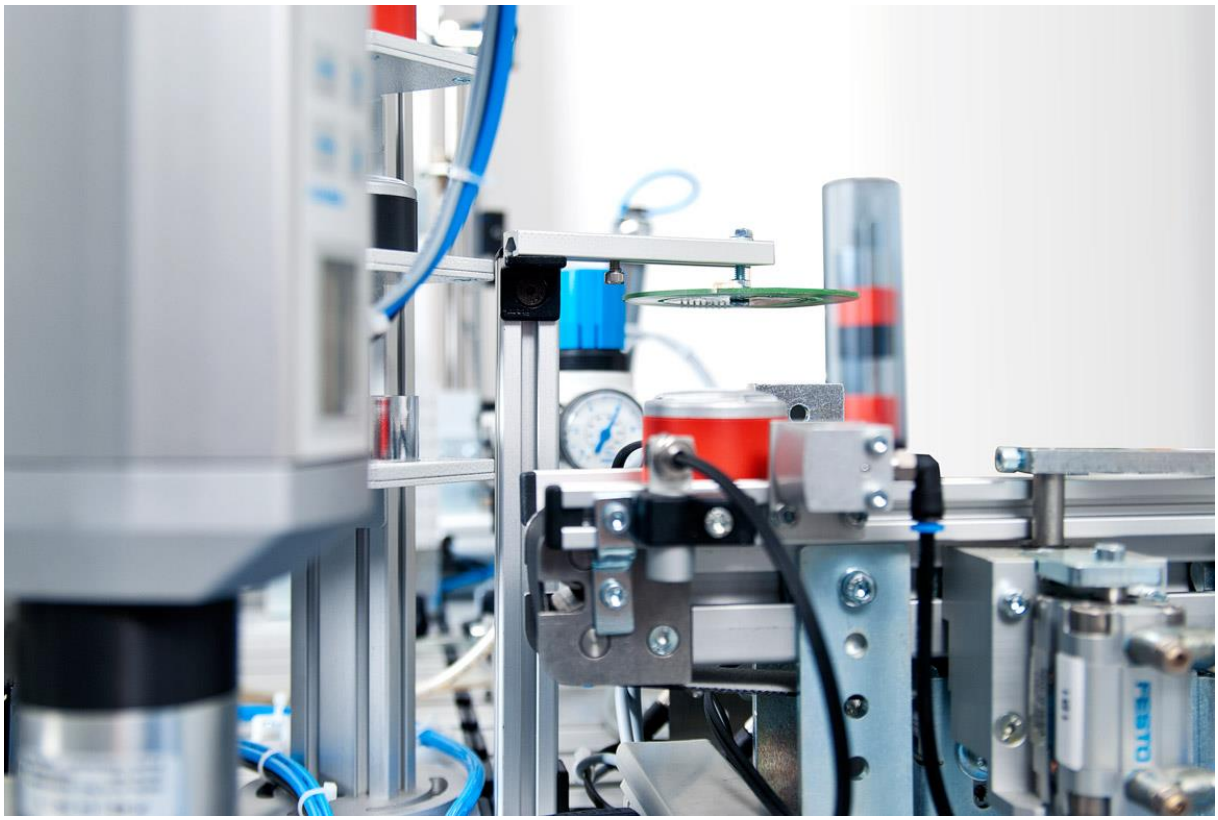


Michael Geisinger, Dr. Chih-Hong Cheng

Industrielle Steuerungsprogramme auf Knopfdruck

Die Idee klingt verlockend: Steuerungsprogramme einfach auf Knopfdruck erzeugen und nicht mehr aufwendig per Hand erstellen. Was steckt konkret hinter diesem Ansatz und wie lässt er sich in der Praxis umsetzen?

Moderne Produktionsanlagen zeichnen sich durch ein komplexes Zusammenspiel von „smarten“ Sensoren und Aktoren sowie leistungsfähigen Steuergeräten aus. Die Programmierung solcher Systeme erfordert spezialisierte Softwarewerkzeuge, die eine geeignete Parametrisierung des Systems erlauben und auf die Anforderungen der Entwickler zugeschnitten sind. Obwohl es dank dieser Werkzeuge noch nie so einfach war, derart komplexe Systeme zu projektieren, zu „orchestrieren“ und zu überwachen, erfordert deren Benutzung dennoch viel Erfahrung. Daher ist der vorherrschende Entwicklungsprozess dadurch gekennzeichnet, dass die Software für die Anlage von Experten erstellt und an neue Anforderungen angepasst wird.

In einer Zeit, in der Produkte zunehmend individualisiert werden und Produktlebenszyklen und damit die Intervalle zwischen der Anpassung von Steuerungsprogrammen immer kürzer werden, stößt diese Entwicklungsmethodik jedoch an ihre Grenzen. Die Steuerungsprogramme enthalten häufig implizite Design-Entscheidungen, die zum Zeitpunkt der Erstellung optimal waren, sich jedoch im Fall von späteren Anpassungen als nachteilig erweisen können. Die Extraktion solcher Design-Entscheidungen aus dem Steuerungsprogramm ist nicht einfach. Zusätzlich wollen heutzutage auch kleine und mittelständische Unternehmen die Fertigung ihrer Kleinserien automatisieren. Da solche Firmen meist keinen Automatisierungsexperten beschäftigen, müssen die eingesetzten Anlagen einen hohen Grad an Modularität und Wandlungsfähigkeit aufweisen, um kosteneffizient zu sein.

Speziell für Szenarien, in denen es häufig zu Veränderungen an der Steuerung des Produktionssystems

kommt, wurde daher von der Forschungseinrichtung Fortiss, einem An-Institut der Technischen Universität München, ein neuartiger Ansatz entwickelt, bei dem das benötigte Steuerungsprogramm aus entsprechenden Modellen auf Knopfdruck synthetisiert wird. Der Knackpunkt ist hierbei, dass durch eine mehrschichtige Modellierung die Komplexität stufenweise reduziert wird. Die höchste Stufe erlaubt es, die in der Anlage durchzuführende Produktionsaufgabe in einem mathematischen Formalismus zu spezifizieren. Auf Basis der Modellierung erfolgt dann die Erstellung des Steuerungsprogrammes.

Die Produktionsanlage spielt „Schach“

Kern der Synthese ist der spielbasierte Problemlöser GAVS+ [1]. Darauf aufbauend wurde das Werkzeug MGSyn (Model, Game, Synthesis) entwickelt, das den GAVS+ mit geeignet transformierten Eingabedaten aus den Modellen versorgt. Man stelle sich dazu die Automatisierungsaufgabe wie ein Schachspiel vor: Die Züge des Spielers mit den weißen Figuren entsprechen den in der Anlage ausgeführten Produktionsschritten, während die Züge des schwarzen Spielers die jeweiligen Reaktionen der Umgebung repräsentieren. Beispielswiese kann Weiß (das Steuerungsprogramm) in einem Zug nach einer physikalischen Eigenschaft eines Werkstücks „fragen“ und Schwarz (die Umgebung) beantwortet die Frage, indem sie einen ihrer möglichen Züge tätigt. Die Umgebung kann auch Fehlerfälle verursachen, sofern diese als gültige Züge modelliert sind. Genau wie zwei Schachspieler strategisch planen und auf die Aktionen des Gegners eingehen müssen, muss auch die Produktionsanlage auf die Eigenschaften des Produkts, das Fortschreiten des Produktionsprozesses und mögliche Fehlerbedingungen reagieren, um die Produktionsaufgabe erfolgreich durchführen zu können. Plakativ gesprochen geht es also darum, Schwarz schachmatt zu setzen – der Umgebung also jegliche Möglichkeit zu nehmen, das Spiel für sich zu entscheiden.

Das Schachspiel ist jedoch nur ein anschaulicher Vergleich. Tatsächlich funktioniert ein solcher Ansatz nur dann, wenn die Komplexität des zu lösenden Problems deutlich geringer ist als bei einem Schachspiel. Denn wie allseits bekannt ist, sind optimale Schachzüge selbst mit sehr schnellen Computersystemen meist nicht zu berechnen.

Glücklicherweise ist es beim Spiel zwischen Automatisierungsanlage und Umgebung bei geeigneten Einschränkungen möglich, in annehmbar kurzer Zeit (wenige Sekunden bei einfachen Problemen bis wenige Minuten bei komplexeren) Steuerungsprogramme zu synthetisieren, die bezüglich der angegebenen Kriterien optimal sind. Konkret heißt das, dass die Steuerungsprogramme die erfolgreiche Abarbeitung der Aufgabe im Rahmen der Modellierung garantieren.

Die Synthesezeit hängt maßgeblich von der Anzahl der möglichen Züge ab, sprich von der Menge der ausführbaren Aktionen der Steuerung und der Umgebung. Modulare Fertigungsstraßen (vgl. Bild 1) mit diskretem Zustandsraum, bei denen sich die Position und Eigenschaften von Werkstücken, die Anzahl der verbauten Aktoren und Sensoren sowie die möglichen Fehlerfälle einfach beschreiben lassen, eignen sich daher besonders gut für diesen Ansatz.

Die Beschreibung der Prozesse

Um automatisch Steuerungsprogramme synthetisieren zu können, müssen zunächst alle relevanten Eigenschaften der Anlage in einem formalen Modell repräsentiert werden. Dabei handelt es sich um eine Sprache, in der die Bedeutung der einzelnen „Wörter“ genau definiert ist und die somit ein Computer automatisch verarbeiten kann.

Das Modell stellt meist eine erhebliche Vereinfachung der Realität dar, da seine Mächtigkeit im Zielkonflikt mit der Synthesezeit steht. Beim vorgestellten Ansatz besteht das Modell aus drei Ebenen: dem Hardwaremodell, dem Anlagenmodell und dem Aufgabenmodell [2].

Das Hardwaremodell beschreibt die Eigenschaften der verbauten Hardwaremodule. Jedes Modul wird dabei durch eine Menge von Arbeitspositionen und einer Menge von Aktionen charakterisiert. Die Arbeitspositionen repräsentieren Orte im jeweiligen Modul, an denen sich Werkstücke befinden können. Die Aktionen erlauben es, die verschiedenen Arbeitsschritte zu modellieren, die das Modul ausführen kann.

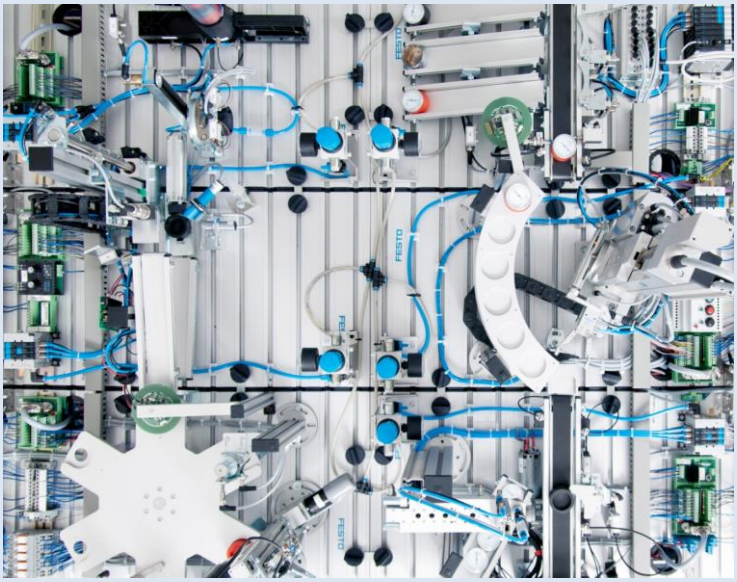


Bild 1: Modulare Fertigungsstraßen eignen sich aufgrund ihrer Wandlungsfähigkeit besonders gut für automatische Synthese.

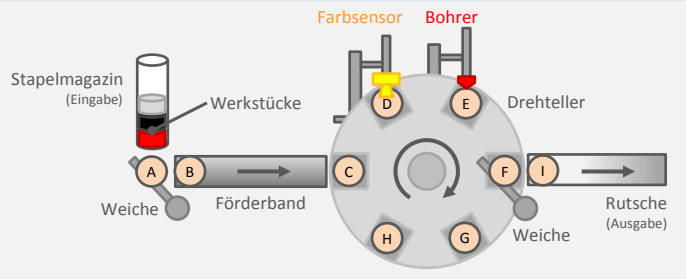


Bild 2: Anlagenmodell eines einfachen Produktionsprozesses.

Ein unidirektionales Förderband ist im einfachsten Fall durch zwei Arbeitspositionen und eine Aktion „bewege-Band“ charakterisierbar, die ein Werkstück von Position 1 nach Position 2 transportiert (siehe Tabelle 1). Entsprechend können einfache Bearbeitungsschritte, wie das Bohren eines Lochs in ein Werkstück, durch ein Modul „Bohrer“ mit einer Arbeitsposition und der Aktion „bohre-Werkstück“ beschrieben werden. Die Modellierung von Sensoreingaben erfolgt ähnlich: Die Erkennung, welche Farbe ein Werkstück hat, wird mittels der Aktion „erkenne-Farbe“ im Modul „Farbsensor“ abgebildet.

Im Anlagenmodell werden die Hardwaremodule zu einer konkreten Anlage kombiniert. Eine Anlage kann zum Beispiel aus den in Bild 1 dargestellten Komponenten bestehen. Um die Anordnung der Module im Modell zu erfassen, wird definiert, welche Arbeitspositionen jeweils „überlappen“. In der Beispielanlage überlappt Position 2 der Weiche mit Position 1 des Förderbands – hier mit „B“ gekennzeichnet. Gleiches gilt beispielsweise für Position 2 des Förderbands und die linke Position des Drehtellers (C). Aus dieser Definition und den zur Verfügung stehenden Aktionen lässt sich automatisch der mögliche Materialfluss ableiten.

Das Aufgabenmodell erlaubt schließlich die Spezifikation einer abstrakten Arbeitsaufgabe. Dabei werden Anfangszustand und gewünschter Endzustand benannt. Die informelle Beschreibung einer möglichen Aufgabe in der Beispielanlage ist: „Gegeben ein Werkstück namens ‚WS‘ an Position ‚A‘ (unteres Ende des Stapelmagazins), soll der Endzustand sein, dass sich das Werkstück an Position ‚I‘ befindet (oberes Ende der Rutsche) und genau dann gebohrt ist, wenn es rot ist.“ Diese Aufgabe kann wie folgt notiert werden:

- Anfangszustand: Position(WS, A)
- Endzustand: Position(WS, I) ∧ (gebohrt(WS) ⇔ Farbe(WS, rot))

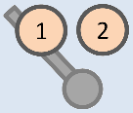
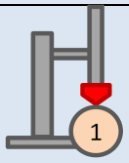
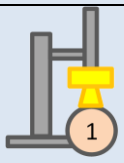
Die Konstrukte „Position“, „Farbe“ und „gebohrt“ stammen dabei aus dem Hardwaremodell und repräsentieren den Zustand der Anlage und der darin befind-lichen Werkstücke zu einem gewissen Zeitpunkt. Für die Spezifikation der drei Modelle wurde das frei verfügbare Entwicklungswerkzeug MGSyn [3] erstellt (siehe Bild 3).

Synthese von Steuerungsprogrammen

Nun kann das „Spiel“ beginnen. Auf Knopfdruck werden die vorhandenen Aktionen dazu zunächst in Züge von Steuerungsprogramm und Umgebung mit entsprechenden Vor- und Nachbedingungen umgewandelt. Im konkreten Fall kann das Steuerungsprogramm die Weichen, das Förderband und den Drehteller bewegen, den Bohrer auslösen und die Umgebung nach der Farbe eines Werkstücks fragen, sofern dieses sich beim Farbsensor befindet. Die Umgebung wartet auf eine Anfrage nach der Farbe und hat dann als Antwort drei „Züge“ zur Verfügung, die den drei möglichen Farben Schwarz, Rot und Silber entsprechen. Der spielbasierte Löser übernimmt nun die Generierung eines Steuerungsprogramms, das in der Lage ist, auf die Eingaben aus der Umgebung (modelliert als Züge des Umgebungsspielers) während der Ausführung zu reagieren. Hier besteht ein wesentlicher Unterschied zur traditionellen Planung, bei der in der Regel alle Parameter bereits vorab bekannt sind und die abzuarbeitenden Schritte im Voraus festgelegt werden.

Schlussendlich wird im Beispiel ein Steuerungsprogramm generiert, das zunächst die Farbe des Werkstücks testet und dann das Werkstück abhängig vom Ergebnis bohrt. Sollte es für eine gegebene Aufgabe keine Lösung geben, so wird dies automatisch erkannt und entsprechend rückgemeldet. Das entwickelte Werkzeug liefert jedoch nicht nur eine Lösung für die gegebene Aufga-

Tabelle 1: Module einer Produktionsanlage werden im Hardwaremodell abstrakt beschrieben.

Hardware-Modul	 Förderband	 Weiche	 Bohrer	 Farbsensor
Verfügbare Aktionen	bewege-Band (von Position 1 nach 2)	aktiviere-Weiche (von Position 1 nach 2)	bohre-Werkstück (an Position 1)	erkenne-Farbe (an Position 1)
Zugeordnet zu	Steuerung (Aktion)	Steuerung (Aktion)	Steuerung (Aktion)	Umgebung (Sensoreingabe)

be, sondern beinhaltet auch Mechanismen, um diese Lösung sowohl zu simulieren als auch auf der Zielhardware zur Ausführung zu bringen. Die Simulation erlaubt es, Anlagen zu evaluieren, die es in der Realität (noch) nicht gibt. Die konkrete Ausführung auf der Zielhardware unterstützt handelsübliche Steuerungen von Siemens und Festo, die mittels OPC und Siemens-MPI angebunden sind. Zusätzlich werden mittels Code-Generierung eingebettete mikrocontrollerbasierte Zielplattformen unterstützt.

Um die synthetisierten Steuerungsprogramme zu optimieren, wurden des Weiteren Mechanismen untersucht, die die gleichzeitige Ausführung von meh-reren Aktionen und die Auswahl eines optimalen Steuerungsprogramms aus der Menge aller möglichen Lösungen auf der Basis von geeigneten Kostenmodellen erlauben.

Die Stärken dieses Verfahrens zeigen sich, wenn sich die auszuführende Aufgabe ändert: Da die Beschreibung der Aufgabe im Gegensatz zur traditionellen Programmierung auf einer sehr abstrakten Ebene stattfindet, reicht eine geringe Einarbeitungszeit, um diese abändern zu können. Wichtig ist hierbei, dass das Aufgabenmodell keine konkrete Umsetzung der Aufgabe benennt, sondern vielmehr deren Anforderungen auflistet. Es wird also spezifiziert, was die Anlage tun soll und nicht wie sie es tun soll. Dies vermeidet, dass Design-Entscheidungen tief in das Steuerungsprogramm eingebettet werden. Die formale Spezifikation im Hardwaremodell könnte dabei der Hersteller der Hardwaremodule liefern, ähnlich wie Hersteller von Computerperipherie entsprechende Treiber zur Verfügung stellen.

Reif für die industrielle Praxis?

Die Einsatzfähigkeit der automatischen Synthese hängt maßgeblich davon ab, wie umfangreich die Modelle sind und wie optimal die resultierenden Steuerungsprogramme sein sollen. Im Allgemeinen steigt die Synthesezeit exponenziell mit der Größe des Zustandsraums. Handelt es sich um komplexe Systeme, so ist es sinnvoll, diese in verschiedene Teilsysteme zu zerlegen, die dann einzeln betrachtet und schneller synthetisiert werden können.

Eins ist sicher: Die automatische Synthese wird in naher Zukunft die manuelle Programmierung sicher nicht ersetzen. Selbst mit solchen Ansätzen sind noch Experten nötig, die das Hardware- und Anlagenmodell erstellen und die lokalen Regelschleifen der modellierten Aktionen umsetzen. Denn Modelle sind nur so gut wie die Informationen darin – und gleiches gilt für die daraus synthetisierten Steuerungsprogramme.

Die Modelle sollten daher als unmissverständliche Dokumentation der auszuführenden Produktionsaufgabe verstanden werden, die sich bei Bedarf auch automatisch verarbeiten lässt. Dennoch eröffnet dieser Ansatz sogar kleinen und mittelständischen Un-ternehmen die Chance, die Software für einfache Automatisierungsaufgaben durch Erstellung beziehungsweise Anpassung des Aufgabenmodells selbst zu konfigurieren und damit Kosten zu sparen.

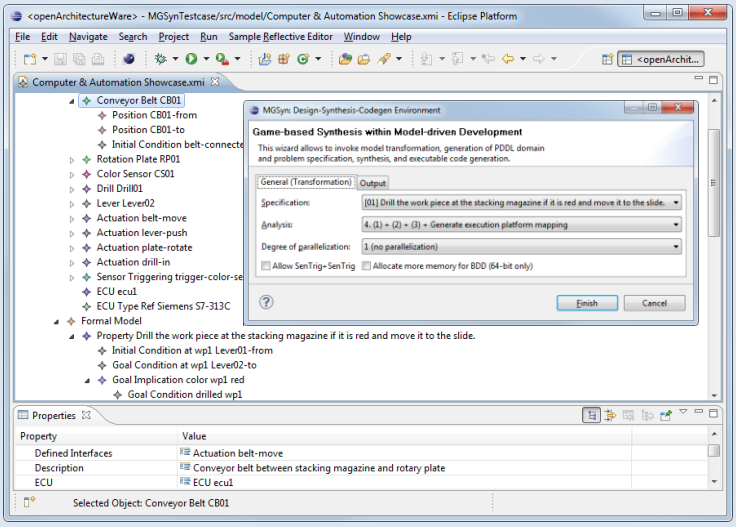


Bild 3: Hardware-, Anlagen- und Aufgabenmodell werden mit dem Entwicklungswerkzeug MGSyn spezifiziert.

Michael Geisinger arbeitet als wissenschaftlicher Mitarbeiter am fortiss-Institut der TU München.

Dr. Chih-Hong Cheng ist am fortiss-Institut der TU München als Gruppenleiter tätig.

[1] C.-H. Cheng, A. Knoll, M. Luttenberger und C. Buckl, „GAVS+: an Open Platform for the Research of Algorithmic Game Solving," in *Proc. 17th Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'11)*, Saarbrücken, Deutschland, 2011.

[2] C.-H. Cheng, M. Geisinger, H. Ruess, C. Buckl und A. Knoll, „Game Solving for Industrial Automation and Control," in *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA'12)*, St. Paul, MN, USA, 2012.

[3] C.-H. Cheng und M. Geisinger, „MGSyn - Automatic Synthesis for Industrial Automation," 2013. [Online]. Available: http://mgsyn.fortiss.org/.