

Einsatz von Augmented Reality im Ramp-Up Prozess von automatisierten Fertigungssystemen

Dipl.-Wirt.-Ing. R. Eckes

Heinz Nixdorf Institut, Universität Paderborn

Fürstenallee 11, D-33102 Paderborn

Tel.: 05251/60-6267, Fax.: 05251/60-6268

E-mail: raimund.eckes@hni.uni-paderborn.de

Dipl.-Inform. Robert Wagner

Institut für Informatik, Fachgebiet Softwaretechnik, Universität Paderborn

Warburger Str. 100, D-33098 Paderborn

Tel.: 05251/60-3307, Fax.: 05251/60-3530

E-mail: robert.wagner@uni-paderborn.de

Zusammenfassung

Ein Problem der produzierenden Industrie ist die Dauer des Ramp-Up Prozesses von automatisierten Fertigungssystemen. Bisher ist es üblich, die nicht vollständig getestete Steuerungssoftware erst im Ramp-Up des Fertigungssystems zu validieren. Im Ramp-Up kommen die ungetestete Steuerungssoftware, teilweise fehlerhaft errichtete Komponenten und schlimmstenfalls Konstruktionsfehler zusammen. Dies und die steigende Komplexität der Fertigungssysteme, die durch den zunehmenden Einsatz von Steuerungstechnik bedingt ist, führen zu einem problembehafteten Ramp-Up Prozess mit einer lange dauernden Fehleranalyse. Zur Unterstützung der interdisziplinären Fehleranalyse wird in diesem Beitrag ein AR-System vorgestellt, das Zustände aus den verschiedenen Disziplinen (Maschinenbau, Elektrotechnik, Informatik) im realen Fertigungssystem visualisiert.

Schlüsselwörter

Augmented Reality (AR), Ramp-Up, Inbetriebnahme, Hochlauf, Fehleranalyse

1 Der Ramp-Up Prozess

Im folgenden Kapitel wird der Ramp-Up Prozess definiert und die Probleme analysiert, die sich beim Ramp-Up automatisierter Fertigungssysteme ergeben.

1.1 Definition

Der Ramp-Up Prozess (Anlaufprozess) setzt sich aus den Phasen Inbetriebnahme und Hochlauf zusammen. Die Inbetriebnahme stellt

„... die Funktionsbereitschaft und das funktionale Zusammenwirken der zuvor montierten Einzelkomponenten her und prüft die Korrektheit der Einzelfunktionen sowie deren funktionales Zusammenwirken. Das Ergebnis der Inbetriebnahme ist eine abnahmefertige, technisch funktionsfähige Anlage.“

In der sich anschließenden Hochlaufphase

„... wird die Anlage beim Nutzer unter ihren nominellen, personellen, organisatorischen und technischen Randbedingungen auf eine dauerhafte Nennleistung gebracht. Dazu werden im Rahmen der Optimierung und Stabilisierung des Betriebsverhaltens insbesondere die jetzt erst erkennbaren technischen, personellen und organisatorischen Unzulänglichkeiten und technischen Frühausfälle behoben.“ [Zeu98]

Der Lebenszyklus von Produktionssystemen in Abbildung 1 bezieht sich auf Produktionssysteme zur Serienproduktion. So werden nach der Fertigung / Vormontage kein Funktionstest und keine Inbetriebnahme beim Hersteller durchgeführt. Die Prozesse Montage, Inbetriebnahme, Hochlauf und Normalbetrieb, werden beim Kunden / Betreiber der Anlage durchgeführt.

Ein Ramp-Up Prozess direkt beim Kunden / Betreiber ist als besonders kritisch zu sehen. Das Inbetriebnahme-Team muss vor Ort sein und hat nur eingeschränkten Zugriff auf die Ressourcen des Herstellers. Jede Verzögerung, offensichtliche Probleme und durchgeführte Arbeiten / Modifikationen sind für den Kunden transparent.

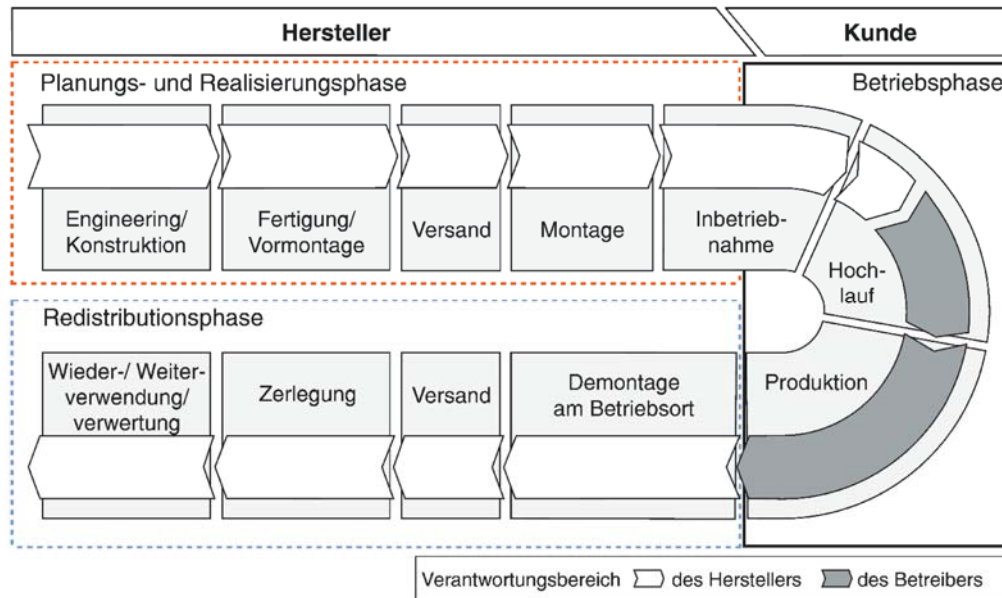


Abbildung 1: Die Betriebsphasen im Lebenszyklus von Produktionssystemen [WES+02]

Für den Ramp-Up Prozess fehlt eine Systematik mit definierten Prozessen und Methoden. Die steigende Komplexität der Produktionssysteme und die fehlende wissenschaftliche Erschließung des Ramp-Up Prozess machen die Durchführung zu einer hoch komplexen Aufgabe [WES+02]. Dazu sind interdisziplinäre Teams notwendig. Die im Ramp-Up Prozess notwendige Qualifikation stellt hohe Anforderungen an die Mitarbeiter. Die fachliche Ausrichtung des Personals spannt sich von den technischen Details für die Inbetriebnahme bis hin zum Know-How über Bedienung und Wartung des Fertigungssystems im Betrieb. Hier ist ein Wissenstransfer von den Entwicklern zu den Betreibern des Systems notwendig. Die intensive Kommunikation der beteiligten Partner und die Unterstützung durch ein Wissensmanagement ist für ein erfolgreiches Anlaufmanagement wichtig [ZM04], [SHK+04], [WES+02].

Die wichtigsten Faktoren, die die Wirtschaftlichkeit des Ramp-Up Prozesses bestimmen, sind Zeit und Kosten. Ein erfolgreicher Anlaufprozess ist die Voraussetzung für den Start der Produktion, d.h. für den wertschöpfenden Einsatz des Produktions- bzw. Fertigungssystems. Die Variantenvielfalt der Produkte nimmt zu und damit einhergehend steigt die Anzahl der Produktanläufe. Daher wird der Ramp-Up innerhalb der Produktentstehung als entscheidende Schlüsselstelle gesehen und bietet ein hohes Optimierungspotenzial. Den nachhaltigen Einfluss des Faktors Zeit im Produktlebenszyklus und dessen Auswirkung auf die Gesamrentabilität eines Produktes sind in Abbildung 2 skizziert.

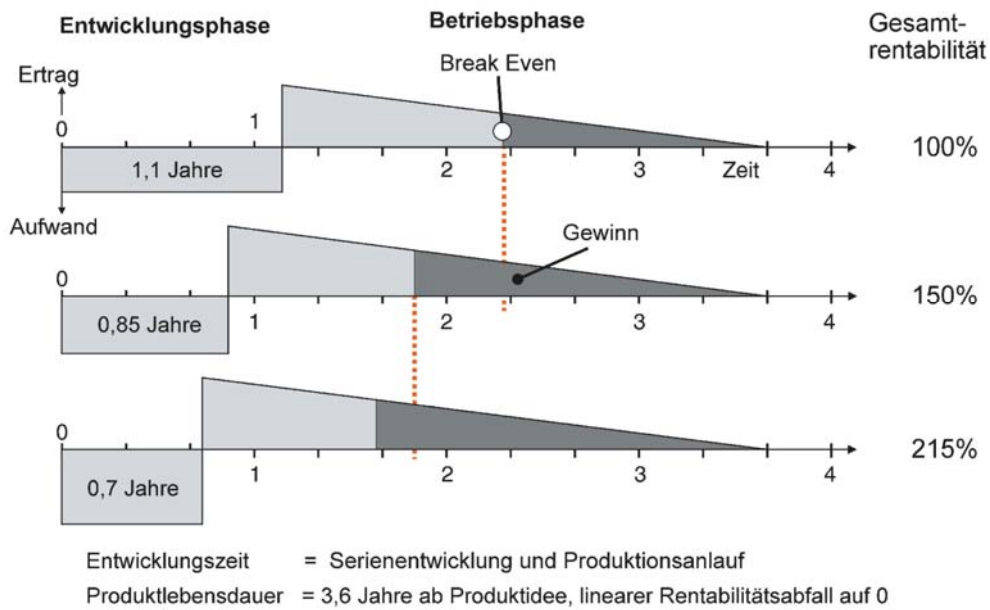


Abbildung 2: Auswirkung der Serienentwicklungs- und Anlaufphase auf die Gesamtrentabilität [WHW02]

Durch die Verkürzung der Anlaufphase kann die Rentabilität deutlich gesteigert werden, weil das Produkt bei konstanter Produktlebensdauer (ab Produktidee) früher am Markt angeboten werden kann. In den frühen Phasen des Produktlebenszyklus werden die höchsten Preise, d.h. die größten Gewinne erzielt [WHW02].

1.2 Problemanalyse

Arbeitsgrundlage für den Ramp-Up Prozess ist eine hinreichende Beschreibung des Fertigungssystems. Zur ordnungsgemäßen Durchführung der Inbetriebnahme und der Fehlersuche werden zahlreiche Informationen über die Funktionsweise des Systems sowie der kinematischen und steuerungstechnischen Abläufe der einzelnen Baugruppen benötigt. Die Entwicklung in der Konstruktion wird komponentenbasiert (mechanisch) und in der Steuerungstechnik signalorientiert durchgeführt. Dies ist ein Bruch, der sich in einer sequenziellen Entwicklung widerspiegelt. Diese sequenzielle Entwicklung der Mechanik und Steuerungstechnik birgt Probleme in sich, die negative Auswirkungen bis in die Inbetriebnahme haben können [Aßm96, S. 14f], [SL00]:

- Hohe Änderungshäufigkeit und unzureichendes Änderungswesen
- Unzureichende Beschreibung der Funktionen (viele Rückfragen und Missverständnisse)

- Ungeregelte Informationshaltung und -weitergabe
- Kommunikationsschwierigkeiten aufgrund organisatorischer und räumlicher Trennung
- Unterschiedliche Ausbildung, Denkweise und Begriffsdefinitionen
- Hoher Zeitdruck wegen zahlreicher „Nebenaufgaben“ (z.B. Angebots-, Inbetriebnahme- und Serviceunterstützung)
- Unzureichende zeitliche Koordination der Arbeiten und überwiegend sequenzielle Arbeitsweise
- Mangelnde Einbeziehung der Elektrokonstruktion / Softwareentwicklung in die Konzeptphase

Die Inbetriebnahme ist die entscheidende Phase in der Realisierung von automatisierten Fertigungssystemen, da erst hier das ordnungsgemäße Zusammenwirken der mechanischen, elektrischen, hydraulischen / pneumatischen und steuerungstechnischen Funktionen des Systems geprüft werden kann. Bei der Inbetriebnahme wird 90% der Zeit für die Steuerungstechnik aufgewendet. Davon entfallen wiederum 70% auf die Suche und Behebung von Fehlern in der Software, meist in SPS¹-Programmen. Ursachen hierfür sind die unzureichende Spezifikation der Steuerungsaufgabe in den frühen Phasen der Entwicklung und unzureichender, abteilungsübergreifender Informationsfluss [Aßm96, S. 16]. Die Verzögerungen sind proportional zum wachsenden Einsatz von Software [ZW05].

Die Validierung der Steuerungssoftware scheitert an der Komplexität und an den eingeschränkten Entwicklungs- und Testwerkzeugen der klassischen verknüpfungsorientierten Programmiermethoden. Dies hat zur Konsequenz, dass Tests mit der entwickelten Steuerungssoftware meist erst im errichteten Fertigungssystem durchgeführt werden können [SL00].

2 Entwicklung der Steuerungssoftware

Im vorangegangenen Kapitel wurden mögliche Ursachen für Verzögerungen bei der Inbetriebnahme aufgezeigt. Dabei wurde die entwickelte Steuerungssoftware als eine Ursache identifiziert, da die Steuerungssoftware meist erst im errichteten Fertigungssystem getestet werden kann. Um die benötigte Zeit für eine Inbetriebnahme zu verkürzen ist es daher besonders wichtig, den Ingenieur auch bei der Inbetriebnahme der zugehörigen Steuerungssoftware zu unterstützen. Hierzu wird im Folgenden an einer Fallstudie eine Methode zur Entwicklung von Steuerungs-

¹ Speicherprogrammierbare Steuerung

software vorgestellt, auf deren Basis eine Unterstützung bei der Inbetriebnahme ermöglicht wird.

2.1 Die Fallstudie

Bei der verwendeten Fallstudie handelt es sich um ein flexibles Fertigungssystem, das aus mehreren CNC-Maschinen, Industrierobotern und einem automatisierten Materialflusssystem besteht. Das verwendete Materialflusssystem wird von der Firma Montech angeboten. Es besteht aus Grundkomponenten, aus denen sich auf die Anwendung angepasste Transportlösungen konfigurieren lassen. Grundelemente sind beispielsweise das Schienensystem, die Transportwagen (Shuttles), die Weichen und die Stationen (vgl. Abbildung 3).

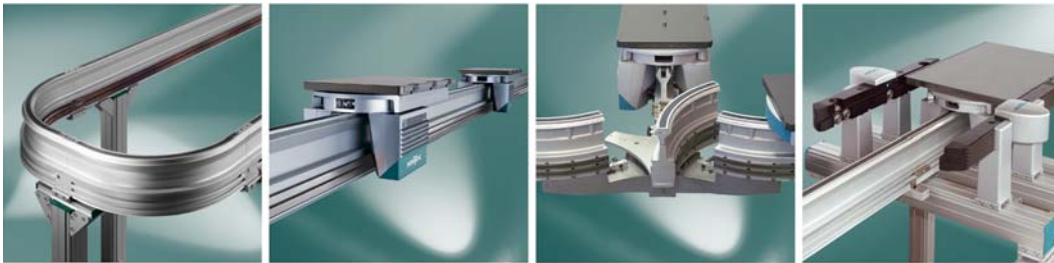


Abbildung 3: *Materialflusssystem Montrac von Montech (www.montech.ch)*

Verzweigungen innerhalb des Montrac Materialflusssystems werden durch Weichen realisiert, die in zwei Richtungen schalten können. In Abbildung 4 ist eine Weiche mit ihren Komponenten dargestellt. Die Richtung wird durch eine Drehbewegung eines gekrümmten Schienenteils gestellt. Eine Weiche hat einen Schnappverschluss (*lock*), der diese in den beiden Endlagen arretiert und zwei Sensoren (*round_sensor*, *straight_sensor*), die die jeweilige Endlage abtasten. Die Drehbewegung wird pneumatisch durch einen doppelt wirkenden Zylinder erzeugt. Der Schnappverschluss hat einen einfach wirkenden Zylinder mit Rückstellfeder. Zum Öffnen dieses Verschlusses wird der Zylinder mit Druckluft beaufschlagt.

Es existieren zwei unterschiedliche Typen der Weiche. Beide können Shuttles auf einer Hauptschleife gerade aus passieren lassen. Die verteilende Weiche (Brancher) führt Shuttles aus der Hauptschleife heraus und die zusammenführende Weiche (Joiner) führt Shuttles wieder in die Hauptschleife zurück. Dieser Unterschied ist auf der betrachteten Steuerungsebene nicht relevant. Das Funktionsprinzip und somit die Steuerung beider Weichen sind identisch. Grundsätzlich befindet sich eine Weiche im arretierten Zustand, damit die korrekte Position während der Durchfahrt eines Shuttles gewährleistet ist.

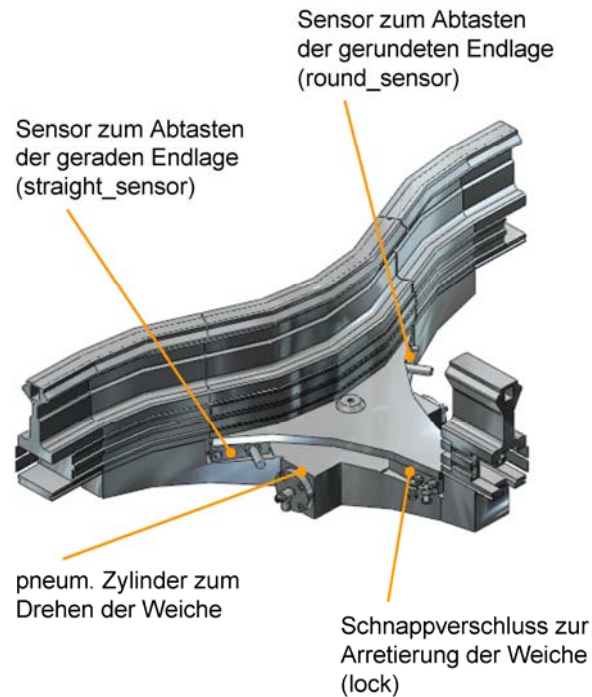


Abbildung 4: Montrac Weiche mit Sensoren und Aktoren

Zur Steuerung eines Schaltvorgangs der Weiche muss also zuerst der Schnappverschluss gelöst und die Drehbewegung durch das Aktivieren des pneumatischen Zylinders initiiert werden. Abhängig von der Drehrichtung muss nach dem Auslösen eines Endlagesensors die Weiche durch Betätigen des Schnappverschlusses wieder arretiert werden. Im Folgenden wird die Entwicklung der Steuerungssoftware für diese Steuerungsaufgabe beschrieben.

2.2 Modellierung der Steuerungssoftware

Die Entwicklung von Steuerungssoftware für reaktive Systeme erfolgt in der Informatik häufig mit Zustandsdiagrammen. Ein Zustandsdiagramm besteht aus Zuständen und Zustandsübergängen. Die Zustände in einem Zustandsdiagramm repräsentieren mögliche Zustände der Steuerungssoftware; die Zustandsübergänge definieren dabei mögliche Zustandsänderungen der Steuerungssoftware, die aufgrund von erfüllten Bedingungen und eintretenden Ereignissen stattfinden können.

Das Zustandsdiagramm zur Steuerung einer Weiche ist in Abbildung 5 dargestellt. Im Folgenden werden die Zustandsübergänge der Weichensteuerung beim Schalten der Weiche von der geraden (Zustand: *straight*) zur gerundeten Endlage (Zustand: *round*) beschrieben. Das Schalten von der gerundeten Endlage zur geraden erfolgt analog. Der Zustand *straight* ist so lang aktiv, bis das Ereignis *round()* ein-

tritt. Dann wird der Schnappverschluss geöffnet (Zustand: *straight unlocked*). Anschließend wird Druck auf den Zylinder zur Drehbewegung gegeben (*valve_def := false*). Die Weiche beginnt mit der Drehbewegung und verlässt die gerade Endlage. Die Weiche führt nun die Drehbewegung aus (Zustand: *switching round*). Der Sensor *round_sensor* liefert das Signal, dass sich die Weiche in der neuen Endlage befindet. Die Weiche wird arretiert (Zustand: *round*). Wird die Drehung (Zustand: *switching round*) nicht innerhalb von 2000 ms abgeschlossen, wird ein Fehlerzustand gewechselt (Zustand: *failure*).

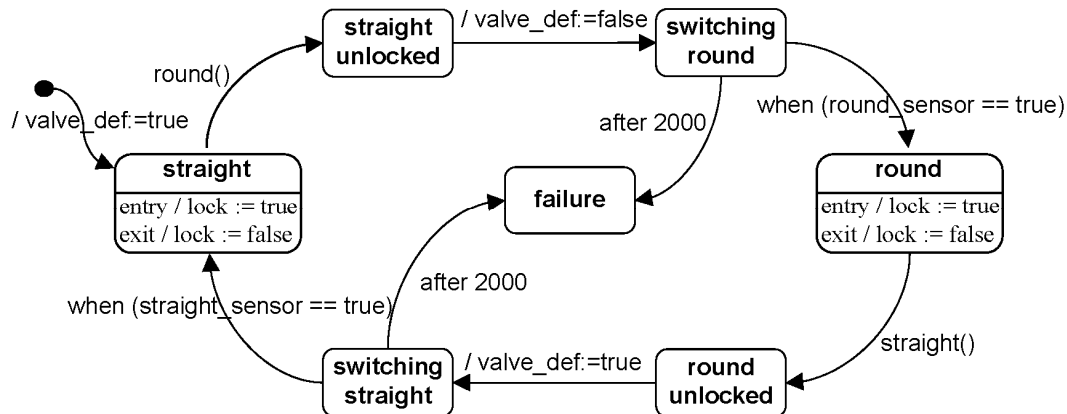


Abbildung 5: Zustandsdiagramm zur Steuerung einer Weiche

Zur Steuerung der Anlage wird aus den so spezifizierten Zustandsdiagrammen automatisch Code für die Steuerungssoftware generiert. Im Folgenden wird dieser Vorgang am Beispiel der Weichensteuerung kurz vorgestellt.

2.3 Codegenerierung

Zur Programmierung von Speicherprogrammierbaren Steuerungen (SPS) werden in der Praxis verschiedene Sprachen eingesetzt, die in der herstellerunabhängigen SPS-Programmiersprache IEC 61131-3 standardisiert wurden. Beispiele hierfür sind die *Anweisungsliste*, welche in den Bereich der maschinennahen Programmiersprachen einzuordnen ist oder der *Kontaktplan*, welcher als graphisches Spezifikationsmittel dem aus der Elektrotechnik bekannten Kontaktplan entspricht. Eine weitere Sprache ist *Strukturierter Text*. Strukturierter Text ist ebenfalls in der IEC 61131-3 definiert und orientiert sich an der höheren Programmiersprache PASCAL. Als klassisch prozedurale Programmiersprache unterstützt sie nicht die typischen Konzepte der Objektorientierung, so dass eine objektorientierte Implementierung der Zustandsdiagramme - z. B. mit Hilfe des Zustandsmusters [GJV95] - nicht möglich ist. Daher wurde zur Implementierung der Zustandsdiagramme auf einfache *switch-case* Anweisungen zurückgegriffen [SWG+04]. In Abbildung 6

ist ein Teil des generierten SPS-Codes für das Zustandsdiagramm aus Abbildung 5 zu sehen.

```

VAR state: INT := 1;           /* state = "straight" */
  timer: TIMER;
  expired: BOOL := FALSE;
END_VAR;

CASE state OF
  ...
3: /* state = "switching round" */
  timer(IN:=TRUE, TV:=T#2000ms); /* start or update timer */
  expired := NOT timer.Q;       /* test for expired timer */

  IF (round_sensor = TRUE) THEN /*when(round_sensor==true) */
    timer(IN:=FALSE, TV:=T#2000ms); /* stop timer */
    state := 4;                  /* state = "round" */
    lock := TRUE;               /* entry action */

  ELSIF (expired = TRUE) THEN   /* after 2000 */
    timer(IN:=FALSE, TV:=T#2000ms); /* stop timer */
    state := 7;                 /* state = "failure" */
  END_IF;

4: /* state = "round" */
  IF (straight = TRUE) THEN     /* event: straight() */
    lock := FALSE;             /* exit action */
    state := 5;                 /* state = "round unlocked" */
  END_IF;
  ...
END_CASE;

```

Abbildung 6: Beispiel für generierten SPS-Code

In dem generierten SPS-Code wird eine ganzzahlige Variable *state* definiert, die den aktuellen Zustand der Steuerungssoftware speichert. Da jeder Zustand eindeutig durch einen ganzzahligen Wert festgelegt ist, kann aufgrund dieses Wertes immer der korrespondierende *case*-Abschnitt ausgeführt und auf Ereignisse im Fertigungssystem reagiert werden. Darüber hinaus bildet die in dieser Variable gespeicherte Information über den aktuellen Zustand der Steuerungssoftware die Grundlage für die im Folgenden beschriebene Unterstützung des Ramp-Up Prozesses mit Hilfe von Augmented Reality.

3 AR-System zur Unterstützung des Ramp-Up Prozesses von automatisierten Fertigungssystemen

In diesem Kapitel wird ein AR-System entwickelt, das den Ramp-Up Prozesses von automatisierten Fertigungssystemen unterstützt. Dazu wird die Architektur des Systems vorgestellt und dessen Einsatz an einem Anwendungsszenario gezeigt.

3.1 Architektur

Die Architektur für den Einsatz des AR-Systems ist in Abbildung 7 dargestellt. Die Komponenten sind ein CASE²-Tool zur Softwarespezifikation, E/A-Konfiguration und Codegenerierung, das Fertigungssystem samt Steuerung und das AR-System.

Die Steuerungssoftware und die E/A Konfiguration werden im CASE-Tool spezifiziert. Die Definition der Ein- und Ausgänge (E/A-Konfiguration) für die Steuerung erfolgt idealer Weise zusammen mit der Mechanikkonstruktion. Dort bekommen Sensoren und Aktoren in der Produktstruktur einen eindeutigen Identifikator. Es wird eine Abbildung von Aktoren / Sensoren und Aus- und Eingängen der Steuerung durchgeführt.

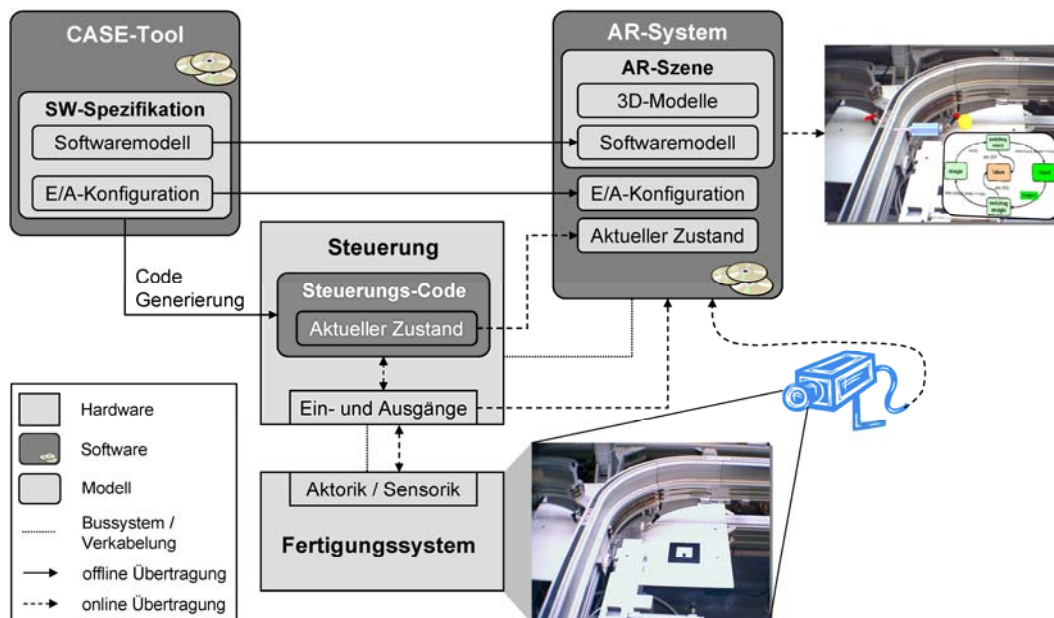


Abbildung 7: Architektur für den Einsatz des AR-Systems zur Unterstützung des Ramp-Up Prozesses von automatisierten Fertigungssystemen

Die Zustände der Ein- und Ausgänge der Steuerung müssen getrennt betrachtet werden. Die Eingänge der Steuerung sind mit Sensoren belegt. Diese liefern Signale aus dem Prozess. Diese Signale können für eine Augmentierung verwendet werden, indem das Signal am realen Sensor überlagert wird. Hier ist zwischen der Visualisierung von digitalen und analogen Signalen zu unterscheiden. Die Ausgänge der Steuerung sind mit Aktoren belegt. Diese erzeugen Kräfte oder Bewegungen, die normalerweise direkt vom Benutzer anhand der Zustandsänderung des realen Fertigungssystems wahrgenommen werden können. Der Aktor selbst

² Computer Aided Software Engineering

ist meist nicht zu sehen. Der Zustand des Aktors kann auch über den entsprechenden Ausgang der Steuerung ermittelt und mit seinem 3D-Modell lagerichtig im Fertigungssystem augmentiert werden. Dies hat den Vorteil, dass der von der Steuerungssoftware vorgegebene Zustand direkt mit der Realität verglichen werden kann.

Das AR-System liest online den Zustand der Steuerungssoftware aus und visualisiert das entsprechende Softwaremodell. Der Vorteil liegt hier insbesondere darin, dass der Zustand der Steuerungssoftware vor Ort direkt zusammen mit der gesteuerten Komponente betrachtet werden kann. Der Zustand der Software wird so direkt mit dem Verhalten der Hardware, den aktuellen Signalen der Sensorik und den resultierenden Ereignissen der Aktorik in Zusammenhang gebracht.

3.2 Anwendungsszenario

Das entwickelte AR-System ermöglicht die Fehleranalyse im Betrieb der errichteten Komponenten. Dabei können sowohl Fehler der Hardware als auch der Software analysiert werden. Der oder die Benutzer können das Verhalten der realen Hardware, die zusätzlich angezeigten Zustände von Aktoren und Sensoren und den Zustand der Steuerungssoftware gleichzeitig wahrnehmen. Durch einen direkten Vergleich zwischen dem erwarteten und tatsächlichen Verhalten werden Fehler identifiziert. Fehler können aus der Mechanik- oder Elektrokonstruktion, der Softwareentwicklung oder der E/A-Konfiguration stammen. Auch die Hardware kann fehlerhaft errichtet worden sein. Benutzer aus den verschiedenen Fachdisziplinen können anhand der augmentierten Informationen ihre Sicht auf das System erklären. Fragen zum Systemverhalten können so unmissverständlich geklärt werden.

Ein Beispiel für die Anwendung eines Prototyps des AR-Systems ist in Abbildung 8 zu sehen. Auf der rechten Seite ist der zeitliche Ablauf beim Schalten einer Weiche eines schienengebundenen Materialflusssystemes aus der Perspektive des Benutzers zu erkennen. Zustände von Sensoren, Aktoren und der Steuerungssoftware werden hier augmentiert. In diesem Beispiel wird nicht mit Occlusion gearbeitet, d.h. die virtuellen Objekte überlagern immer reale Komponenten bzw. das reale Bild. Der doppelt wirkende pneumatische Zylinder wird als 3D-Modell dargestellt, da er in die Weiche integriert ist und von außen nicht wahrgenommen werden kann. Auch die Position und die Zustände der Näherungssensoren werden mit entsprechenden 3D-Modellen angezeigt.

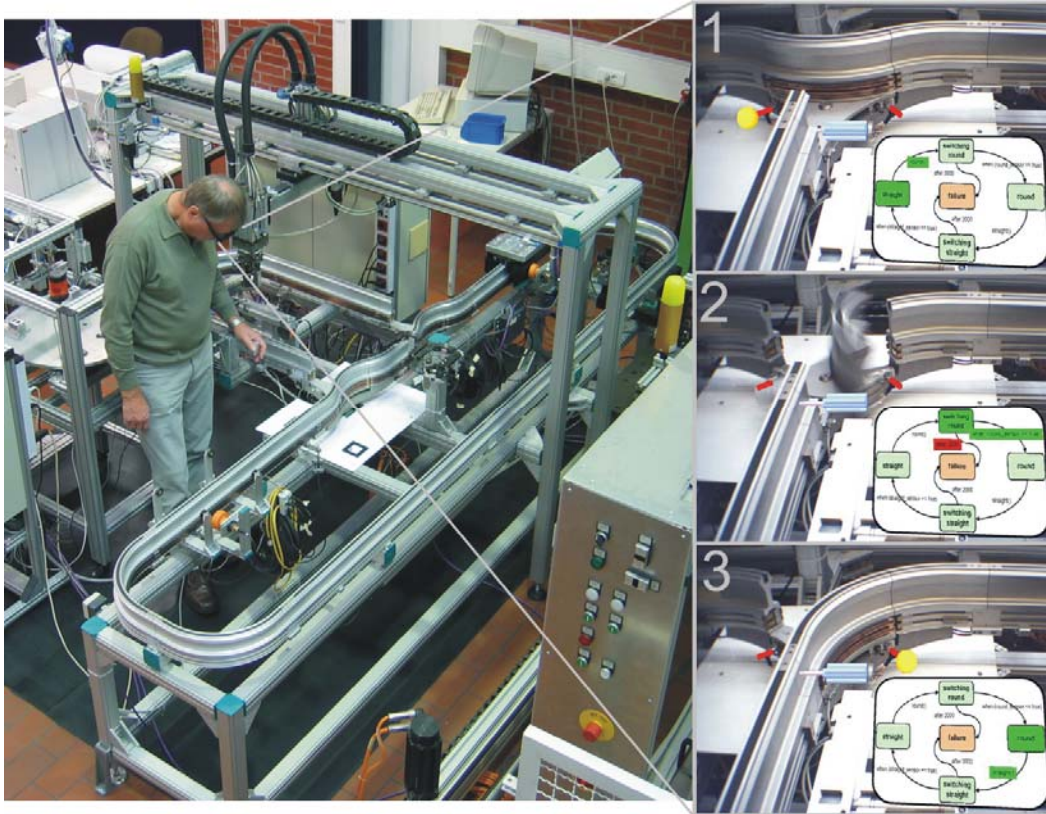


Abbildung 8: Einsatz des AR-Systems zur Inbetriebnahme einer Weiche

In der Anwendung des AR-Systems sind drei Fälle zu unterscheiden. Dazu sind die Zustände der realen Komponenten, deren augmentierten Zustände und der Zustand der Steuerungssoftware zu beachten.

Im ersten Fall stimmen die überlagerten Zustände mit den realen Zuständen überein. Das Verhalten entspricht aber nicht den Erwartungen. So kann zum Beispiel ein Näherungssensor die Endlage der Weiche nicht detektiert. Dies kann gut über den augmentierten Zustand erkannt werden. Die Schlussfolgerung ergibt, dass in der Installation (Hardware) etwas defekt oder falsch eingestellt sein muss. Im zweiten Fall stimmen die augmentierten Zustände nicht mit denen der realen Hardware überein. Beispielsweise detektiert ein Näherungssensor, dass die Weiche eine Endlage erreicht hat, aber der entsprechende augmentierte Zustand stimmt damit nicht überein. Dies kann ein Fehler in der Verkabelung, Buskonfiguration oder E/A-Konfiguration vorliegen. Im dritten Fall stimmt das im Zustandsdiagramm spezifizierte Verhalten nicht mit dem erwarteten Verhalten der realen Hardware überein. Dann sind Fehler in der Softwarespezifikation gemacht worden. Beispielsweise kann der Time-out des Schaltvorgangs der Weiche so kurz spezifiziert worden sein, dass die Steuerung immer in den Fehlerzustand springt.

4 Zusammenfassung

In diesem Beitrag wurde gezeigt, wie der Ramp-Up Prozess mit Augmented Reality unterstützt werden kann. Das vorgestellte AR-System erweitert das Blickfeld des Ingenieurs mit konkreten (Hardware) und abstrakten Zuständen (Software) der betrachteten Komponenten. Es unterstützt die interdisziplinäre Fehleranalyse, indem Zustände aus den verschiedenen Disziplinen (Maschinenbau, Elektrotechnik, Informatik) im realen Fertigungssystem visualisiert werden.

Literatur

- [Aßm96] ABMANN, S.: Methoden und Hilfsmittel zur abteilungsübergreifenden Projektierung komplexer Maschinen und Anlagen. Dissertation, Fakultät für Maschinenwesen, Rheinisch-Westfälische Technische Hochschule Aachen, Berichte aus der Produktionstechnik, Bd. 13/96, Shaker Verlag, Aachen, 1996
- [GJV95] GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J.: Design Patterns: Elements of Reusable Object Oriented Software. Addison-Wesley, Reading, MA, 1995
- [SHK+04] SCHOLZ-REITER, B.; HÖHNS, H.; KRUSE, A.; KÖNIG, F.: Hybrides Änderungsmanagement im Serienanlauf. In: Industrie Management, Jahrgang 20 (2004), Heft 4, Gito-Verlag, S. 21-24
- [SL00] SPATH, D.; LANDWEHR, R.: 3-D-Projektierung und Simulation von Ablaufsteuerungen. In: wt Werkstatttechnik online, Heft 7/8, Jahrgang 90 (2000), S. 292-296
- [SWG+04] SCHÄFER, W.; WAGNER, R.; GAUSEMEIER, J.; ECKES, R.: An Engineer's Workstation to support Integrated Development of Flexible Production Control Systems. In: Integration of Software Specification techniques for Applications in Engineering, Priority Program SoftSpez of the German Research Foundation (DFG) Final Report, Springer-Verlag, Berlin, LNCS Band 3147, 2004
- [WES+02] WIENDAHL, H.-P.; EVERSHEIM, W.; SCHUH, G., KUHN, A. (HRSG.): Schneller Produktionsanlauf von Serienprodukten, Ergebnisbericht der Untersuchung „fast ramp-up“. Dortmund, 2002
- [WHW02] WIENDAHL, H.-P.; HEGENSCHIEDT, M.; WINKLER, H.: Anlaufrobuste Produktionssysteme. In: wt Werkstatttechnik online, Heft 11/12, Jahrgang 92 (2002), S. 650-655
- [Zeu98] ZEUGTRÄGER, C.: Anlaufmanagement für Großanlagen. Dissertation, Universität Hannover, VDI-ortschriftsbericht, Reihe 2, Nr. 470, VDI-Verlag, Düsseldorf, 1998
- [ZM04] ZÄH, M. F.; MÖLLER, N.: Risikomanagement bei Produktionsanläufen. In: Industrie Management, Jahrgang 20 (2004), Heft 4, Gito-Verlag, S. 13-16
- [ZW05] ZÄH, M. F.; WUENSCH, G.: A New Method for Fast Plant Start-Up. In: Proceedings of 1st International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2005), September 22-23, Technical University of Munich, Germany, Herbert Utz Verlag GmbH, München, 2005