Web*, IOS Press, 2022, pp. 51–65. URL: https://doi.org/10.3233/SSW220008. doi:10.3233/SSW220008.

[10] M. De Boer, Q. Smit, M. van Bekkum, A. Meyer-Vitali, T. Schmid, Modular design patterns for generative neuro-symbolic systems, in: Joint Proceedings of the ESWC 2024 Workshops and Tutorials co-located with 21th European Semantic Web Conference (ESWC 2024), 2024.

[11] T. Mossakowski, Modular design patterns for neural-symbolic integration: refinement and combination, in: Proceedings of the 16th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 2nd International Joint Conference on Learning & Reasoning (IJCLR), 2022. URL: https://arxiv.org/abs/2206.04724.

[12] A. Ellis, B. Dave, H. Salehi, S. Ganapathy, C. Shimizu, Implementing snoop-ai in comodide, in: NAECON 2024-IEEE National Aerospace and Electronics Conference, IEEE, 2024, pp. 101–104.

[13] P. Slattery, A. K. Saeri, E. A. Grundy, J. Graham, M. Noetel, R. Uuk, J. Dao, S. Pour, S. Casper, N. Thompson, The ai risk repository: A comprehensive meta-review, database, and taxonomy of risks from artificial intelligence, arXiv preprint arXiv:2408.12622 (2024).

[14] N. AI, Artificial intelligence risk management framework (ai rmf 1.0), 2023. URL: https://doi.org/10.6028/NIST.AI.100-1.

[15] Q. Lu, L. Zhu, X. Xu, J. Whittle, Responsible-ai-by-design: A pattern collection for designing responsible artificial intelligence systems, IEEE Software 40 (2023) 63–71. URL: https://api.semanticscholar.org/CorpusID:247218356.

[16] Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Zowghi, A. Jacquet, Responsible ai pattern catalogue: A collection of best practices for ai governance and engineering, ACM Comput. Surv. 56 (2024). URL: https://doi.org/10.1145/3626234. doi:10.1145/3626234.

[17] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, T. Gebru, Model cards for model reporting, in: Proceedings of the conference on fairness, accountability, and transparency, 2019, pp. 220–229.

[18] D. Reisman, J. Schultz, K. Crawford, M. Whittaker, Algorithmic impact assessments: a practical framework for public agency, AI Now 9 (2018).

[19] D. Golpayegani, H. J. Pandit, D. Lewis, To be high-risk, or not to be—semantic specifications and implications of the ai act's high-risk ai applications and harmonised standards, in: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 2023, pp. 905–915.

[20] D. Golpayegani, H. J. Pandit, D. Lewis, Airo: An ontology for representing ai risks based on the proposed eu ai act and iso risk management standards, in: Towards a Knowledge-Aware AI, IOS Press, 2022, pp. 51–65.

[21] G. Booch, The unified modeling language user guide, Pearson Education India, 2005.

[22] K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, J. Zhao, PROV-O: The PROV Ontology, 2013.

[23] B. Kollmann, Towards a Workflow-based AI System Documentation, Master's thesis, WU Wien, Vienna, AT, 2024. URL: https://semantic-systems.org/wp-content/uploads/2024/11/Kollmann.pdf.

[24] M. Al-Emran, V. Mezhuyev, A. Kamaludin, Technology acceptance model in m-learning context: A systematic review, Computers & Education 125 (2018) 389–412.