

# Large Language Models für SPS Codegenerierung

OTH Amberg-Weiden | Fakultät EMI

---



**01** | Einführung

**02** | Ansätze

**03** | Antragsidee

**04** | Förderung

**01** | Einführung

02 | Ansätze

03 | Antragsidee

04 | Förderung

## Warum sind Coding-Copiloten heute selbstverständlich – aber nicht im SPS-Umfeld?

- Coding-Copiloten sind heute Stand der Technik (Python, C, Java, ...)
- Umfangreiche Trainingsdaten aus Open-Source-Plattformen (GitHub, StackOverflow, ...)
- Im SPS-Umfeld bislang nur begrenzt öffentlich verfügbare Daten
- Zunehmende Verbesserung der Datenlage durch aktuelle Forschungsarbeiten
- Enge industrielle Kooperation als Schlüssel für praxisnahe Lösungen

## Warum brauchen wir einen Copiloten für die SPS-Programmierung?

- SPS-Engineering ist zeitaufwendig und wissensintensiv
- Lösungen basieren stark auf individueller Erfahrung
- Fachkräftemangel und demografischer Wandel verschärfen die Situation
- Hohes Risiko von Wissensverlust bei Personalwechsel und Renteneintritt
- Ansatz: **gemeinsamer Förderantrag** für eine praxistaugliche Lösung

01 | Einführung

02 | **Ansätze**

03 | Antragsidee

04 | Förderung

Proprietäre Ansätze

Forschungsansätze

## Pro:

- Generierung und Erklärung von SPS-Code (Fokus auf SCL/ST)
- Erstellung und Anpassung einfacher Visualisierungen
- Beantwortung von Fragen zur Herstelldokumentation
- Automatisierte Erstellung von Unit-Tests

## Pro:

- Generierung und Erklärung von SPS-Code (Fokus auf SCL/ST)
- Erstellung und Anpassung einfacher Visualisierungen
- Beantwortung von Fragen zur Herstelldokumentation
- Automatisierte Erstellung von Unit-Tests

## Contra:

- Starke Bindung an das geschlossene Ökosystem
- Keine Anpassbarkeit an firmenspezifische Wissensquellen und Standards
- Fokus auf Unterstützung für SCL/ST
- Kostenpflichtig

- Erste öffentliche Datensätze für Training & Benchmarking von SPS-Code
- Datenbasis aktuell begrenzt und oft stark vereinfacht
- Fokus überwiegend auf einzelne SPS-Sprachen (v. a. Structured Text)
- Erste Arbeiten zu grafischen Sprachen (FBD, SFC), meist mit einfachen Beispielen
- Evaluation häufig über wenige Fallstudien statt standardisierter Benchmarks

01 | Einführung

02 | Ansätze

03 | **Antragsidee**

04 | Förderung

Allgemein

1. Retrieval

2. Planning

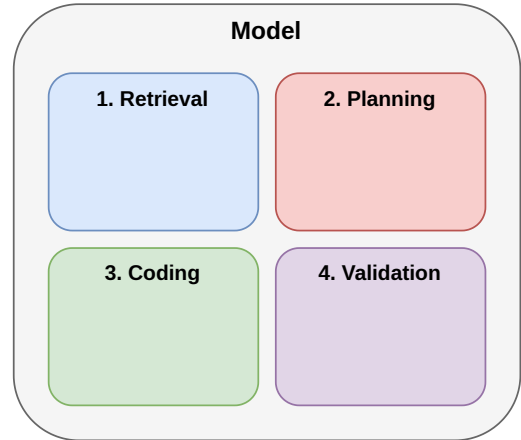
3. Coding

4. Validation

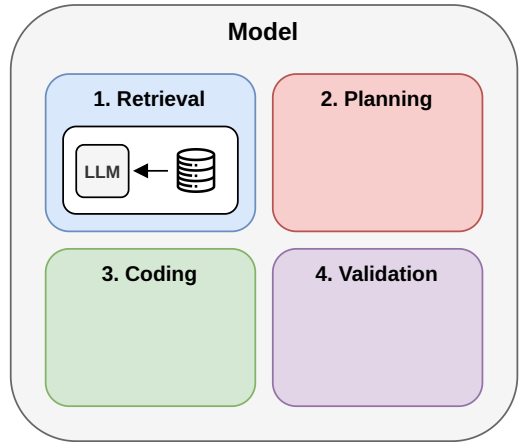
Übersicht

Weitere Projektideen

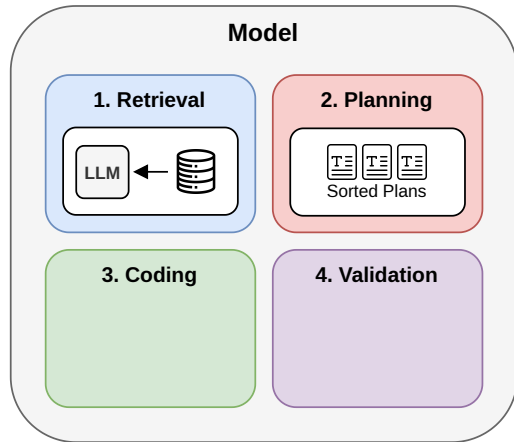
- KI-Unterstützung für SPS-Engineering
- Kombination aus Large Language Models (LLMs) und Firmenwissen
- Mensch bleibt im Engineering-Prozess involviert
- Modell verarbeitet Eingaben und erzeugt daraus kontextbezogene Vorschläge und Ergebnisse



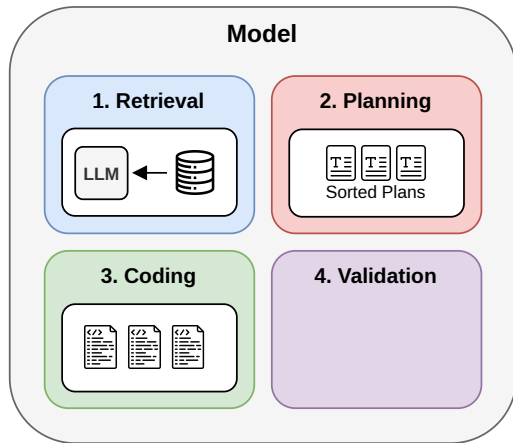
- Firmendaten als Wissensbasis
- Semantische Suche nach ähnlichen Informationen
- Wissen unabhängig von Einzelpersonen verfügbar machen



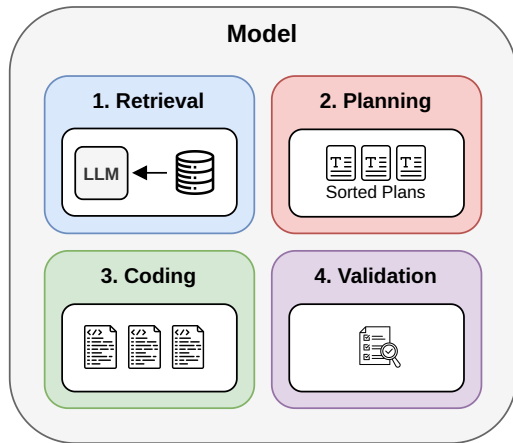
- Analyse der Anforderungen
- Zerlegung der Spezifikation in Teilaufgaben
- Strukturierung des Prozesses



- Codegenerierung aus Planung & Wissen
- Nutzung von Guidelines & Libraries
- Textuelle **und** grafische SPS-Sprachen

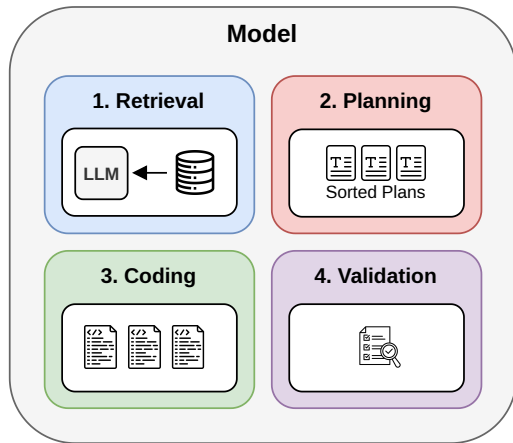


- Syntax-Prüfung des Codes
- Logik- & Funktionsprüfung
- Iterativer Verbesserungsprozess



Quelle Icons: <https://www.flaticon.com>

- Gemeinsames Basismodell
- Firmenspezifisches Wissen durch Retrieval integrieren
- Mehrsprachige SPS-Unterstützung



Quelle Icons: <https://www.flaticon.com>

- Kürzere Engineering-Zeiten
- Sicherung von Expertenwissen
- Hersteller- und sprachenunabhängig
- Schutz geistigen Eigentums (IP-Schutz)
- Höhere Qualität und Konsistenz von SPS-Programmen
- Schnellere Einarbeitung neuer Mitarbeiter
- Standardisierung von Engineering-Prozessen
- Risikominimierung bei Personalwechsel und Fachkräftemangel



- Automatisierte Erstellung von Projektdokumentationen
- Natürlichsprachliche Analyse und Auswertung von CAD-Dateien
- Durchsuchung technischer Dokumentationen per Spracheingabe
- Unterstützung bei Wartung und Instandhaltung
- Automatisierte Qualitätsprüfung und Fehlererkennung
- Simulation und Validierung von SPS-Programmen



01 | Einführung

02 | Ansätze

03 | Antragsidee

**04 | Förderung**

Förderlinien

- Förderlinie „**Künstliche Intelligenz und Data Science**“ von Informations- und Kommunikationstechnik (IuK) Bayern
- Themenoffen
- Typisches Fördervolumen: **700.000 € bis 800.000 €** pro Vorhaben
- Breites Spektrum förderfähiger Kosten: Personal, Materialien, externe Leistungen sowie technische Ausstattung
- Förderquote von **bis zu 50 %** (Siehe nachfolgendes Beispiel)
- Hohe Attraktivität für **Verbundprojekte mit mehreren Industriepartnern**

Weitere Informationen: <https://www.iuk-bayern.de/>

Vier Schritte der Festlegung:

1. Arbeitspaketbewertung
2. Umverteilung (bei Hochschul-/Forschungsbeteiligung)
3. Verbundbonus (unter Voraussetzungen)
4. Zustimmung der Projektpartner

\* (gemäß Richtlinie vom 15.05.2019); Quelle: <https://www.iuk-bayern.de/foerderquotenberechnung>

### Bewertungskriterien

- Industrielle Forschung: bis **50 %**
- Experimentelle Entwicklung: bis **25 %**
- Administrative Tätigkeiten: **nicht förderfähig**

### Beispiel – Förderquote nach Arbeitspaketbewertung

| Partner        | Projektk. [T€] | PM IF | PM EE | Förderq. | Zuwendung |
|----------------|----------------|-------|-------|----------|-----------|
| Unternehmen U1 | 300            | 32    | 4     | 47,22 %  | 141,7     |
| Unternehmen U2 | 400            | 18    | 27    | 35,0 %   | 140,0     |
| Hochschule     | 200            | 30    | 0     | 50,0 %   | 100,0     |

Ergebnis: individuelle, gewichtete Förderquote je Projektpartner

### Grundsatz

- Nur bei Beteiligung von Hochschulen/Forschungseinrichtungen, da diese keine Eigenmittel einbringen
- Erhöhung deren Förderquote bis max. **100 %**
- Finanzierung durch Absenkung der Industriequoten
- Gesamtförderquote des Verbunds bleibt konstant

### Beispiel – Förderquoten nach Umverteilung

- U1: **30,4 %** (91,2 T€)
- U2: **22,5 %** (90,0 T€)
- Hochschule: **100 %** (200,0 T€)

Gesamtförderquote des Verbunds bleibt bei 42,4 %

### Verbundbonus

- Zuschlag bis zu **+15 Prozentpunkte** für gewerbliche Partner
- Maximal **50 %** Förderung je Industriepartner

### Voraussetzung (mind. eine erfüllt)

- Forschungseinrichtungen tragen mind. 10 % der Kosten
- oder: KMU beteiligt und kein Unternehmen >70 % der Kosten

### Beispiel – Förderquoten inkl. Verbundbonus

- U1: **45,4 %** (136,2 T€)
- U2: **37,5 %** (150,0 T€)
- Hochschule: **100 %** (200,0 T€)

Für den Verbund ergibt sich so eine Förderquote von 54,0 %

### Zustimmung erforderlich, wenn

- bewilligte Förderquote unter beantragter Förderquote liegt

### Bedeutung

- Projektpartner bestätigt Durchführung zu den bewilligten Konditionen
- Voraussetzung für die Freigabe des Projektstarts

### Ergebnis des Gesamtbeispiels

| Empf.        | Projektk. [T€] | Förderq. | Förderq. n. U. | Förderq. m. B. | Zuwendung [T€] |
|--------------|----------------|----------|----------------|----------------|----------------|
| U1           | 300            | 47,22 %  | 30,4 %         | 45,4 %         | 136,2          |
| U2           | 400            | 35,0 %   | 22,5 %         | 37,5 %         | 150,0          |
| Hochschule   | 200            | 50,0 %   | 100 %          | 100 %          | 200,0          |
| <b>Summe</b> | <b>900</b>     |          |                |                | <b>486,2</b>   |

- Förderlinie „**Strukturimpuls Forschungseinstieg**“ für erstberufene Professorinnen und Professoren
- Themenoffen
- Gesamtfördersumme: **270.000 €** pro Vorhaben
- Keine direkte Förderung von Unternehmen; **indirekter Nutzen** durch Kooperation und Verwertung von Forschungsergebnissen

- **Zentrales Innovationsprogramm Mittelstand (ZIM)** des BMWK
- Förderung von **F&E-Projekten kleiner und mittlerer Unternehmen (KMU)**
- Einzel- oder **Kooperationsprojekte** zwischen Unternehmen sowie mit Forschungseinrichtungen
- Förderquote: **bis zu 60 %** für KMU, Forschungseinrichtungen als Unterauftragnehmer
- **Themenoffen** und kontinuierlich beantragbar

- **Natural Language Processing (NLP)** – datengetriebene Sprach- und Textverarbeitung
- **Robotik** – KI-gestützte Wahrnehmung, Planung und Interaktion
- **Cybersecurity** – intelligente Erkennung und Abwehr von Angriffen

Alle Themen sind grundsätzlich **KI-basiert** und passen zu den vorgestellten Förderlinien. Bei Interesse an einem der Themen freuen wir uns über eine **unverbindliche Kontaktaufnahme**, auch unabhängig von dieser Veranstaltung.

## Beginn gemeinsamer Ideenaustausch



Quelle Icons: <https://www.flaticon.com>

# Vielen Dank!

---

## **OTH Amberg-Weiden**

Ostbayerische Technische Hochschule (OTH) Amberg-Weiden  
Kaiser-Wilhelm-Ring 23 | 92224 Amberg

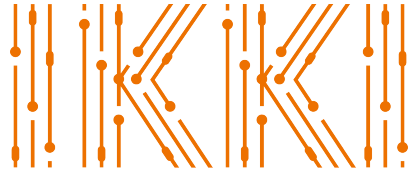
Tel.: +49 (9621) 482-3749  
[a.kestler@oth-aw.de](mailto:a.kestler@oth-aw.de)



# Innovations- und Kompetenzzentrum Künstliche Intelligenz (IKKI)



<https://www.oth-aw.de/ikki>



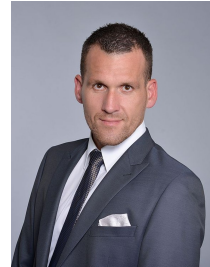
INNOVATIONS - UND  
KOMPETENZZENTRUM  
KÜNSTLICHE INTELLIGENZ



**Prof. Dr.-Ing. Ulrich Schäfer**

Telefon: +49 (9621) 482-3623

E-Mail: [u.schaefer@oth-aw.de](mailto:u.schaefer@oth-aw.de)

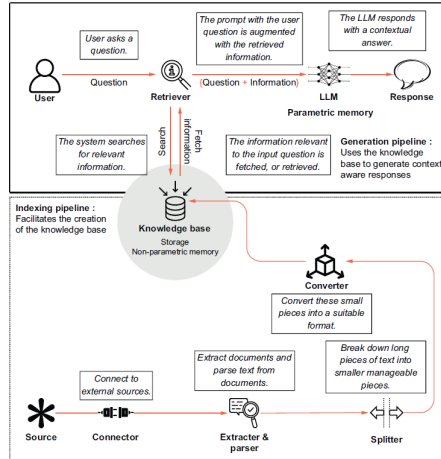


**Prof. Dr.-Ing. Christian Bergler**

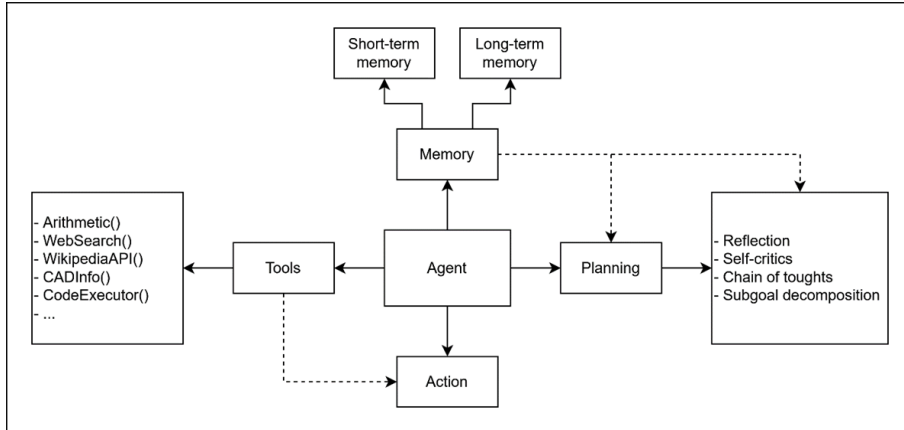
Telefon: +49 (9621) 482-3653

E-Mail: [c.bergler@oth-aw.de](mailto:c.bergler@oth-aw.de)

## Modelle

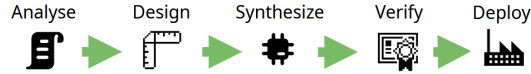


Quelle: A Simple Guide to Retrieval Augmented Generation [1]



Quelle: „LLM-powered Autonomous Agents“ [2]

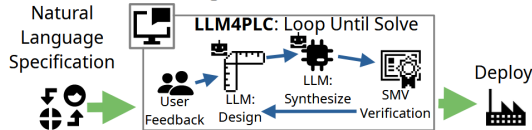
### Typical Workflow: High Engineering Effort



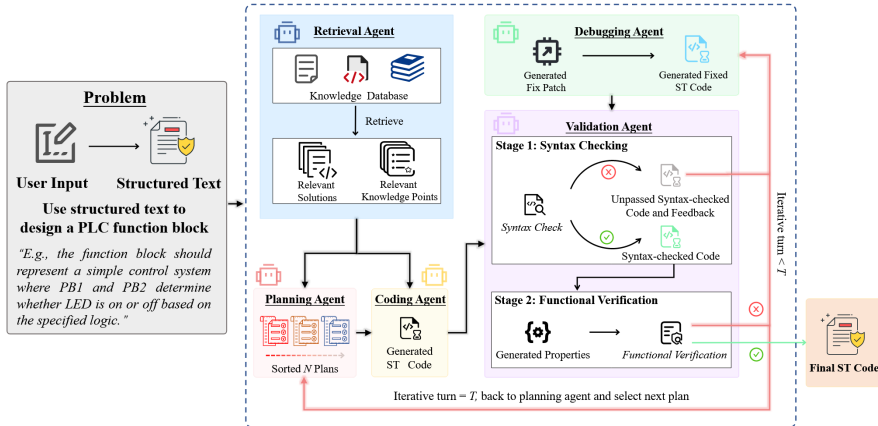
### Naive LLM Generation: Catastrophic Failure



### LLM4PLC: Engineer-Guided Automation



Quelle: „LLM4PLC: Harnessing Large Language Models for Verifiable Programming of PLCs in Industrial Control Systems“ [3]



Quelle: Agents4PLC: Automating Closed-loop PLC Code Generation and Verification in Industrial Control Systems using LLM-based Agents [4]

- [1] A. Kimothi, *A Simple Guide to Retrieval Augmented Generation*, First edition. Shelter Island, NY: Manning Publications, 2025, ISBN: 978-1-63343-585-8. Adresse: <https://learning.oreilly.com/library/view/-/9781633435858/>.
- [2] L. Weng, „LLM-powered Autonomous Agents“, *Lil'Log*, 06/2023. Adresse: <https://lilianweng.github.io/posts/2023-06-23-agent/>.
- [3] M. Fakihi, R. Dharmaji, Y. Moghaddas, G. Quiros, O. Ogundare und M. A. Al Faruque, „LLM4PLC: Harnessing Large Language Models for Verifiable Programming of PLCs in Industrial Control Systems“, in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*, Ser. ICSE-SEIP '24, Lisbon, Portugal: Association for Computing Machinery, 2024, S. 192–203, ISBN: 9798400705014. DOI: 10.1145/3639477.3639743. Adresse: <https://doi.org/10.1145/3639477.3639743>.
- [4] Z. Liu u. a., *Agents4PLC: Automating Closed-loop PLC Code Generation and Verification in Industrial Control Systems using LLM-based Agents*, 2024. arXiv: 2410.14209 [cs.SE]. Adresse: <https://arxiv.org/abs/2410.14209>.