

Tudor Ionescu, Martina Merz<sup>1</sup>

## **Cyber-physische Produktion: Modelle und Inszenierung der Smart Factory**

**Abstract:** Der Artikel rekonstruiert die Entwicklung eines Smart Factory-Demonstrators in einem Großkonzern. Er zeigt auf, wie den dabei auftretenden Herausforderungen in der Praxis begegnet und eine bestimmte Vision der intelligenten Fabrik in Demonstrationen für ein Expertenpublikum inszeniert wurde. Der Demonstrator entspricht, in vereinfachter Form, der Vorstellung einer weitgehend automatisierten Fabrik, in der Roboter und andere „smarte“ Produktionsmaschinen kleinteilige Geräte nach vorgegebenen „Rezepten“ selbstorganisierend und dezentral produzieren. Aus Perspektive einer praxisorientierten Wissenschafts- und Technikforschung (STS) nimmt die zugrunde liegende ethnographische Untersuchung Aspekte der sozialen Organisation und technische Abläufe gleichermaßen in den Blick. Die Studie zeigt auf, dass Demonstratoren – hier als Modelle zukünftiger Smart Factories betrachtet – mehrere Zielsetzungen haben. Die identifizierten Zielsetzungen sind: die mit der Industrie 4.0-Vision assoziierten Versprechen in einer realen Produktionsumgebung auf Machbarkeit und Plausibilität zu prüfen; das Demonstrator-Projekt konzernintern als Lernumgebung zu begreifen; die Vision einer intelligenten Fabrik zu vergegenständlichen, wobei die Anliegen sowohl der Systementwickler als auch der potentiellen Abnehmer berücksichtigt werden.

### **1 Einleitung<sup>2</sup>**

In der Ecke einer Fabrik, die kleinteilige Geräte für den industriellen Bereich produziert, befindet sich ein Demonstrator, bestehend aus zwei Robotern und einem kleinen Förderband, die auf eine Tischplatte montiert sind. Gespannt wartet eine Gruppe von Fabrikbesuchern auf den Beginn der Demonstration. Ein Manager setzt den Produktionsprozess in Gang. Ein Roboter greift das erste Werkstück auf. Wie eine tanzende Schlange bewegt er sich auf unerwarteten Pfaden und vermeidet dabei – das Publikum ins Staunen versetzend – mehrere Hindernisse, bevor er das Werkstück dem zweiten Roboter übergibt, der dieses an einem genau definierten Ort ablegt. Die Vision einer solchen, durch ein Cyber-Physisches Produktionssystem realisierten automatisierten Montage von Werkstücken ist eines der Kernelemente des heutigen Diskurses um die „Smart Factory“ (oder: „Intelligente Fabrik“). Mit diesen Themen befasst sich der vorliegende Text.

Die Vision einer durch Informationstechnologie angetriebenen „vierten industriellen Revolution“ (Industrie 4.0) wurde 2012 in Deutschland von einer Gruppe von Universitätsprofessoren, hochrangigen Vertretern führender Technologieunternehmen und Regierungsexperten in Form eines Konzeptpapiers skizziert (vgl. Kagermann et al. 2013). In diesem Dokument wird insbesondere Cyber-Physischen Produktionssystemen (CPPS) eine große Bedeutung für die Fabrik der Zukunft zugeschrieben.

---

<sup>1</sup> Dr. Tudor Ionescu, Technische Universität Wien. E-Mail: tudor.ionescu@tuwien.ac.at. Prof. Dr. Martina Merz, Alpen-Adria-Universität Klagenfurt. E-Mail: martina.merz@aau.at.

<sup>2</sup> Für Anregungen und kritische Kommentare danken wir Ingo Schulz-Schaeffer und den anderen Herausgebern des Sonderbandes. Wir danken ebenfalls für das Feedback, das wir anlässlich der Präsentation der Arbeit auf der Tagung „Arbeit und Technik *revisited*: Gemeinsame Frühjahrstagung der Sektionen ‚Arbeits- und Industriesoziologie‘ und ‚Wissenschafts- und Technikforschung‘ der Deutschen Gesellschaft für Soziologie (DGS)“ (April 2018, WZB) sowie am Institut für Wissenschaftskommunikation und Hochschulforschung der Alpen-Adria-Universität Klagenfurt (Juni 2018, Wien) erhalten haben.

CPPS bestehen aus vernetzten Maschinen mit unterschiedlicher Rechenleistung und Funktion, die in engen Rückkopplungsschleifen miteinander sowie mit der physischen Umgebung verbunden sind. Dadurch ermöglichen sie die zentrale Steuerung vieler intelligenter Produktionsmaschinen bis hin zu einer gesamten intelligenten Fabrik. Die vierte industrielle Revolution, so die Vorstellung, wird von der Entwicklung intelligenter Produktionssysteme geleitet und erlaubt es bzw. droht (je nach Perspektive), (menschliche) Arbeit weitgehend zu ersetzen. Diese Vision erinnert an einen Trend aus den 1980er Jahren, der unter dem Namen Computer-Integrated Manufacturing (CIM) bekannt wurde und aufgrund der Ungenauigkeit von Montagerobotern und der Komplexität computerintegrierter Fertigungssysteme damals nicht weiterverfolgt wurde (vgl. Heßler 2014; Pfeiffer 2015). Das Konzept Industrie 4.0 legt somit einen erneuten Versuch hochgradiger Automatisierung nahe, der dank der technologischen Fortschritte der letzten drei Jahrzehnte in den Bereichen der Informationstechnologie, der Robotik, des 3D-Drucks etc. Erfolg durch Kostenminderungen verspricht. Anders jedoch als beim CIM wird die dystopische Vorstellung einer menschenleeren Fabrik bewusst vermieden. Es sind zum Beispiel (wie im betrachteten Fall) Arbeitsplätze in unmittelbarer Nähe der Roboter vorgesehen.

Im vorliegenden Text wird ein Projekt zur Entwicklung eines CPPS-Demonstrators in einem Großkonzern vorgestellt und diskutiert. Der Demonstrator entspricht der Umsetzung der Vorstellung einer weitgehend automatisierten Fabrik, in der Roboter und andere „smarte“ Produktionsmaschinen kleinteilige Geräte nach vorgegebenen „Rezepten“ selbstorganisierend und dezentral produzieren. Der von uns verfolgte analytische und empirische Ansatz ist der praxisorientierten Wissenschafts- und Technikforschung (STS) verpflichtet, die Technikentwicklung in einer Mikroperspektive und in ihrer Prozesshaftigkeit mittels eines ethnographischen Zugangs untersucht und dabei soziale Organisation und technische Abläufe gleichermaßen fokussiert. Eine solche Herangehensweise ergänzt die vorliegenden Analysen des Industrie 4.0-Diskurses und lenkt den Blick auf konkrete Forschungs- und Entwicklungsprojekte, in denen die Visionen der Smart Factory umgesetzt werden sollen – mit einem besonderen Interesse für die Arbeitspraktiken und Artefakte, die sozialen Konstellationen und die zeitliche Entfaltung.

Ein CPPS-Demonstrator lässt sich aus Sicht der Wissenschafts- und Technikforschung mit Blick auf (mindestens) drei Konzepte und damit assoziierte Literatur näher bestimmen.

*Erstens* besteht ein Bezug zum Phänomen der *Demonstration* neuer Technologien. Technologiedemonstrationen richten sich, in Marketing-Events wie etwa Industriemessen, an potentielle Käufer und Konkurrenten sowie in anderen Settings auch an eine breitere Öffentlichkeit (vgl. z. B. Coopmans 2010; Simakova 2010). Solche Demonstrationen werden durch ihre Initiatoren mit größter Sorgfalt inszeniert. Zentraler Bestandteil dieser Inszenierung ist die strategische Gestaltung des Verhältnisses von Vorder- und Hinterbühne (Smith 2009), von „revelation“ und „concealment“ (Coopmans 2010). Auch im betrachteten Fall wird die der Technikentwicklung zugrunde liegende Arbeit in der Demonstration unsichtbar gemacht, um die Innovationskraft des neuen Artefakts besonders hervorzuheben. Das Verbergen technischer

Spezifikationen kann ebenfalls dem Schutz vor einer möglichen Verletzung der Eigentumsrechte (vgl. „spy management practices“, ebd., S. 162) dienen.

*Zweitens* lassen sich Demonstratoren mit Blick auf *prototype scenarios* (Schulz-Schaeffer/Meister 2017) diskutieren. Letztere konstituieren „Verhandlungsarenen“ zwischen einer fiktionalen und einer empirischen Realität; sie verkörpern eine vereinfachte und fragmentarische Version einer imaginierten zukünftigen Realität (ebd., S. 11). Auch Demonstratoren lassen sich als Materialisierung von Imaginationen verstehen, gewissermaßen als greifbares Ergebnis der Umsetzung solcher Szenarien. In organisationsinternen Kontexten spielt sich die Verhandlung zwischen Gegenwart und Zukunft auf einer praktischen (können wir so etwas bauen?) und einer pragmatischen (können wir uns so etwas leisten?) Ebene ab. Im Rahmen der konzerneigenen Innovationsstrategie dienen übergeordnete „prototype scenarios“ als Referenz und Richtlinie. Die vorliegende Studie fokussiert hingegen auf die Realisierung und Inszenierung von Demonstratoren auf der Ebene der Entwicklungs- und Demonstrationspraktiken.

*Drittens* kann ein Demonstrator als *Modell* aufgefasst werden. Die Ansicht ist weit verbreitet, dass die Wirksamkeit von Modellen in ihrer Eigenschaft gründet, einen Gegenstand möglichst genau zu „repräsentieren“. Praxisorientierte Ansätze der STS lenken den Blick hingegen stärker auf die konkrete Entwicklungsarbeit, den Gebrauch und die Rolle von Modellen in verschiedenen Situationen und Settings (vgl. Merz/Hinterwaldner 2012; Merz 2016). Fragen nach der Repräsentation durch Modelle (bzw. den Demonstrator) werden in dieser Perspektive erweitert und derart reformuliert, dass nicht nur interessiert, welche (existierende, anvisierte) Wirklichkeit ein Demonstrator repräsentiert, sondern ebenfalls, auf welche Weise und in welcher Form er diese erst hervorbringt und welche Möglichkeiten er bietet, am Modell zu lernen oder gewisse Aspekte zu untersuchen.

CPPS-Demonstratoren werden im Folgenden als Modelle zukünftiger cyber-physischer Fabriken betrachtet. Bei ihrer Entwicklung handelt es sich um aufwändige und kostspielige Formen des Modellbaus, die sowohl die schöpferischen Grenzen als auch die Potentiale der in einem Unternehmen arbeitenden Forschungsingenieure zum Ausdruck bringen.<sup>3</sup> Mit Demonstratoren, so eine erste Beobachtung, werden ganz unterschiedliche Ziele verfolgt. Für den hier betrachteten Fall der Entwicklung eines CPPS-Demonstrators in einem großen Technologiekonzern sind die identifizierten Zielsetzungen: die mit der Industrie 4.0-Vision assoziierten Versprechen in einer realen Produktionsumgebung auf Machbarkeit und Plausibilität zu prüfen; das Demonstrator-Projekt als Lernumgebung zu begreifen, insbesondere hinsichtlich der Arbeit mit kollaborativen Robotern, die bei verschiedenen nach wie vor von Menschen ausgeführten Montagearbeiten eingesetzt werden sollen; die Vision einer intelligenten Fabrik in Demonstrationen für ein Expertenpublikum zu inszenieren. Im Verlauf des Textes wird diese Multifunktionalität weiter exploriert.

---

<sup>3</sup> Auch das Industrie 4.0-Konzeptpapier betont den großen Wert von „Planungs- und Erklärungsmodellen“ bestehender und zukünftiger Produktionssysteme und -prozesse bei dem Versuch, „ihre zunehmende Komplexität zu meistern“ (Kagermann et al. 2013, S. 86).

Im Folgenden werden zunächst die Fallstudie und das empirische Vorgehen dargestellt (2). Im Hauptteil wird die Entwicklung des Demonstrators im Kontext der dabei auftretenden vielfältigen Herausforderungen anhand selektiver Aspekte nachgezeichnet; ein erster Abschnitt ist der Softwareentwicklung (3), ein zweiter der Integration von Software und Hardware (4) gewidmet. Daran anschließend wird die Aufmerksamkeit auf die Demonstration gerichtet, und es wird geschildert, wie der Demonstrator im Dienste der Inszenierung der Smart Factory in Aktion tritt (5). Es folgt ein kurzer Schlussteil (6).

## **2 Fallstudie und empirisches Vorgehen**

Die vorliegende Studie rekonstruiert die Entwicklung eines Demonstrators für ein Cyber-Physisches Produktionssystem (CPPS) in einem Technologiekonzern – der aus Gründen der Anonymisierung nicht weiter gekennzeichnet wird – über einen Zeitraum von zwei Jahren: von der Konzeption des Demonstrators bis zu seinem ersten Einsatz in Demonstrationen. Die Entwicklung des Demonstrators hat im Konzern den Status eines Forschungstransferprojekts, im Rahmen dessen ein Machbarkeitsnachweis für bestehende CPPS-Bausteine, -konzepte und -patente für einen konzerninternen Abnehmerkreis erbracht werden soll. Das Projekt wird durch mehrere firmeninterne Forschungszuschüsse finanziert. Der Demonstrator soll in der Fabrik platziert werden, wodurch der Eindruck entsteht, dass Robotik- und Softwareexperten mitten in einer typischen Produktionsumgebung Forschungs- und Entwicklungstätigkeiten ausüben können. In zeitlicher Perspektive soll der Demonstrator einen Ausblick hinsichtlich einer weitgehend durch CPPS automatisierten Produktionsstätte geben. Man muss sich eine Vielzahl solcher nebeneinander angeordneter oder inselartiger Produktionszellen vorstellen, die durch komplexe Software und Datenmodelle (sog. Digitale Zwillinge) sowie durch Arbeiter und (fahrerlose) Transportfahrzeuge miteinander vernetzt sind.

Laut Konzept besteht der Demonstrator aus mehreren Produktionsmaschinen, darunter zwei kollaborative Roboter (Cobots) und ein kleines Förderband, die an einer Tischplatte befestigt sind. Neben den drei genannten aktiven Einheiten soll er auch passive Elemente enthalten, wie Kanban-Rutschen und 3D-gedruckte Werkstückhalter. Auf einer Seite der Produktionszelle ist ein Arbeitsplatz in unmittelbarer Nähe zu den Robotern vorgesehen. Anders als bei traditionellen Automatisierungsanwendungen sind in dieser Konfiguration Mensch und Roboter nicht physisch getrennt. Es liegt die Erwartung zugrunde, dass die Roboter die Sicherheit der Zusammenarbeit normgerecht gewährleisten können.

Empirisch beruht unsere Studie auf den, durch eine STS-Perspektive sensibilisierten, ethnographischen Beobachtungen eines der am Projekt mitwirkenden Entwickler, der zugleich einer der Verfasser des vorliegenden Textes ist. Umfangreiche Feldprotokolle wurden von ihm, auf der Grundlage von ad hoc-Notizen, nach Beendigung der Mitarbeit an der Entwicklung des Demonstrators verfasst. Die Protokolle decken die gesamte Entwicklungsperiode des Demonstrators ab, von der Antragsphase bis zu den ersten Demonstrationen, mit besonderer Aufmerksamkeit für die Arbeit der Softwarearchitekten und Entwickler. Die Distanzierung von der „nativen“

Teilnehmerperspektive wird durch die gemeinsame Analyse des Falls und des Datenmaterials mit der Co-Autorin sowie durch die Einbettung der Beobachtungen in relevante STS-Debatten befördert (vgl. Amann/Hirschauer 1997). Die vorliegenden Daten werden derzeit durch qualitative Interviews ergänzt, um die Perspektiven weiterer am Projekt beteiligter Personen in der Analyse adäquat berücksichtigen zu können. Der vorliegende Text ist insofern ein erster Bericht aus einem noch andauernden Forschungsprojekt.

### **3 Softwareentwicklung: Pfadabhängigkeit im Kontext der Organisation**

Die in der Studie fokussierte Arbeit der Softwarearchitekten und Entwickler ist durch eine hybride, dynamisch wechselnde Kombination von Aktivitäten in der Fabrikhalle, der Arbeit im Büro und Online-Kommunikation im Rahmen einer großen Organisation gekennzeichnet. In diesem Kontext spielt die spezifische Organisationskultur eine bedeutende Rolle, die in den Diskursen, Erwartungen und Praktiken der Projektbeteiligten zum Ausdruck kommt.

Eine Ecke der Fabrik wird für Industrie 4.0-orientierte F&E-Aktivitäten und damit für den Demonstrator reserviert. Sobald der Standort des Fabriklabors bestimmt ist, wird er von Stahlstangen umzingelt, die eine Art Zaun bilden. Der Ort ist etwa 3x3 Meter groß und befindet sich in einer Ecke im hinteren Teil der Fabrik, wo verschiedene Rahmenkonstruktionsarbeiten zur Unterstützung der Montagetätigkeit anderer Arbeiten abgewickelt werden. Das CPPS-Labor befindet sich demnach in einer typischen Fabrikhalle, in der mit einer gewissen Regelmäßigkeit grobe Geräusche zu hören sind: Metall- und Holzschneiden, Bohren, Schrauben und Hämmern.

Als die Softwareentwickler die Fabrik, ihr zukünftiges Fabriklabor, erstmals betreten, sind lediglich der Tisch und einer der Roboter bereits montiert. Es ist die Zeit der Pläne und Konzepte, und es herrscht eine Stimmung der Aufregung, kombiniert mit dem Nervenkitzel des Unbekannten, das vor ihnen liegt. Der Projektleiter bemüht sich, die anderen Mitglieder des ca. fünfköpfigen Teams von der Machbarkeit des Demonstrators zu überzeugen, indem er auf seine Erfahrung in früheren Forschungsprojekten verweist. Allerdings hatten diese nicht das Ziel verfolgt, 'echte' Produkte zu assemblieren, sondern lediglich einfache Lego-ähnliche Bausätze. Die notwendige Verbesserung der Montagepräzision sowie die Validierung des in den Vorgängerprojekten entstandenen Softwaredesigns in einer realen Produktionsumgebung stellen die besondere Herausforderung und den Forschungscharakter des aktuellen Projekts dar.

Die von früheren Projekten 'geerbte' Softwarearchitektur wird im Unternehmen als revolutionär bezeichnet. Zum Zeitpunkt, an dem die Arbeit am Demonstrator beginnt, sind bereits viele Funktionen zur Patentierung angemeldet. Mehrere führende System- und Softwarearchitekten des Konzerns haben eine Referenzarchitektur konzipiert, indem sie die Richtlinien der Industrie 4.0-Vision genau zu befolgen suchen.

Die Realisierung eines CPPS erfordert die Entwicklung – und anschließende Integration – von Software- und Hardwarekomponenten (Roboter, Sensoren, industrielle Rechner etc.) unterschiedlicher Herkunft, gewissermaßen einer komplexen Soft-

wareökologie mit heterogenen Elementen. Im verfolgten Projekt werden dafür zum einen „Open source“-Komponenten herangezogen, wie etwa die Robotertreiber sowie eine Reihe von Bibliotheken für inverse Kinematik und Roboterpfadplanung. Zum anderen müssen viele anwendungsspezifische Softwarekomponenten neu entwickelt werden, die der Zusammenführung der aus dritter Hand bezogenen Software und Hardware dienen. Hinzu kommt noch die proprietäre *Firmware* der einzelnen Produktionseinheiten, die als unterste Softwareschicht beibehalten werden muss. Zunächst gibt der Projektleiter den Plan aus, die in den Vorgängerprojekten entwickelte Software wiederzuverwenden, die auf dem Fundament einer verbreiteten Robotik-Middleware<sup>4</sup> aufgebaut ist. Allerdings stellt sich dies bald als unmöglich heraus, weil die bestehende Software damals für ganz bestimmte Demo-Szenarien hartkodiert<sup>5</sup> worden war und daher nun nicht für neue Anwendungen eingesetzt werden kann. D. h. anstatt die Roboterbewegungen über Parameter zu steuern, waren die Anfangs- und Endkoordinaten fast aller Bewegungen direkt in den Code eingegeben worden. Aus diesem Grund muss der gesamte Code für den neuen Demonstrator neu entworfen und geschrieben werden. Allerdings hat anfangs keiner der Projektbeteiligten die notwendigen Kenntnisse im Bereich der Roboterprogrammierung. Die Hoffnung, Codes aus vorherigen Demonstratoren wiederverwenden zu können, führt auch zur Wahl von C++ als „Integrationssprache“ für alle anderen Systemkomponenten, während beide Roboter ihre je eigenen Programmierumgebungen mitbringen. Die firmeneigene Softwareentwicklungsumgebung mitgezählt, müssen daher insgesamt vier unterschiedliche Programmiersprachen zur Umsetzung des Demonstrators verwendet werden.

Die Auswahl der im Projekt verwendeten Entwicklungswerkzeuge ist folglich von großer Heterogenität geprägt. Diese Heterogenität entspricht einer Organisationslogik, der zufolge Softwareprodukte möglichst aus eigenem Hause zu verwenden sind, um die Anpassungsfähigkeit des Konzerns hinsichtlich der vierten industriellen Revolution zu demonstrieren. Und da einige der aus Vorgängerprojekten entstandenen CPPS-Konzepte und Ideen bereits patentiert wurden, gibt es von Seiten des Managements ebenfalls die Präferenz für den neuen Demonstrator betreffende Designentscheidungen, die diese begünstigen: Die patentierten Konzepte hochrangiger Experten des Konzerns sollen praktisch erprobt und demonstriert werden, um deren Überführung in Produkte zu ermöglichen. Dadurch entstehen Abhängigkeiten bezüglich der notwendigen Expertise (z. B. Kenntnisse firmeneigener proprietärer Programmiersprachen), die über die Einbindung einer Vielzahl öffentlich verfügbarer Softwarebausteine mit einem unübersichtlichen Softwaresammelsurium verwoben werden. Ähnlich werden auch vor Projektanfang getroffene und patentierte Designentscheidungen in den Demonstrator eingebettet, wie etwa die Auswahl der Programmiersprachen, Bibliotheken und Middleware, die, wie sich später herausstellen wird, nachhaltige Konsequenzen für das gesamte Projekt haben werden.

---

<sup>4</sup> Unter Middleware versteht man eine Softwareschicht, die bestimmte technologiespezifische Funktionen (z. B. Roboterbewegungen) abstrahiert, sodass eine technologieunabhängige Entwicklung möglich wird.

<sup>5</sup> Unter „hartkodiert“ versteht man, dass bestimmten, normalerweise durch Variablen repräsentierten Parametern unveränderbare Werte im Quellcode zugewiesen werden. Somit verliert ein hartkodierter Parameter seinen ursprünglichen Zweck zugunsten der Einfachheit.

Die Pfadabhängigkeit bezieht sich nicht nur auf die getroffenen Designentscheidungen und CPPS-Konzepte, sondern ebenfalls auf die Formen der Arbeitsorganisation. Auch die agile Entwicklungsmethodologie wird aus Vorgängerprojekten übernommen. Die Aufgaben werden mittels eines webbasierten Kanban-Taskboards im Rahmen zweimal wöchentlich stattfindender Entwicklermeetings verteilt und überwacht. Neue Aufgaben werden auch außerhalb dieser Meetings vergeben. Das Kanban-Board füllt sich schon in den ersten Monaten mit hunderten von Tasks unterschiedlicher Granularität, die von der Änderung eines Konfigurationsfiles bis hin zur Integration eines gesamten Robotersystems reichen. Anfangs sind die meisten Tasks in den Spalten „To Do“ oder „In Progress“ eingetragen, während die Spalte der beendeten Tasks relativ dünn besiedelt ist. Dies ändert sich im Laufe des Projekts, während immer mehr Tasks in allen Spalten hinzukommen. Dabei verweilen einige Tasks monatelang in der Spalte „In Progress“, weil beispielsweise die Integration der *Open source*-Robotertreiber zu einem unerwarteten Verhalten der Roboter führt und es für sie keine offizielle technische Unterstützung seitens des Autors des Treibers gibt. Die mit agilen Methoden assoziierten Erwartungen einer Produktivitätserhöhung in der Softwareentwicklung werden konterkariert durch unvorhergesehene Schwierigkeiten bei der Integration extern bezogener Software („third party“). Die panoptische Transparenz des Kanban-Boards sorgt für Spannungen zwischen dem Projektleiter, den Architekten und den Entwicklern. Einschätzungen bezüglich der voraussichtlichen Dauer vieler Aufgaben erweisen sich als unrealistisch, und deren langes Verweilen in der Spalte „In Progress“ im Kanban-Board führt zu Verzögerungen, für die sich die Entwickler rechtfertigen müssen.

Die skizzierte Entwicklung scheint im Widerspruch zur Tatsache, dass die agilen Entwicklungsmethoden bei ihrer Konzeption und Einführung einen „practice turn“ in der Softwareentwicklung kennzeichneten, demzufolge Programmieren als kooperativer und kommunikativer Prozess aufzufassen ist (vgl. Schmidt 2008; Schulz-Schaeffer/Bottel 2018). Die Diskrepanz zum untersuchten Fall lässt sich vermutlich damit erklären, dass die praktische Anwendung des Verfahrens entgegen dem originären Konzept der agilen Methoden erfolgt und statt der Kooperationsorientierung stärker die Kontrolle und die forcierte Mikrofragmentierung in den Vordergrund rücken, die Flexibilität eher behindern.

#### **4 Umzug in die Fabrik: Integration und Tests**

Mehr als ein halbes Jahr nach Projektbeginn verbringt das nun um einzelne Industrieingenieure erweiterte Projektteam – aufgrund der Notwendigkeit, die neu entwickelten Softwarekomponenten nicht nur in der Simulationsumgebung, sondern auch mit der realen Hardware zu integrieren – zunehmend mehr Zeit in der Fabrik. Nur dort ist es möglich, das Verhalten der Roboter, die von einer sich ständig verändernden Software gesteuert werden, unter realen Bedingungen und in Zusammenwirkung mit allen anderen Produktionseinheiten zu testen. Da jede Einheit von einer anderen Person betreut wird, erfordern die Integrationstests eine intensivere und regelmäßige mündliche oder über Chats vermittelte Kommunikation zwischen den Teammitgliedern.

Das Testen einer neuen oder geänderten Funktion des Demonstrators erfordert eine ganze Reihe arbeitsintensiver Schritte, die in kleinen Teams (zwei bis drei Personen) durchgeführt werden. Zunächst muss die Funktion kompiliert werden, und da C++ die bindende Programmiersprache ist, dauert dies zwei bis drei Minuten. Nach der Kompilierung wird die neue Version der Software auf den Steuerungs-PC des Roboters heruntergeladen. Darauf müssen die Roboteranwendung neu gestartet und die Roboter in ihre Ausgangspositionen gebracht werden. Die Werkstücke werden in verschiedene Werkstückhalter und Kanban-Rutschen eingelegt. Als nächstes muss die zentrale Anlagensteuerungssoftware zusammen mit allen anderen Einheiten, insbesondere den Robotern, neu gestartet werden. Anschließend muss man warten, bis alle Einheiten in einer Zustandsüberwachungsliste als „aktiv“ auftauchen. Nach der erfolgreichen Initialisierung aller Einheiten muss eine Prozess- und Stückliste entsprechend einem Produkt ausgewählt werden. Der Montageablauf wird dann in Form einer baumähnlichen Struktur am Bildschirm dargestellt, wobei die Blätter dieses Baumes für einzelne Werkstücke und die internen Knoten für die notwendigen Produktionsfertigkeiten der Cyber-Physischen Produktionseinheiten stehen. Dann übernimmt ein Algorithmus die Machbarkeitsprüfung für das ausgewählte Produkt, und anschließend lässt sich die Produktion, d. h. der Test, starten. Vom Beginn der Kompilierung bis zum Ende der Machbarkeitsprüfung vergehen rund zehn bis 15 Minuten. Bis kurz vor der Demonstration läuft keiner der Integrationstests zu Ende, da immer wieder neue Fehler auftreten. Insgesamt werden pro Test etwa 30 Minuten benötigt. Für das volle Programm, in dem zwei verschiedene Produkte „produziert“ werden sollen, dauert der gesamte Testzyklus noch länger. Die Analyse und Behebung eines komplexen Problems erfordert bis zu zehn solcher Testzyklen.

Die typischerweise in Büros tätigen Entwickler arbeiten nun manchmal eine ganze Schicht lang Seite an Seite mit den Fabrikarbeitern, für die erstere jedoch ein fremdes Völkchen bleiben. In der Fabrik muss man als Büroarbeiter oder -arbeiterin einen weißen Arbeitsmantel tragen. Ansonsten unterliegt das CPPS-Labor den gleichen Regeln wie alle anderen Fabrikbereiche. Allerdings ist die Fabrik für die Produktionsarbeit optimiert (z. B. hinsichtlich Ergonomie, Belichtung, Raumaufteilung) und daher wenig geeignet für die stundenlange Arbeit am Computer. Der Fabriklärm wird ab und zu von Diskussionen zwischen den Entwicklern überdeckt, die mit ihren Computern zu zweit oder dritt an zwei kleinen Schreibtischen neben dem Roboterisch sitzen. Es scheint ihnen zuweilen, als würde die Produktionswelt sie „verschlingen“ (Feldtagebuch), als wäre kein Ausweg aus der Fabrik möglich, solange sie den Demonstrator nicht dazu brächten, die Industrie 4.0-Versprechen einzulösen.

In dieser räumlichen, physischen und organisatorischen Konstellation wird das CPPS-Labor (incl. dem Demonstrator) zum „Mediator“: es vermittelt zwischen den Arbeitswelten der Softwareentwicklung und der Produktionsarbeit, zwischen der virtuellen Welt des Codes und der materiellen Welt der Fabrik und Produktion.<sup>6</sup> Dabei

---

<sup>6</sup> Die Vorstellung des Demonstrators als Vermittler ist inspiriert von der entsprechenden Debatte zu „Models as Mediators“ (Morrison/Morgan 1999). Zunächst bezog sich das Konzept primär auf die Vermittlungsrolle der Modelle zwischen Theorie und der Welt (ibid.). Später wurde das Konzept auf andere Vermittlungsprozesse ausgeweitet, z. B. jene zwischen Wissens-, Expertise- und Handlungsfeldern.



scheint die Erste in die Zweite einzudringen mit dem Potenzial sie umzugestalten (für eine ähnliche Dynamik vgl. Miller/O'Leary 1994), doch davon ist zu diesem Zeitpunkt noch nichts zu merken.

Etwa eine Woche vor der ersten geplanten Demonstration ist die Lage kritisch. Keines der Demo-Szenarien funktioniert, weil die einzelnen Systemkomponenten nicht richtig integriert sind. Wegen anderer Verzögerungen bei der Implementierung kann bis kurz vor dem Demo-Termin nur unzureichend getestet werden. Überall gibt es offene ‚Baustellen‘: ungelöste Defekte und Anomalien, Konfigurationsfehler, vergessene oder fehlende Features. Zu diesem Zeitpunkt besteht der Demonstrator aus der Summe seiner Teile, und obwohl jeder dieser Teile mehr oder weniger funktioniert, scheitert das Gesamtsystem an der für die Demonstration konzipierten Aufgabe, einige Produktionsschritte an zwei Produkten auszuführen.

Eine der größten Herausforderungen besteht darin, die sog. Ports zu messen. Ports sind jene Positionen im Raum (meistens auf dem Tisch oder einer anderen stabilen Unterlage), an denen die Übergabe von Werkstücken von einer Produktionseinheit zu einer anderen erfolgt. Da die Roboter „blind“ sind, müssen diese Ports sowie die Werkstücke darin ganz präzise vermessen und positioniert werden, damit die Roboter nicht danebengreifen. Das Messen der Ports ist mühsam. Der Roboter muss manuell zur sogenannten Greifposition gefahren werden. Anschließend wird die Roboterpose vorsichtig nachjustiert, damit die Greiffinger das Werkstück so genau wie möglich fassen können. Einmal erreicht, müssen die sechs Eulerkoordinaten der Greifposition am Steuerungsgerät des Roboters abgelesen, in das globale Koordinatensystem transformiert und schließlich in Quaternionen umgerechnet werden. Insgesamt gibt es mehr als 15 Ports, wobei jede Portmessung samt Umrechnungen etwa eine halbe Stunde in Anspruch nimmt. Die Entwickler benutzen ein webbasiertes Tool für die Umrechnung von Eulerwinkel in Quaternionen – bis sie feststellen, dass die drei Dezimalstellen, die der Web-Computer zur Verfügung stellt, nicht präzise genug sind. Jede Portmessung muss bis zu acht Dezimalstellen genau sein, damit der Roboter die Werkstücke richtig greifen und montieren kann. Ein Entwickler vertritt sogar die Auffassung, dass zehn Dezimalstellen notwendig sind. Es kommt das Problem hinzu, dass jede unvorsichtige Interaktion mit den Tischplatten zumindest einen Teil der Messungen ungültig macht. Dies hat zur Folge, dass die Ports während der gesamten Integrationsphase durchgehend gemessen und wieder gemessen werden müssen, obwohl dies einmal als finite Aufgabe im Kanban-Board definiert worden war. Das System ist fragil: Legt man ein Werkstück auch nur einen Zehntelmillimeter entfernt von seiner idealen Position im Port ab, greift der Roboter knapp daneben und wird somit der Test fehlschlagen. Um erfolgreich zu sein, muss ein Cyber-Physisches Produktionssystem folglich eine perfekte Mischung aus Hardware, Software und Mensch sein.

In den letzten Wochen vor der ersten geplanten Demo teilt das Team jeden Tag in zwei Phasen ein. Es gibt eine Testphase mit den einzelnen Systemkomponenten (sog. Unit Tests) und anschließend eine Integrationstestphase mit allen Produktionseinheiten. Dabei werden die geplanten Demo-Szenarien getestet. Allerdings schafft das Team nicht mehr als zwei oder drei Integrationstests am Tag, weil nach jedem

gescheiterten Test die Probleme zuerst behoben werden müssen, bevor ein neuer Test gestartet werden kann. Die Integrationstests machen viele früher parallel durchführbare Aufgaben unmöglich. Schließlich wird der Demo-Termin um ca. vier Wochen nach hinten verschoben. Zwei Tage vor dem neuen Termin gelingt dem Team ein erster erfolgreicher Integrationstest. Die entsprechend große Freude nach mehr als einem Jahr erfolgloser Arbeit, in der Wahrnehmung der Entwickler, stellt sich jedoch als kurzlebig heraus, weil am Folgetag die Tests erneut scheitern. Das System ist nach wie vor äußerst instabil. Die intensive Arbeit der letzten zwei Tage vor der Demonstration ermöglicht jedoch die Stabilisierung des Demonstrators, sodass die Verantwortlichen eine nochmalige Verschiebung nicht mehr für notwendig erachten.

Die praktische Erfahrung mit den Integrationstests legt zwei Hauptgründe nahe, weshalb die Entwicklung eines autonomen CPPS die optimistischen Erwartungen der Systemarchitekten bezüglich der Machbarkeit und Umsetzungsdauer nicht gänzlich erfüllen kann. Erstens sind die Erwartungen an die Interaktion zwischen hochpräzisen Robotern und passiven Hardwareelementen dominiert von der Vorstellung, dass die Präzision der Roboter auf ihre Umgebung übertragbar sei. Wie die Portmessungen zeigen, ist dies jedoch nicht der Fall. Nur wenn die Umgebung genaueste Maße hätte oder es eine Messtechnik gäbe, mithilfe derer sich solche Portmessungen automatisieren ließen, könnte man das Gesamtsystem eventuell mit weniger Zeit stabilisieren. Zweitens beruht das an die „Computer Integrated Manufacturing“ (CIM) erinnernde Paradigma der autonomen Cyber-Physischen Produktionssysteme auf dem Modell der Agenten-basierten Simulation und Interaktion. Obwohl dieses Modell für manche Arten von Simulationen gut geeignet zu sein scheint, erweist es sich für die Umsetzung eines CPPS als ungeeignet, da die Autonomie der beteiligten Produktionseinheiten (oder Agenten) eine fehlerfreie Interaktion mit sich selbst sowie zwischen ihnen voraussetzt. Dabei tauchen immer wieder Fehler bei der Initialisierung einzelner Einheiten sowie bei der Synchronisierung der Kommunikation zwischen ihnen auf. Dies sorgt dafür, dass die ersten etwa 50 Integrationstests scheitern. Erst danach erreicht man eine gewisse fehlerfreie Wiederholbarkeit der Tests.

## **5 Demonstration: Inszenierung der Smart Factory**

Trotz aller Schwierigkeiten bei der Entwicklung des Demonstrators eröffnet der erste erfolgreiche Integrationstest die Möglichkeit, eine bestimmte Vision der Smart Factory in Demonstrationen vor einem Publikum zu inszenieren. Damit konnte eines der Hauptziele des Projekts (neben dem Aspekt des Expertise- und Erfahrungsaufbaus) erfüllt werden.

Inzwischen wird der Demonstrator regelmäßig hochrangigen Führungskräften und Experten präsentiert. Dies geschieht als Nebenschau während des Werkbesuchs einer hochrangigen Delegation, oder es bekommen ein Manager oder eine Professorin eine gesonderte Vorführung. Ziel dieser Demonstrationen ist es, das jeweilige Publikum davon zu überzeugen, dass die vollautomatisierte Fabrik der Zukunft tatsächlich mit den schon heute verfügbaren, firmeneigenen sowie Open Source-Technologien realisierbar ist. Dank der koordinierten Interaktionen zwischen dem Demonstrator, dem Vortragenden, den anwesenden Forschungsingenieuren und

verschiedenen Zielgruppen bietet die Demonstration einen beschränkten, aber nicht weniger faszinierenden Einblick in die imaginierte Zukunft der Arbeitswelt. Jede Demonstration erfordert umfangreiche Vorbereitungen, an denen fast das gesamte Projektteam beteiligt ist, sowie eine sorgfältige Inszenierung, die im Folgenden entlang der betroffenen rhetorischen Kategorien geschildert wird.

### **Decorum: Roboter und Forschungsingenieure in einer Fabrik**

Um zum CPPS-Demonstrator zu gelangen, müssen die Besucher in der Fabrik zunächst einen langen Korridor durchqueren, der von Arbeitern auf der rechten und Industrieingenieuren auf der linken Seite belebt wird. Der Demonstrator wird von einem ein Meter hohen Zaun aus Stahlrohren umschlossen, der symbolisch und physisch eine Abgrenzung der Forschungsarbeiten von Produktionsarbeiten markiert. Ein Schild warnt: „Nur autorisiertes Personal.“ Jeder der sechs Sektoren des aus Furnier gefertigten Tisches, auf den zwei Roboter montiert sind, ist einer bestimmten Produktionseinheit gewidmet; einer der Sektoren ist für eine Arbeiterin bzw. einen Arbeiter vorgesehen. Neben dem Robotertisch stehen Schreibtische mit Computerbildschirmen. Unter ihm sind unzählige Kabel sowie mehrere Industrierechner zu sehen. Alle Maschinen stehen noch still, während die Mitglieder des Entwicklungsteams die Werkstücke in verschiedene Ports stellen und die Software initialisieren. Das Fabriklabor ist ein Grenzobjekt zwischen Forschung, Entwicklung, Produktion und Schau. Vom Entwicklungsmodus kann jederzeit in den Präsentationsmodus und wieder zurück gewechselt werden. Sobald die Besucher am Demonstrationsort eintreffen und ihre Stehplätze um das Stahlgerüst einnehmen, beenden die Entwickler die laufenden Vorbereitungen rasch. Ein Moment der Stille setzt ein.

### **Logos: Der Prototyp in Aktion**

Die Demonstration beginnt typischerweise mit einer kurzen Einführung anhand von zwei bis drei Präsentationsfolien durch den als Zeremonienmeister agierenden Manager. Die Vermittlung harter Fakten und Daten zielt darauf ab, das Vertrauen des Publikums zu gewinnen. Nach wenigen Minuten wird der erste Produktionsprozess gestartet, während der Vortragende, abhängig von Rang und Expertise der Zuschauenden, in mehr oder weniger technischen Details erklärt, was hinter den Kulissen passiert. Die Erklärungen enthalten eine Reihe von mit großen Erwartungen aufgeladenen Begriffen und Ausdrücken, z. B. „das Produkt steuert seine eigene Produktion“, „digitale Zwillinge“, „Losgröße eins“, Flexibilität, Autonomie, Selbstorganisation und Selbstoptimierung etc., kombiniert mit technischen Details über Ports, automatische Pfadplanung, Sensoren, sensitive Robotik, Ontologien und automatische Produktionsplanung. Sobald einer der Roboter seinen ersten automatisch geplanten Weg hinter sich gelegt und das erste Werkstück aufgegriffen hat, scheint das Publikum von der Performance absorbiert zu sein. Die Erklärungen des Vortragenden dienen als Kohäsionsmittel für Ideologie, Technologie und den visuellen Diskurs der kooperativen Roboter, die die gleichen Produkte assemblieren können wie die hinter dem Publikum arbeitenden Männer und Frauen. Im Hintergrund ertönen die üblichen Geräusche der Fabrik, die ununterbrochen produziert.

Der Höhepunkt der Demonstration ist erreicht, wenn einer der Roboter ein Produkt in einem stark eingeschränkten Raum aufgreift und platziert – eine echte Herausforderung für die automatische Pfadplanung – und dabei die eingangs skizzierte Schlangenbewegung ausführt. Die Zuschauer fragen: Warum wählt er diesen Weg? Ist er optimal? Warum macht er nicht immer die gleichen Bewegungen? Der Vortragende nutzt die Gelegenheit, einige Details zur automatischen Pfadplanung zu vermitteln. Automatisierte Wege werden benötigt, um eine flexible Produktion zu ermöglichen und somit die Anforderung der „Losgröße eins“ an die Industrie 4.0 zu erfüllen. Losgröße eins bedeutet, dass die Fabrik der Zukunft in der Lage sein muss, kleine Mengen bis hin zu Einzelstücken zu produzieren, um den steigenden Flexibilitätsanforderungen der Kunden gerecht zu werden.

Der letzte Produktionsschritt wird von einem Mitglied des Entwicklungsteams ausgeführt, um zu zeigen, dass Menschen in der Automationsschleife bleiben und in Zukunft mit Robotern zusammenarbeiten werden. Doch der Großteil der Arbeiten wird voraussichtlich von den in weißen Arbeitsmänteln gekleideten Experten erledigt werden, die vermutlich die Programmierung und Instandhaltung der Roboter und der anderen Cyber-Physischen Produktionsmaschinen langfristig betreuen müssen. Während einige Arbeitsplätze verschwinden werden, so der begleitende Diskurs, entstehen neue. Dabei müsse man für Neues offen bleiben.

### **Ethos und Kairos: Für die Gelegenheit passende Expertise**

Nachdem das Publikum davon überzeugt scheint, dass der Demonstrator nicht hartkodiert wurde – die seltsamen, immer wieder neuen, gelungenen Transportbewegungen des Roboters legen diese Einschätzung nahe – schaltet der Vortragende in einen lockeren Präsentationsmodus. Es geht jetzt nicht mehr darum, die Machbarkeit der Smart Factory zu demonstrieren, sondern vielmehr das Know-how des Entwicklungsteams und dessen Verfügbarkeit für andere Projekte bekannt zu machen. Dabei preist er die von der Industrie 4.0-Vision gebotene einzigartige Chance an, sich mit den Herausforderungen der totalen Automatisierung erneut zu beschäftigen. Die anderen Mitglieder des Entwicklungsteams, die bisher eine eher passive Rolle in der Performance spielten, werden nun zum zentralen Fokus des Vortrags. Ihre Expertise sei mehr denn je am Markt gefragt. Angeregt durch Fragen aus dem Publikum skizziert der Vortragende einen weiteren Forschungskontext, in dem der vorliegende Demonstrator lediglich als der neueste und am weitesten fortgeschrittene aus einer ganzen Reihe vergleichbarer Demonstratoren erscheint. Dank des Know-hows der Experten, die in diesen Vorgängerprojekten ebenfalls beteiligt waren, sei das Unternehmen jetzt auf dem richtigen Weg, den Wettbewerb am Industrie 4.0-Markt aufzunehmen.

Zum Abschluss bietet die Präsentation einen Ausblick auf zukünftige Projekte zur Beantwortung noch ungelöster Fragestellungen. Wie erreicht man die notwendige Präzision in der roboterunterstützten Montage? Wie kann man CPPS produktiver und gleichzeitig kostengünstiger machen? Wie wird eine nahtlose Mensch-Maschine-Interaktion ermöglicht? Die Lösung all dieser Probleme wird vorerst den Versprechen des maschinellen Lernens und der künstlichen Intelligenz zugeschrieben. Durch ma-

schinelles Lernen könnten Präzision, Produktivität, Energieverbrauch, Produktionsabläufe und vieles mehr qualitativ verbessert und optimiert werden. 3D-Kameras ermöglichen die automatische Erstellung digitaler Zwillinge für Werkstücke und Produktionseinheiten. Selbstoptimierende CPPS würden das Potenzial maschineller Lerntechniken nutzen, um eine höhere Präzision und Flexibilität zu erreichen. Dafür sei offensichtlich mehr Forschungsarbeit notwendig.

## 6 Schluss

Der vorliegende Text ging von Fragen nach Konzept und Realisierung von Demonstratoren sowie deren Entwicklung, Inszenierung und Bedeutung aus. Die schrittweise empirische Begleitung und anschließende Reflexion eines CPPS-Demonstrator-Projekts in der alltäglichen Praxis eines Technologiekonzerns führte zu Beobachtungen, von denen einige hier herausgegriffen werden.

Es zeigte sich, dass eine bedeutende Funktion der Demonstratoren darin besteht, eine Vision zu vergegenständlichen, sie zu verkörpern, materialisieren, konkretisieren und auszuagieren. Diese Vergegenständlichung adressiert sowohl die Anliegen der Systementwickler als auch die Interessen potentieller Abnehmer intelligenter Produktionsmaschinen sowie neuartiger CPPS-Technologien und Konzepte. Im Rahmen dieser Funktion können Demonstratoren in erster Linie als wirkmächtige, wegweisende Modelle für die Machbarkeit und Plausibilisierung der Industrie 4.0-Vision angesehen werden. Ein erfolgreicher Demonstrator zeigt einen gangbaren Weg in Richtung eines potentiell noch weit entfernten Horizontes.

Trotz ihrer zukunftsorientierten Ausrichtung erfolgt die Entwicklung der Demonstratoren mit den aktuell verfügbaren Mitteln. Dabei wurden im untersuchten Projekt gewisse mimetische Tendenzen in Bezug auf die Übernahme agiler Entwicklungsmethoden und Prinzipien aus anderen Kontexten (z. B. andere Projekte, agile Softwarehäuser) beobachtet. Dazu zählen Artefakte wie etwa das virtuelle Kanban-Board und das zweimal wöchentlich stattfindende „developer meeting“, das nach dem immer gleichen Muster ablief und im Team zu einem unbeliebten Ritual geworden war. Aus Sicht der Entwickler des CPPS-Demonstrators wurde schnell deutlich, dass sich die physische Umgebung der sonst rechenstarken Roboter und anderen Gerätschaften nicht so einfach in Form von Software erfassen ließ. Die arbeitsintensiven Portmessungen und Integrationstests zum Beispiel zeigten auf, dass manche neu belebten Prinzipien agentenbasierter Systeme (Dezentralität, Selbstorganisation, Flexibilität etc.) sich in einem realen Entwicklungs- und Produktionskontext weiterhin als problematisch erwiesen.

Heute werden immer mehr Demonstratoren sowohl in öffentlichen als auch in organisationsinternen Kontexten, wie es das hier behandelte Beispiel zeigt, entwickelt. Dabei scheint sich das Muster zu wiederholen: Ein Team von Programmierern, Forschungsingenieuren oder Studierenden entwickelt ein neuartiges System, um es anschließend einem strategischen Publikum bestehend aus Entscheidungsträgern oder Meinungsbildnern (Manager, Kunden, Politiker, Journalisten) im Rahmen eines Präsentationsrituals vorzuführen. Anschließend wird der Demonstrator verbessert und

mit neuen Features ausgestattet, damit – so wird zumindest erhofft – ein höherer „wow“-Effekt in weiteren Demonstrationen erzielt werden kann. Diese Vorgehensweise zeigt, dass sowohl die Promotoren der Industrie 4.0-Vision als auch deren Abnehmer Praktiken der Modellentwicklung (dem Bau von Demonstratoren) und deren Demonstration vor verschiedenen Öffentlichkeiten (Firmeninterne vs. Verbände und größere Kreise von Universitäten und Firmen) adoptiert haben. Als Modelle für die Fabrik der Zukunft haben Demonstratoren folglich eine performative Funktion, indem sie den bisher weitgehend diskursiven Charakter dieser oftmals kritisch beäugten Vision etwas näher an eine fassbare Realität zu bringen versuchen. Demonstratoren, die für bestimmte Adressatenkreise gezielt entwickelt werden, bieten nicht nur einen wichtigen Zugang zum aktuellen Stand der Technik in Bezug auf die Smart Factory für Entwickler und potentielle Abnehmer solcher Produktionssysteme. Sie zeigen auch einen Weg auf, wie die Vision praktisch zu erreichen ist, nicht zuletzt für eine breitere interessierte Öffentlichkeit.

## Literatur

- Amann, K./Hirschauer, S. 1997: Die Befremdung der eigenen Kultur: Ein Programm. In: Hirschauer, S./Amann, K. (Hg.): Die Befremdung der eigenen Kultur. Zur ethnographischen Herausforderung soziologischer Empirie, Frankfurt a. M.: Suhrkamp, S. 7-52
- Coopmans, C. 2010: ‚Face value‘: New medical imaging software in commercial view. In: *Social Studies of Science*, 41 (2), S. 155-176
- Heßler, M. 2014: Die Halle 54 bei Volkswagen und die Grenzen der Automatisierung. In: *Zeithistorische Forschungen*, 11, S. 56-76
- Kagermann, H./Helbig, J./Hellinger, A./Wahlster, W. (Hg.) 2013: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Deutschlands Zukunft als Produktionsstandort sichern; Abschlussbericht des Arbeitskreises Industrie 4.0, Forschungsunion, Frankfurt a. M.
- Merz, M./Hinterwaldner, I. 2012: Neue Modelle, Bilder und Simulationen: Zwischen Repräsentativität und Produktivität. In: Maasen, S./Kaiser, M./Reinhart, M./Sutter, B. (Hg.): *Handbuch Wissenschaftssoziologie*, Wiesbaden: Springer VS, S. 303-316
- Merz, M. 2016: Epistemische Innovation: Zur Entstehung des Neuen in der Wissenschaft aus Sicht der Science Studies. In: Rammert, W./Windeler, A./Knoblauch, H./Hutter, M. (Hg.): *Innovationsgesellschaft heute. Perspektiven, Felder, Fälle*, Wiesbaden: Springer VS, S. 355-371
- Miller, P./O’Leary, T. 1994: The factory as laboratory. In: *Science in Context*, 7 (3), S. 469-496
- Morrison, M./Morgan, M. S. 1999: Models as Mediating Instruments. In: Morgan, M. S./Morrison, M. (Hg.): *Models as Mediators: Perspectives on Natural and Social Science*, Cambridge UK: Cambridge University Press, S. 10-37
- Pfeiffer, S. 2015: Warum reden wir eigentlich über Industrie 4.0? Auf dem Weg zum digitalen Despotismus. In: *Mittelweg*, 36 (6), S. 14-36
- Schmidt, R. 2008: Praktiken des Programmierens. Zur Morphologie von Wissensarbeit in der Software-Entwicklung. In: *Zeitschrift für Soziologie*, 37 (4), S. 282-300

- Schulz-Schaeffer, I./Bottel, M. 2018: Die Herstellung transnational mobiler Arbeitstätigkeiten in der Softwareentwicklung. In: Quack S./Schulz-Schaeffer I./Shire K./Weiß A. (Hg.): Transnationalisierung der Arbeit, Wiesbaden: Springer VS, S. 99-127
- Schulz-Schaeffer, I./Meister, M. 2017: Laboratory settings as built anticipations – prototype scenarios as negotiation arenas between the present and imagined futures. In: Journal of Responsible Innovation, 4 (2), S. 197-216
- Simakova, E. 2010: RFID 'Theatre of proof': Product launch and technology demonstration as corporate practices. In: Social Studies of Science, 40 (4), S. 549-576
- Smith, W. 2009: Theatre of use: A frame analysis of information technology demonstrations. In: Social Studies of Science, 39 (3), S. 449-480