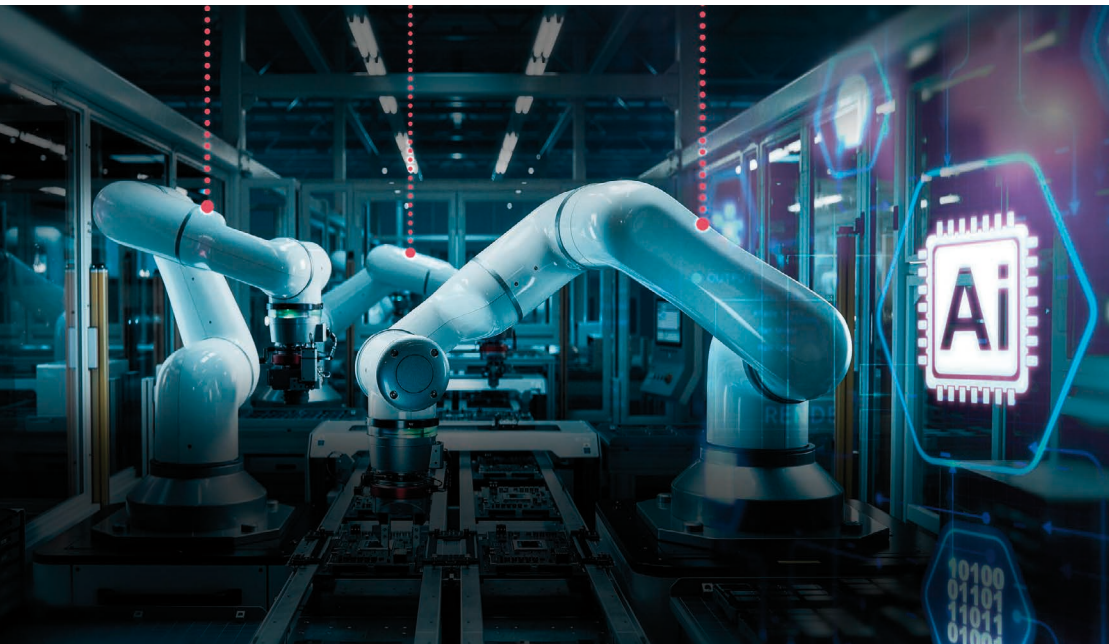


Warum KI die SPS-Entwicklung verändert

Von klassischer SPS-Programmierung zu KI-basierter Software Defined Automation



Warum programmieren wir SPS immer noch wie 1995?

Diese provokante Frage stelle ich mir, wenn ich mir das momentan vorherrschende Vorgehen bei der SPS-Programmierung anschau. Wer an die 90er-Jahre denkt, erinnert sich an AWL (Anweisungsliste), manuelle Speicheradressierung und starre Hardware-Kopplungen. Doch während sich die IT-Welt in den letzten drei Jahrzehnten mehrfach neu erfunden hat, dominiert in der SPS-Entwicklung trotz mittlerweile virtuellen Steuerungen oft noch immer die klassische „Fleißarbeit“. Es wird Zeit für neue Techniken und Innovationen.

Die Komplexität und die Anforderungen moderner Automatisierungslösungen wachsen exponentiell. Cloud-Anbindungen, IoT-Schnittstellen, Cybersecurity und immer mehr Software-Varianten treiben den Aufwand in die Höhe. Die Engineering-Budgets und vor allem die verfügbaren Fachkräfte wachsen jedoch, wenn überhaupt, nur linear. Die Produktivität wird zum Flaschenhals. Ingenieure verbringen zu viel Zeit mit dem Schreiben von Boilerplate-Code und Standard-Routinen, statt sich auf die Prozessoptimierung zu konzentrieren.

Wir brauchen moderne Werkzeuge, die den heutigen Anforderungen stand halten. KI kann hier viele Aufgaben übernehmen und damit dem Ingenieur Zeit ersparen. Sie will nicht den Menschen ersetzen, weil dessen Urteilsvermögen im Prozess dringend gebraucht wird.

Die Rolle von KI und Mensch

sind klar getrennt und ergeben zusammen das Ganze. Die KI übernimmt sich wiederholende Aufgaben, bedient Standardlogiken und erstellt Grundstrukturen. Der Ingenieur kümmert sich um die Architektur, die Sicherheit und die Validierung. Er trägt die Verantwortung, d. h. er hat mehr Zeit für echte Ingenieursarbeit.

Die Lösung

Die SPS wird Software: Der neue Lösungsansatz, den Unternehmen wie logiccloud verfolgen, bricht mit zwei Paradigmen der klassischen Automatisierung: der Bindung an proprietäre Hardware und der lokalen Entwicklungsumgebung. Die Steuerung wird zur „App“. Losgelöst von dedizierter SPS-Hardware läuft die Runtime auf beliebigen Industrie-PCs oder Edge-Gateways.

Die gesamte Entwicklungsumgebung wandert außerdem in den Browser, ist also per (deutscher oder privater) Cloud zugänglich. Es ist keine lokale Installation mehr nötig. Erst dieser Schritt in die Cloud schafft die Datengrundlage, um den nächsten Evolutionssprung zu zünden: den Einsatz von generativer künstlicher Intelligenz (GenAI) im Engineering-Prozess.

Architektur: Erst der Plan, dann der Code

Viele Entwickler nutzen bereits Tools wie ChatGPT oder andere Coding-Agents, um Code-Schnipsel zu erzeugen. Doch für die industrielle Automatisierung greift das zu kurz. Einer generischen KI fehlt das Verständnis für die IEC 61131-3 Normen, die Hardware-Restriktionen und vor allem für den spezifischen Projektkontext.

Warum reicht generische KI nicht aus?

Oder Was ist der Unterschied zwischen ChatGPT und einer Automatisierungs-KI? Generisch bedeutet allgemein. Die generische KI ist zu allgemein. Es fehlt der Bezug zur Automatisierungsaufgabe und der echten SPS-Umgebung. Außerdem gibt es keinen Kontext zur Programmiersprache. Da diese Bezüge fehlen, ist keine sinnvolle Validierung möglich. Die Automatisierung braucht eine domänenspezifische KI. Erschwerend kommt hinzu, dass ein plausibler Code nicht automatisch ein sicherer Code ist. Vor dem blinden Vertrauen in Standard-LLMs sei an dieser Stelle gewarnt.

Wie verhindert man das Halluzinieren der KI?

Die Automatisierung braucht domänenspezifische KI. Um der KI das Halluzinieren auszutreiben, befindet sich die KI direkt auf der Automatisierungsplattform. Sie wird speziell für die industrielle Automatisierung trainiert. Sie kennt das Projekt, die Codebasis und die Architektur.

Sie arbeitet im Kontext des Projektes und nicht im luftleeren Raum. Grundlage ist eine Pyramide aus internem Know-how, einer technischen Knowledge Base und darauf aufgesetzten KI-Modellen. Der Workflow ähnelt dabei der Arbeitsweise eines menschlichen Ingenieurs.

Der Prozess beginnt mit einer Projektbeschreibung in natürlicher Sprache (Text) oder einem Q&A-Dialog. Bevor eine Zeile Code geschrieben wird, werden die Anforderung in logische Teilbereiche zerlegt. Die KI entscheidet, welche Artefakte benötigt werden: Das eigentliche SPS-Programm, wiederverwendbare Komponenten, I/O-Definitionen und, ein entscheidender Unterschied zu reinen Code-Generatoren, auch die passenden HMI-Screens.

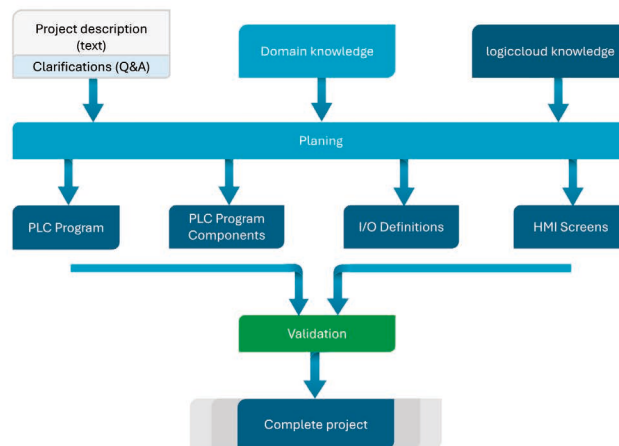
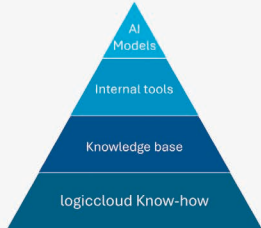
Danach wird validiert. Der generierte Output wird nicht ungeprüft übernommen. Nur so wird verhindert, dass syntaktisch korrekter, aber funktional unsinniger oder unsicherer Code in die Runtime gelangt.

Autor:
Bernhard Böhler
Gründer
logiccloud
<https://logiccloud.com/>

logiccloud Project Assistant



logiccloud Assistant



vertiefte SPS-Programmierkenntnisse ihre Anforderungen in Prosa beschreiben und eine funktionierende Basis-Applikation erhalten. Zudem soll die KI künftig nicht nur Software schreiben, sondern auch Hardware-Empfehlungen aussprechen. Basierend auf der Komplexität des Codes und den benötigten Zykluszeiten könnte das System vorschlagen, welche Steuerungshardware für das spezifische Projekt am wirtschaftlichsten ist.

Es ist der Schritt von der manuellen Manufakturarbeit hin zur industrialisierten Software-Erstellung. Oder anders gesagt: Das endgültige Update für das Jahr 1995.

Fazit

KI in der SPS-Entwicklung ist kein Trend. Es ist ein strategischer Wandel hin zu Effizienz, Innovation und Wettbewerbsfähigkeit in der industriellen Automatisierung. Während diese Tools sich weiterentwickeln, wird ihre Integration in Entwicklungsprozesse essenziell für Organisationen, die in einem Markt bestehen wollen, der Geschwindigkeit und Präzision belohnt.

Die Frage ist nicht, ob man sich mit KI-gestützter Entwicklung auseinandersetzen sollte – sondern ob man zu denen gehört, die lernen, sie gut zu nutzen, oder zu denen, die die nächsten fünf Jahre behaupten, es sei eine Modeerscheinung, während ihre Wettbewerber schneller liefern, schneller iterieren und die Märkte erobern, die man selbst aus Vorsicht verpasst hat. ◀

Der Netzwerkeffekt der Cloud

Ein oft unterschätzter technischer Aspekt ist die Datenhaltung. In der klassischen Automation schlummern Terabytes an wertvollem Projektwissen und Lösungsstrategien auf lokalen Festplatten von Laptops, isoliert in Silos. Für das Training von KI-Modellen sind diese Daten unzugänglich.

Durch den Cloud-Ansatz werden die Projekte zentralisiert. Das ermöglicht, natürlich unter Wahrung des Datenschutzes und mit Einverständnis, das Trainieren der Modelle auf einer immer breiteren Basis realer Automatisierungsmuster. Die KI lernt mit jedem Projekt dazu („Network Effect“), erkennt Muster in der Hardware-Anbindung oder bei Standard-Regelkreisen und wird so kontinuierlich präziser. Das unterscheidet das System von lokalen On-Prem-Lösungen, die jedoch genauso möglich seien.

Über die Reverse-Engineering-Funktionen kann der Assistent bestehenden Legacy-Code analysieren und automatisch verständliche Kommentare hinzufügen.

Das Pareto-Prinzip: Vom Coder zum Validierer

Das Ziel ist nicht, den Programmierer zu ersetzen, sondern ihn massiv zu entlasten. Es gilt das Pareto-Prinzip: Etwa 80 Prozent des Codes, darunter Standardfunktionen, I/O-Mappings und Grundstrukturen, übernimmt der KI-Assistent. Die verbleibenden 20 Prozent, also

Feinjustierung, Sicherheitsabnahmen und komplexe Prozesslogik, bleiben in der Hand des Ingenieurs.

Die Aufteilung ist klar: Die KI übernimmt das Mechanische, der Mensch das Bedeutungsvolle.

Dieser Workflow verändert seine Rolle: Er muss weniger tippen, aber mehr prüfen. Die Validierung der KI-Ergebnisse wird zur Kernkompetenz. Gleichzeitig löst die KI ein weiteres leidiges Thema der Branche: die Dokumentation. Programmierer programmieren lieber, als dass sie dokumentieren.

Die Vision

Das Projekt kann mit all seinen Anforderungen im Klartext geschrieben werden. Die KI versteht die Beschreibung sucht in den bereits

vorhandenen Daten nach ähnlichen und verwendbaren Passagen. Daraus erstellt sie das SPS-Programm mit seinen Funktionsbausteinen, definiert Ein- und Ausgänge und berücksichtigt auch das HMI. Das Ergebnis ist eine komplette Lösung, die dann vom Ingenieur validiert und anschließend kompiliert werden kann. Anschließend erfolgt die Umsetzung. So kann einfach und schnell ein ganzes Projekt erzeugt werden.

Ausblick: Demokratisierung der Automatisierung

Langfristig könnte die KI die Hürden für den Einstieg in die Automatisierung deutlich senken. Die Vision ist, dass künftig auch Verfahrenstechniker oder Heizungsbauer ohne

logiccloud ChatBot



User Interactions

Interactive prompt interface (chatbot)

Integrated IDE actions

Context

logiccloud Project

Knowledge and tools

logiccloud Knowledge Base

Internal tools

logiccloud Know-how

Results

Code analysis & optimization (What's wrong?)

Code generation - vibe coding (Add this new feature)

Logic reverse-engineering (What does this do?)

Tailored guidance (How to do this?)