



Peer-Review: 25.10.2022

Software in Cyberphysischen Produktionssystemen

Herausforderungen zur Umsetzung in der Industrie

Kevin Feichtinger, Johannes Kepler University Linz; Kristof Meixner, Felix Rinker, TU Wien; István Koren, RWTH Aachen University; Holger Eichelberger, University of Hildesheim; Tonja Heinemann, Universität Stuttgart; Jörg Holtmann, Chalmers | University of Gothenburg; Marco Konersmann, Judith Michael, RWTH Aachen University; Eva-Maria Neumann, Technical University of Munich; Jérôme Pfeiffer, Universität Stuttgart; Rick Rabiser, Johannes Kepler University Linz; Matthias Riebisch, Universität Hamburg; Klaus Schmid, University of Hildesheim

Um den effektiven und effizienten Betrieb von Cyberphysischen Produktionssystemen (CPPSen) sicherzustellen, spielt Software eine zunehmend wichtige Rolle. Die enormen Fortschritte bei Softwareentwicklungsmethoden, welche in den letzten Jahren erzielt wurden, scheinen jedoch die aktuellen Herausforderungen der Industrie nicht zu erfüllen, weil diese die Industrie nicht oder nur langsam erreichen. In diesem Beitrag werden die Herausforderungen für die Softwareentwicklung in CPPSen aus Sicht von neun Industrievertretern aus acht europäischen Unternehmen unterschiedlicher Größe diskutiert. Um den digitalen Transformationsprozess für eine zukunftsfähige Produktion zu begleiten, wurden aus den beschriebenen Herausforderungen Perspektiven für die Forschung erarbeitet. Die Umsetzung dieser Ziele ist vor dem Hintergrund von ökonomischen, sozialen und Nachhaltigkeitsanforderungen notwendig.

#Digitale Transformation #CPPS Engineering #Forschungsherausforderungen #Industrie 4.0

Software in Cyber-Physical Production Systems

Challenges for implementation in the industry

To ensure the effective and efficient operation of Cyber-Physical Production Systems (CPPSs), software plays an increasingly important role. Tremendous progress has been made in recent years in the development of advanced software engineering methods. However, they do not seem to meet the current challenges of the industry, as they are not or only slowly reaching the industry. In this paper, the challenges for software engineering in CPPSs are discussed from the perspective of nine industry representatives from eight European companies of different sizes. To accompany the digital transformation process for future-proof production, perspectives for research were developed from the described challenges. The implementation of these goals is necessary considering economic, social, and sustainability requirements.

#digital transformation #CPPS engineering #research challenges #Industry 4.0

1. Einleitung

Cyberphysische Produktionssysteme (CPPSe) werden von unterschiedlichen Disziplinen, wie z. B. der Mechatronik, Elektronik und Softwaretechnik, entwickelt, die anpassbare und auf Kundenbedürfnisse zugeschnittene Produkte herstellen [1]. CPPSe passen sich mit Hilfe modernster Informations- und Kommunikationstechnologie an veränderliche Bedingungen ihrer physischen Umgebung an [1]. In der Praxis wirft die Entwicklung von CPPSen Herausforderungen auf, die multidisziplinäre Lösungen erfordern, um die zunehmende Komplexität zu bewältigen und den Aufwand für Implementierung und Wartung zu reduzieren. Diese Herausforderungen betreffen die Entwicklung und den Betrieb von Software für CPPSe.

In mehreren Arbeiten wurden Herausforderungen für CPPSe erörtert [2] und Forschungsprogramme zu deren Bewältigung

vorgestellt [3, 4]. Moderne Softwareentwicklungsansätze, wie agile Methoden, Modellierungstechniken, kontinuierliche Integration und Bereitstellung sowie Variabilitätsmodellierung, könnten dabei helfen, diese Herausforderungen zu bewältigen. Nach den Erfahrungen der Autor:innen und neuen empirischen Ergebnissen [5] scheinen diese Ansätze jedoch die Industrie nicht zu erreichen.

Deshalb initiierten wir 2021 den *Software Engineering in Cyber-Physical Production Systems Workshop* (SECPPS-WS) [6, 16], um das Thema näher zu untersuchen. Ziel des SECPPS-WS ist es, Ansätze und Methoden zur Softwareentwicklung für CPPSe zu diskutieren und die Herausforderungen bei der Anwendung dieser Ansätze und Methoden auf die CPPS-Domäne zu analysieren. Nach zwei Workshops wurde eine regelmäßige Lesegruppe eingerichtet, um einen Überblick über die aktuelle Literatur zu erhalten und (Teil-)Lösungen

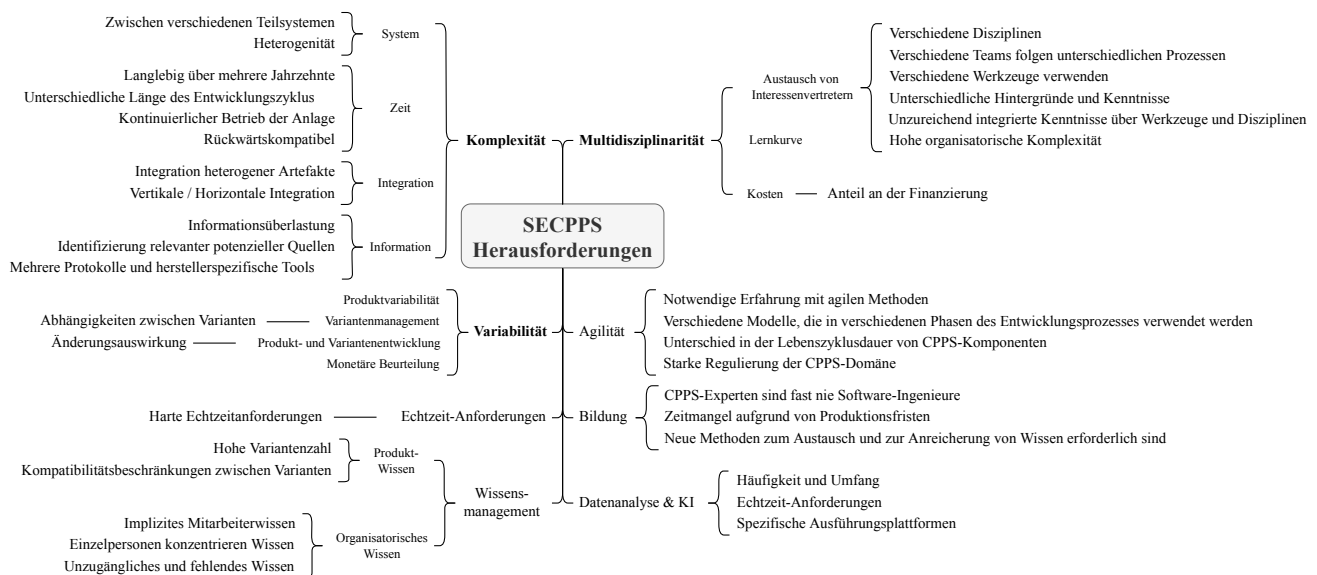


Abbildung 1: Übersicht über die identifizierten SECPPS-Herausforderungen. Adaptiert von [8].

für bereits beschriebene Herausforderungen zu sammeln. Zum dritten SECPPS-WS luden wir Industrievertreter ein, um aktuelle Herausforderungen aus der industriellen Praxis der Softwareentwicklung zu erarbeiten. Neun Industrievertreter aus acht europäischen Unternehmen unterschiedlicher Größe, die CPPSe oder Lösungen für CPPSe entwickeln, nahmen teil. Die Domänen reichten von der Automobilherstellung über die medizinische Versorgungstechnik bis hin zu Lösungsanbietern und Integratoren für CPPSe. Auch wenn die Industrievertreter:innen nicht die gesamte Branche abdecken konnten, achteten wir auf eine sorgfältige Auswahl, um eine möglichst repräsentative Gruppe von Expertinnen und Experten zu erhalten [7]. Im Anschluss erörterten die Autor:innen des Artikels die beschriebenen Herausforderungen und untersuchten die von der Lesegruppe gefundene Literatur, um festzustellen, inwieweit diese auf die Herausforderungen eingeht. Die Ergebnisse wurden im Rahmen der ETFA 2022 Konferenz [8] präsentiert. Im Folgenden bieten wir einen Überblick über die angesprochenen Themen, zeigen Forschungsperspektiven auf und skizzieren mögliche Lösungen.

2. Softwaretechnische Herausforderungen

Abbildung 1 zeigt die Kategorisierung der Herausforderungen [8], wobei der Fokus dieses Artikels auf den Herausforderungen Komplexität, Multidisziplinarität und Variabilität liegt.

2.1 Komplexität

Komplexität in der Softwareentwicklung für CPPSe wurde entlang mehrerer Dimensionen berichtet.

- » **Systemkomplexität:** Zwei Workshopteilnehmer aus der Prozesstechnik und Systemintegration berichteten über Systemkomplexität. Diese wird durch die große Anzahl verschiedener Subsysteme und deren Heterogenität, wie

Sensor-, Echtzeit- oder Prozessoptimierungssysteme, die synchronisiert miteinander agieren müssen, induziert.

- » **Zeitliche Komplexität:** Ein Unternehmen aus dem Bereich Sondereinsatzfahrzeuge berichtete, dass ihre CPPSe über Jahrzehnte genutzt werden. Diese Langlebigkeit stellt eine Herausforderung für die Wartung und Weiterentwicklung von CPPSen und deren Software dar. Mehrere Workshop-Teilnehmer wiesen darauf hin, dass unterschiedlich lange Entwicklungszyklen der beteiligten Disziplinen diese Herausforderung wesentlich bestimmen. Dabei sind die Entwicklungszyklen, etwa der Mechanik, üblicherweise bedeutend länger als die der Software. Ein Teilnehmer aus dem Anlagenbau beschrieb die Notwendigkeit eines kontinuierlichen Anlagenbetriebs (24/7/365). Dies erschwert im Vergleich zur Softwareentwicklung Experimente in der Produktion erheblich. Zusätzlich müssen CPPSe über ihre Nutzungszeit weiterentwickelt werden und dennoch rückwärtskompatibel bleiben. Modularität und klar definierte Schnittstellen können zwar helfen, erhöhen die Komplexität aber weiter, etwa bei Schnittstellenänderungen.

- » **Integrationskomplexität.** Die Komplexität der CPPSe erfordert Fachleute unterschiedlicher Bereiche. Jeder Bereich nutzt diverse Entwicklungsartefakte, um die unterschiedlichen Sichten zu realisieren. Die Integration dieser Artefakte, z. B. mittels modellbasiertem Systems Engineering, erfordert eine vertikale Integration, d. h. über mehrere Abstraktions- oder Produktionsschichten (wie Komponente, Subsystem und System) und eine horizontale Integration, d. h. über Systeme derselben Schicht hinweg. Diese Integrationsherausforderungen erstrecken sich auch auf Werkzeuge, Software, Schnittstellen und andere Komponenten, die an komplexen Systems-of-Systems beteiligt sind. Für einige Bereiche finden sich Lösungsansätze, wie etwa Module Type Packages aus

der Prozesstechnik [9]. Jedoch kann die Verallgemeinerung, abhängig von der gewählten Architektur oder dem Protokoll, eingeschränkt werden.

- » **Informationskomplexität:** CPPSe produzieren große Datenmengen, die zu Informationsüberlastung führen können. Ein von den Workshopteilnehmern genanntes Beispiel ist eine hohe Anzahl von Ereignismeldungen in Mensch-Maschine-Schnittstellen. Ferner unterscheiden sich die Daten je nach Quelle und Kontext in ihrer Übertragungsgeschwindigkeit, Qualität und Gültigkeit. Aufgrund der Menge und der zeitlichen Anforderungen von Ereignisdaten wird die Skalierbarkeit von Lösungen zu einem Problem. Es wird etwa berichtet, dass die Identifizierung von Fehlerquellen nicht trivial ist, z. B. wenn ein Produktfehler in der Qualitätskontrolle erkannt wird. Ein weiterer Aspekt, der die Komplexität der in CPPSen übertragenen Informationen erhöht, ist die Vielzahl von Protokollen, einschließlich der unterschiedlichen Semantik. Herstellerspezifische Werkzeuge zur Verwaltung dieser Protokolle erschweren zusätzlich die Kommunikation.

Fazit: Komplexität ist ein mehrdimensionales Problem

Zeitliche, System-, Integrations- und Informationskomplexität sind ein integraler Aspekt im CPPS-Kontext. Modellierungstechniken und domänenspezifische Sprachen können helfen, die Komplexität zu bewältigen. Ähnlich kann eine zusätzliche Automation durch geeignete Softwaretechniken, etwa kontinuierliche Integration, die Handhabung der Komplexität unterstützen. Dabei ist darauf zu achten, dass die Informationsflut für Mitarbeitende durch geeignete Informationsaufbereitung und Visualisierung im Rahmen gehalten wird.

2.2 Multidisziplinarität

Entwickler:innen multipler Disziplinen sind an der Realisierung von CPPSen beteiligt, wodurch sich einige Herausforderungen ergeben.

- » **Wissens- und Werkzeugintegration:** Softwareingenieur:innen, die für die Entwicklung von CPPS-Software verantwortlich sind, benötigen Wissen aus anderen Disziplinen [4]. Wie mehrere Workshopsteilnehmer aus der Prozesstechnik und dem Industrieconsulting bestätigten, haben Teams aus verschiedenen Disziplinen diverse Hintergründe und Kenntnisse und benutzen unterschiedliche Werkzeuge, z. B. bei der Entwicklung von (Steuerungs-)Software. Die Werkzeuge und das Fachwissen sind schlecht integriert, was zu einer hohen organisatorischen Komplexität führt (vgl. Abschnitt 2.1). Die Werkzeugunterstützung und -akzeptanz von wissenschaftlichen Integrationsansätzen in Industrieprojekten ist jedoch noch nicht ausreichend.
- » **Stakeholder-Austausch:** In der Praxis müssen beim CPPS-Engineering wichtige Systemmerkmale über mehrere

Disziplinen hinweg definiert werden. Industrieteilnehmer vermissen regelmäßige Abstimmungstreffen in der Praxis. Die Informationsweitergabe basiert in der Regel auf manuell erstellten und teils aus verschiedenen Tools exportierten Dokumenten und Tabellen. Auf der Grundlage des in diesen Dokumenten enthaltenen Wissens entwickeln und pflegen Softwareingenieur:innen die CPPS-Software.

- » **Lernkurve:** Viele Branchen verwenden alte Industriestandards anstelle moderner Softwareentwicklungsansätze. Ein Teilnehmer aus der Prozesstechnik erklärte, dass es mehrere Jahre Erfahrung benötigt, bis Softwareingenieur:innen mit diesen vertraut sind. Forschende arbeiten daher daran, moderne Ansätze wie DevOps in die Industrie zu bringen [10]. In der Praxis kann die Anwendung neuer Ansätze jedoch zu Problemen führen, z. B. kommt es durch mangelnde Tests zu Integrationsproblemen.
- » **Softwarekosten:** Laut einem Teilnehmer aus dem Anlagenbau ist der Anteil der Finanzierung von Software im Vergleich zum Gesamtbudget eines CPPS-Projekts minimal. Dies betrifft insbesondere die Basisautomatisierung, die bei einigen Projekten etwa 0,5 bis 2 % des Budgets ausmacht.

Fazit: CPPSe erfordern echte multidisziplinäre Ansätze und Werkzeuge

In der Praxis sind die Werkzeuge und Artefakte oft disziplinspezifisch und schwer zu integrieren. Daher kann der Informationsaustausch über gemeinsame Formate wie AutomationML nur ein erster Schritt sein. Außerdem ist eine angemessene (disziplinenübergreifende) Interaktion zwischen Stakeholdern in der Praxis trotz der Werkzeuge schwer zu realisieren.

2.3 Variabilität

Über Jahrzehnte hinweg gepflegte CPPSe führen zu einer Variantenflut, um spezifische Kundenanforderungen zu erfüllen. Wie von allen Teilnehmern bestätigt wurde, erwarten Kunden und Kundinnen ein System, das auf ihre spezifischen Bedürfnisse zugeschnitten ist. Der Umgang mit Variabilität in der Industrie hängt jedoch stark von meist implizitem Fachwissen und kundenspezifischen Tools ab, die sich auf bestimmte Artefakte und Software- und Hardwareplattformen konzentrieren [4, 3]. Den Berichten zufolge können die Herausforderungen den Themen Produktvariabilität, Variantenmanagement und Entwicklung sowie deren monetären Bewertung zugeordnet werden.

- » **Produktvariabilität:** Zwei Teilnehmer aus dem Sonderfahrzeugbau und dem Verpackungsmanagement erwähnten die gewünschte hohe Produktvariabilität in der modernen Produktion. Während ein Teilnehmer anmerkte, dass seine Produkte sehr individuell sind, fügte ein zweiter hinzu, dass spezifische Geschäftsprozesse zu

ihrer Modellierung fehlen. Es fehlen Ansätze, Variabilität aus mehreren Perspektiven, insbesondere Produkt-, Prozess- und Geschäftsperspektiven, zu betrachten.

- » **Variantenmanagement:** Mehrere Teilnehmer des Workshops aus unterschiedlichen Domänen betonten, dass die hohe Anzahl unterschiedlicher CPPS-Varianten und insbesondere das Management von deren Kompatibilität eine besondere Herausforderung ist. Zu diesem Zweck müssen die Abhängigkeiten zwischen Varianten modelliert werden. Hierzu können Ansätze zur Variabilitätsmodellierung verwendet werden [11]. Jedoch scheinen nur wenige dieser Ansätze in Bezug auf CPPSe in der Industrie angekommen zu sein [5].
- » **Variantenentwicklung:** Die Teilnehmer berichteten über die Produktevolution und die Auswirkungen auf das Produktionssystem als eine wesentliche Herausforderung. Die Auswirkungen einer Produktevolution sollten idealerweise im Voraus sichtbar gemacht werden können. Die Evolution von Software ist immer noch eine Herausforderung für die Softwareentwicklung [11].
- » **Monetäre Bewertung:** Ein Teilnehmer aus der Prozesstechnik stellte fest, dass die monetäre Bewertung von Produktvarianten herausfordernd ist. Wie kann etwa der Wert einer zusätzlichen Produktvariante automatisch bewertet werden, und sind Kunden und Kundinnen (des CPPS) bereit, zusätzliche Kosten zu zahlen? Derzeitige Ansätze zur Variabilitätsmodellierung konzentrieren sich weitgehend auf die Produktmerkmale, berücksichtigen aber weder die Anzahl zu produzierender Produkte noch den monetären Wert der Merkmale. Kostenmodelle berücksichtigen die Frage der physischen Produktionsprozesse und ihrer Variabilität nicht, was jedoch für CPPSe essenziell ist.

Fazit: Akademische Ansätze des Variabilitätsmanagement erreichen die Industrie nicht

Variabilitätsmanagement beeinflusst sowohl Entwicklungs- als auch Geschäftsprozesse im CPPS-Engineering, jedoch erreichen akademische Ansätze hierfür nicht die Industrie. Daher ist es notwendig, solche Ansätze gezielter auf Industriebedürfnisse anzupassen und etwa die monetäre Bewertung von Variabilität und physische Produktionsprozesse zu berücksichtigen.

2.4 Weitere Herausforderungen

Im Folgenden werden weitere Herausforderungen der Praxis, die über die vorigen Themen hinausgehen, beschrieben. So wurden vor allem Herausforderungen zu den Themen Agilität, Aus- und Weiterbildung, Wissensmanagement, Echtzeitanforderungen, sowie Datenanalyse und KI genannt. Eine detaillierte Diskussion dazu ist in [8] zu finden.

- » **Agilität:** Einem Teilnehmer aus der Systemintegration zufolge besteht häufig ein Mangel an Erfahrung mit agilen

Methoden und insbesondere mit modellbasierten Entwicklungsmethoden. Die Modellverifizierung und -validierung, etwa mit kontinuierlicher Integration, ist aufgrund der Komplexität und Vielfalt der Modelle unerlässlich. Agile Softwareentwicklungsprozesse für CPPSe müssen unterschiedliche Skalen (kurze Produktlebenszyklen und Software-Update-Zyklen vs. Langlebige Produktionsanlagen) berücksichtigen, was aber auch in der CPPS-Entwicklung kurze Innovationszyklen erfordert. Dem Teilnehmer zufolge werden jedoch diese Innovationszyklen häufig durch die starke Regulierung behindert, welche je nach Domäne auch notwendig ist. Nichtsdestotrotz gibt es Versuche, die Integration von (Teil-)Prozessen besser zu unterstützen [10].

- » **Aus- und Weiterbildung:** Obwohl inzwischen viele Fachleute in der CPPS-Entwicklung im Rahmen ihrer Ausbildung programmieren lernen, sind sie selten ausgebildete Softwareingenieur:innen. Studiengänge wie Mechatronik und Softwaretechnik enthalten bereits multidisziplinäre Elemente, was auch für die CPPS-Entwicklung vorteilhaft ist. Jedoch wären mehr Bildungsangebote und Weiterbildungsmöglichkeiten für CPPS-Entwickler:innen notwendig. Berichten aus dem Workshop zufolge ist die Wissensweitergabe erfahrener Kolleginnen und Kollegen sowie Softwareingenieur:innen eine Herausforderung. Grund ist oft ein Zeitmangel bei der Ausbildung am Arbeitsplatz, um Projektfristen gerecht zu werden. In dieser Hinsicht werden neue Methoden zum Austausch und zur Anreicherung von Wissen während der CPPS-Entwicklung benötigt.
- » **Wissensmanagement:** CPPSe müssen domänenspezifisches Wissen aus heterogenen Informationsquellen integrieren, um die Grundlage etwa für Analysen, Simulationen und Entscheidungen zu schaffen. Organisatorisches Wissen wird als implizites Mitarbeiterwissen oft nicht explizit gemacht. Dies ist eine Herausforderung, da Einzelpersonen über die Zeit Wissen akkumulieren, es aber von diesen Personen abhängt, ob der Wissenstransfer funktioniert.
- » **Echtzeitanforderungen:** Ein Teilnehmer aus dem Sonderfahrzeugbau berichtete über Herausforderungen durch Echtzeitanforderungen von CPPSen, wie etwa Reaktionszeiten von Subsystemen unter 10 ms. Insbesondere die Heterogenität der Subsysteme erzwingt Klassifizierungen von Echtzeitanforderungen, wobei Anwendungen unterschiedlicher Klassen und somit unterschiedlich harter Echtzeitanforderungen miteinander interagieren müssen.
- » **Datenanalyse und KI:** Der vermehrte Einsatz von Datenanalyse und Künstlicher Intelligenz (KI) führt dazu, dass qualitativ hochwertige Daten in ausreichender Menge von CPPSen erhoben werden. Häufigkeit und Umfang von Maschinendatenströmen aus spezifischen Ausführungsplattformen mit harten Echtzeitanforderungen bei hohen Abstraten führen zu Herausforderungen in der

Verarbeitung der Datenmengen. Ein wesentliches Problem im Umgang mit KI ist die Verständlichkeit und damit die Vertrauenswürdigkeit der Ergebnisse, etwa aus Sicht der Operateure [12]. Hier können sich in Entwicklung befindliche Ansätze zur Erklärbarkeit Künstlicher Intelligenz beitragen.

Fazit: Der Faktor Mensch, die Echtzeitanforderungen und komplexe Datenanalysen stellen Herausforderungen für die Industrie dar

Die Berücksichtigung menschlicher Faktoren ist von entscheidender Bedeutung. Ingenieur:innen sind eine wichtige Quelle für Informationen über Produkte und Varianten, das Unternehmen und seine Prozesse. Ihr Wissen zu modellieren, es in der Software zu nutzen, sachkundige Mitarbeitende zu halten und ihre Kenntnisse (z. B. bezüglich agiler Methoden) auf den neuesten Stand zu bringen, wird auch in Zukunft ein wesentlicher Faktor sein. Echtzeitanforderungen führen zu Herausforderungen in der Kalibrierung und Steuerung von CPPSen, aber auch in der Modellierung und Analyse von Produktionsdaten, denen Wissenschaft und Praxis gemeinsam begegnen müssen.

3. Forschungsperspektiven

Für die Forschung zur Softwareentwicklung ergeben sich folgende Perspektiven, um den digitalen Transformationsprozess zu einer zukunftsfähigen Produktion zu begleiten. Um den involvierten Disziplinen die Beibehaltung eigener Werkzeuge, Artefakte und Arbeitsweisen zu ermöglichen, müssen Mittel zur Wissensintegration entwickelt werden. Auch die Unterstützung der Variabilität in Raum (Varianten) und Zeit (Versionen) ist unerlässlich. Bei der CPPS-Entwicklung werden im Gegensatz zur Softwareentwicklung (grafische) technische Modelle statt textbasierten Artefakten verwendet. Daher müssen bestehende Varianten- und Versionsverwaltungssysteme an den CPPS-Kontext angepasst werden.

Eine grundlegende Eigenschaft vieler Herausforderungen ist, dass die Disziplinen „nicht dieselbe Sprache sprechen“. Jedoch erfordert die Entwicklung von Software für CPPSe Kenntnisse all diese Disziplinen. Ein Lösungsansatz sind Ontologien, die Konzepte verschiedener Disziplinen darstellen, die das erforderliche Wissen und seine Metainformationen abdecken. Ein weiterer Lösungsansatz ist die Verwendung von verbundenen, konsistenzunterstützten Modellierungsansätzen, etwa dem *Common Concept Glossary*-Ansatz [13], mit domänenspezifischen Modellierungssprachen als Teil von integrierten Wissensgraphen. Darüber hinaus muss die Forschung multidisziplinäre Ansätze zur Wissens- und Werkzeugintegration – statt einer einfachen Verschmelzung bestehender Werkzeuge – vorantreiben. So sind bestehende Werkzeuge für das Lebenszyklusmanagement sowie die Werkzeugintegration in mehrfacher Hinsicht unzureichend. Während etwa Werkzeuge für Produktlebenszyklusmanagement aus dem Maschinenbau für die Verwaltung von Modellen und Daten physischer Teile geeignet sind, aber Artefakte der Softwareentwicklung zu sehr abstrahieren, eignen sich etwa

Werkzeuge des Anwendungslebenszyklusmanagement aus der Softwareentwicklung nicht, um Artefakte unterschiedlicher Disziplinen zu integrieren. Des Weiteren fehlt das gegenseitige Verständnis der Arbeitsweise, sodass die Adaptierung von bestehenden Ansätzen nicht vorangetrieben werden kann [5].

Zur Verbindung von disziplinären Modellen und Konzepten eignen sich insbesondere Digitale Zwillinge, die weitere Möglichkeiten zur Simulation, Integration, Prüfung und Eigenschaftsvorhersage bieten. Außerdem ermöglichen sie Rückwärtskompatibilität und den Dauerbetrieb. Offene Fragen für die Forschung hierzu sind die horizontale und vertikale Integration verschiedener Teilsysteme und die Berücksichtigung des Zeitverhaltens. Aktuelle Forschung betrifft etwa die Verwaltungsschale [14], Module Type Packages [9] oder Fähigkeiten und Fertigkeiten [15], welche um Abstraktion und Entkoppelung bemüht sind. Die Integration neuer Technologien bzw. die Umsetzung neuer Standards ist durchaus mit Risiken behaftet. Beispielsweise könnte sich ein massiver Einsatz von *Verwaltungsschalen* auf die Skalierbarkeit des Systems auswirken. Zudem können neue Versionen eines Standards oder davon abgeleiteter Formate wie bei Verwaltungsschalen oder *OPC UA Companion Specifications* unerwartete Evolutionen auslösen oder sogar erzwingen. Die Softwareentwicklung kann hier insbesondere mit neuen Technologien unterstützen, z. B. durch generative Ansätze die Effizienz der Integration erhöhen, mit domänenspezifischen Sprachen oder der Modellverifikation, um Modellierungsfehler zu vermeiden, oder mit Analysen bzw. mit Werkzeugen, welche Skalierbarkeit bzw. Performanz vorhersagen [2].

Die Forschung muss im Rahmen von langfristigen Leuchtturmprojekten zeigen, dass Agilität und modellgetriebene Entwicklungsmethoden in der industriellen Praxis gut angewendet werden können. Dazu werden mehr Feldversuche mit neuartigen Methoden benötigt, z. B. modellbasierte Ansätze und (Software-)Produktlinienansätze. Dafür wären Industriepartner, die sich auf eine langfristige und unternehmensübergreifende Zusammenarbeit einlassen, wünschenswert.

4. Fazit

Dieser Beitrag stellt Herausforderungen für die Softwareentwicklung im CPPS-Engineering vor, motiviert durch Stimmen von neun Praktikern aus acht verschiedenen europäischen Unternehmen. Zu den wichtigsten Herausforderungen für die Softwareentwicklung gehören Komplexität, Multidisziplinarität und Variabilität. Ungelöste Herausforderungen reichen von der Notwendigkeit von Entwicklungsansätzen und -Werkzeugen, die ein echtes multidisziplinäres Umfeld fördern über akademische Ansätze, die die Industrie nicht erreichen, bis hin zur Entwicklung von CPPS-Varianten/-Artefakten. Diese Herausforderungen müssen von Forschenden und Praktizierenden der Softwaretechnik im Hinblick auf ökonomische, soziale und nachhaltige Anforderungen stärker berücksichtigt werden. Basierend auf den beschriebenen Herausforderungen wurden Forschungsperspektiven definiert, welche die kooperative Zusammenarbeit von Forschenden und Praktizierenden erfordern.

Danksagung

Wir bedanken uns bei allen Teilnehmern der SECPSS-Workshopserie. Wir bedanken uns für die finanzielle Unterstützung durch das Bundesministerium für Digitalisierung und Wirtschaftsstandort und die Nationalstiftung für Forschung, Technologie und Entwicklung sowie die Christian Doppler Forschungsgesellschaft. Das diesem Bericht

zugrundeliegende Vorhaben wurde teilweise gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (DFG) durch das Exzellenzcluster Internet of Production (EXC-2023 – 390621612), dem Bundesministerium für Bildung und Forschung (XAPS – 01IS19084D, EXPLAIN – 01IS22030E) und dem Bundesministerium für Wirtschaft und Klimaschutz (IIP-Ecosphere – 01MK20006C, SofDCar – 19S21002L).

Referenzen

- [1] Biffl, S., Gerhard, D., Lüder, A. (2017). Introduction to the multi-disciplinary engineering for cyber-physical production systems. *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*, 1-24.
- [2] Vogel-Heuser, B., Fantuzzi, C., Wimmer, M., Böhm, M., Fay, A. (2019). Herausforderungen in der interdisziplinären Entwicklung von Cyber-Physischen Produktionssystemen. *at-Automatisierungstechnik*, 67(6), 445-454.
- [3] Galster, M., Zdun, U., Weyns, D., Rabiser, R., Zhang, B., Goedicke, M., Perrouin, G. (2017). Variability and complexity in software design: Towards a research agenda. *ACM SIGSOFT Software Engineering Notes*, 41(6), 27-30.
- [4] Rabiser, R., & Zoitl, A. (2021). Towards mastering variability in software-intensive cyber-physical production systems. *Procedia Computer Science*, 180, 50-59.
- [5] Berger, T., Steghöfer, J. P., Ziadi, T., Robin, J., Martinez, J. (2020). The state of adoption and the challenges of systematic variability management in industry. *Empirical Software Engineering*, 25, 1755-1797.
- [6] Rabiser, R., Vogel-Heuser, B., Wimmer, M., Wortmann, A., Zoitl, A. (2022). Workshop on Software Engineering in Cyber-Physical Production Systems (SECPSS), 2nd Edition. In *Software Engineering*, ser. LNI, vol. P-320. Gesellschaft für Informatik e.V., 2022.
- [7] Felderer, M., Travassos, G. H. (2020). *Contemporary empirical methods in software engineering* (pp. 285-311). Berlin/Heidelberg, Germany: Springer.
- [8] Feichtinger, K., Meixner, K., Rinker, F., Koren, I., Eichelberger, H., Heineemann, T., ... Schmid, K. (2022). Industry Voices on Software Engineering Challenges in Cyber-Physical Production Systems Engineering. In *Proceedings of the 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022.
- [9] VDI/VDE/NAMUR 2658-4. (2022). Automation engineering of modular systems in the process industry - Modelling of module services. VDI: www.beuth.de.
- [10] Brauner, P., Dalibor, M., Jarke, M., Kunze, I., Koren, I., Lakemeyer, G., ... Ziefle, M. (2022). A computer science perspective on digital transformation in production. *ACM Transactions on Internet of Things*, 3(2), 1-32.
- [11] Raatikainen, M., Tiihonen, J., Männistö, T. (2019). Software product lines and variability modeling: A tertiary study. *Journal of Systems and Software*, 149, 485-510.
- [12] Liao, Q. V., Gruen, D., Miller, S. (2020). Questioning the AI: informing design practices for explainable AI user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1-15).
- [13] Rinker, F., Waltersdorfer, L., Meixner, K., Biffl, S. (2019). Towards support of global views on common concepts employing local views. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1686-1689). IEEE.
- [14] Bundesministerium für Wirtschaft und Klimaschutz. (2022). *Details of the Asset Administration Shell - Part 1*. Abgerufen von: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html
- [15] Järvenpää, E., Siltala, N., Hylli, O., Lanz, M. (2021). Capability matchmaking software for rapid production system design and reconfiguration planning. *Procedia CIRP*, 97, 435-440.
- [16] Rabiser, R. (2022). *Welcome to the Homepage of the SECPSS Workshop Series*. Abgerufen von: <https://rickrabiser.github.io/secpps-ws/>

AUTOREN

Kevin Feichtinger ist Doktorand am LIT Cyber-Physical Systems Lab an der Johannes Kepler Universität Linz, Österreich (JKU). Er schloss 2019 sein Masterstudium in Software Engineering an der JKU ab. Seine aktuellen Forschungsinteressen umfassen Variabilitätsmodellierung, Modelltransformationen, Softwareevolution und Softwareentwicklung für Cyberphysische (Produktions-) Systeme.

Kevin Feichtinger, M.Sc.

Johannes Kepler Universität Linz
LIT Open Innovation Center
Altenberger Str. 69
4040 Linz
☎ +43 732-2468-9489
@ kevin.feichtinger@jku.at

Kristof Meixner ist Doktorand am Christian Doppler Labor SQI der TU Wien. Seine Forschungsinteressen umfassen Aspekte der Wiederverwendung und Variabilität in der Konstruktion von Cyberphysischen Produktionssystemen. Er ist auch Mitorganisator der Spezialsession über Software Engineering und fähigkeits- und fertigkeitsbasiertes Engineering auf der ETFA und des Workshops für Evolution in der Variabilität auf der SPLC.

Felix Rinker ist Doktorand am Christian Doppler Labor SQI der TU Wien. Seine Forschungsinteressen umfassen Aspekte der Multi-View-Risiko- und Ingenieurwissensintegration, einschließlich des Änderungsworkflowmanagements für Cyberphysische Produktionssysteme. Er ist Mitorganisator der Spezialsession über Software Engineering für Cyberphysische Produktionssysteme (SECPSS) auf der ETFA.

István Koren ist Postdoc am Lehrstuhl für Process and Data Science (Prof. Wil van der Aalst) an der RWTH Aachen University und dort stellvertretender Koordinator des Bereichs digitale Infrastruktur im Exzellenzcluster „Internet of Production“. Seine Forschungsarbeiten befassen sich mit der Entwicklung und dem Betrieb komplexer software- und datenintensiver Informationssysteme.

Holger Eichelberger ist promovierter Mitarbeiter am Institut für Informatik der Universität Hildesheim und dort stellvertretender Arbeitsgruppenleiter. Seine Forschungsinteressen umfassen Software-Produktlinien, modellbasiertes Software-Engineering und Performance-Engineering. Er ist Projektleiter im IIP-Ecosphere Projekt und maßgeblich verantwortlich für die Entwicklung der IIP-Ecosphere Plattform.

Tonja Heinemann ist wissenschaftliche Mitarbeiterin im Bereich Industrielle Steuerungstechnik am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen der Universität Stuttgart. Ihr Studium hat sie mit einem M.Sc. in Mechatronik an der Universität Stuttgart und einem M.Sc. in Systems, Control and Mechatronics an der Chalmers University of Technology abgeschlossen.

Jörg Holtmann ist PostDoc an der Chalmers University of Technology | Universität Göteborg (Schweden). Zuvor war er Seniorexperte am Fraunhofer IEM (Paderborn, Deutschland), nachdem er seine Dissertation an der Universität Paderborn abgeschlossen hat. Seine Forschungsinteressen liegen im modellbasierten Requirements/Systems/Software Engineering für softwareintensive Systeme.

Marco Konersmann ist Postdoc am Lehrstuhl für Software Engineering an der RWTH Aachen, Deutschland. Seine Forschungsschwerpunkte sind modellbasierte Softwarearchitektur und Software-Engineering sowie die Beziehung zwischen Modellen, Implementierungs- und Laufzeitsichten. In seiner Dissertation beschäftigte er sich mit der Integration von Softwarearchitektur-Modellen in Code.

Judith Michael ist PostDoc und Teamleiterin am Lehrstuhl für Software Engineering der RWTH Aachen, Aufsichtsratsmitglied der Lakeside Science & Technology Park GmbH und Präsidiumsmitglied der Gesellschaft für Informatik. Sie erforscht das Engineering komplexer, langlebiger, softwareintensiver Systeme, wie z. B. für Digitale Zwillinge, und entwickelt Software-Sprachen und modellgetriebene Methoden für Informations- und Assistenzsysteme.

Eva-Maria Neumann schloss 2018 ihr Masterstudium im Maschinenbau an der Technischen Universität München

(TUM) ab. Derzeit promoviert sie am Lehrstuhl für Automatisierungs- und Informationssysteme der TUM. Ihre Forschungsschwerpunkte sind statische Code-Analyse und Softwaremetriken zur Quantifizierung der Qualität von Steuerungssoftware sowie der funktionsorientierte Entwurf von modularen Steuerungssoftwarearchitekturen zur Verbesserung der Wiederverwendbarkeit.

Jérôme Pfeiffer ist ein wissenschaftlicher Mitarbeiter am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) der Universität Stuttgart. Seine Forschungsinteressen liegen im Bereich Software Language Engineering und der modellgetriebenen Entwicklung von Digitalen Zwillingen.

Rick Rabiser ist Universitätsprofessor für Software Engineering in Cyber-Physical Systems und Leiter des LIT Cyber-Physical Systems Lab und des Christian Doppler VaSiCS an der Johannes Kepler Universität Linz. Gemeinsam mit seinem Team forscht er an der Beherrschung der Software-Variantenvielfalt in (Cyberphysischen) (Produktions-)Systemen, Softwareproduktlinien, Softwarewartung und -evolution und Usability von Software Engineering Werkzeugen.

Matthias Riebis hat Automatisierungstechnik an der TU Dresden studiert. Anschließend promovierte und habilitierte er an der TU Ilmenau im Fachgebiet Softwaresysteme/Prozessinformatik. Von 2012 bis 2023 forschte und lehrte er am Fachbereich Informatik der Universität Hamburg als Professor für Softwareentwicklungs- und -konstruktionsmethoden. Er verband dabei insbesondere die Themengebiete Software-Architektur, Software-Reengineering und Software-Modellierung.

Klaus Schmid ist Universitätsprofessor und leitet die Arbeitsgruppe Software Systems Engineering an der Universität Hildesheim. Im Fokus der Arbeitsgruppe stehen Softwarequalität, Industrie 4.0, Intelligente Softwaresysteme und E-Learning-Systeme. Diese Themen werden in einer Vielzahl von Forschungsprojekten, insbesondere in Kooperation mit Unternehmen, verfolgt.