

Webbasiertes Werkzeug für das modellbasierte Engineering von KI-Anwendungen für Automatisierungssysteme

Marvin Schieseck¹, Philip Topalis¹ und Alexander Fay²

Abstract:

In diesem Beitrag wird ein Konzept vorgestellt, welches es der Industrie erlaubt, eigenständig KI-Anwendungen für Automatisierungssysteme samt Fertigungs- oder Verfahrensprozess zu modellieren und deren Entwicklung, Integration, Dokumentation und Betrieb auf Basis dieses Modells zu koordinieren. Das Konzept basiert auf der Kombination eines grafischen Beschreibungsmittels und einer Entwicklungsmethode, welche in ein webbasiertes Werkzeug zur grafischen Modellierung überführt werden. Anschließend wird zur Evaluation des Werkzeugs ein exemplarischer Anwendungsfall aus der Industrie modelliert. Die Evaluation zeigt den Nutzen des Werkzeugs bei der Koordination von Entwicklung, Integration und Betrieb industrieller KI-Anwendungen auf.

Keywords: KI-Engineering, Web-basierte Werkzeuge, grafische Modellierung

1 Einleitung

In den letzten Jahren wurde die Integration von Künstlicher Intelligenz (KI) in Automatisierungssysteme wie Produktions- und Prozessanlagen seitens der Unternehmen, der Forschung und der Politik vorangetrieben [Bu20; Bu23]. Dabei wurden eine Vielzahl von Anwendungsfällen identifiziert, bei denen der Einsatz von KI entweder die Effizienz bestehender Ansätze übertreffen konnte oder eine Lösung überhaupt erst ermöglichte. Beispiele dafür finden sich im Bereich der Bild- und Signalklassifikation sowie der Textverarbeitung [ISO/IEC TR 24030]. In der Folge konnten durch die Nutzung von KI industrielle Anwendungsfälle der prädiktiven Instandhaltung oder der optischen Qualitätsüberwachungen umgesetzt werden [Di20].

Trotz dieser Potenziale und den vielversprechenden Anwendungsfällen wird KI in der Industrie dennoch vergleichsweise wenig eingesetzt [MG23]. Häufig werden KI-Anwendungen im Rahmen von Forschungs- und Pilotprojekten entwickelt, diese können aber im Anschluss nur bedingt wirtschaftlich in die Praxis überführt werden. Dies kann insbesondere darauf zurückgeführt werden, dass die Integration von KI-Systemen in Automatisierungssysteme im Vergleich zur Integration von KI-Systemen in konventionelle Softwaresysteme mit zusätzlichen Herausforderungen konfrontiert ist. Selbiges gilt neben der Integration auch

¹ Helmut-Schmidt-Universität, Institut für Automatisierungstechnik, Holstenhofweg 85, 22043 Hamburg, Deutschland, {marvin.schieseck, philip.topalis}@hsu-hh.de,

² Ruhr-Universität Bochum, Lehrstuhl für Automatisierungstechnik, Universitätsstraße 150, 44801 Bochum, Deutschland, alexander.fay@rub.de,

für die Entwicklung von KI-Systemen für Automatisierungssysteme. Von Merkel-Kiss und Garrel [MG23] wurden diese Herausforderungen im Rahmen einer quantitativen Metastudie durch Auswertung von 24 nationalen Studien klar herausgestellt. Nach den Autoren sind die größten Herausforderungen, insbesondere für kleine und mittelständische Unternehmen: 1.) *Mangel an Kompetenzen in den Unternehmen und Mangel an Fachkräften*, 2.) *hohe Aufwände für die Konzeptionierung, Entwicklung, Implementierung und den Betrieb sowie damit einhergehend hohe Kosten und lange Amortisationszeiten*, 3.) *komplexe, technischen Infrastrukturen*, 4.) *komplexe Entscheidungsprozesse für die Unternehmen*, 5.) *niedrige Technologiereife und fehlende Standards*.

Um die Entwicklung und Integration von KI-Systemen zu unterstützen, existieren bereits verschiedene Ansätze. Beispielsweise existieren diverse Methoden, die eine zeitliche und logische Abfolge von Handlungen vorschlagen und so den Entwicklungsprozess durch gezielte Festlegung der Entwicklungsschritte unterstützen, wie beispielsweise der *Cross-Industry Standard Process for Data Mining (CRISP-DM)* [WH00], *Data mining methodology for engineering applications (DMME)* [Hu19] oder das *Process Model for AI Systems Engineering (PAISE)* [HS22]. Ein weiterer Ansatz sind grafische Beschreibungen des KI-Systems sowie des technischen Systems [KWS21] [STF23]. Neben Methoden und Beschreibungsmitteln existieren auch Werkzeuge, welche bei der Entwicklung von KI-Systemen unterstützen. Der Großteil dieser KI-Werkzeuge unterstützt den Anwender, indem sie es erleichtern, ein KI-Modell zu erstellen, den Code zu kapseln und zu versionieren und die Ressourcen für das Training bereitzustellen und zu verwalten [Ru21]. Beispiele dafür sind KI-Werkzeuge wie *MLFlow*³, *ClearML*⁴, *Knime*⁵ oder *RapidMiner*⁶, aber auch KI-Plattformen wie *Vertex AI*⁷ von Google, *Azure AI*⁸ von Microsoft oder *SageMaker*⁹ von Amazon. Diese Werkzeuge sind darauf spezialisiert, den Aufbau und das Training von KI-Modellen sowie die damit verbundene Datenvor- und -nachverarbeitung zu erleichtern und zu verwalten [KKH22]. Sie ermöglichen es jedoch nicht, die technische Anlage oder die technischen Prozesse eines Unternehmens konkret zu berücksichtigen und in den Entscheidungs- oder Modellierungsprozess mit einfließen zu lassen. Die bestehenden Werkzeuge nutzen keine für industrielle Anwendungen geeigneten Beschreibungsmittel und unterstützen nicht alle Schritte, die für die industrielle KI-Entwicklung erforderlich sind. Zusammenfassend folgt, dass weder die bestehenden KI-Werkzeuge noch die vorgestellten Methoden noch die Beschreibungsmittel den vorgestellten Herausforderungen in ausreichendem Maße begegnen können, insbesondere den hohen Aufwänden, den langen Amortisationszeiten sowie dem Umgang mit der komplexen technischen Infrastruktur. Keiner der bestehenden Ansätze ist hinreichend geeignet, ein verständliches Konzept einer KI-Anwendung für ein

3 <https://mlflow.org/>

4 <https://clear.ml/>

5 <https://www.knime.com/>

6 <https://docs.rapidminer.com>

7 <https://cloud.google.com/vertex-ai?hl=de>

8 <https://azure.microsoft.com/de-de/solutions/ai>

9 <https://aws.amazon.com/de/sagemaker/>

Automatisierungssystem zu entwickeln, die Entwicklung zu koordinieren, die Integration der KI in das Automatisierungssystem zu ermöglichen und abschließend den Betrieb zu gewährleisten.

Es fehlt ein Lösungsansatz, der es der Industrie erlaubt, eigenständig KI-Anwendungen für Automatisierungssysteme samt Fertigungs- oder Verfahrensprozesses zu modellieren und deren Entwicklung auf Basis des Modells zu koordinieren. Dieser Lösungsansatz muss als industrielle Schnittstelle zwischen den verschiedenen an der Entwicklung beteiligten Domänen geeignet sein und die Integration der bestehenden KI-Werkzeuge in die industrielle Entwicklung von KI ermöglichen. Dieser Handlungsbedarf wird auch ausführlich vom Forschungsbeirat der Plattform Industrie 4.0 [Ma21a] dargelegt.

In diesem Beitrag wird der Handlungsbedarf adressiert und ein Konzept zur Schließung dieser Lücke vorgestellt. Das Konzept basiert auf der Anwendung des Beschreibungsmittels von Schieseck et al. [STF23], welches in ein web-basiertes Werkzeug zur grafischen Modellierung überführt wird, um speziell die industrielle KI-Entwicklung nach DMME zu unterstützen. Es wird dabei nach dem von Schnieder [Sc99] vorgestellten Prinzip der Kombination von Beschreibungsmittel, Methode und Werkzeug vorgegangen. Das Werkzeug wurde mit React Flow¹⁰, einer Bibliothek für interaktive und browserbasierte grafische Modellierung, realisiert¹¹.

2 Stand der Technik und Forschung

In diesem Abschnitt des Beitrags werden verschiedene Ansätze, wie Entwicklungsmethoden, grafische Modellierungssprachen, KI-Werkzeuge, Community-Plattformen, Automated Machine Learning oder Design Patterns vorgestellt. Nachfolgend werden diese Ansätze erläutert und daraus ein gezielter Handlungsbedarf abgeleitet.

2.1 Entwicklungsmethoden

Unter einer Methode wird ein auf einem Regelsystem aufbauendes Verfahren zur Erlangung von theoretischen oder praktischen Ergebnissen verstanden. Im Kontext der Systementwicklung zeigen Methoden auf, wie eine Lösung planmäßig erarbeitet werden kann, welche Entwicklungsschritte in welcher Reihenfolge vollzogen werden sollten und welche Aktivitäten in den jeweiligen Schritten auszuführen sind. [Sc99]

Es existieren Methoden, welche darauf ausgelegt sind, die allgemeine KI-Entwicklung zu unterstützen, wie beispielsweise CRISP-DM [WH00], *Machine Learning Operations* [KKH23] (MLOps) oder *Engineering data driven applications* [He19] (EDDA). Darüber hinaus existieren Methoden, welche speziell für die industrielle KI-Entwicklung entwickelt

¹⁰ <https://reactflow.dev/>

¹¹ <https://github.com/schiesem/ai-modeling-tool>

wurden, wie beispielsweise *Process Model for AI Systems Engineering* PAISE [HS22], *Machine Learning for Production* [Ei22] (ML4P) oder DMME [Hu19].

Als de-facto Standard für allgemeine Data Mining- aber auch KI-Projekte im Allgemeinen hat sich in den letzten zwei Jahrzehnten der *Cross-Industry Standard Process for Data Mining* (CRISP-DM) [WH00] durchgesetzt [Ma21b; VDI/VDE 3714-1]. Das Vorgehen nach CRISP-DM ist in sechs Schritte unterteilt: 1. *Business Understanding*, indem die Geschäfts- und Businessziele definiert werden, 2. *Data Understanding*, indem Hypothesen auf der Grundlage von Erfahrungen und qualifizierten Annahmen mithilfe der Daten definiert werden, 3. *Data Preparation*, indem relevante Daten aufbereitet werden, 4. *Modeling*, indem eine geeignete KI-Methode ausgewählt und ein Modell trainiert wird, 5. *Evaluation*, indem das trainierte Modell mithilfe von Testdaten evaluiert wird und 6. *Deployment*, indem das Modell sowie die Datenvor- und Nachverarbeitung in Betrieb genommen wird.

CRISP-DM bezieht sich auf die Entwicklung von KI-Anwendungen für konventionelle Softwarebereiche und ist nur eingeschränkt für technische Systeme und damit die industrielle KI-Entwicklung geeignet, weil es die Gegebenheiten des technischen Systems und des zugrundeliegenden technischen Prozesses nicht ausreichend berücksichtigt [Hu19; Ma21b]. Deshalb wurde CRISP-DM von Huber et al. [Hu19] für technische Systeme angepasst, durch die Ergänzung von drei Schritten. Diese adaptierte Methode bezeichnen die Autoren als *Data mining methodology for engineering applications* (DMME) (siehe Abb. 1).

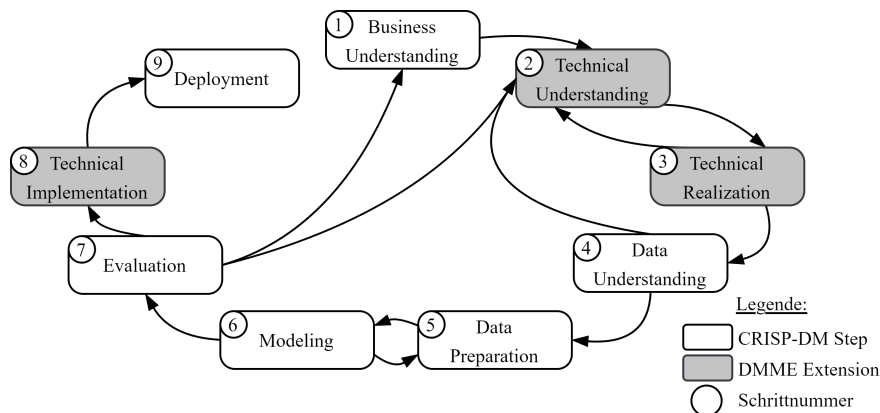


Abb. 1: DMME Methode nach Huber et al. [Hu19]

Zunächst werden nach dem Business Understanding zwei aufeinanderfolgende Schritte, das *Technical Understanding* und die *Technical Realization* ergänzt. Im *Technical Understanding* werden Informationen über die technische Anlage sowie die prozess- und verfahrenstechnischen Zusammenhänge zusammengetragen. Mithilfe dieser Informationen werden aus den Geschäftszielen dann technische Ziele formuliert. Danach werden im Schritt *Technical Realization* verschiedene Konzepte für mögliche KI-Anwendungen entworfen und bewertet. Parallel dazu werden die erforderlichen Daten innerhalb des technischen Systems

aufgenommen und gegebenenfalls Veränderungen der Anlage durchgeführt, beispielsweise das Installieren oder Kalibrieren von Sensorik. Der dritte Schritt, die *Technical Implementation*, wird nach der Evaluation ergänzt. Er dient dazu, das trainierte Modell sowie die dazugehörige Datenversorgung in das technische System zu integrieren. DMME kann als angepasster de-facto Standard für die industrielle KI-Entwicklung angesehen werden und stellt ein Mindestmaß an absolut notwendigen Schritten für die industrielle KI-Entwicklung bereit.

2.2 Grafische Modellierungssprachen und Beschreibungsmittel

Durch grafische Darstellungen kann der Kern eines Problems formal und verständlich für unterschiedliche Beteiligte dargestellt werden [Ha19; VDI/VDE 3681]. Grafische Beschreibungsmittel und Modellierungssprachen ermöglichen die Definition von Symbolen, Regeln und Semantik einer solchen grafischen Darstellung und damit eine verständliche Modellierung für alle beteiligten Experten. Das übergeordnete Ziel einer grafischen Modellierung ist, die Aufgabe und Lösung des Problems möglichst formal abzubilden und dadurch die Lösungsfindung sowie die Kommunikation aller an der Entwicklung beteiligten Akteure zu erleichtern [Sc99].

Grafische Modellierungssprachen werden in allgemeine Modellierungssprachen (General Purpose Modeling Language, GPMLs) und in domänenspezifische Modellierungssprachen (Domain Specific Modelling Languages, DSMLs) unterteilt. GPMLs wie die Unified Modeling Language (UML) oder die Systems Modeling Language (SysML) erlauben es Experten, verschiedene Probleme aus unterschiedlichen technischen Bereichen auf der Grundlage allgemeiner Konzepte wie Klassen, Attribute oder Vererbung zu modellieren. Diese Flexibilität verringert jedoch gleichzeitig auch die Produktivität des Modellierungsprozesses und die Verständlichkeit der grafischen Darstellung, weil die technischen Konzepte auf Domänenebene auf der Grundlage der allgemeinen Konzepte von Grund auf rekonstruiert werden müssen. DSMLs sind weniger flexibel, bieten dafür aber bereits die technischen Konzepte auf Domänenebene, die der Fachsprache des betreffenden Fachbereichs entsprechen. Sie steigern daher die Produktivität und Verständlichkeit der Modellierung. [Fr10; Ro19]

Ein Ansatz speziell für die grafische Beschreibung von KI-Systemen wurde von Kaymakci et al. [KWS21] vorgestellt, welcher auf der Darstellung der Datenquellen, der Datensinken sowie der datenverarbeitenden Elemente beruht. Dieser Ansatz wurde von Schieseck et al. [STF23] aufgegriffen und mit Elementen anderer Beschreibungsmittel wie der formalisierten Prozessbeschreibung [VDI/VDE 3682] kombiniert. Auf dieser Basis wurde ein grafisches Beschreibungsmittel für KI-Systeme innerhalb von Automatisierungssystemen vorgestellt, welches auf drei grundlegenden Elementgruppen basiert: Systemkomponenten, Systemfunktionen und Systemrelationen.

Die Systemkomponenten stellen die Bausteine dar, aus denen sich das technische System zusammensetzt. Sie werden weiter in technische Ressourcen und Produkte unterteilt.

Die technischen Ressourcen, wie beispielsweise Steuerungen, Edge-Geräte, Computer, Cloud, Aktoren und Sensoren führen die Systemfunktionen aus. Die Produkte hingegen sind die Gegenstände der realen Welt, die während des technischen Prozesses umgewandelt werden, beispielsweise Halbzeuge zu Fertigteilen. Systemfunktionen wandeln eine Eingabe in eine entsprechende Ausgabe um und ermöglichen damit die Zielerreichung des technischen Systems. Es werden sieben verschiedene Systemfunktionen definiert: Automatisieren, Transformieren von Produkten, Aufnehmen von Daten, Speichern von Daten, Verarbeiten von Daten, Trainieren und Inferieren. Systemrelationen beschreiben die Verbindungen und Beziehungen zwischen den Systemelementen und erlauben es, die Abhängigkeiten zu modellieren. Es werden drei Kategorien von Systemrelationen definiert: Kommunikation, Zuordnung und Material-/Energiefluss. Eine Kommunikation ist eine informationstechnische Verbindung zwischen zwei Systemkomponenten. Eine Zuordnung ist eine strukturelle Zuweisung von Systemfunktionen zu Systemkomponenten. Ein Material- oder Energiefluss stellt eine Verbindung zwischen Produkten und Transformationsprozessen dar. In Abb. 2 ist die definierte Symbolik der Elemente dargestellt. Die Systemfunktionen werden durch Rechtecke dargestellt. Die technischen Ressourcen werden durch Rechtecke mit abgerundeten Ecken repräsentiert und für die Produkte werden Kreise verwendet. Die Systemrelationen werden durch Linien repräsentiert. Die Zuweisung von Systemkomponenten und Funktionen erfolgt mithilfe einer Strichlinie. Der Materialfluss wird durch eine Punktlinie dargestellt. Die Symbolik und die Semantik für diese Elemente wird aus der VDI 3682 [VDI/VDE 3682] übernommen. Eine Kommunikation wird mithilfe einer Volllinie dargestellt. Zusätzlich haben Schieseck et al. [STF23] ein Meta-Modell vorgeschlagen, um das grafische Beschreibungsmittel in eine vollwertige DSML zu überführen. Durch dieses Meta-Modell werden die Klassenrelationen formal definiert.

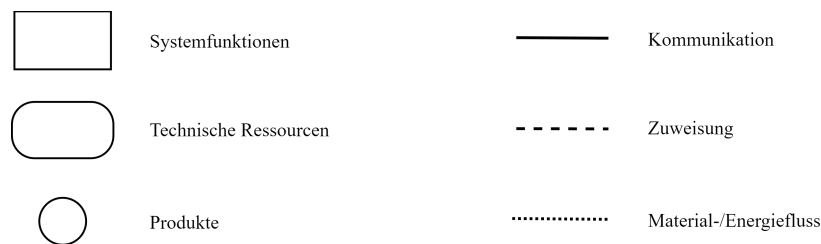


Abb. 2: Symbole des Beschreibungsmittels nach Schieseck et al. [STF23]

In Abb. 3 ist die Modellierung eines exemplarischen KI-Anwendungsfalls zur Überwachung eines Stanzprozesses abgebildet. Bei diesem Stanzprozess wird das Eingangsprodukt Blechplatte durch den Transformationsprozess Stanzen in ein fertiges Fertigteil und Restmaterial umgewandelt. Der Stanzprozess wird durch einen Antriebsmotor ausgeführt. Durch einen rechten und einen linken Lagesensor werden die Positionsdaten der jeweiligen Seite der Hubmechanik erfasst. Diese Daten werden von den Sensoren an die Steuerung kommuniziert. Dort erfolgt auf Basis dieser Werte die Lagesteuerung der Stanze, indem der Antriebsmotor

entsprechende Stellwerte erhält. Gleichzeitig kommuniziert die Steuerung die Positionsdaten in eine Cloud. Dort werden zunächst Merkmale aus den zeitabhängigen Positionsdaten erzeugt und als Merkmalreihe gespeichert. Diese Daten können für das Training des Modells verwendet werden oder für die Ausführung eines bereits trainierten Modells. Das Ausführen und Anwenden eines bereits trainierten Modells unter Verwendung von Produktionsdaten wird in diesem Kontext auch als Inferenz bezeichnet. Weil die Cloud von der Steuerung stetig neue Batches von Daten erhält, können diese kontinuierlich und parallel zum Stanzprozesses in der Cloud ausgewertet werden. Bei entsprechender Eignung der Daten und der Auswahl eines geeigneten Modells könnte über diesen Aufbau zum Beispiel eine KI-Anwendung zur prädiktiven Instandhaltung der Stanze realisiert werden.

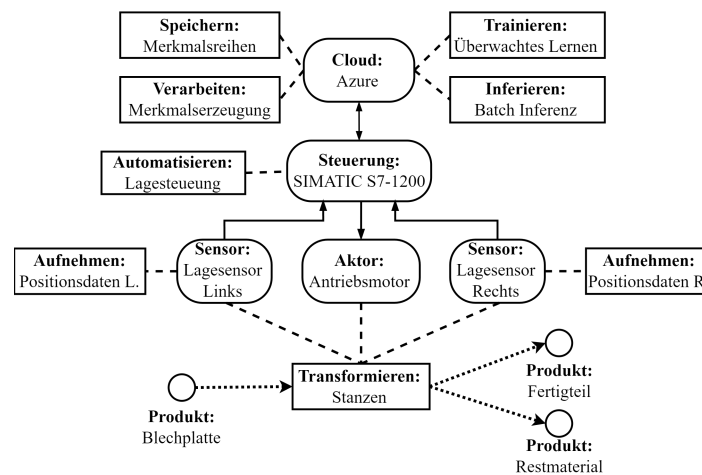


Abb. 3: Modellierung eines exemplarischen Anwendungsfalls mithilfe des Beschreibungsmittels nach Schieseck et al. [STF23]

2.3 KI-Werkzeuge

Ein Werkzeug ist nach Schnieder [Sc99] ein Hilfsmittel, das die Anwendung einer Methode sowie die Verwendung eines Beschreibungsmittels unterstützt, um die Entwicklungsaufwände und damit verbundenen Kosten zu reduzieren. Für die Entwicklung von KI-Anwendungen gibt es bereits eine Vielzahl derartiger Werkzeuge, sowohl kommerziell genutzte als auch open source Ansätze.

Ein bekanntes open source Werkzeug ist *KubeFlow*, welches auf die Orchestrierung von KI-Pipelines unter Verwendung von Kubernetes spezialisiert ist [Ru21]. *KubeFlow* ermöglicht, die Datenvor- und Nachverarbeitung, das Training und das Deployment des ausführbaren Modells zu kapseln und mithilfe einer webbasierten Plattform zu visualisieren. Ein weiteres open source Werkzeug, welches ebenfalls die Orchestrierung von KI-Pipelines

ermöglicht, ist *ClearML*. *ClearML* erlaubt wie *KubeFlow* die Kapselung der Datenvor- und Nachverarbeitung, des Trainings und des Deployments sowie eine webbasierte Visualisierung [CI23]. Neben *KubeFlow* und *ClearML* ist *MLFlow* ein Werkzeug, welches auf die teilautomatisierte Dokumentation und Versionierung des Trainings, der Evaluation und der Datenverarbeitung spezialisiert ist. Dieses Tool stellt ebenfalls eine webbasierte Visualisierung bereit.

Weitere open source Werkzeuge, die nicht webbasiert sind, sondern eine lokale Installation erfordern, sind *RapidMiner* und *Knime*. Beide Werkzeuge sind auf das Data-Mining und auf erweiterte Datenanalyse und spezialisiert und nicht auf die Entwicklung vollständiger KI-Anwendungen. Sie basieren auf einer grafischen Visualisierung und Modellierung von Data-Mining-Pipelines, bestehend aus Datenvor- und -nachverarbeitung und Training sowie Ausführung von Modellen.

Kommerzielle Werkzeuge sind beispielsweise *Azure AI* von Microsoft, *SageMaker* von Amazon oder *VertexAI* von Google. Diese Werkzeuge basieren auf einem nutzungsbasierten Preismodell (Pay-per-Use) und sind ebenfalls webbasiert. Sie erlauben ebenfalls die Orchestrierung von KI-Pipelines wie *KubeFlow* und *ClearML*. Darüber hinaus stellen sie auch teilautomatisierte Dokumentation und Versionierung wie *MLFlow* bereit. Im Gegensatz zu den open source Werkzeugen können die kommerziellen Werkzeuge nicht im gleichen Umfang angepasst und erweitert werden [HM22]. Dafür stellen sie einen umfangreichen Support und flexible Rechenleistungen zur Verfügung. Eine Übersicht über die KI-Werkzeug-Landschaft wird von der *AI Infrastructure Alliance*¹² gegeben und aktuell gehalten.

2.4 Community Plattformen

Community Plattformen ermöglichen einen Zusammenschluss von Nutzern und stellen spezielle Dienste für den Informationsaustausch zwischen den Mitgliedern bereit. Eine verbreitete Community-Plattform im Bereich der KI-Entwicklung stellt *OpenML*¹³ [Va13] dar. Dort können KI-Modelle, Daten sowie Evaluationsergebnisse hochgeladen und zwischen den Nutzern ausgetauscht werden. Zusätzlich stellt *OpenML* verschiedene Bibliotheken, beispielsweise für Python [Fe19], bereit, um die geteilten Inhalte möglichst einfach in individuelle Entwicklungs-Projekte integrieren zu können.

Eine weitere und insbesondere in der wissenschaftlichen Community verbreitete Plattform ist *paperswithcode*¹⁴. Dort werden Veröffentlichungen inklusive des dazugehörigen Codes hochgeladen, damit die entsprechenden Inhalte den Mitgliedern der Community zur Verfügung stehen. Darüber hinaus werden KI-Anwendungen nach verschiedenen

¹² <https://ai-infrastructure.org/>

¹³ <https://www.openml.org/>

¹⁴ <https://paperswithcode.com/>

Anwendungsfällen kategorisiert, beispielsweise Computer Vision, Natural Language Processing, Audio- oder Spracherkennung, um den Mitgliedern der Community die Suche zu erleichtern.

Eine der weltweit größten Community-Plattformen für Data-Science und KI-Anwendungen ist *Kaggle*¹⁵. Dort können ebenfalls Datensätze und Modelle hochgeladen und unter den Mitgliedern ausgetauscht werden. Eine Besonderheit bei Kaggle ist, dass auch Unternehmen eigene Datensätze hochladen können und anschließend einen Wettbewerb mit einer datenspezifischen Aufgabe ausschreiben können. Diese Wettbewerbe sind teilweise mit fünfstelligen Preisgeldern dotiert und jedes Mitglied von Kaggle kann an dem Wettbewerb teilnehmen.

2.5 Automated Machine Learning

Automated Machine Learning (AutoML) zielt darauf ab, Anwendungen für maschinelles Lernen automatisch zu erstellen, ohne dass umfassende Kenntnisse in Statistik und maschinellem Lernen erforderlich sind. Bei AutoML wird versucht, die derzeitige Art der Erstellung von ML-Anwendungen durch Automatisierung zu verbessern. Automatisiert werden vor allem das Auswählen, das Erstellen, das Trainieren und das Optimieren eines ML-Ansatzes sowie die Datenvor- und -nachverarbeitung. Aber auch ML-Experten können von AutoML profitieren, indem mühsame Aufgaben wie die Hyperparameter-Optimierung und Datenverarbeitung automatisiert werden. Fachfremde können von AutoML profitieren, weil sie dazu befähigt werden, selbständig diese Aufgaben zu erfüllen, ohne auf einen KI-Experten angewiesen zu sein. [ZH21]

2.6 Design-Patterns

Design-Patterns (dt. Entwurfsmuster) erfassen bewährte Lösungskonzepte für wiederkehrende Probleme in technischen Disziplinen. In der Softwareentwicklung sind Design-Patterns üblicherweise Entwürfe für bewährte Softwarearchitekturen, welche von Softwareentwicklern als Grundlage für eine eigene Lösung verwendet werden können. [Ga94]

Von Lakshmanan et al. [LRM20] werden derartige Design-Patterns speziell für Machine Learning Anwendungen vorgestellt, welche den Entwicklungsprozess von KI-Anwendungen unterstützen können. Es werden bewährte Entwürfe für die Aufbereitung und Verarbeitung von Daten, für das Training, das Verwalten und das Integrieren eines ML-Modells sowie für das Überwachen einer ML-Anwendung gezeigt. Daran anknüpfend werden von

¹⁵ <https://www.kaggle.com/>

Huyen [Hu22] verschiedene Softwarearchitekturen für KI-Anwendungen beschrieben und diskutiert, beispielsweise Edge-basierte, Cloud-basierte und hybride Architekturen. Diese Architekturen werden auch in der [ISO/IEC 22989] vorgestellt.

2.7 Bewertung und Handlungsbedarf

Schnieder [Sc99] argumentiert, dass der Einsatz von Methoden oder Beschreibungsmitteln ohne unterstützende Werkzeuge nur von begrenztem Nutzen für das industrielle Engineering ist, weil in diesem Fall die Mehraufwände und damit die Mehrkosten steigen. Im Idealfall sollten also für eine wirtschaftliche Entwicklung alle Schritte einer Methode durch Werkzeugen unterstützt werden. Diese Unterstützung kann entweder a.) durch ein Werkzeug für alle Schritte oder b.) durch mehrere untereinander kompatible Werkzeuge für einzelne Schritte erfolgen. Damit die Industrie beispielsweise auf der Basis von DMME eine wirtschaftliche KI-Entwicklung durchführen kann, ist also eine durchgängige Werkzeugunterstützung entweder nach a.) oder b.) wünschenswert. Aus diesem Grund wird nachfolgend bewertet, inwieweit die vorgestellten Ansätze die Industrie bereits in den erforderlichen Schritten nach DMME unterstützen und ob diese Unterstützung ausreichend ist.

Die bestehenden KI-Werkzeuge sind darauf ausgelegt, das Training der Modelle (S6), deren Evaluation (S7), die Datenvor- und Nachverarbeitung (S5) und das Produktivsetzen/Deployment (S9) der trainierten Modelle zu unterstützen sowie die Versionierung der Softwareelemente zu vereinfachen (S9). Beim Datenverständnis (S4) können die Werkzeuge durch statistische Methoden und grafische Darstellungen unterstützen. Die Werkzeuge setzen auf das Bereitstellen und Verwalten von Rechenressourcen und auf web-basierte grafische Oberflächen, um Softwareelemente zu kapseln und miteinander zu verbinden. Es besteht jedoch keine Möglichkeit, zusätzliche Informationen und Randbedingungen des zugrundeliegenden technischen Systems mit einfließen zu lassen. Dadurch können diese Werkzeuge nicht beim Identifizieren von Anwendungsfällen im eigenen Unternehmen (S1), dem Formulieren von technischen Zielen (S2), beim Entwurf von verschiedenen technischen Konzepten und Softwarearchitekturen (S3) sowie der Anpassung des technischen Systems (S3) helfen. Bei der Implementierung der KI-Anwendung in das technische System können die Werkzeuge eingeschränkt unterstützen, weil sie verschiedene Rechenressourcen bereitstellen, welche in die Anlage und das technische System integriert werden können (S8).

Das grafische Beschreibungsmittel nach Schieseck et al. [STF23] ermöglicht durch die definierte Symbolik zusammen mit dem Metamodell eine abstrakte Beschreibung von KI-Anwendungen auf Systemebene durch Darstellung der wechselseitigen Abhängigkeiten zwischen KI, technischen Systemkomponenten und dem unterlagerten Fertigungs- sowie Produktionsprozess. Prinzipiell ist es deshalb in der Lage, beim Identifizieren von Anwendungsfällen (S1) zu unterstützen. Darüber hinaus erlaubt die grafische Darstellung

der technischen Zusammenhänge eine Formulierung von technischen Zielen (S2). Darauf aufbauend können verschiedene technische Konzepte und Softwarearchitekturen entworfen (S3), grafisch dargestellt, verglichen, bewertet und getestet und die technische Anlage dahingehend angepasst werden (S3). Ebenso wird die Koordinierung und Durchführung der technischen Implementierung (S8) unterstützt. Das Beschreibungsmittel hilft nur eingeschränkt beim Datenverständnis, weil es zwar technische Zusammenhänge innerhalb der Anlage aufzeigt, jedoch keine konkrete Untersuchung der Daten durchführt (S4). Weil das Beschreibungsmittel keine algorithmischen Inhalte ausführt, hilft es nicht bei der Produktivsetzung (S9), nicht bei der Datenvorverarbeitung (S5), nicht bei der Modellierung (S6) und nicht bei der Evaluation (S7).

Die vorgestellten Community Plattformen erlauben es, die aufgezeichneten Daten des eigenen Anwendungsfalls innerhalb der Community bereitzustellen. Dadurch können die Mitglieder der Community an dem Anwendungsfall mitarbeiten. Darüber hinaus kann auf bestehende Ansätze und Modelle innerhalb der Community Plattformen zugegriffen werden, um diese für den eigenen Anwendungsfall zu nutzen. Diese Plattformen unterstützen deshalb durch Bereitstellung von Daten und Modellen die Datenvorverarbeitung (S5), das Modellieren (S6) und die Evaluation (S7) und in Teilen auch das Datenverständnis.

AutoML zielt darauf ab, die Datenvorverarbeitung (S5), die Modellierung (S6) und die Evaluation (S7) zu automatisieren, weshalb AutoML auch hauptsächlich in diesen Schritten unterstützt. Viele der bestehenden KI-Werkzeuge haben bereits AutoML-Ansätze integriert. Die bestehenden KI Design-Patterns unterstützen beim Entwurf von KI-Systemen auf abstrakter Systemebene (S3) sowie der späteren Integration (S8), zum Beispiel durch die Bereitstellung von Softwarearchitekturen für industrielle Anwendungsfälle unter Einbezug funktionaler und nichtfunktionaler Anforderungen.

Zusammenfassend zeigt die Bewertung auf, dass eine ausreichende Werkzeugunterstützung für die industrielle KI-Entwicklung bisher nicht für alle Entwicklungsschritte nach DM-ME ausreichend gewährleistet ist. Während die Schritte (S5), (S6) und (S7) bereits von bestehenden KI-Werkzeugen unterstützt werden, existiert für die Schritte (S4) und (S8) keine ausreichende und für die Schritte (S1) (S2) und (S3) gar keine Werkzeugunterstützung. Gleichzeitig sind insbesondere die Schritte (S1) und (S2) für die Initialisierung eines KI-Projekts und der Schritt (S3) für die Konzeptionierung der KI-Anwendung von entscheidender Bedeutung [Hu19]. Darüber hinaus ist der Schritt (S8) für die Integration und die Inbetriebnahme der KI-Anwendung und damit für die Wirtschaftlichkeit der Anwendung entscheidend [Hu19]. Die Bewertung wird in Tabelle 1 zusammengefasst. Zumindest in Teilen lassen sich die von Merkel-Kiss et al. [MG23] aufgezeigten Herausforderungen auf diesen Mangel einer entsprechender Werkzeugunterstützung zurückführen. Eine geeignete Werkzeugunterstützung könnte beispielsweise bei der Initiierung und Koordinierung der Entwicklung einer KI-Anwendung unterstützen und die damit die hohen Aufwände, Kosten und Amortisationszeiten reduzieren sowie die komplexen Entscheidungsprozesse innerhalb der Unternehmen vereinfachen.

Tab. 1: Bewertung des Stands der Technik und Wissenschaft. (BU=Business Understanding, TU=Technical Understanding, TR=Technical Realization, DU=Data Understanding, DP=Data Preparation, M=Modeling, E=Evaluation, TI=Technical Implementation, D=Deployment)

		BU	TU	TR	DU	DP	M	E	TI	D
Ansatz	Schritt	S1	S2	S3	S4	S5	S6	S7	S8	S9
	KI Werkzeuge		○	○	○	◐	●	●	●	◐
Beschreibungsmittel		●	●	●	◐	○	○	○	●	◐
Community Plattformen		○	○	○	◐	●	●	●	○	○
AutoML		○	○	○	○	●	●	●	○	○
KI Design-Patterns		○	○	●	○	○	○	○	●	○
		● Geeignet ◐ Teilweise geeignet ○ Nicht geeignet								

3 Anforderungsanalyse

Die Bewertung des Stands der Technik und Wissenschaft hat gezeigt, dass für die Industrie nicht in allen erforderlichen Entwicklungsschritten nach DMME eine ausreichende Werkzeugunterstützung vorhanden ist. Dies betrifft insbesondere die Schritte (S1), (S2), (S3) und (S8). Gleichzeitig hat die Bewertung auch gezeigt, dass diese Schritte durch Anwendung des grafischen Beschreibungsmittels nach [STF23] sowie durch den Einsatz von KI-Design-Patterns unterstützt werden können. Keiner von beiden Ansätzen wurde jedoch bisher in ein Werkzeug integriert und diese sind daher für die industrielle Praxis gegenwärtig nur eingeschränkt von Nutzen. Deshalb ist ein neues Werkzeug erforderlich. Nachfolgend werden die Anforderungen herausgearbeitet, die ein Werkzeug dafür erfüllen muss:

- A1 *Interaktive und grafische Modellierung*: Das Werkzeug muss eine interaktive und grafische Modellierung ermöglichen. Interaktiv bedeutet, dass der Anwender die grafische Modellierung direkt in der grafischen Darstellung, beispielsweise via Drag-and-Drop der grafischen Symbole, durchführen kann. Dadurch sollen die modellierungsbedingten Mehraufwände und die daraus resultierenden Kosten niedrig gehalten werden.
- A2 *Berücksichtigung der Formalisierung des Beschreibungsmittels*: Das Werkzeug muss auf die Formalisierung des Beschreibungsmittels begrenzt werden können, also eine definierte Menge von Symbolen sowie Regeln für deren zulässige Kombination. Dadurch soll das Werkzeug ausschließlich eine Modellierung mit den Symbolen und Regeln ermöglichen, die durch das Beschreibungsmittel vorgegeben sind. Dadurch soll die Reproduzierbarkeit und die Verständlichkeit der Modellierung gesteigert werden.

- A3 *Serialisierte Repräsentation der grafischen Modellierung*: Das Werkzeug muss eine serialisierte Repräsentation der Modellierung bereitstellen, um die Interoperabilität mit bestehenden domänenspezifischen Werkzeugen ermöglichen. Dadurch soll das Werkzeug eine Möglichkeit bereitstellen, bestehende KI-Werkzeuge und verwandte Ansätze an das Werkzeug anzubinden.
- A4 *Einfacher Zugang zum Werkzeug*: Es muss ein einfacher Zugang zu dem Werkzeug für alle an der Entwicklung beteiligten Akteure bestehen. Dadurch soll die interdisziplinäre Zusammenarbeit gefördert und gleichzeitig die mit der Verwendung des Werkzeugs verbundenen Aufwände und Kosten niedrig gehalten werden.

4 Auswahl von Beschreibungsmittel, Methode und Werkzeug

Im Stand der Technik wurde DMME mit den neun Schritten bereits als angepasster de-facto Standard für die industrielle KI-Entwicklung herausgestellt. DMME stellt ein Mindestmaß an absolut notwendigen Schritten für die industrielle KI-Entwicklung bereit. Ebenfalls wurde dargelegt, dass das Beschreibungsmittel nach Schieseck et al. [STF23] eine geeignete Möglichkeit zur grafischen Modellierung von KI-Anwendungen in Automatisierungssystemen ist. Darüber hinaus ist es in der Lage, eben die Entwicklungsschritte von DMME zu unterstützen, die gegenwärtig noch nicht ausreichend unterstützt werden.

Noch nicht betrachtet wurden, welche verschiedene Möglichkeiten zur praktischen Umsetzung und Ausgestaltung eines Werkzeugs existieren. Anhand der Anforderungen werden deshalb nachfolgend verschiedene Implementierungs- und Realisierungsmöglichkeiten bewertet.

Allgemein kann zwischen lokaler Installation und web-basierter **Ausführung des Werkzeugs** unterschieden werden. In beiden Fällen kann prinzipiell eine interaktive und grafische Modellierung bereitgestellt werden (A1). Auch die Berücksichtigung der Formalisierung (A2) sowie die serialisierte Repräsentation (A3) ist in beiden Fällen möglich. Bei einer web-basierten Ausführung kann der Nutzer mithilfe eines beliebigen Browsers auf das Werkzeug zugreifen. Der Aufwand einer lokalen Installation ist nicht vorhanden (A4). Zusätzlich kann eine web-basierte Ausführung einfach um eine Nutzerverwaltung erweitert werden, sodass mehrere Akteure der unterschiedlichen Domänen gleichzeitig Zugriff auf dieselbe Modellierung erhalten können. Darüber hinaus sind viele der bestehenden KI-Werkzeuge ebenfalls web-basiert, wodurch eine Integration in diese KI-Werkzeuge durch eine web-basierte Lösung im Vergleich zu einer lokalen Installation vereinfacht wird.

Bei der **technischen Umsetzung des Werkzeuges** stehen konfigurierbare Werkzeuge, wie beispielsweise Meta Edit, oder Frameworks, wie beispielsweise ReactFlow oder DiagramJS, zur Verfügung. Eine interaktive und grafische Modellierung kann in beiden Fällen realisiert werden (A1). Auch die Berücksichtigung der Formalisierung ist in beiden Fällen möglich (A2). Die konfigurierbaren Werkzeuge sind jedoch bei der Bereitstellung einer serialisierten Repräsentation eingeschränkt, weil diese Serialisierung dort nicht beliebig gestaltbar ist.

Bei Frameworks hingegen ist die serialisierte Repräsentation beliebig gestaltbar (A3). Ein einfacher Zugang wird von web-basierten Frameworks wie ReactFlow und DiagrammJs ermöglicht. Die konfigurierbaren Werkzeuge verfügen teilweise auch über Dienste, welche einen web-basierten Zugang zur Modellierung ermöglichen, beispielsweise EMF Cloud. Allerdings sind diese Dienste und damit die web-basierte Bereitstellung der Modellierung nicht in vergleichbarem Umfang gestaltbar, wie es bei den Frameworks der Fall ist (A4). Für das Werkzeug können verschiedene **grafische Benutzerschnittstellen** verwendet werden. Die einfachste Schnittstelle stellt hierbei ein Command-Line-Interface dar. Dort erfolgt die Interaktion mit dem Programm durch Befehle innerhalb eines Terminals. Eine nutzerfreundlichere Schnittstelle ist ein Editor-Interface, welches aus Textfeldern und Buttons aufgebaut ist, wie beispielsweise beim *AutomationML Editor*¹⁶. Dort erfolgt die Interaktion mit dem Programm durch das Betätigen von Buttons und das Ausfüllen von Textfeldern. Darüber hinaus besteht auch die Möglichkeit eines grafischen Diagramm-Interfaces, welche sich neben Buttons und Textfeldern vor allem durch eine interaktive Arbeitsfläche auszeichnet, auf der verschiedene grafische Elementen erstellt und durch von Drag-and-Drop-Funktionen verbunden werden können. Beispiele für derartige Diagramm-Interfaces sind MetaEdit, ReactFlow oder das Eclipse Modeling Framework. Eine interaktive, grafische Modellierung im Sinne von (A1) kann nur durch ein Diagramm-Interface gewährleistet werden. Zusammenfassend folgt, dass ein web-basiertes Werkzeug mit Diagramm-Interface unter Zuhilfenahme eines Frameworks eine anforderungsgerechte Lösung darstellt. Dieses Werkzeug soll die Methodenschritte (S1), (S2), (S3) und (S8) von DMME unter Anwendung des Beschreibungsmittels von Schieseck et al. [STF23] unterstützen.

5 Konzept des Werkzeugs und Implementierung

Im vorangegangenen Kapitel wurde herausgearbeitet, wie ein anforderungsgerechtes Werkzeug gestaltet werden kann. Nachfolgend wird ein Konzept für ein derartiges Werkzeug und eine Implementierung vorgestellt. Die Implementierung erfolgt auf Basis von React¹⁷ unter Verwendung des Frameworks React Flow und der JavaScript-Laufzeitumgebung Node.js¹⁸. Ein Einblick in die Oberfläche des Werkzeugs wird in Abb. 4 gegeben.

Die grundlegenden grafischen Elemente sind Funktion, Ressource, Produkt, Produkt-/ und Materialfluss, Zuweisung und Kommunikation. Die Elemente Funktion, Ressource und Produkt kann der Anwender mithilfe von Buttons auf der Oberfläche instanzieren. Den instanziierten Elementen können Schnittstellen hinzugefügt werden, über die mithilfe von Drag-and-Drop dann Systemrelationen, also Zuweisung, Kommunikation oder Produkt-/ und Materialfluss, instanziiert werden können. Beispielsweise wird durch das Verbinden von zwei Systemkomponenten automatisch eine Kommunikationsbeziehung zwischen beiden

¹⁶ <https://www.automationml.org/>

¹⁷ <https://react.dev/>

¹⁸ <https://nodejs.org/en>

Systemkomponenten instanziiert. Jedes grafische Element erhält eine eindeutige ID, welche ebenfalls automatisch erstellt wird. Notwendige Attribute für die Kennzeichnung der verwendeten Elemente werden automatisch in einem Datenmodell gespeichert. Dadurch wird ein konsistentes Datenmodell entsprechend des Meta-Modells von Schieseck et al. [STF23] gewährleistet. Bei der Modellierung werden die Modellierungsvorschriften entsprechend der Formalisierung des Beschreibungsmittels nach Schieseck et al. [STF23] berücksichtigt und Fehlmodellierungen durch die Software abgefangen. Mithilfe eines Drop-Down Menüs können für die Systemkomponenten und die Systemfunktionen die entsprechenden Unterklassen ausgewählt werden, beispielsweise die Unterklasse Sensor oder Aktor für die Klasse Systemkomponente. Zusätzlich können weitere Merkmale wie Namen vergeben werden.

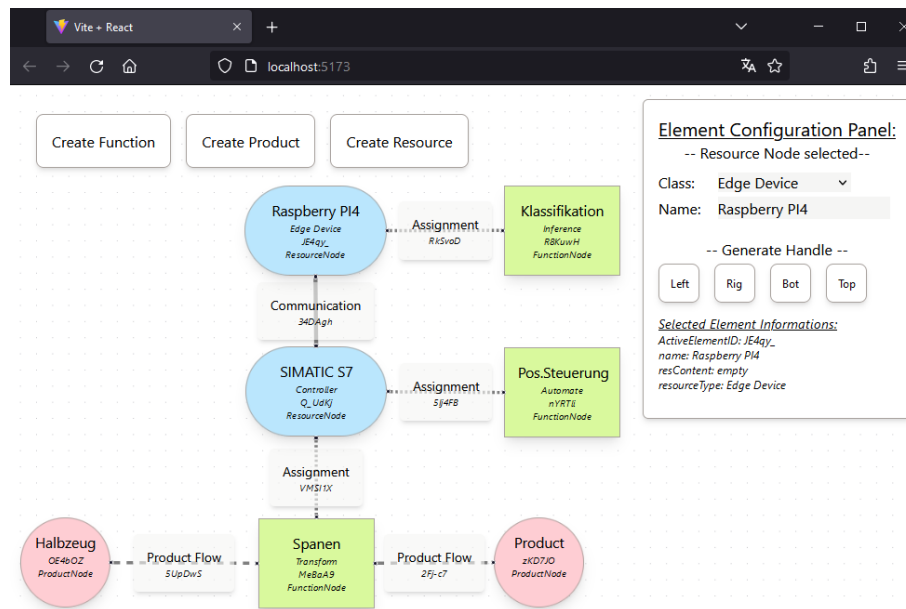


Abb. 4: Grafische Oberfläche des webbasierten Werkzeugs.

Der grafischen Modellierung wird ein JSON-Datenmodell zugrunde gelegt, welches sich schematisch an dem Meta-Modell von Schieseck et al. [STF23] orientiert und parallel während der Modellierung instanziiert wird. In Listing 1 ist ein exemplarischer Auszug des JSON-Datenmodells für die beispielhafte Modellierung aus Abb. 5 dargestellt. Die drei Elemente Funktionen, Produkte und Ressourcen werden in einem Array unter dem Key nodes abgelegt. Die Relationen werden in einem Array unter dem Key edges abgelegt und die Schnittstellen in einem Array unter dem Key handles. Welche Klasse von Element vorliegt, wird mit dem Key type spezifiziert. Die Unterklassen und weitere Attribute jedes Elements werden in einem Array unter dem Key data spezifiziert. Mithilfe weiterer

Keys wie `position`, `width` und `height` werden Informationen zur Positionierung der grafischen Elemente gespeichert. Diese werden jedoch aus Gründen der Übersichtlichkeit hier nicht genauer erläutert und sind in Listing 1 nicht mit aufgeführt. Das instanziierte JSON-Datenmodell kann exportiert werden, um die Modellierung zu speichern, zu teilen oder an andere Werkzeuge anzubinden.

```

{"state": {
  "nodes": [
    {
      "id": "JE4qy_",
      "type": ResourceNode,
      "data" {
        "name": Raspberry PI4,
        "resourceType": Edge Device
      },
      ... ,
    },
    {...},
  ],
  "edges": [
    {
      "id": "34DAgh",
      "type": CommunicationEdge,
      "source": "JE4qy_",
      "sourceHandle": "oPrLTc",
      "target": "Q_Udkj",
      "targetHandle": "uSziUr",
      ... ,
    },
    {...},
  ],
  handles: [...],
}
}

```

List. 1: Auszug des JSON-Modells

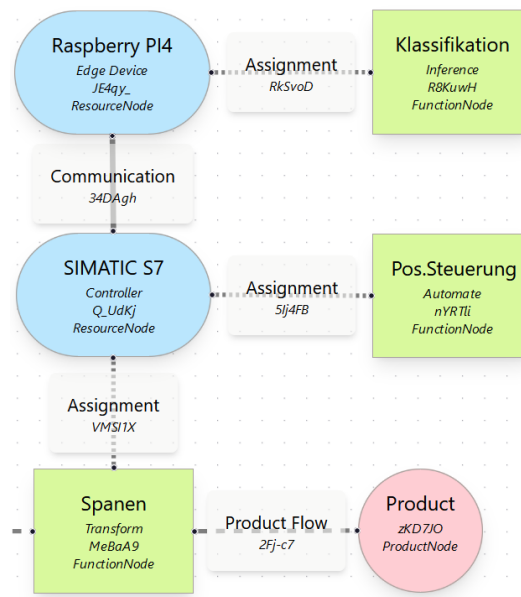


Abb. 5: Beispielhafte Modellierung mit dem Werkzeug

6 Evaluation am Fallbeispiel

Das Werkzeug wird zur Evaluation auf einen realen Anwendungsfall aus der Industrie angewendet. Bei dem Anwendungsfall handelt es sich um eine KI-Anwendung innerhalb einer technischen Anlage zur Überwachung der Produktqualität. Dieser Anwendungsfall wird im Rahmen des Forschungsprojekts *Engineering von künstlichen Intelligenzen*¹⁹ (EKI) umgesetzt und die Anwendung des Werkzeugs erfolgt in Zusammenarbeit mit einem Partner aus der Industrie.

Beschreibung des Anwendungsfalls: In dem Anwendungsfall führt eine technische Anlage

¹⁹ <https://dtecbw.de/home/forschung/hsu/projekt-eki>

einen Fertigungsprozess aus, bei dem ein Schaum aus Polyurethan (PUR) auf den Rand einer Glasscheibe aufgetragen wird. Bevor das PUR aufgetragen werden kann, muss ein Primer auf die Glasscheibe aufgetragen werden. Dieser dient als Haftvermittler zwischen dem PUR und der Scheibe. Der Primerauftrag erfolgt in einer Fertigungszelle bestehend aus einem Roboter, einer Auftragseinheit und einer Kamera. Die Auftragseinheit besteht aus einem Schwamm für den Auftrag des Primers und einer Dosiereinheit zur Befeuchtung des Schwammes. Um die Haftung des PUR auf der Glasscheibe zu gewährleisten, muss die Auftragsfläche des Primers vollständig und ohne Fehlstellen benetzt sein. Ansonsten besteht die Gefahr, dass sich das PUR von der Glasscheibe löst. Das Ziel der KI-Anwendung ist es deshalb, den Primerauftrag zu überwachen und dadurch die Produktqualität der mit Polyurethan (PUR) umschäumten Glasscheibe zu gewährleisten.

Beschreibung des Fertigungsprozesses und der Modellierung: Zu Beginn des Auftragsprozesses nimmt ein Roboter die Scheibe auf und führt diese an dem Schwamm der Auftragseinheit entlang. Jede Scheibe besitzt einen eigenen RFID-Tag und kann so eindeutig zugeordnet werden. Hinter der Auftragseinheit ist eine Kamera montiert, mit deren Hilfe ein Bildstrom des Auftragsprozesses aufgenommen wird. Die Robotersteuerung übernimmt die Steuerung des Roboters und der Dosiereinheit. Die Auswertung der Bilddaten erfolgt auf einem Edge-Gerät mithilfe eines KI-Modells. Wenn die Auswertung der Bilddaten eine Fehlstelle identifiziert, dann erfolgt ein erneuter, punktueller Primerauftrag an der entsprechenden Fehlstelle. Da es sich hierbei um eine zeitkritische Auswertung der Daten bezogen auf den Produktionszyklus handelt, ist es sinnvoll, die Inferenz des KI-Modells nahe am Prozess auf dem Edge-Steuergerät auszuführen. Bevor das KI-Modell ausgeführt werden kann, ist allerdings ein rechenintensives Training des Modells erforderlich. Dieses Training kann nicht auf dem Edge-Gerät durchgeführt werden, weil es nicht über ausreichend Rechenressourcen verfügt. Deshalb werden die aufgenommenen Bilder über das Edge-Steuergerät ebenfalls in die Cloud übertragen und dort vorverarbeitet und gespeichert. Das Training des Modells erfolgt dann in der Cloud, weil dort flexible Rechenressourcen verfügbar sind. Anschließend werden das trainierte Modell und die Datenvorverarbeitung auf das Edge-Device für die Inferenz verschoben und können dort ausgeführt werden. Der beschriebene Anwendungsfall wurde mithilfe des Werkzeugs gemeinsam mit dem Industriepartner modelliert, das Ergebnis ist in Abb. 6 dargestellt.

Durchführung der Modellierung und des Entwicklungsprojekts: Der Industriepartner konnte das Werkzeug nutzen, um zunächst seinen Fertigungsprozess sowie die dazugehörigen Produkte darzustellen. Anschließend konnte er die am Fertigungsprozess beteiligten technischen Systemkomponenten sowie deren Relationen einzeichnen. Dadurch konnte sich der Industriepartner einen Überblick über das technische System verschaffen und zur Koordination und Initiierung des Entwicklungsprojektes nutzen. Der Industriepartner hat darauf aufbauend weitere Akteure wie Softwareentwickler und KI-Experten gezielt in die Konzeptionierung und Entwicklung eingebunden. So konnte in domänenübergreifender Zusammenarbeit mithilfe des Modells geklärt werden, an welchen Stellen und aus welchen Gründen welche Funktionen und Softwarekomponenten sinnvoll sind. Beispielsweise, dass

die Inferenz in diesem Anwendungsfall auf einem Edge-Gerät ausgeführt werden sollte, gleichzeitig das Training allerdings in der Cloud erfolgen muss. Anschließend konnten die jeweiligen Experten der Domänen ihre Entwicklungsaufgaben mit domänenspezifischen Werkzeugen fortführen. Beispielsweise haben die KI-Experten ihre Modelle unter Verwendung von MLFlow entwickelt und evaluiert. Die Automatisierungstechniker haben die Pfadsteuerung des Roboters innerhalb eines domänenspezifischen Steuerungstools angepasst. In regelmäßigen Terminen koordiniert durch das Industrieunternehmen war es so möglich, dass auf Basis der Modellierung verschiedene Experten parallel an der Umsetzung der jeweiligen Funktionen arbeiteten und die KI-Anwendung erfolgreich entwickelt wurde.

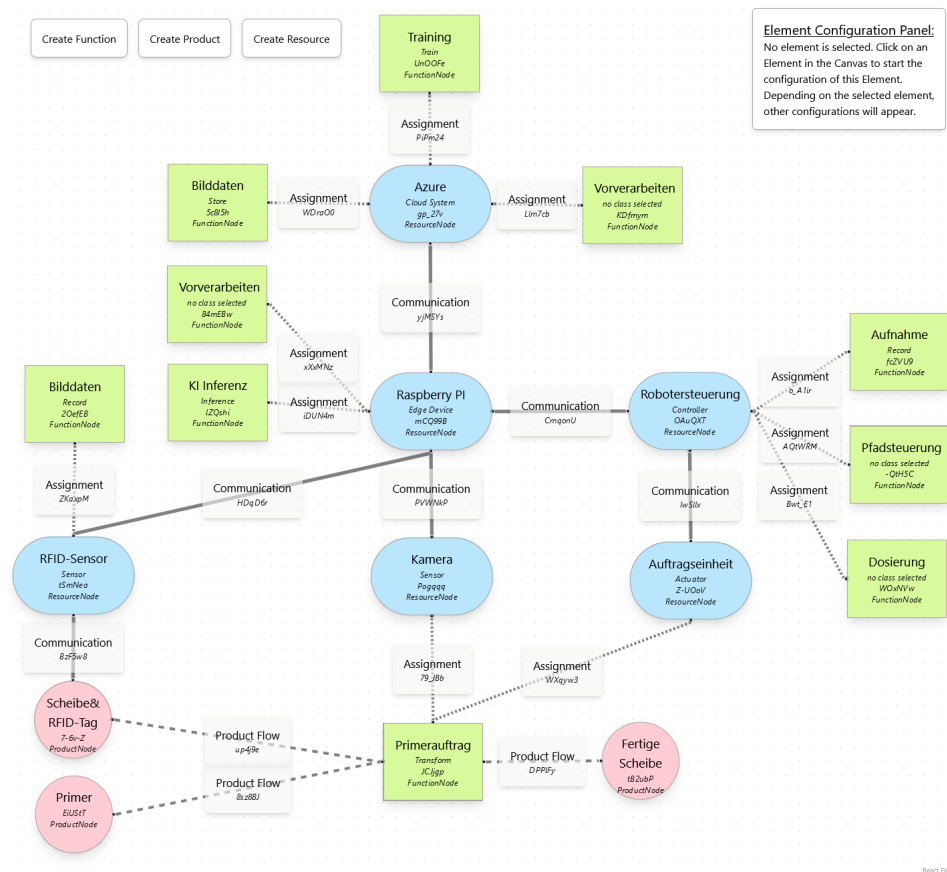


Abb. 6: Modellierung des Anwendungsfalls aus dem Evaluationsbeispiel

7 Diskussion

Auf Basis der Evaluation am Fallbeispiel wird diskutiert, ob das Werkzeug die gestellten Anforderungen A1 bis A4 erfüllt und welchen Beitrag das Werkzeug zum Lösen des Handlungsbedarfs leistet.

Das Werkzeug ermöglicht dem Anwender eine interaktive grafische Modellierung durch Buttons, Drop-Down-Menüs, Eingabefelder und Drag-and-Drop-Funktionen (A1).

Die Modellelemente und Verbindungen sind auf die von Schieseck et al. [STF23] vorgegebenen Elemente beschränkt. Zusätzlich können nur Verbindungen oder Elemente erstellt werden, welche dem Meta-Modell und damit Formalisierung entsprechen. Davon abweichende und fehlerhafte Verbindungen und Elemente werden automatisch vom Werkzeug abgefangen und verhindert (A2).

Während der Modellierung wird automatisch ein Datenmodell im JSON-Format instanziiert und dadurch eine serialisierte Repräsentation der Modellierung erstellt (A3). Das Datenmodell ermöglicht es, die Modellierungsinformationen zu speichern und zwischen verschiedenen Entwicklern und Werkzeugen auszutauschen. Innerhalb bestehenden KI-Werkzeuge wie *KubeFlow* könnten Schnittstellen geschaffen oder angepasst werden, um die Informationen im JSON-Format einzulesen. Der KI-Entwickler würde dann innerhalb von *KubeFlow* über Informationen der geplanten Hardware- und Kommunikationsbedingungen verfügen und könnte diese bei der Entwicklung und Auswahl des KI-Algorithmus berücksichtigen. Beispielsweise könnte es sein, dass die Inferenz in einem Anwendungsfall auf einem Edge Gerät ausgeführt werden muss. Durch den Import könnten die Kennwerte der Hardware und damit etwaige Einschränkungen hinsichtlich der Rechenleistung innerhalb von *KubeFlow* berücksichtigt und für Simulationen genutzt werden.

Über einen Webbrowser haben verschiedene Entwickler die Möglichkeit, auf das Werkzeug zuzugreifen und die grafische Modellierung einzusehen (A4).

Die Evaluation hat gezeigt, dass das Modell und das Werkzeug als Grundlage für die Entwicklung der einzelnen Funktionen und Softwarekomponenten sowie für die Koordination des Projektes verwendet werden kann. Das Werkzeug und die Modellierung können vor allem dabei unterstützen, den Anwendungsfall und das Geschäfts- sowie das technische Ziel zu beschreiben (S1), verständlich darzustellen (S2), ein domänenübergreifendes ein Lösungskonzept zu erarbeiten (S3) und dieses zu integrieren (S8). Durch die webbasierte Bereitstellung des Werkzeugs sowie den interaktiven grafischen Modellierungsansatz waren die Mehraufwände, die durch die Verwendung des Werkzeugs entstanden sind, gering. Insofern wird das Werkzeug als geeigneter Ansatz bewertet, um die Industrie in den bisher nicht ausreichend werkzeuggestützten Entwicklungsschritten der industriellen KI-Entwicklung zu unterstützen. Die Evaluation erfolgte nur an einem Anwendungsfall mit nur einem Industriepartner. Für eine präzisere Evaluierung des Werkzeugs und dessen Nutzens müssen weitere KI-Entwicklungsprojekte von weiteren Industrieunternehmen durchgeführt werden.

8 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein webbasiertes Werkzeug zur Modellierung von KI-Anwendungen für Automatisierungssysteme vorgestellt. Es erlaubt der Industrie eine interaktive, grafische Modellierung von KI-Anwendungen auf Basis eines Beschreibungsmittels für KI-Anwendungen. Die grafische Modellierung besteht aus Systemkomponenten, Systemfunktionen und Systemrelationen, welche durch definierte Symbole dargestellt werden, sowie Regeln zu deren Kombination. Durch die webbasierte Implementierung des Werkzeugs und die interaktive Modellierung wird der zusätzliche Aufwand durch die Modellierung selbst gering gehalten. Das Modell kann in einer JSON-Datei gespeichert werden und dadurch zwischen Entwicklern und Werkzeugen ausgetauscht werden. Durch das Werkzeug werden insbesondere die Schritte der industriellen KI-Entwicklung unterstützt, die durch bestehende KI-Werkzeuge gegenwärtig nicht ausreichend oder gar nicht unterstützt werden. Dadurch wird die Initiierung und Koordination der Entwicklung von KI-Anwendungen aus der Industrie heraus gefördert.

Das Werkzeug stellt gegenwärtig noch ein Funktionsmuster dar und ist ein erster Schritt, um das modellbasierte Engineering von KI-Anwendungen zu ermöglichen. Für die Zukunft ist geplant, das Werkzeug zu erweitern und dem Anwender die Möglichkeit zu bieten, weitere Elemente und Attribute flexibel hinzuzufügen, beispielsweise durch Angabe des Kommunikationstyps oder der Kommunikationsgeschwindigkeit. Zusätzlich ist geplant, das Werkzeug um ein Expertensystem zu erweitern, welches dem Anwender parallel zur Modellierung Hinweise gibt, indem es beispielsweise geeignete Design-Patterns für den aktuellen Anwendungsfall vorschlägt. Außerdem ist für die Zukunft geplant, die Evaluation auszuweiten. Einerseits durch die Modellierung von weiteren realen Anwendungsfällen durch verschiedene Industrieunternehmen und andererseits durch die Durchführung von Umfragen unter Fachleuten.

Acknowledgements

Diese Forschung wird von dtec.bw - Forschungszentrum Digitalisierung und Technik der Bundeswehr gefördert. dtec.bw wird von der Europäischen Union - NextGenerationEU gefördert.

Literatur

- [Bu20] Bundesministerium für Wirtschaft und Klimaschutz (BMWK): Strategie Künstliche Intelligenz der Bundesregierung, 2020.
- [Bu23] Bundesministerium für Bildung und Forschung (BMBF): Aktionsplan Künstliche Intelligenz, 2023.

- [CI23] ClearML: ClearML - Your entire MLOps stack in one open-source tool, Software available from <http://github.com/allegroai/clearml>, 2023, URL: <https://clear.ml/>, Stand: 24.01.2024.
- [Di20] Diemer, J.; Elmer, S.; Gaertler, M.; Gamer, T.; Görg, C.; Grotepass, J.; Kalhoff, J.; Kramer, S.; Legat, C.; Meyer-Kahlen, J.-P.; Nettsträter, A.; Niehörster, O.; Schmidt, B.; Schweichhart, K.; Ulrich, M.; Weitschat, R.; Winter, J.: KI in der Industrie 4.0: Orientierung, Anwendungsbeispiele, Handlungsempfehlungen, Techn. Ber., Plattform Industrie 4.0, 2020.
- [Ei22] Eisenträger, M.; Frey, C.; Herzog, A.; Moghiseh, A.; Morand, L.; Pfrommer, J.; Stephani, H.; Stoll, A.; Wessels, L.: ML4P-Vorgehensmodell Machine Learning for Production, 2022.
- [Fe19] Feurer, M.; van Rijn, J. N.; Kadra, A.; Gijsbers, P.; Mallik, N.; Ravi, S.; Mueller, A.; Vanschoren, J.; Hutter, F.: OpenML-Python: an extensible Python API for OpenML. arXiv 1911.02490/, 2019, URL: <https://arxiv.org/pdf/1911.02490.pdf>.
- [Fr10] Frank, U.: Outline of a method for designing domain-specific modelling languages, Techn. Ber., ICB-research report, 2010.
- [Ga94] Gama, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design patterns, elements of reusable object-oriented software. Addison-Wesley, 1994.
- [Ha19] Habermann, R.; Nagel, P.; Becker, M.; Büchel, A.; von Massow, H.: Systems engineering. Springer, 2019.
- [He19] Hesenius, M.; Schwenzfeier, N.; Meyer, O.; Koop, W.; Gruhn, V.: Towards a software engineering process for developing data-driven applications. In: 2019 IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE). IEEE, S. 35–41, 2019.
- [HM22] Hewage, N.; Meedeniya, D.: Machine Learning Operations: A Survey on MLOps Tool Support. arXiv 2022. arXiv preprint arXiv:2202.10169/, 2022.
- [HS22] Hasterok, C.; Stompe, J.: PAISE®–process model for AI systems engineering. at-Automatisierungstechnik 70/9, S. 777–786, 2022.
- [Hu19] Huber, S.; Wiemer, H.; Schneider, D.; Ihlenfeldt, S.: DMME: Data mining methodology for engineering applications—a holistic extension to the CRISP-DM model. Procedia CIRP 79/, S. 403–408, 2019.

- [Hu22] Huyen, C.: Designing machine learning systems. O'Reilly Media, 2022.
- [ISO/IEC 22989] International Organization for Standardization (ISO): Artificial Intelligence concepts and terminology, ISO/IEC TR 22989, 2022.
- [ISO/IEC TR 24030] International Organization for Standardization (ISO): Information technology — Artificial intelligence (AI) — Use cases, ISO/IEC TR 24030, 2021.
- [KKH22] Kreuzberger, D.; Kühl, N.; Hirschl, S.: Machine Learning Operations (MLOps): Overview, Definition, and Architecture. arXiv 20222205/, 2022.
- [KKH23] Kreuzberger, D.; Kühl, N.; Hirschl, S.: Machine learning operations (mlops): Overview, definition, and architecture. IEEE Access/, 2023.
- [KWS21] Kaymakci, C.; Wenninger, S.; Sauer, A.: A Holistic Framework for AI Systems in Industrial Applications. In: Innovation Through Information Systems. Springer International Publishing, 2021.
- [LRM20] Lakshmanan, V.; Robinson, S.; Munn, M.: Machine learning design patterns. O'Reilly Media, 2020.
- [Ma21a] Manoury, M.; Brünnhäuser, J.; Geiger, A.; Lindow, K.; Stark, R.: Modellierungs- und Simulationsbedarfe der intelligenten Fabrik, Techn. Ber., Forschungsbeirat der Plattform Industrie 4.0 / acatech – Deutsche Akademie der Technikwissenschaften, 2021.
- [Ma21b] Martínez-Plumed, F.; Contreras-Ochando, L.; Ferri, C.; Orallo, J. H.; Kull, M.; Lachiche, N.; Quintana, M. J. R.; Flach, P. A.: CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. IEEE Transactions on Knowledge and Data Engineering 33/, S. 3048–3061, 2021, URL: <https://api.semanticscholar.org/CorpusID:212932421>.
- [MG23] Merkel-Kiss, M.; Garrel, J.: Systematische Literaturanalyse zum KI-Einsatz und KI-basierten Geschäftsmodellen in produzierenden kleinen und mittleren Unternehmen. Zeitschrift für Arbeitswissenschaft 77/3, S. 453–468, 2023.
- [Ro19] Roy Chaudhuri, S.; Natarajan, S.; Banerjee, A.; Choppella, V.: Methodology to develop domain specific modeling languages. In: Proceedings of the 17th ACM SIGPLAN International Workshop on Domain-Specific Modeling. S. 1–10, 2019.
- [Ru21] Ruf, P.; Madan, M.; Reich, C.; Ould-Abdeslam, D.: Demystifying mlops and presenting a recipe for the selection of open-source tools. Applied Sciences 11/19, S. 8861, 2021.

- [Sc99] Schnieder, E.: Methoden der Automatisierung. Vieweg+Teubner Verlag, Wiesbaden, 1999.
- [STF23] Schieseck, M.; Topalis, P.; Fay, A.: A Graphical Modeling Language for Artificial Intelligence Applications in Automation Systems. In: 2023 IEEE 21st International Conference on Industrial Informatics (INDIN). Lemgo, DE, 2023.
- [Va13] Vanschoren, J.; van Rijn, J.N.; Bischl, B.; Torgo, L.: OpenML: Networked Science in Machine Learning. SIGKDD Explorations 15/2, S. 49–60, 2013, URL: <http://doi.acm.org/10.1145/2641190.2641198>.
- [VDI/VDE 3681] Verein Deutscher Ingenieure e.V. (VDI): Einordnung und Bewertung der Beschreibungsmittel aus der Automatisierungstechnik, VDI/VDE 3681, 2022.
- [VDI/VDE 3682] Verein Deutscher Ingenieure e.V. (VDI): Formalisierte Prozessbeschreibungen, VDI/VDE 3682, 2015.
- [VDI/VDE 3714-1] Verein Deutscher Ingenieure e.V. (VDI): Implementation and operation of Big Data application in the manufacturing industry - Implementation of Big Data projects, VDI/VDE 3714-1, 2022.
- [WH00] Wirth, R.; Hipp, J.: CRISP-DM: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. Bd. 1, S. 29–39, 2000.
- [ZH21] Zöllner, M.-A.; Huber, M. F.: Benchmark and survey of automated machine learning frameworks. Journal of artificial intelligence research 70/, S. 409–472, 2021.