

Systemmodelle aus Texten extrahieren

Warum KI-Chat nicht genug ist

Dr. Michael Jastram
Formal Mind GmbH



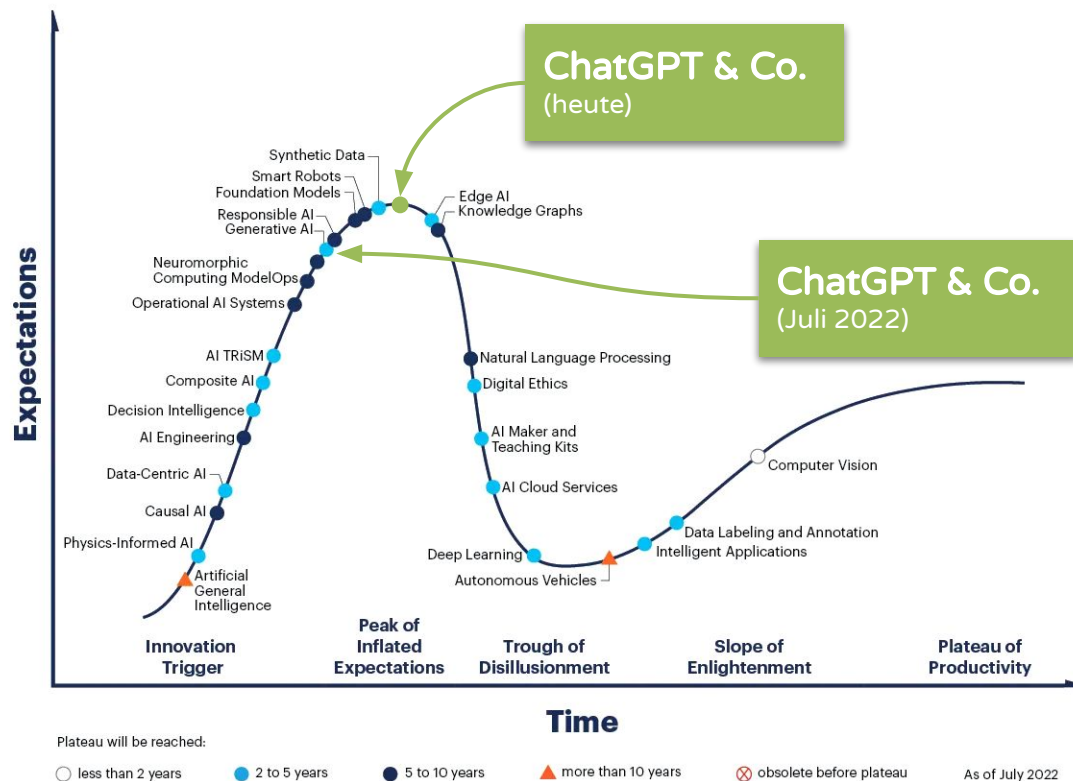
ChatGPT

HYPE ODER NUTZEN





Hype Cycle for Artificial Intelligence, 2022



gartner.com

Source: Gartner
 © 2022 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner and Hype Cycle are registered trademarks of Gartner, Inc. and its affiliates in the U.S. 1957302

Gartner®



Agenda

- Motivation & Scope
- KI im Systems Engineering heute
- Modellierung: Was und warum?
- 4 Thesen zu KI + MBSE
- Die Zukunft
- Ausblick & Fragen

Zur Person: Michael Jastram



- Promotion 2012 zum Thema Anforderungsmodellierung
- Selbstständig seit 2011 (Formal Mind GmbH): Systems Engineering
- Artikelreihe im HOOD-Blog: SEcKI – Systems Engineering mit KI

Systems Engineering Trends

www.se-trends.de



Jetzt anmelden

Semiant AI Quality Assistant

www.semiant.com




Jetzt anmelden

Warum KI in der Produktentwicklung?




KI im Produkt wird heute nicht behandelt


Engineers develop complex products 10x faster with Trace.Space

Requirements management for the next generation 

 Follow us!

 Careers

 Try the API

 Early access

Introducing the ValiAssistant

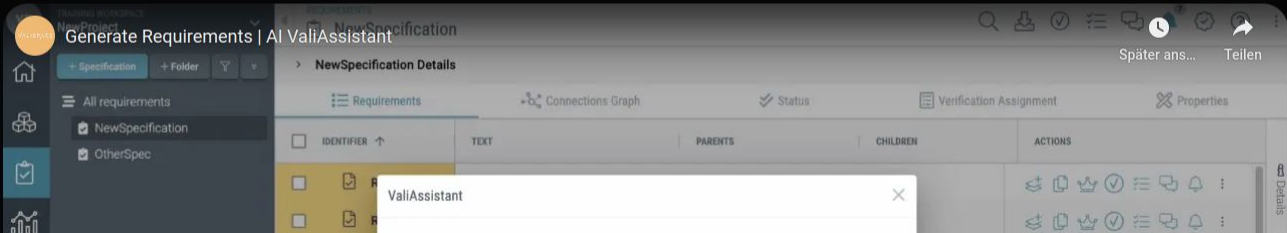
Access the limitless power of AI, right inside Valispace.

Develop great products. Faster.

Try It Free

Generate Requirements

From 0 to a specification in no time.





Warum KI* in der Produktentwicklung?

Um Komplexität zu beherrschen

*und MBSE

<https://www.cutter.com/article/a's-role-accelerating-product-development>



KI im Systems Engineering heute

KI kann...

Automatisieren

- Effizienz erhöhen
- Sicherheit (für Menschen) erhöhen
- Mühsame Aufgaben reduzieren
- Ermöglichung neuer Möglichkeiten

Anreichern

- Motivation steigern
- Mehr Kontrolle bei der Automatisierung ermöglichen
- Erfolg erhöhen und dabei die Verantwortung beim Nutzer belassen
- Die Leistung der Nutzer skalieren
- Kreativität erhöhen

Beispiel 1: Prüfung der Anforderungsqualität

reQlab

Requirements

This document is used to present the current state of reQlab:

- The system reQlab shall define five states of quality for requirements.
- Is no modal verb used, the system can warn the user. is no modal verb used, the system shall warn the user by visual features.
- Is no modal verb used, the system shall warn the user by visual features.
- Encapsulated sentences shall be avoided Multiple sentences in one requirement shall be marked as wrong by the system, if multiple sentences are used, the system shall provide a help text.
- reQlab shall mark multiple sentences as wrong.
- if multiple sentences are used, the system shall provide a help text.
- if the user uses weak words, reQlab shall provide a good help text.
- if the user uses "Weak Words", reQlab shall provide the help text according to the a
- reQlab shall warn the user if a condition is at the end of the sentence.
- if a condition is at the end of the sentence, reQlab shall warn the user.
- The user shall have the ability to identify his failure.
- reQlab shall provide the user the ability to identify his failure.
- if the ending of a sentence is wrong, shall the system identify this?
- if the ending of a sentence is a question mark, reQlab shall identify this.
- reQlab shall classify multiple requirements fast and check if they are well written ac
- reQlab shall classify multiple requirements.
- reQlab shall check the quality of requirements according to the IEEE definition.
- Ich moechte ein rotes Auto kaufen, Dieses auto muss schnell fahren.
- Das System moechte ein rotes Auto bestellen.
- This is a new requirement.
- This is a checked Requirement
- The system shall be super cool!
- reQlab shall classify this requirement.
- This is a test Requirement!
- The system shall open the window.
- Das System muss das Fenster öffnen.
- Das System kann vieles gut beantworten?
- The system shall classify requirements.
- This is a Sentence. Please check it.
- Das ist ein Test.
- Das System reQlab muss Anforderungen klassifizieren.
- Nach dem Abschließen muss das System herunterfahren.
- Das ist eine neue Anforderung

Work Item Properties

ReqLab-931 - reQlab shall classify multiple requirements fast and check if they are well writ...

properties

Severity: **Must Have**

Status: **Draft**

Requirement: reQlab shall classify multiple requirements fast and check if they are well written according to the IEEE definition

Rating: 3

Result: reQlab found multiple possible improvements.

- 1. Outside of conditional sentences, the use of "and" / "or" often means that a sentence should be separated into two distinct requirements.
- 2. Conditions should be expressed at the beginning of the sentence.
- 3. You should avoid imprecise or ambiguous terms.

reQlab (Polarion)

Jama Requirements Optimizer

jama software

LABS

SHARE CONTACT

Your Requirements Analysis

Your overall score: **90%**

Analysis summary:

- 80 rows reported from Jama here
- 70 rows recognized as valid requirements and analyzed against 13 RCOSE rules
- 10 rows marked as invalid and ignored

Tips to improve your requirements:

- Leverage guidelines from RCOSE and EASE to increase accuracy
- Add traceability, real-time collaboration, forced review, test management, hazard analysis and more with Jama Behaviors

The following requirements had an analysis score <= 60%.

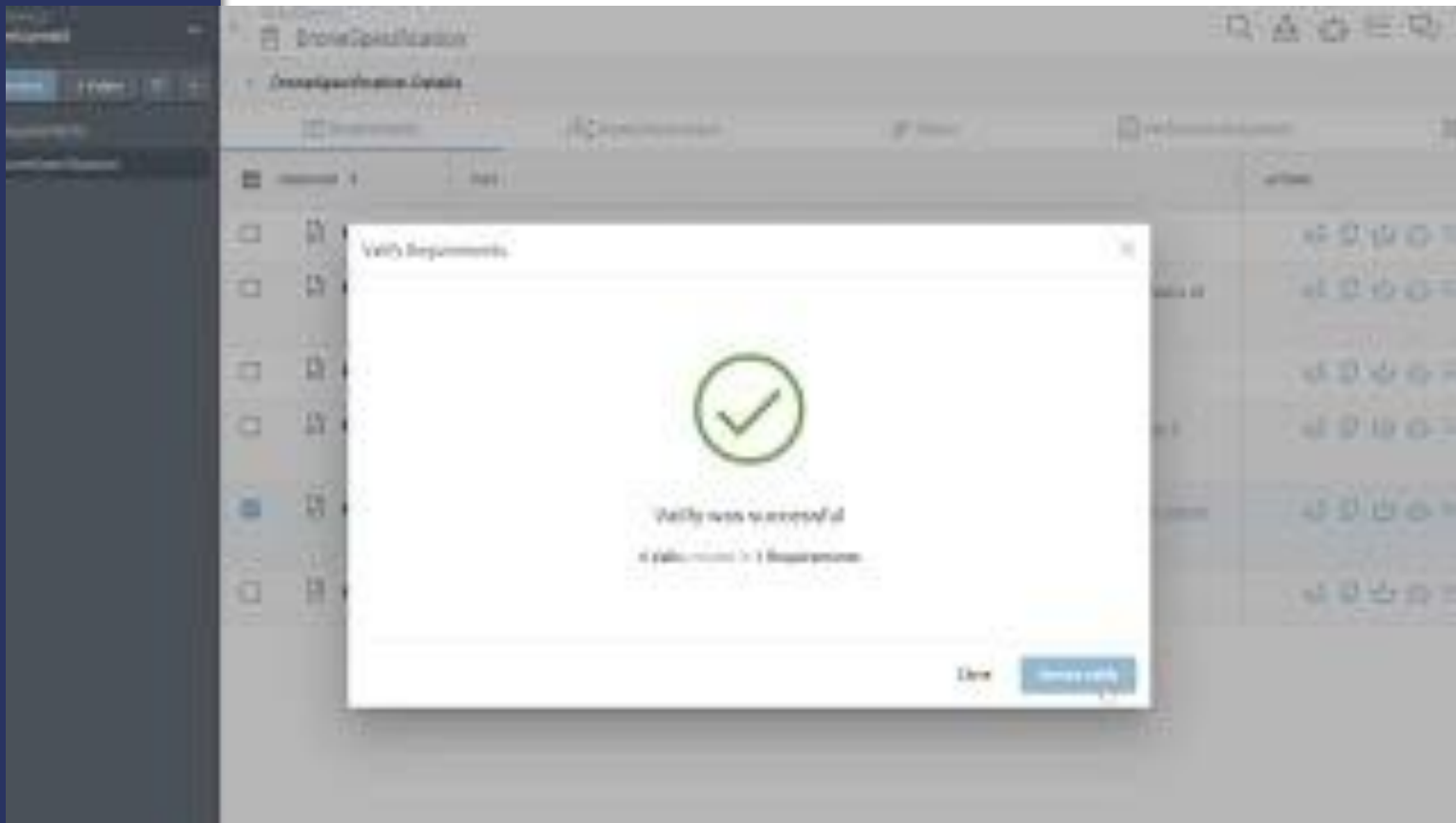
Index	Requirement	Score
15	The <class Administration or national government department> shall not enter into a settlement which involves acceptance of such liability as is mentioned in the first or second sub-paragraph to the paragraph without the consent of the Recognized Organisation which shall not unreasonably be withheld.	60%
19	If it is established that the is not the legal entity responsible for the performance of the duties of the Naval Administration, the term "" as used on the preceding sub-paragraphs shall be replaced by the name or designation of the appropriate Agency, body or organization as determined by the end-agreed by the appropriate Agency, body or organization.	60%

The following requirements had an analysis score between 60% - 70%.

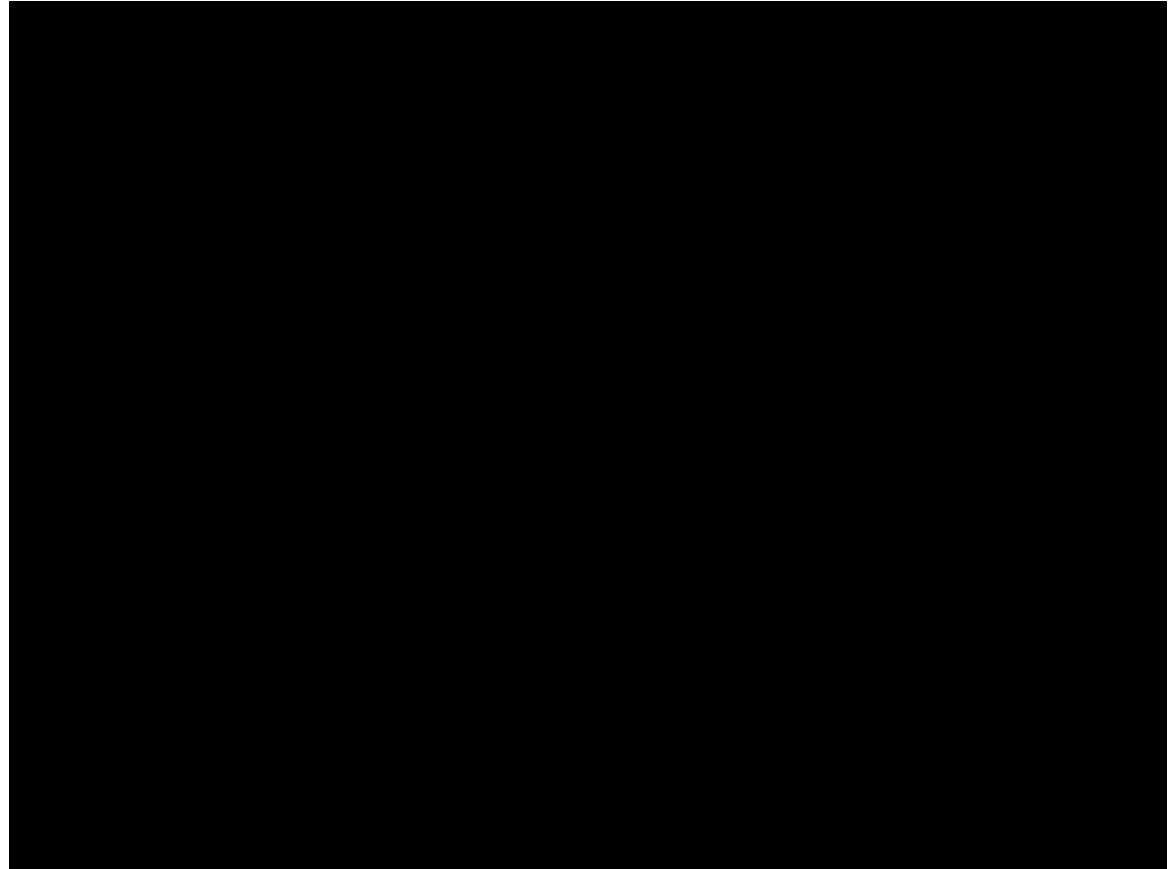
Index	Requirement	Score
7	The Flight_Information_System shall usually be on line.	60%
31	The controller shall send the driver his itinerary for the day, it shall be delivered at least 8 hours prior to his shift.	60%
39	A lamp or LED test or dual lamp/ filament reliability shall be provided for switches if the mean time between failure is less than 100,000 hours.	60%
40	The Mars Rover shall always drive at 10 inch on natural terrain up to 10 degree slope when at maximum gross vehicle weight.	60%
45	While on Mars the Mars Rover shall be controlled from earth with its remote.	60%

Beispiel 2: Valispace (Automatisieren)

VALISPACE



Beispiel 3: trace.space (Anreichern)





Beeindruckend! Aber...

- Wie kann die KI meine Architektur nutzen?
- Muss ich alles überprüfen?
- Landet meine Innovation im KI-Modell?
- Kann ich effektiv Wiederverwendung anwenden?
- Können andere (Wettbewerber) Aspekte meines Designs sehen?
- Müssen meine Produktdaten in die Cloud?
- ... und vieles mehr

customer organization research performance complex data retention process risk analysis impact opportunity project

CONTEXT

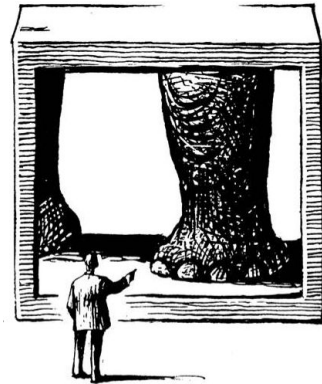


treatment organization impact assessment context organization
strategy performance evaluation performance organization
resources important sales risk monitor research sales assessment plan scope

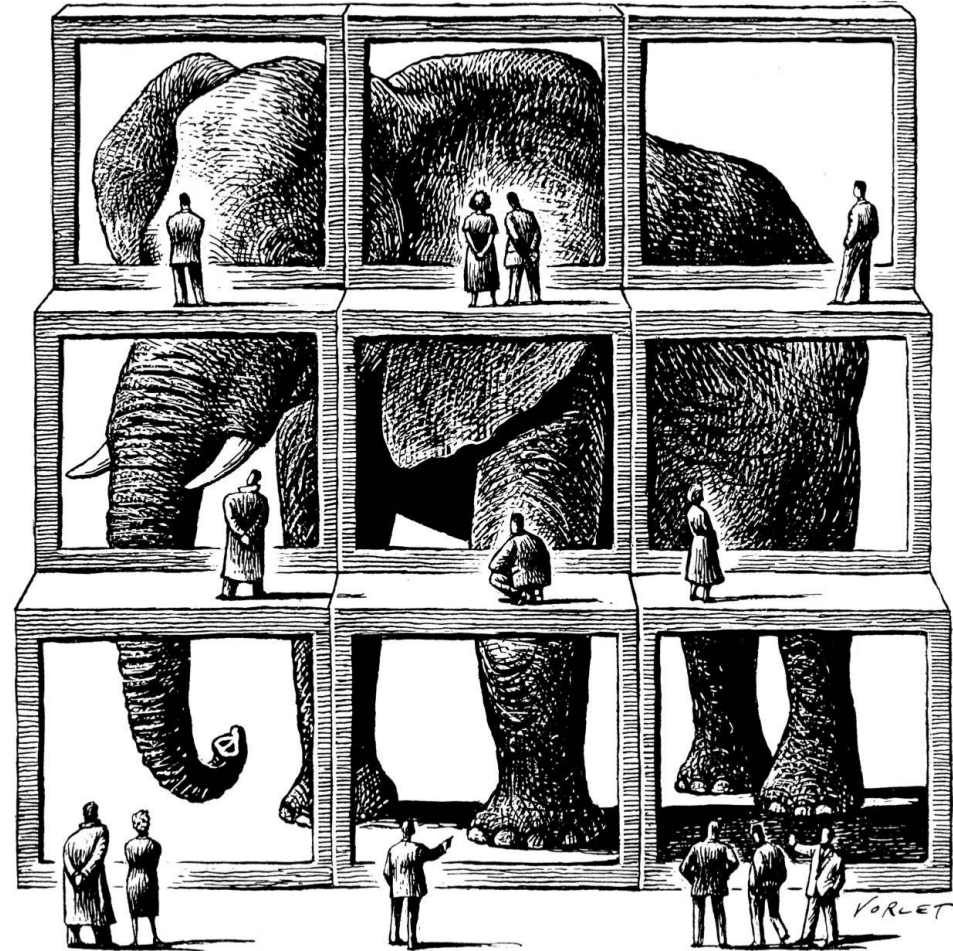


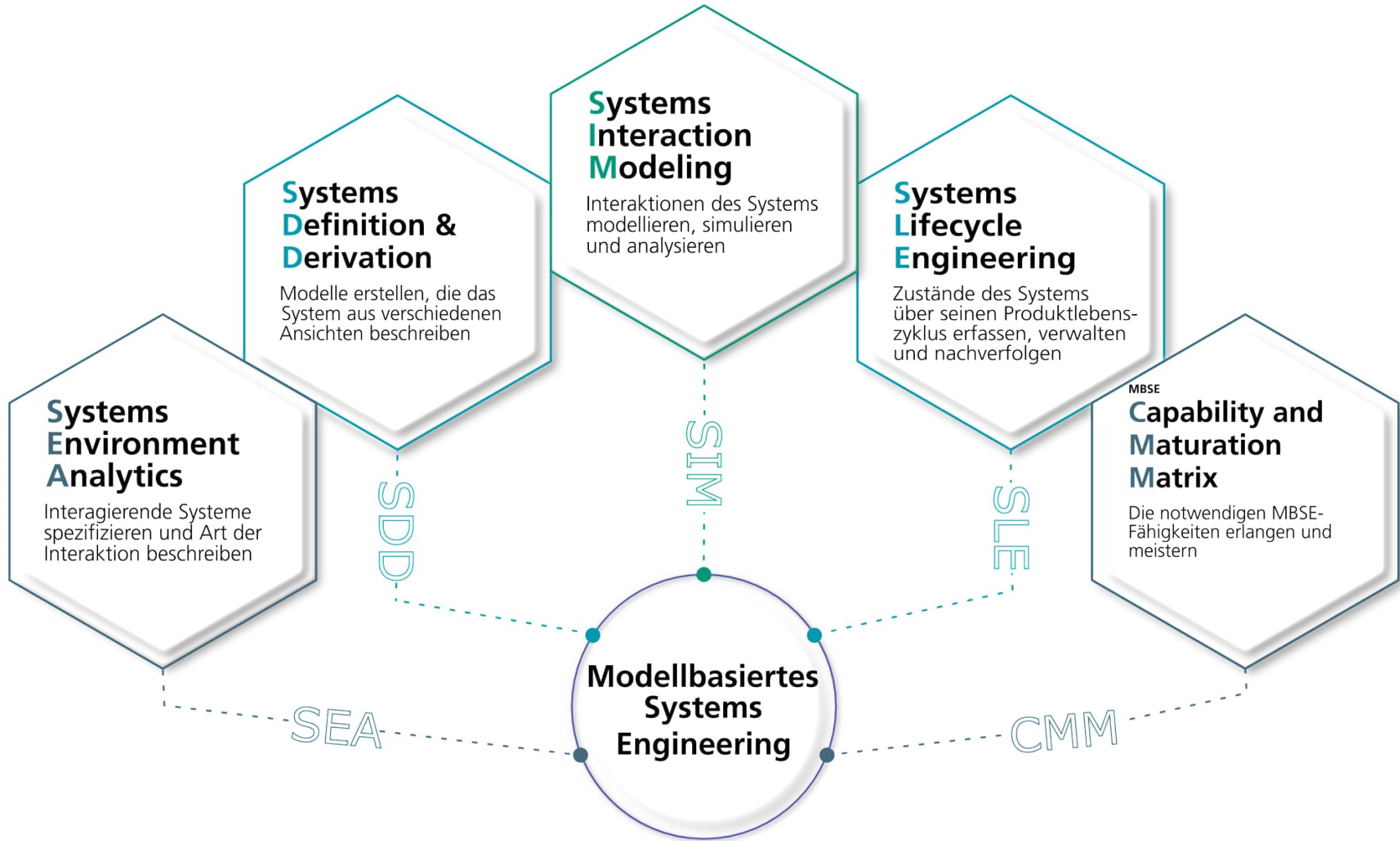
MBSE: KI-Kontext durch das Systemmodell

Ohne Kontext



Mit Kontext



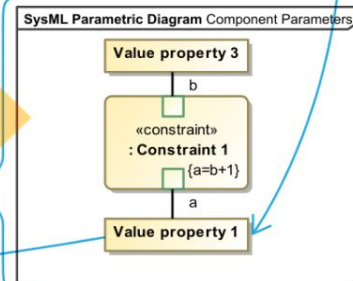
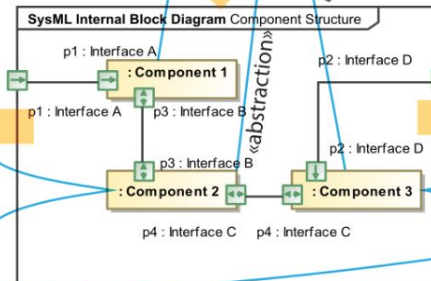
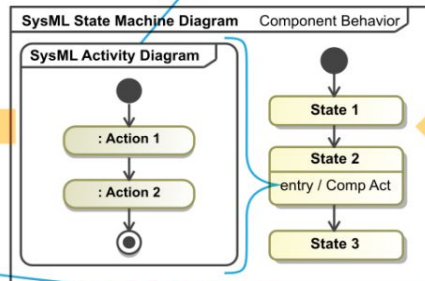
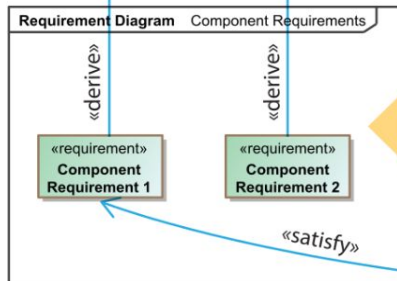
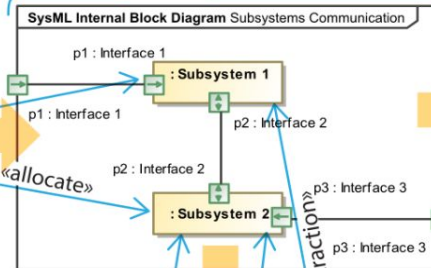
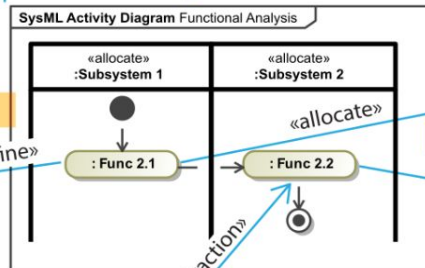
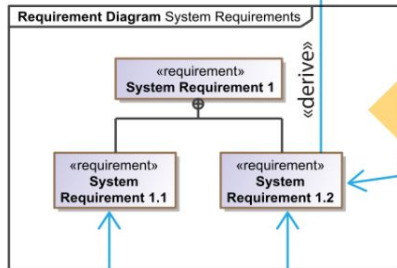
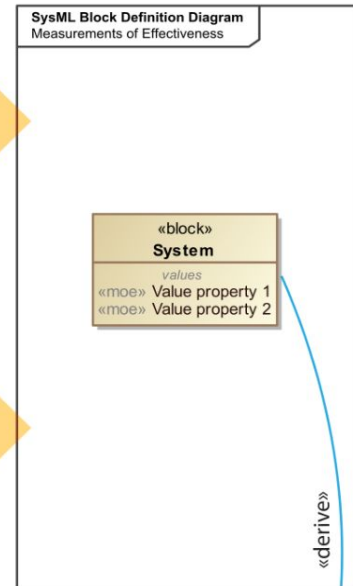
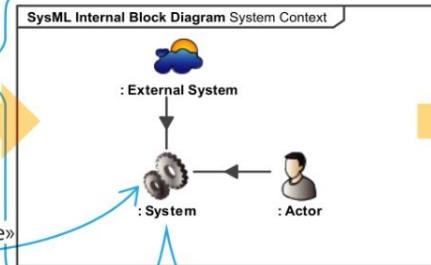
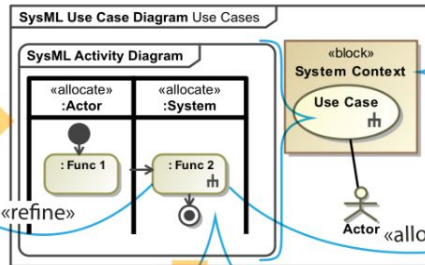
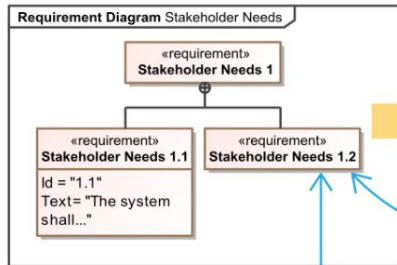


Requirements

Behavior

Structure

Parametrics



Modelle unterstützen viele Aktivitäten

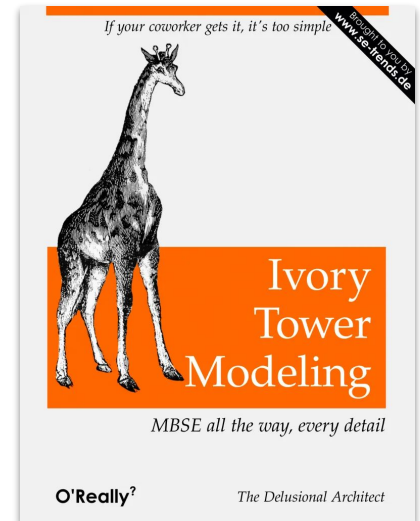
18 Dinge, die Sie mit Systemmodellen machen können

1. Kommunizieren
2. Dokumente generieren
3. Sichten erstellen
4. Testfälle generieren
5. Code generieren
6. Fortschritt verfolgen
7. Vollständigkeit prüfen
8. Informationen zusammenfassen
9. Automatische Verifikation
10. Auswirkungen von Änderungen verstehen
11. Risiko bewerten
12. Konflikte erkennen
13. Inkonsistenzen erkennen
14. Das Modell animieren (Simulation)
15. Das Modell als Spezifikation
16. Das Modell zur Strukturierung
17. Wissen verwalten
18. Wiederverwendung

Quelle: <https://www.se-trends.de/einsatz-von-systemmodellen/>

Das Problem mit Systemmodellierung

- Akzeptanz
- Methodik
- Integration
- Elfenbeinturm-Modelle
- Spaghetti-Modelle



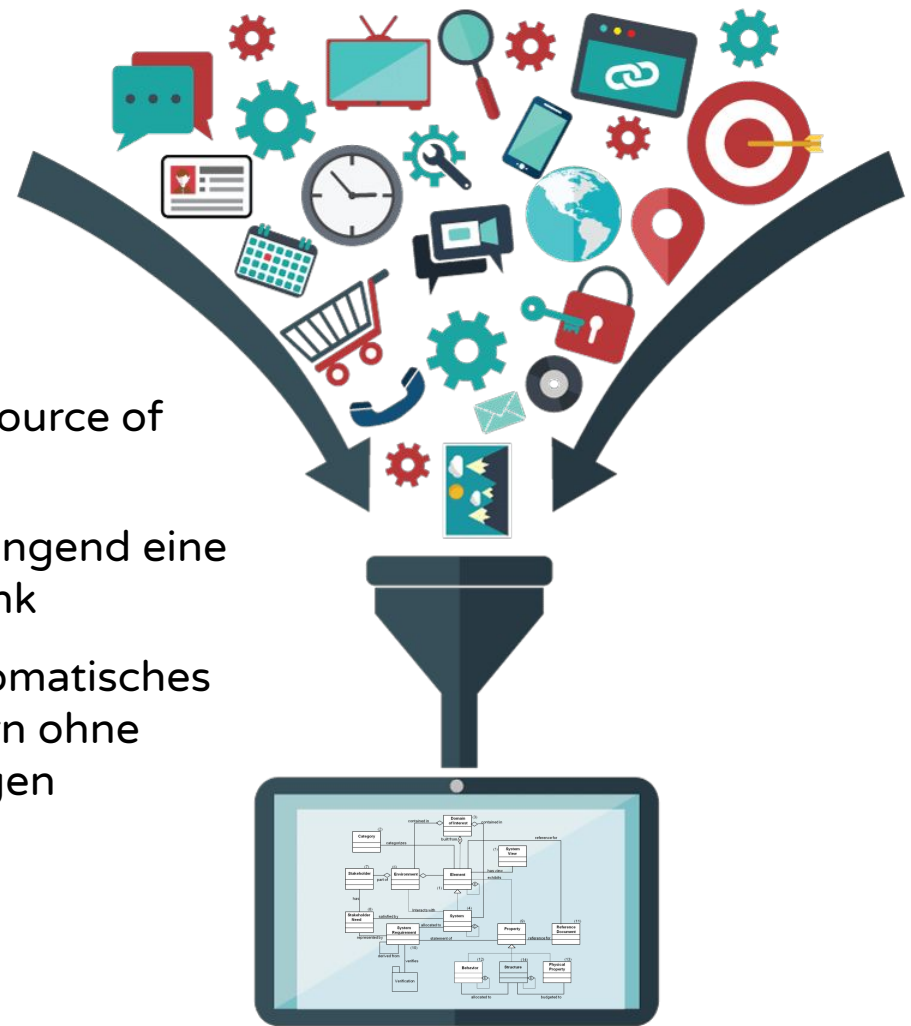
4 Thesen

Für einen Quantensprung in der KI-gestützten Produktentwicklung

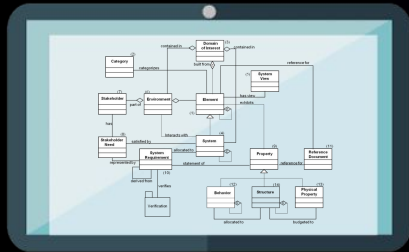
1. Das Systemmodell repräsentiert “die Wahrheit”
2. Die KI vermittelt zwischen Mensch und Modell
3. Der Mensch braucht nichts vom Modell zu wissen (kann aber)
4. Die KI muss den Modell-Kontext verstehen

These 1: Das Systemmodell repräsentiert “die Wahrheit”

- “Single Source of Truth”
- Nicht zwingend eine Datenbank
- Kein automatisches Verändern ohne Nachfragen



These 2: Die KI muss zwischen Mensch und Modell vermitteln



Of course, I'd be happy to help. Can you provide me with more information about your company? What kind of image or message do you want to convey through the logo?

Ai

I want the logo to convey a sense of innovation and cutting-edge technology. And I'd like the color scheme to be blue and silver.

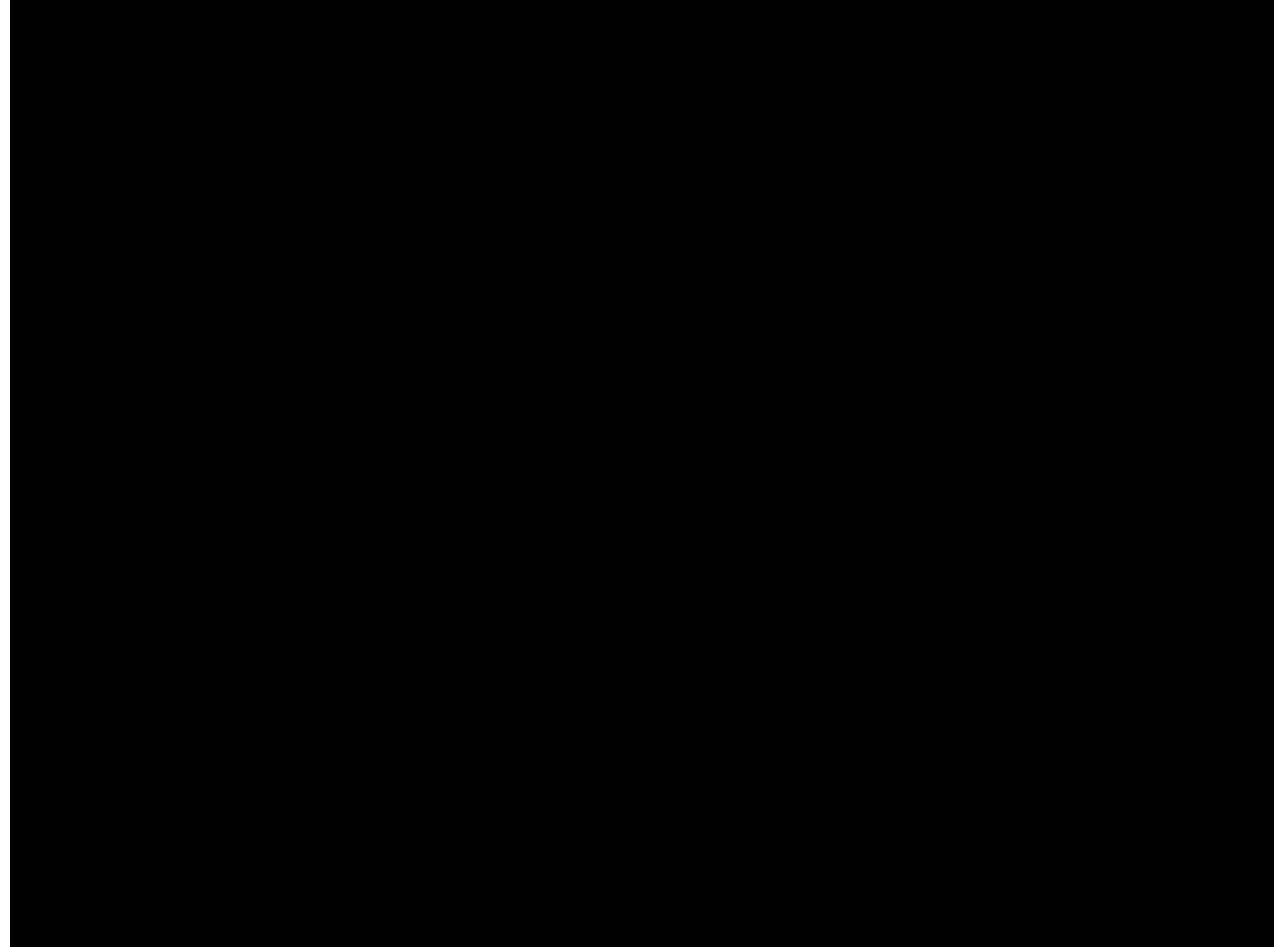


Understood. Let me generate some initial designs based on your specifications.

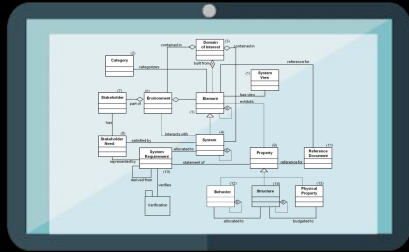
Ai

I really like the th

These 2:
**Wir brauchen
ein
Übersetzungs
werkzeug!**



These 3: Der Mensch braucht nichts vom Modell zu wissen (sollte aber bei Interesse Zugriff haben)



Of course, I'd be happy to help. Can you provide me with more information about your company? What kind of image or message do you want to convey through the logo?



I want the logo to convey a sense of innovation and cutting-edge technology. And I'd like the color scheme to be blue and silver.



Understood. Let me generate some initial designs based on your specifications.



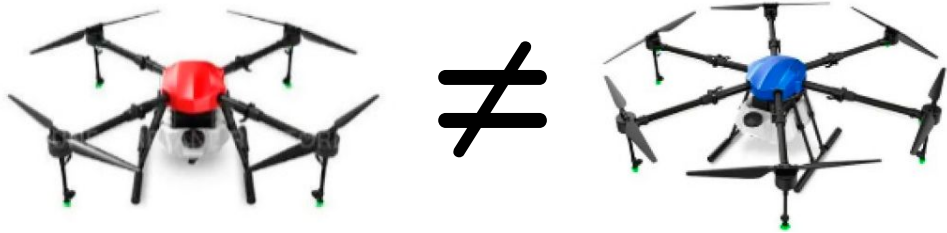
I really like the th

These 4: Die KI muss den Modell-Kontext verstehen

Essentiell, damit...

- Keine Produkte vermischt werden
- Architektur respektiert wird
- Wiederverwendung ermöglicht wird
- Das Modell an der richtigen Stelle verändert wird

... und vieles mehr



**Gibt es das
schon?**



**Noch nicht —
Wir (und andere)
arbeiten dran!**

Die Zukunft



**Modelle als Dreh- und Angelpunkt
für KI-basierte Produktentwicklung**

Die Zukunft

- KI-Chat erreicht bald das “Tal der Enttäuschungen”
- SysML v2 könnte ein Enabler sein (frühestens 2025)
- Bis dahin:
 - Punktuelle Lösungen ohne Modell-Kontext
 - Kunden- und Anwendungsfall-spezifische Lösungen mit Kontext-Bewusstsein



KI im MBSE

<https://www.selive.de/ai-in-mbse/>

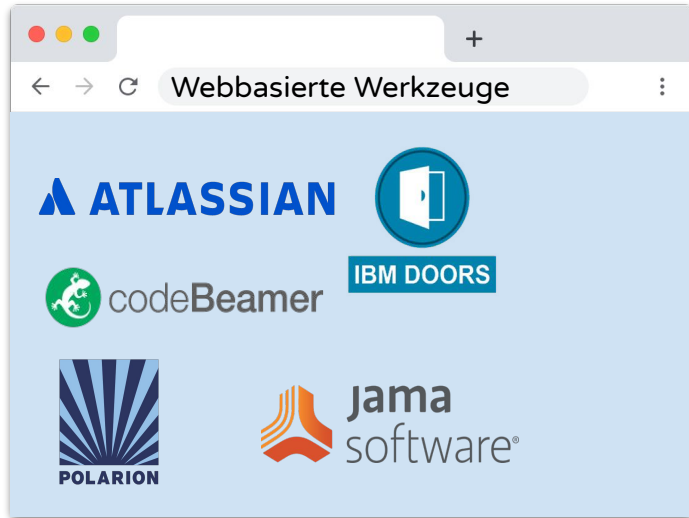
- Analyze stakeholder requirements
- Analyze system
- Derive system requirements
- Specify functions
- Specify system architecture
- Discipline-specific development
- Life Cycle Management Process
- ...

- Hierarchize system
- Structure system
- Specify system architecture
- Specify interfaces
- Evaluate system architectures
- Link system elements to functions
- Specify requirements from system elements

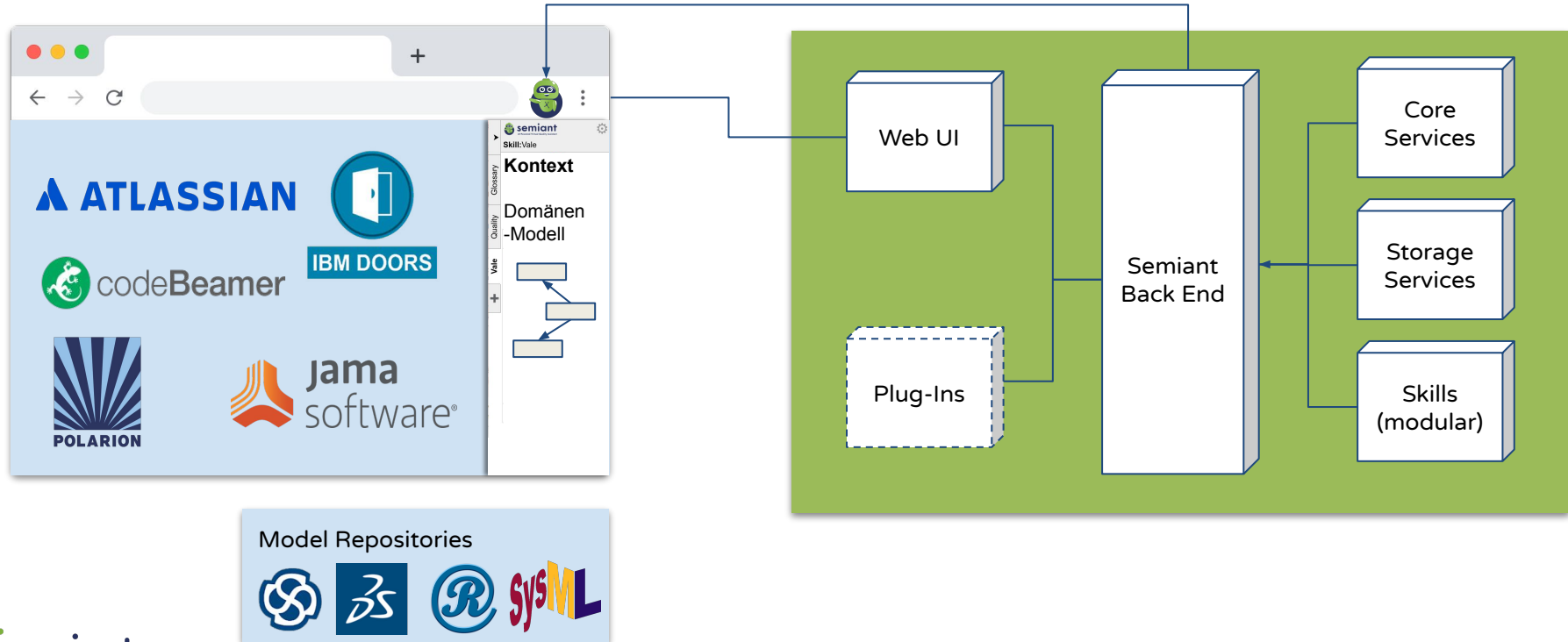
Analyze stakeholder requirements	Elicit stakeholder requirements	Identify stakeholders Describe stakeholders Categorize stakeholders Include stakeholder requirements
	Define stakeholder requirements	Describe stakeholder requirements Link stakeholder requirements
	Analyze stakeholder requirements	Review stakeholder requirements Refine requirements
	Manage stakeholder requirements	Structure stakeholder requirements Determine responsibilities
Analyze services of the system (use cases)	Identify use cases	Identify use cases Describe use cases
	Define activities to achieve the system services	Structure use cases Link use cases with stakeholders Identify activities Describe activities Structure activities Define activities Identify activities

System Element	Requirement ID	Requirement Text	Priority	State	Origin	Target	Traceability
System Requirements	SR-001	System shall be available 24/7	High	Open	Customer	System	SR-001
	SR-002	System shall support 1000 concurrent users	Medium	Open	Customer	System	SR-002
	SR-003	System shall be secure against unauthorized access	High	Open	Customer	System	SR-003
	SR-004	System shall be easy to use	Low	Open	Customer	System	SR-004
	SR-005	System shall be scalable	Medium	Open	Customer	System	SR-005
	SR-006	System shall be reliable	High	Open	Customer	System	SR-006
	SR-007	System shall be maintainable	Medium	Open	Customer	System	SR-007
	SR-008	System shall be flexible	Low	Open	Customer	System	SR-008
	SR-009	System shall be interoperable	Medium	Open	Customer	System	SR-009
	SR-010	System shall be cost-effective	Low	Open	Customer	System	SR-010
Functional Requirements	FR-001	System shall provide a user interface	High	Open	System	System	FR-001
	FR-002	System shall provide a search function	Medium	Open	System	System	FR-002
	FR-003	System shall provide a reporting function	Medium	Open	System	System	FR-003
	FR-004	System shall provide a notification function	Medium	Open	System	System	FR-004
	FR-005	System shall provide a logging function	Medium	Open	System	System	FR-005
	FR-006	System shall provide a backup function	Medium	Open	System	System	FR-006
	FR-007	System shall provide a recovery function	Medium	Open	System	System	FR-007
	FR-008	System shall provide a configuration function	Medium	Open	System	System	FR-008
	FR-009	System shall provide a monitoring function	Medium	Open	System	System	FR-009
	FR-010	System shall provide a maintenance function	Medium	Open	System	System	FR-010

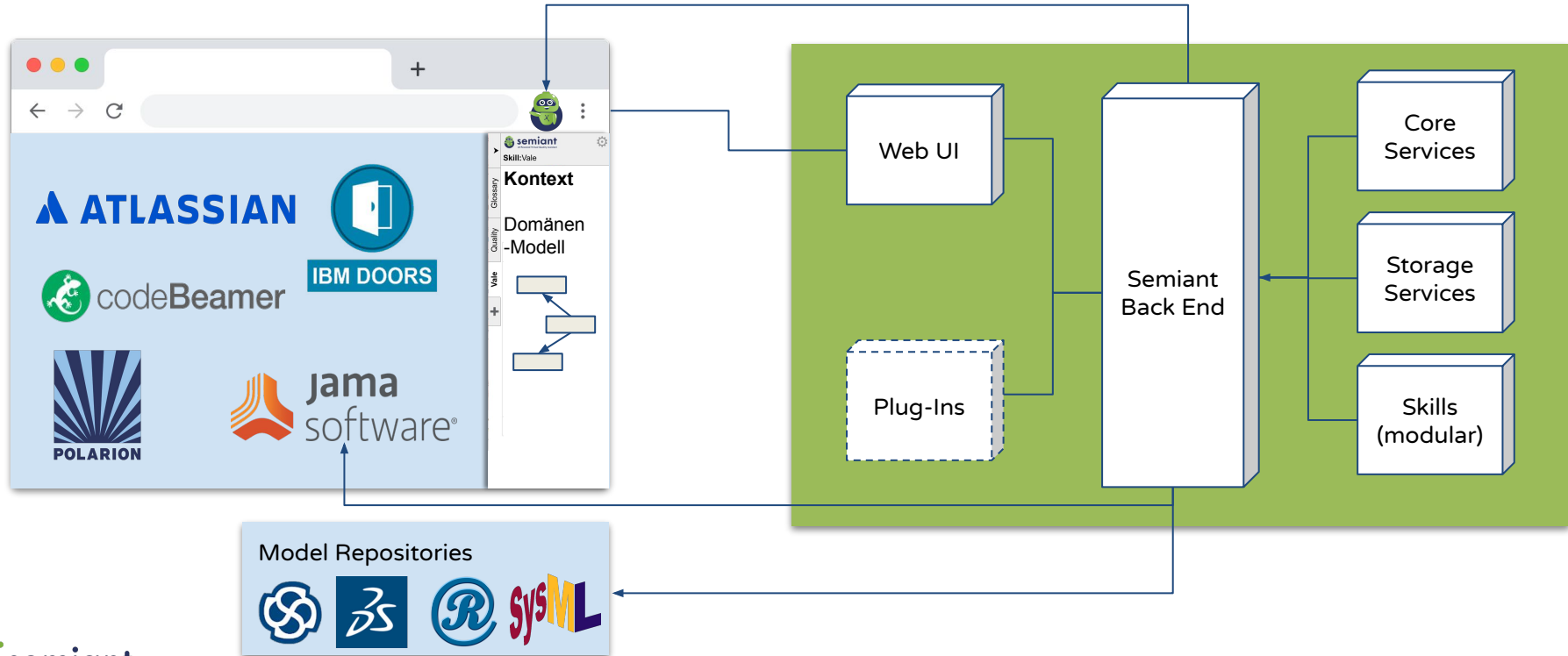
Typisches Systems Engineering Ecosystem



Semiant: UI-Integration über Browser Extension

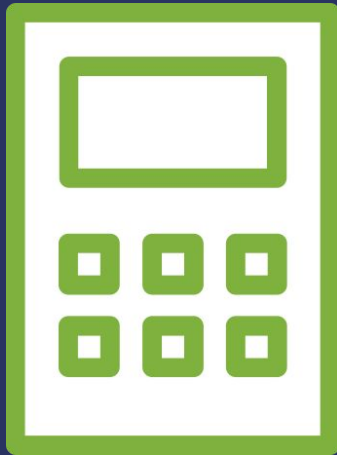


Semiant: Bestehende Repositories nutzen



Beispiel-Service: Deterministische Modellextraktion aus Text

Arbeit von Arora et.al.: Extraktion von Domänen-Modellen



Rule Id	Description
C1	All noun phrases in the requirements are candidate concepts.
A1	Transitive verbs suggest associations.
A2	A verb with a preposition suggests an association.
A3	A relative clause modifier of nouns suggests an association.
A4	A verbal clausal modifier suggests an association.
A5	A non-finite verbal modifier suggests an association.
A6	A prepositional dependency (Link Path) suggests an association between indirectly-related concepts.
Ag1	Genitive cases suggest aggregations.
Ag2	Terms "contain", "include", [...] suggest aggregations.
At1	An intransitive verb with an adverb suggests an attribute.
G1	Premodifiers of noun phrases suggest generalizations.

Arora, C., Sabetzadeh, M., Nejati, S., & Briand, L. (2019). An active learning approach for improving the accuracy of automated domain model extraction. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 28(1), 1-34.

An Active Learning Approach for Improving the Accuracy of Automated Domain Model Extraction

CHETAN ARORA, SoT Centre for Security, Reliability and Trust, University of Luxembourg
MEHRDAD SABETZADEH, SoT Centre for Security, Reliability and Trust, University of Luxembourg
SHIVA NEJATI, SoT Centre for Security, Reliability and Trust, University of Luxembourg
LIONEL BRIAND, SoT Centre for Security, Reliability and Trust, University of Luxembourg

Domain models are a ubiquitous way of representing requirements more precisely than natural language. However, alongside the benefits of domain models, the manual effort required to create them is high. In this paper, we propose an active learning approach for improving the accuracy of automated domain model extraction. To this end, we provide feedback over the extracted models to the system to provide it with more examples of correct and incorrect extractions. The results are promising, showing that the system can improve its accuracy over time. The results are also compared with a baseline system that does not use active learning.

1. INTRODUCTION
Natural language is a rich and expressive way of communicating domain knowledge, as we show in our evaluation.

Using Domain-specific Corpora for Improved Handling of Ambiguity in Requirements

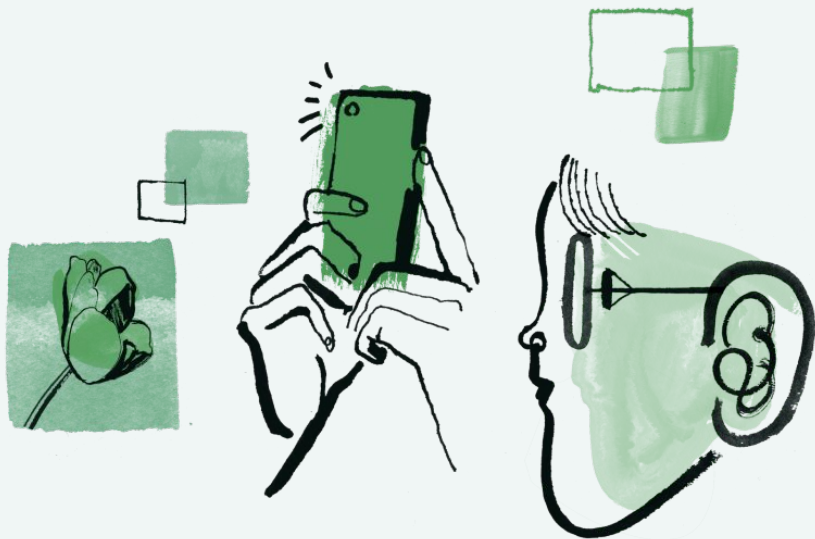
Saad Ezzini¹, Sallam Abualhajja¹, Chetan Arora¹, Mehrdad Sabetzadeh¹, Lionel C. Briand^{1*}
¹SoT Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg

²Deakin University, Geelong, Australia
³School of Electrical Engineering and Computer Science, University of Otago, New Zealand
Email: {saad.ezzini, sallam.abualhajja}@uni.lu, chetan.arora@deakin.edu.au, {m.sabetzadeh, lbriand}@outlook.com

Abstract—Ambiguity in natural-language requirements is a pervasive issue that has been studied by the requirements engineering community for more than two decades. A fully manual approach for addressing ambiguity in requirements is tedious and time-consuming, and may further overlook *unacknowledged ambiguity* – the situation where different stakeholders perceive a requirement as unambiguous but, in reality, interpret the requirement differently. In this paper, we propose an automated approach that uses natural language processing for handling ambiguity in requirements. Our approach is based on the automatic generation of a domain-specific corpus from Wikipedia. Integrating domain knowledge, as we show in our evaluation,

This has an impact on the likely interpretations of the requirements and consequently on what can be considered as ambiguous. Second, ambiguity can be *unacknowledged*, meaning that multiple readers, being unaware of such ambiguity, may have different interpretations for the same requirement. In contrast to *acknowledged* ambiguity where the reader recognizes ambiguity, *unacknowledged* ambiguity might lead to serious problems due to unconscious misunderstandings [7]. A fully manual analysis of ambiguity is expensive and also likely to overlook unacknowledged ambiguity. There is therefore a need

Leitfaden zum Selbermachen



People + AI Research

<https://pair.withgoogle.com>

User Needs + Defining Success



Even the best AI will fail if it doesn't provide unique value to users.

[Read more](#) →

Data Collection + Evaluation



Decide what data are required to meet your user needs, source data, and tune your AI.

[Read more](#) →

Mental Models



Introduce users to the AI system and set expectations for system-change over time.

[Read more](#) →

Explainability + Trust



Explain the AI system and determine if, when, and how to show model confidence.

[Read more](#) →

Feedback + Control



Design feedback and control mechanisms to improve your AI and the user experience.

[Read more](#) →

Errors + Graceful Failure



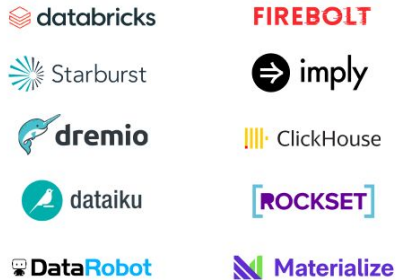
Identify and diagnose AI and context errors and communicate the way forward.

[Read more](#) →

Data Startups

<https://future.com/data50/>

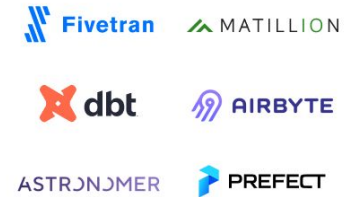
QUERY & PROCESSING



AI/ML



ELT & ORCHESTRATION



DATA GOVERNANCE & SECURITY



CUSTOMER DATA ANALYTICS



BI & NOTEBOOKS



DATA OBSERVABILITY



Fazit



- Der KI-Hype wird bald abklingen
- KI kann heute punktuell die Produktentwicklung unterstützen
- Langfristig muss die KI ein Systemmodell kennen und nutzen
- Der Markt ist noch stark in Bewegung, fertige Lösungen gibt es noch nicht (selber bauen!)

THANK YOU

Fragen?

Zur Person: Michael Jastram



- Promotion 2012 zum Thema Anforderungsmodellierung
- Selbstständig seit 2011 (Formal Mind GmbH): Systems Engineering
- Artikelreihe im HOOD-Blog: SEcKI – Systems Engineering mit KI

Systems Engineering Trends

www.se-trends.de



Jetzt anmelden

Semiant AI Quality Assistant

www.semiant.com



Jetzt anmelden