Ansätze zur Beherrschung der Variabilität in Software-intensiven Cyber-Physischen Produktionssystemen

A. Gutierrez, L. Sonnleithner, A. Zoitlo, R. Rabiser

Software-intensive Cyber-Physische Produktionssysteme (SiCPPS) sind hochkomplexe vernetzte Systeme, die autonom miteinander und mit ihrer Umgebung interagieren, um flexibel eine Vielzahl an Produkten zu produzieren. Solche Systeme haben typischerweise eine lange Lebensdauer, über die eine Vielzahl von Varianten entwickelt und gewartet werden muss. Diese Systeme bestehen aus vielen heterogenen Hard- und Softwarekomponenten, welche auf verschiedene Arten konfiguriert und kombiniert werden können. Variabilität betrifft dabei nicht nur diese Komponenten, sondern auch Entwicklungsprozesse, Anwendungsdomänen sowie Methoden und Werkzeuge. In der Industrie hängt der Umgang mit Variabilität jedoch aktuell stark von Expertenwissen als auch von selbst entwickelten Werkzeugen ab. Diese funktionieren oft nur mit ganz bestimmten Artefakten bzw. Hard- und Softwareplattformen. Existierende Forschungsansätze aus dem Gebiet der SiCPPS beschäftigen sich nicht explizit mit der Beherrschung dieser Variabilität. In diesem Beitrag definieren wir eine Forschungsagenda zur Beherrschung der Variabilität in SiCPPS. Wir konzentrieren uns dabei auf die Entwicklung von Techniken (i) zur Erhebung und Modellierung von Variabilität und (ii) zur Automatisierung sowohl der Generierung von Artefakten in SiCPPS als auch der Generierung und Wartung von Modellen auf Basis bestehender Artefakte.

Schlüsselwörter: Variabilität; Cyber-Physische Produktionssysteme; Variabilitätsmodellierung; Variability Mining; Evolution

Approaches to mastering variability in software-intensive Cyber-Physical Production Systems.

Software-intensive Cyber-Physical Production Systems (SiCPPS) are highly complex networked systems that autonomously interact with each other and their environment to flexibly produce a variety of products. Such systems typically have long lifetimes over which a large number of variants must be developed and maintained. These systems consist of many heterogeneous hardware and software components, which can be configured and combined in various ways. Variability affects not only these components, but also the development processes, application domains, and methods and tools. In industry, however, the handling of variability currently depends heavily on expert knowledge as well as on tools developed in-house. These often only work with very specific artifacts or hardware and software platforms. Existing research approaches in the field of SiCPPS do not explicitly address the management of this variability. In this paper, we define a research agenda for mastering variability in SiCPPS. We focus on developing techniques (i) for eliciting and modeling variability and (ii) for automating both the generation of artifacts in SiCPPS and the generation and maintenance of models based on existing artifacts.

Keywords: variability; Cyber-Physical Production Systems; variability modeling; variability mining; evolution

Eingegangen am 12. Mai 2021, angenommen am 20. Juli 2021, online publiziert am 11. August 2021 © The Author(s) 2021



1. Einleitung und Motivation

Ein Cyber-Physisches System (CPS) ist ein System in dem physische Einheiten und Softwarekomponenten eng miteinander vernetzt und abhängig von ihrem Betriebskontext und ihrer Umgebung interagieren. CPS sind hochkomplexe, verteilte und Software-intensive Systeme, d.h. Software trägt einen wesentlichen Teil zum Entwurf, der Konstruktion, der Inbetriebnahme und der Wartung und Weiterentwicklung (Evolution) solcher Systeme bei. Software-intensive Cyber-Physische Produktionssysteme (SiCPPS) sind hochvariable Systeme von Systemen [36], die sich ständig weiterentwickeln [20].

Variabilität meint "die Fähigkeit eines Softwaresystems oder -artefakts effizient erweitert, geändert, angepasst oder konfiguriert zu werden" [8]. In SiCPPS unterstützt Software den Aufbau hochkonfigurierbarer Produktionssysteme, die in der Lage sind, Produkte fle-

xibel an die Anforderungen des Kunden angepasst zu produzieren. Der Entwicklungsprozess von SiCPPS umfasst mehrere technische Disziplinen (z.B. Mechanik, Elektrik, Mechatronik, Software), die während der Systementwicklung zusammenarbeiten müssen, um mit der Vielfalt an Produktvarianten umgehen zu können [14]. Dies erfordert insbesondere eine klare Definition der Abhängigkeiten zwischen den Artefakten in diesen Disziplinen.

Wie in Abb. 1.a dargestellt, werden die meisten der heterogenen Artefakte (z.B. Software, Elektroschaltpläne, Hydraulikpläne, Stücklisten, Spezifikationen) von verschiedenen Personen aus den unterschiedlichen Disziplinen mit unterschiedlichen Werkzeugen manuell erstellt und gepflegt. Abhängigkeiten zwischen den Artefakten re-

Gutierrez, Antonio, CDL VaSiCS, LIT CPS Lab, Johannes Kepler Universität Linz, Linz, Österreich; Sonnleithner, Lisa, CDL VaSiCS, LIT CPS Lab, Johannes Kepler Universität Linz, Linz, Österreich; Zoitl, Alois, CDL VaSiCS, LIT CPS Lab, Johannes Kepler Universität Linz, Altenberger Straße 69, 4040 Linz, Österreich (E-Mail: alois.zoitl@jku.at); Rabiser, Rick, CDL VaSiCS, LIT CPS Lab, Johannes Kepler Universität Linz, Linz, Österreich

¹US National Science Foundation Cyber-Physical Systems Program: https://www.nsf.gov/pubs/2021/nsf21551/nsf21551.htm.

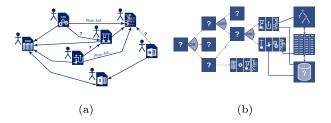


Abb. 1. In SiCPPS werden heterogene Artefakte aus verschiedenen Disziplinen mit unterschiedlichen Abhängigkeiten von verschiedenen Personen meist manuell, mit verschiedenen Werkzeugen gepflegt (a); Variabilität wird in SiCPPS typischerweise durch verschiedene, oft eigens entwickelte Mechanismen und Werkzeuge verwaltet (b)

sultieren aus physikalischen Anforderungen oder aus Geschäftsentscheidungen. Diese Abhängigkeiten sind jedoch häufig nicht dokumentiert, insbesondere nicht zwischen Artefakten verschiedener Disziplinen. Damit hängt der Erfolg der Entwicklung stark vom Domänenwissen der Mitarbeiter ab. Darüber hinaus folgen Werkzeugen aus verschiedenen Disziplinen unterschiedlichen Ontologien, was ihre Interoperabilität behindert.

Die Industrie hat daher ein Interesse daran besser mit der Variabilität der Artefakte umzugehen und vor allem systematische Wiederverwendung zu unterstützen, da damit Entwicklungs- und Wartungskosten gesenkt und die Markteinführungsdauer reduziert werden kann [10]. In einem kürzlich erschienenen Artikel haben Berger et al. [3] die Ergebnisse einer Studie mit zwölf mittelgroßen bis großen Firmen in Bezug auf den Einsatz von Variabilitätsmanagementtechniken vorgestellt. Sie kommen zu dem Schluss, dass Hardware und der Druck des Marktes zur Individualisierung die wichtigsten Treiber für Variabilität in der Industrie sind. Außerdem folgern sie, dass Industrie 4.0 und Digitalisierung das Variabilitätsmanagement erschweren bzw. weitere Forschung motivieren.

Systematisches Variabilitätsmanagement wurde in der bisherigen CPS-Forschung bisher weitgehend vernachlässigt [60]. Im Bereich des Software Engineering, insbesondere im Bereich der Softwareproduktlinien [10], konzentrierte sich die frühe Forschung zwar auf software-intensive Systeme [39, 46], jedoch fokussiert sich die aktuelle Forschung mehr und mehr auf die Variabilität von Software, Services und Betriebssystemen. Leider werden Evaluierungen von Forschungsansätzen oft mit kleinen, konstruierten Beispielen und im Labor statt unter realen Bedingungen durchgeführt [46].

Wie in Abb. 1.b angedeutet, muss in SiCPPS die Variabilität verschiedener heterogener Artefakte verwaltet werden. Ein "one-sizefits-all"-Ansatz zur Variabilitätsmodellierung ist daher von vornherein zum Scheitern verurteilt. Stattdessen erfordern verschiedene Disziplinen unterschiedliche Ansätze zur Verwaltung der Variabilität und einen Mechanismus, der hilft die verschiedenen Variabilitätsebenen miteinander zu verbinden [19].

In unserer Forschung konzentrieren wir uns auf die Beherrschung der Variabilität in SiCPPS, d.h. diese zu verstehen, zu modellieren bzw. zu dokumentieren und vor allem zur automatisierten Konfiguration von SiCPPS zu verwenden. Im Rahmen dieses Artikels fassen wir den Stand der Technik und Praxis zusammen, diskutieren die besonderen Herausforderungen im Umgang mit Variabilität in SiCPPS und beschreiben unsere Forschungsziele und Forschungsagenda.

2. Stand der Technik

Unsere Arbeit stützt sich zum einen auf bestehende Forschung im Bereich CP(P)S und zum anderen auf Forschung zu Variabilität und Softwareproduktlinien (SPL) sowie Vorarbeiten zu Variabilität und

Harrison et al. [21] geben beispielsweise eine ausführliche Zusammenfassung bestehender Entwicklungsstandards, -methoden und -werkzeuge für CP(P)S. Die Autoren betonen: "Es wurde viel über die potenziellen Vorteile der Einführung von CPSen in der Fertigungsindustrie veröffentlicht. Es wurde jedoch weniger darüber geschrieben, wie solche Automatisierungssysteme effektiv konfiguriert und über ihren Lebenszyklus hinweg unterstützt werden können und wie die Anwendungsmodellierung, Visualisierung und Wiederverwendung solcher Systeme am besten erreicht werden kann." Biffl et al. [4] fassen bestehende Arbeiten zum multidisziplinären Engineering für SiCPPS zusammen. Die Autoren stellen außerdem einen modellgetriebenen, auf AutomationML basierenden SiCPPS Entwicklungsansatz inklusive Versionierungs- und Verknüpfungsunterstützung vor. In der Praxis verwenden Domänenexperten jedoch nach wie vor verschiedene selbst entwickelte und zugekaufte Entwicklungswerkzeuge. Dies führt zu einer heterogenen Werkzeugslandschaft mit einer großen Anzahl an Entwicklungswerkzeuge, die auf unterschiedlichen Datenstrukturen arbeiten [16]. Modellentwicklung, -pflege und -integration sowie Datenaustausch und -freigabe stellen in der Praxis immer noch große Herausforderungen dar [26]. Im Zuge von Industrie 4.0 und daraus dem RAMI4.0 resultierenden Verwaltungsschale steigt die Verfügbarkeit von maschinenlesbaren integrierten Datenmodellen [25].

Bestehende Arbeiten beschäftigen sich zwar mit dem Entwicklungsprozess von SiCPPS, einschließlich der Versionierung und Verknüpfung von Modellen und Artefakten, jedoch wurde ein systematisches Variabilitätsmanagement jedoch bisher meist vernachläs-

Seit Anfang der 1990er Jahre wird zu SPL und Variabilität geforscht [39, 46]. Eine SPL wurde definiert als "eine Familie von (software-intensiven) Systemen, die aus einer gemeinsam verwalteten Menge von Artefakten durch systematische Wiederverwendung entwickelt werden und durch Erweiterungen und Konfiguration (Variabilität) an die Bedürfnisse spezifischer Kunden oder Marktsegmente angepasst werden." [10]. Es gibt erste Forschungsarbeiten zum Thema Variabilität und CPS. Zum Beispiel stellen ter Beek et al. [2] ein Featuremodell (Modell der Variabilität) einer Familie von Zugsteuerungssystemen vor. Brings et al. [7] berichten über eine explorative Fallstudie zur Variabilitätsmodellierung in der Industrieautomatisierung. Krüger et al. [27] schlagen in ihrem Positionspapier eine Klassifizierung der für CPS relevanten Variabilitätsaspekte vor. Sie weisen auf aktuelle Herausforderungen bei der Variabilitätsmodellierung hin und skizzieren offene Forschungsfragen und -ziele. Vor allem weisen sie darauf hin, dass die Variabilität eines CPS typischerweise heterogene Aspekte umfasst, die in ein gemeinsames Modell integriert werden müssen. Fang [15] stellt einen halbautomatischen, modellbasierten Ansatz vor, der darauf abzielt, die Wiederverwendbarkeit bestimmter Komponenten während des Entwicklungsprozesses zu verbessern (Ebene 4 Software der Automatisierungspyramide). Bezüglich der Variabilität in CPS untersuchten Safdar et al. [48] die Verwendung von existierenden Variabilitätsmodellierungstechniken und argumentieren, dass es notwendig ist, bestehende Ansätze zu erweitern oder neue Ansätze vorzuschlagen.

Dies wollen wir erreichen, indem wir unsere eigenen früheren Arbeiten zu SPL und Variabilität [12, 43, 44, 56] mit unseren Arbeiten im Bereich Steuerungstechnik und Produktionsautomatisierung [58, 61-63] zusammenführen. Seit kurzem arbeitet einer der Autoren auch mit Kollegen an der TU Wien an der Modellierung der Variabilität von Produkten, Prozessen und Ressourcen in SiCPPS [17, 31] was ein wichtiger zusätzlicher Input für die in diesem Beitrag vorgeschlagene Arbeit ist.

322

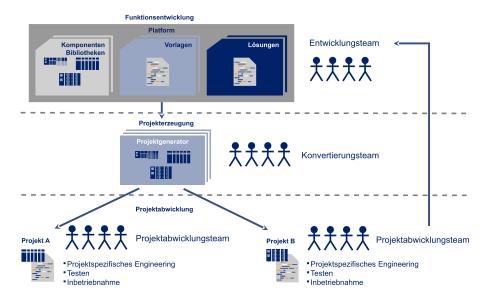


Abb. 2. Übersicht über einen typischen Entwicklungsprozess, der von vielen Maschinen- und Anlagenbauern angewandt wird

3. Stand der Praxis

Aufgrund der unterschiedlichen Arten von automatisierten Prozessen und der für die industrielle Automatisierung verwendeten Steuerungssoftware wurden der Systementwicklungsprozess und das Lebenszyklusmodell bisher nicht vereinheitlicht [64]. Im Gegensatz zu typischen Software-Engineering-Lebenszyklusmodellen sind in SiCPPS verschiedene Disziplinen (Mechanik, Elektrik, Software, etc.) an verschiedenen Schritten der Maschinen- und Anlagenentwicklung beteiligt. Entscheidungen in einer Disziplin können die anderen in großem Umfang sowohl kurzfristig als auch langfristig beeinflussen. Beispiele dafür sind Anpassungen einer Maschine an die örtlichen Gegebenheiten des Kunden, der Wechsel von pneumatischen Achsen auf servo-elektrische oder der Wechsel eines Komponentenlieferanten.

Abbildung 2 zeigt einen typischen, die hohe Variabilität unterstützenden Entwicklungsprozess von Steuerungssoftware in SiCPPS. Er besteht aus drei Hauptphasen: Funktionsentwicklung, Projekterzeugung und Projektabwicklung. Im Rahmen der Funktionsentwicklung wird die Kernfunktionalität in Form einer Plattform entwickelt und gepflegt. Eine solche Plattform besteht aus Komponentenbibliotheken, Vorlagen und Lösungen für Maschinen oder Anlagen. Für ein bestimmtes Projekt werden die entsprechenden Elemente der Plattform verwendet (unter Verwendung eines Clone-and-Own-Reuse-Ansatzes) und auf das im konkreten Projekt verwendete Steuerungssystem umgesetzt (erweitert). Da sich Steuerungssysteme stark unterscheiden, muss dieser Projekterzeugungsprozess die Besonderheiten des Zielsystems berücksichtigen und kann daher langwierig und fehleranfällig sein. Außerdem müssen die Elemente der Plattform in einer für das eingesetzte Steuerungssystem spezifischen Version gepflegt werden. Dies sind die Gründe, warum nur eine Teilmenge der Steuerungssysteme für Projekte verwendet werden kann. In der Projektdurchführungsphase wird die erstellte Steuerungssoftware manuell an die spezifische Maschinen- oder Anlagenkonfiguration angepasst, getestet und schließlich in Betrieb genommen. Oft werden in der Projektdurchführungsphase auch zusätzliche projektspezifische Implementierungen durchgeführt.

Dieses Vorgehen reduziert den Entwicklungsaufwand zwar bereits erheblich, dennoch gibt es immer noch offene Herausforderungen. Die hohe Variabilität von Maschinen- oder Anlagenkonfiqurationen wird noch überwiegend manuell gehandhabt. Darüber

hinaus müssen Änderungen an Maschinen- oder Anlagenkonfigurationen, die während der Projektabwicklung auftreten, manuell nachverfolgt und bearbeitet werden. Dadurch steigt der Aufwand in der Projektabwicklung. Derzeit gibt es keine Verbindung zwischen dem Code der Plattform und dem projektspezifischen Code. Daher ist es sehr schwierig, Korrekturen und Verbesserungen, die während der Projektabwicklung identifiziert werden, in die Plattform rückzuführen. Dies muss derzeit manuell erfolgen und es gibt keinen systematischen Roundtrip-Prozess.

Das Ziel unserer Forschung ist es daher, die Maschinen- und Anlagenvariabilität in der Plattform besser zu handhaben, den Projekterzeugungsprozess zu verbessern, um den projektspezifischen Entwicklungsaufwand zu reduzieren und ein systematischeres Round-Trip-Engineering zu unterstützen. Davon motiviert forschen wir an methodischer Unterstützung für die Beherrschung von Variabilität mit Unterstützung von semantischer Datenintegration in SiCPPS und wenden Ansätze und Werkzeuge auf Anwendungsfälle aus der Industrie an.

4. Forschungsziele

Abbildung 3 zeigt einen Überblick über unsere Forschungsziele im Rahmen eines SiCPPS Entwicklungsprozesses unter der Annahme der Verfügbarkeit einer gemeinsamen Plattform (vgl. Abschn. 3), von der Produkte abgeleitet werden.

(RG1) Modellierung der Variabilität in SiCPPS. Die Anforderungen an SiCPPS sind oft nicht vollständig dokumentiert, sondern existieren teilweise nur in den Köpfen der Ingenieure und sind somit implizites Wissen. Darüber hinaus sind die Anforderungsinformationen auf mehrere Artefakte verteilt, z.B. auf detaillierte technische Spezifikationen, Designmodelle, diverse Pläne, Softwareartefakte und sogar Benutzerhandbücher oder technische Benutzerdokumentation. Aufgrund der Heterogenität dieser Artefakte und weil verschiedene Rollen (oft mit ganz unterschiedlichem Hintergrund und unter Verwendung verschiedener Techniken und Werkzeuge) aus verschiedenen Disziplinen sie erstellen und pflegen, ist es fast unmöglich, einen vollständigen Überblick über die Anforderungen von SiCPPS zu erhalten. Speziell für die Variabilität führt dies "oft zu einer Situation, in der niemand im Unternehmen einen umfassenden Überblick über die vorhandene Variabilität hat" [6]. Ein Team aus sehr erfahrenen

323

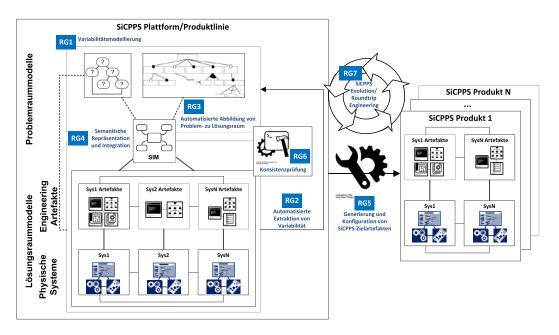


Abb. 3. Forschungsziele zur Beherrschung der Variabilität in SiCPPS

Mitarbeitern kann SiCPPS dennoch erfolgreich planen, entwerfen, entwickeln, in Betrieb nehmen und warten. Der fehlende oder zumindest unvollständige Überblick über die Anforderungen und insbesondere die System- und Softwarevariabilität macht die Lernkurve für neue Mitarbeiter jedoch sehr steil und bremst auch erfahrenes Personal häufig aus. Dies spricht für den Einsatz von Variabilitätsmodellen, um implizites Expertenwissen explizit darzustellen. Solche Modelle sind auch eine Voraussetzung, um Wiederverwendung und Konfiguration besser zu unterstützen. Variabilitätsmodelle [11], z.B. Feature-Modelle oder Entscheidungsmodelle, könnten zu diesem Zweck verwendet werden. Feature-Modelle erfassen Features – das Verständnis der Endbenutzer über die allgemeinen Fähigkeiten von Systemen in der Domäne – und die Beziehungen zwischen ihnen. Entscheidungsmodelle definieren die Entscheidungen, die getroffen werden müssen, um ein Produkt einer SPL zu spezifizieren. Die Besonderheiten von SiCPPS werden in bestehenden Ansätzen noch nicht berücksichtigt. Dies betrifft insbesondere den Umgang mit verschiedenen Engineering-Disziplinen und damit verbundenen Artefakten und Prozessen. Multi-View-Modellierungsansätze [14, 15] und orthogonale Variabilitätsmodellierungsansätze [12, 38] scheinen uns hier ein vielversprechender Ansatzpunkt zu sein.

(RG2) Automatisierte Extraktion von Variabilität. Jeder Modellierungsansatz, der die manuelle Erstellung von Modellen und insbesondere die manuelle Pflege und Weiterentwicklung der Modelle voraussetzt, wird in der Praxis wahrscheinlich scheitern. Selbst wenn der zusätzliche Aufwand für das Re-Engineering und die Modellierung von der Industrie akzeptiert würde, würden die Mitarbeiter in einer zeitlich begrenzten, projektgetriebenen Umgebung weiterhin mit den Artefakten und Werkzeugen arbeiten, an die sie gewöhnt sind, und nicht jedes Mal, wenn sie etwas in ihren Dokumenten, Plänen, dem Quellcode oder ähnlichem ändern, manuell zusätzliche Modelle einpflegen. Bestehende Arbeiten gehen jedoch typischerweise davon aus, dass jemand manuell Variabilitätsmodelle erstellt und pflegt. Damit ein Ansatz zur Variabilitätsmodellierung für die Praxis nützlich ist, muss die Modellerstellung und -wartung daher zumindest teilweise automatisiert werden, d.h. die Extraktion von Variabilitätsinformationen aus bestehenden Artefakten, wie

zum Beispiel aus oft als Spreadsheets vorhandenen Komponentenund Templatelisten, muss unterstützt werden. Siehe den Abschnitt über verwandte Arbeiten in [31]. Das automatische Extrahieren von Variabilität in SiCPPS erfordert das Parsen der heterogenen Artefakte und die richtige Interpretation der Informationen auf der Grundlage vordefinierter Regeln oder Heuristiken. Domänenexperten sollten nur dann hinzugezogen werden, wenn etwas nicht vollständig automatisiert werden kann. Von diesen Experten angebotene Lösungen müssen später zu programmierten Regeln werden, die die Aufgabe in Zukunft weiter automatisieren. Forschung aus dem Bereich der künstlichen Intelligenz (insbesondere Information Retrieval und maschinelles Lernen) könnte hier hilfreich sein [55].

(RG3) Automatisierte Abbildung von Problem- zu Lösungsraum. Variabilitätsmodelle bieten typischerweise eine Problemraumsicht (Anforderungen, Features) auf bestehende Systeme und wiederverwendbare Artefakte. Bestehende Arbeiten zur Modellierung des Lösungsraums, z.B. unter Verwendung der Unified Modeling Language,² der IEC 61499 [61] und insbesondere der Systems Modeling Language³ und der AutomationML⁴ berücksichtigen Variabilität nur teilweise. Insbesondere für Ingenieure, die für die Wartung verantwortlich sind, ist es oft schwer zu verstehen, welche Auswirkungen eine Änderung, die sie an einem Artefakt vornehmen, auf bestehende Modelle hat und umgekehrt. Gleichzeitig müssen Ingenieure schnell reagieren, um Produktionssysteme am Laufen zu halten oder sie nach Ausfällen wieder zum Laufen zu bringen. Bestehende Ansätze müssen an den SiCPPS-Kontext angepasst werden. Insbesondere müssen sie so flexibel gestaltet werden, dass neue (Arten von) Artefakten und Modellen integriert werden können. Bestehende Standards wie AutomationML bieten bereits eine Möglichkeit zur Verknüpfung von Lösungsraum-Artefakten. Wenn ein solches explizites Modell der Entwicklungsartefakte und ihrer

²http://www.uml.org

³http://svsml.org.

⁴https://www.automationml.org.

ORIGINAL ARBEIT

Beziehungen bereits vorhanden ist (vgl. RG4), sollten Verknüpfungen vom Problem (Features) zum Lösungsraum (Artefakte) auf der Basis dieses Modells erstellt werden können. Techniken zur Feature-Lokalisierung [47], die auf statischer Analyse, dynamischer Analyse, Information Retrieval oder hybriden Techniken basieren, könnten als Ausgangspunkt verwendet werden. Zusätzlich könnten Traceabilitybasierte Ansätze [9, 23] helfen.

(RG4) Semantische Repräsentation und Integration. Die Entwicklung von SiCPPS ist durch eine Umgebung mit sehr heterogenen Disziplinen and Standards gekennzeichnet [4, 57]. Dennoch sind die Entwicklungsdaten aus den verschiedenen Disziplinen stark voneinander abhängig [33]. Die Daten werden jedoch meist nur innerhalb der einzelnen Disziplinen gepflegt. Aufgrund fehlender formaler Definitionen und Richtlinien bzw. Regeln für Ingenieure wird derzeit entweder eine manuelle Synchronisation vorgenommen, die zeitaufwändig und fehleranfällig ist, oder es werden Arbeitsabläufe sequentiell organisiert. Letzteres verlangsamt den Fortschritt eines Projekts erheblich [34]. Die bisherigen Erfahrungen im Bereich des modellbasierten Systems-Engineering [59] und der Werkzeugintegration von Entwicklungswerkzeugen [58] haben gezeigt, dass eine wesentliche Anforderung darin besteht, die vorhandenen Entwicklungsdaten aus den verschiedenen Entwicklungsdisziplinen (wieder) zu verwenden. Es wurde versucht, Ansätze aus dem Software Engineering als dedizierte Sprachen zur Modellierung dieser Informationen einzuführen, z.B. SysML und AADL [18]. Allerdings ist deren Akzeptanz bei Ingenieuren für viele SiCPPS gering, da wieder zusätzliche Modelle entwickelt und gepflegt werden müssen. Wir schlagen ein Semantic Integration Model (SIM) basierend auf einer Ontologie vor. Ontologien unterstützen die explizite Spezifizierung des Wissens in einer Disziplin, erhöhen den Grad der Spezifikation von Wissen durch die Einbindung von Semantik in die Daten und fördern deren Austausch in einer explizit verständlichen Form. Ausgangspunkt für das SIM sind aktuelle Entwicklungsmodelle der verschiedenen Disziplinen sowie bestehende Modellierungstechniken für das Systems Engineering, wie SysML oder AADL, und entsprechende Datenintegrationsstandards wie AutomationML. Das SIM soll genutzt werden um z.B. Konsistenzprüfungen (vgl. RG6) und die Generierung von Zielartefakten zu unterstützen (vgl. RG5).

(RG5) Generierung und Konfiguration von SiCPPS-Zielartefakten. Die Generierung von Zielartefakten auf Basis von Entwicklungsmodellen erfolgt aktuell typischerweise mit einzelnen Eingabemodellen und wird von Fall zu Fall entwickelt. Dies ist eine langwierige, fehleranfällige Aufgabe, die für jedes neue Zielartefakt erneut durchgeführt werden muss [52]. Die semantische Integration von Entwicklungsdaten, wie sie unser SIM bieten würde (vgl. RG4), wird den Prozess der Zielartefaktgenerierung vereinfachen. Als Zielartefakte sehen wir z.B. Steuerungsprogrammfragmente, automatisierte Testszenarien, Steuerungskonfigurationen oder anlagenspezifische multidisziplinäre Simulationsskripte vor. Wir planen zusätzlich die in Variabilitätsmodellen enthaltenen Informationen zu nutzen, um eine schnelle Ableitung mehrerer Varianten von generierten Artefakten zu unterstützen. Im SiCPPS-Kontext besteht eine besondere Herausforderung darin, dass das für die Produktableitung benötigte Wissen oft in den Köpfen von Personen aus verschiedenen Disziplinen verteilt ist, die während des Ableitungsprozesses möglicherweise nicht verfügbar sind. Darüber hinaus können die Variabilitätsmodelle in der realen Welt sehr komplex werden. Die Integration von wiederverwendeten (sowohl selbst entwickelten als auch "off-the-shelf") Artefakten mit neu erstellten Artefakten [41, 42] ist daher eine zentrale Herausforderung, die es hier zu bewältigen gilt.

(RG6) Konsistenzüberprüfung für SiCPPS. SiCPPS müssen zu jeder Zeit Sicherheit und Korrektheit gewährleisten. Dies gilt auch

für alle Modelle, die SiCPPS-Artefakte und deren Variabilität darstellen [11]. Selbst bei einem hohen Automatisierungsgrad (vgl. RG2) können leicht Inkonsistenzen in Variabilitätsmodelle eingeführt werden. Bestehende Arbeiten zur Überprüfung der Konsistenz von Design- bzw. Variabilitätsmodellen und Software-Engineering-Artefakten [5, 13, 56] könnten ein vielversprechender Ansatzpunkt sein. Die Definition von Konsistenzeinschränkungen kann im Kontext von SiCPPS besonders herausfordernd sein, vor allem, wenn Stakeholder, die nicht aus dem Software-Engineering kommen, Einschränkungen, zum Beispiel als Logische Ausdrücke, definieren müssen. Dies motiviert eine domänenspezifische Sprache, vielleicht eine, die an verschiedene (Gruppen von) Benutzern angepasst werden kann. Außerdem müssen wir einen besonderen Schwerpunkt auf die Nutzbarkeit der Werkzeugunterstützung legen [45].

(RG7) SiCPPS Evolution/Round-trip Engineering. In der Praxis ist es kaum möglich, Kundenanforderungen allein durch die Wiederverwendung bestehender Artefakte zu befriedigen weil Kunden spezifische Anforderungen haben, vor allem im Kontext von SiCPPS. Neu entwickelte (projektspezifische) Artefakte haben jedoch das Potenzial, auch in zukünftigen Projekten nützlich (und wiederverwendet) zu werden. Es ist daher essentiell, den vollen Round-Trip zu unterstützen, d.h. zu entscheiden, welche neu entwickelten Artefakte dieses Potenzial haben und diese dann auch richtig in die Platform bzw. Produktlinie zu integrieren. Dies ist ein wichtiger Aspekt zur Unterstützung der Produktlinienevolution. In einem SiCPPS-Kontext muss eine Evolutionsunterstützung darüber hinaus die Heterogenität von (neu entwickelten) Artefakten und deren Abhängigkeiten berücksichtigen.

5. Forschungsagenda

Wir planen, die im vorigen Abschnitt diskutierten Forschungsziele mit dem in Abb. 4 dargestellten Arbeitsplan im Rahmen des Christian Doppler Labors Mastering Variability in Software-intensive Cyberphysical Production Systems (VaSiCS), in Zusammenarbeit mit einem Industriepartner anzugehen. Wir werden unsere Forschung auf die bestehenden SiCPPS Entwicklungsartefakte unseres Industriepartners stützen, die individuell von Personen aus verschiedenen beteiligten Disziplinen entwickelt und gewartet werden. Explizite Variabilitätsmodelle werden vom Industriepartner derzeit nicht verwendet. Variabilitätsinformationen sind vor allem in den Köpfen der beteiligten Ingenieure vorhanden, aber auch in verschiedenen Spreadsheets und Konfigurationsdateien dokumentiert, die uns zur Analyse und Nutzung zur Verfügung stehen. Zusätzlich haben verschiedene Abteilungen bereits diverse eigene Werkzeuge entwickelt, die sie bei ihrer täglichen Arbeit unterstützen. Wir betrachten diese Werkzeuge, und insbesondere die von ihnen produzierten Daten, als einen sehr relevanten Input, um die Variabilität zu modellieren. Diese Fallstudie ist somit eine sehr gute Grundlage für die Evaluierung unserer Forschung und für die Entwicklung erster Prototypen.

Ein wichtiger erster Schritt in unserer Agenda besteht darin, die Variabilität der von unserem Industriepartner entwickelten und gepflegten SiCPPS zu erheben und zu modellieren. Zu diesem Zweck planen wir Top-Down-Variabilitätsanalysen [40] und Bottom-Up-Variabilitätsanalysen [12, 24] von bestehenden Artefakten. Um sicherzustellen, dass sich unsere Forschung nicht nur auf die SiCPPS eines einzigen Partners konzentriert, planen wir, mindestens ein weiteres, von einer anderen Firma entwickeltes System zu analysieren.

Basierend auf diesen Untersuchungen werden wir den vorhandenen Quellcode der SiCPPS unseres Industriepartners analysieren und abgeschlossen Module und deren Abhängigkeiten identifizieren [54]. Wir werden Modellierungsansätze für SiCPPS Steuerungssoftware entwickeln, welche die Modularität und Entkopplung von

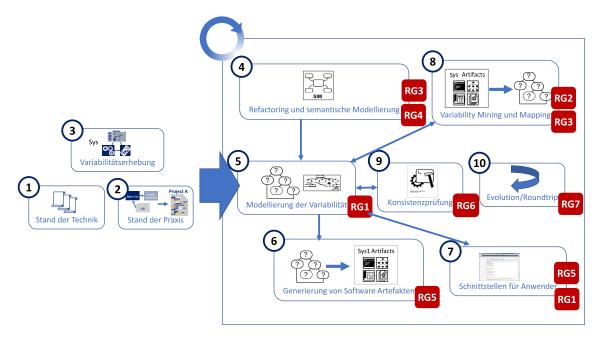


Abb. 4. Forschungsansatz

Komponenten unterstützen, die für die Beherrschung der Variabilität erforderlich sind [32, 51]. Dies bedeutet auch Refactoring Methoden zur Umwandlung der bestehenden Modelle in modulare Architekturen als Grundlage für die Variabilitätsmodellierung und -analyse zu entwickeln [50, 53]. Zusätzlich werden wir die wichtigsten Domänenmodelle für die Extraktion des Wissens aus den bestehenden technischen Modellen der verschiedenen Disziplinen überprüfen, um das SIM [22, 37] abzuleiten. Wir planen, einen iterativen Ansatz zu verfolgen, d.h. mit jedem integrierten Modell wird das SIM weiter ausgebaut.

Basierend auf den definierten Modellen werden wir an der Entwicklung von geeigneten Werkzeugen arbeiten, um (i) SiCPPS-Zielartefakte zu generieren und zu konfigurieren, (ii) vorhandene Artefakte automatisch zu analysieren, um unsere Modelle zu befüllen und (iii) die Konsistenz der Modelle zu überprüfen. Wir werden auf bestehenden Forschungsarbeiten zu Feature-Facets [28], Feature-Identifikation und -Extraktion [1, 30, 47], Constraint-Extraktion [35, 55], und Feature-Modell-Synthese [29, 49] zur (halbautomatischen) Extraktion von Variabilitätsinformationen aufbauen und diese anpassen. Wie wir festgestellt haben, ist es in der Industrie üblich, für jeden Kunden spezifische Artefakte zu entwickeln (und dies auch noch "vor Ort"/bei der Inbetriebnahme), daher ist es wichtig, den "Round-Trip" zu unterstützen und zu identifizieren, welche neu entwickelten Artefakte für neue Kunden wiederverwendet werden können und Teil der Platform bzw. Produktlinie werden sollen. Damit kann mit der Zeit der Wiederverwendungsgrad erhöht werden und der Aufwand für projektspezifische Entwicklungen reduziert werden.

Das Feedback unseres Industriepartners, durch wissenschaftliche Kollaborationen und von der Forschungsgemeinschaft wird uns helfen, unsere Pläne weiter zu verfeinern. Als ersten Meilenstein werden wir einen prototypischen SiCPPS-Modellierungsansatz mit Unterstützung für die Generierung und Konfiguration ausgewählter SiCPPS-Artefakten entwickeln, der nach zwei Jahren erreicht werden soll. Anschließend werden wir den Ansatz auf weitere Artefakttypen und andere SiCPPS unseres Industriepartners sowie auf

mindestens ein weiteres, nicht von unserem Industriepartner entwickeltes SiCPPS verallgemeinern. Parallel dazu werden wir an Schnittstellen/Werkzeugen für Anwender, Unterstützung für automatisches Variability Mining und Mapping, Konsistenzprüfung und Evolution/Roundtrip Engineering arbeiten (Aktionen 7-10). Als zweiten Meilenstein werden wir einen werkzeuggestützten Ansatz zur Beherrschung der Variabilität in SiCPPS, einschließlich Unterstützung für automatisches Variability Mining und Mapping, Konsistenzprüfung und Evolution/Roundtrip Engineering vorstellen.

6. Zusammenfassung

In diesem Beitrag haben wir beschrieben, warum der Umgang mit Variabilität in SiCPPS in der Praxis eine Herausforderung darstellt. Es mangelt an systematischen Ansätzen, Methoden und Lösungen. Wir haben die damit verbundenen Forschungsprobleme diskutiert, insbesondere (i) die prekäre Abhängigkeit von implizitem Expertenwissen; (ii) die unzureichende Automatisierungsunterstützung; (iii) die Schwierigkeiten bei der Bewertung der Auswirkungen von (geänderten) Anforderungen und Artefakten; (iv) den Umgang mit voneinander abhängigen Artefakten und Daten in verschiedenen technischen Disziplinen; und (v) den Umgang mit Inkonsistenzen. Basierend auf diesen Herausforderungen haben wir unsere Forschungsziele und Forschungsagenda skizziert. Konkret planen wir in unserem langfristigen Forschungsprojekt Christian Doppler Labor Va-SiCS die Entwicklung eines werkzeuggestützten Ansatzes zur Beherrschung von Variabilität in SiCPPS, einschließlich Unterstützung für automatisiertes Variability Mining und Mapping, Konsistenzprüfungen und Evolution/Roundtrip Engineering.

Danksagung

Wir bedanken uns für die finanzielle Unterstüzung durch das Bundesministerium für Digitalisierung und Wirtschaftsstandort und die Nationalstiftung für Forschung, Technologie und Entwicklung sowie die Christian Doppler Forschungsgesellschaft. Wir möchten uns auch explizit bei unserem Industriepartner für die Unterstützung bedanken

ORIGINAL ARREIT

Funding Note Open access funding provided by Johannes Kepler University Linz.

Hinweis des Verlags Der Verlag bleibt in Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutsadressen neutral.

Open Access Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quele ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden. Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen. Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf http://creativecommons.org/licenses/by/4.0/deed.de.

Literatur

- Bakar, N. H., Kasirun, Z. M., Salleh, N. (2015): Feature extraction approaches from natural language requirements for reuse in software product lines: a systematic literature review. J. Syst. Softw., 106, 132–149.
- ter Beek, M. H., Fantechi, A., Gnesi, S. (2018): Product line models of large cyberphysical systems: the case of ERTMS/ETCS. In Proc. of the 22nd int. systems and software product line conf., SPLC 2018, Gothenburg (Bd. 1, S. 208–214). New York: ACM
- Berger, T., Steghöfer, J., Ziadi, T., Robin, J., Martinez, J. (2020): The state of adoption and the challenges of systematic variability management in industry. Empir. Softw. Eng., 25(3), 1755–1797.
- Biffl, S., Lüder, A., Gerhard, D. (Hrsg.) (2017): Multi-disciplinary engineering for cyberphysical production systems, data models and software solutions for handling complex engineering projects. Berlin: Springer.
- Blanc, X., Mounier, I., Mougenot, A., Mens, T. (2008): Detecting model inconsistency through operation-based model construction. In 30th int. conf. on software engineering (ICSE 2008), Leipzig, Germany (S. 511–520). New York: ACM.
- Bosch, J., Capilla, R., Hilliard, R. (2015): Trends in systems and software variability. IEEE Softw., 32(3), 44–51.
- Brings, J., Daun, M., Bandyszak, T., Stricker, V., Weyer, T., Mirzaei, E., Neumann, M., Zernickel, J. S. (2019): Model-based documentation of dynamicity constraints for collaborative cyber-physical system architectures: findings from an industrial case study. J. Syst. Archit., 97, 153–167.
- 8. Capilla, R., Bosch, J., Kang, K. (2013): Systems and software variability management: concepts, tools and experiences. Berlin: Springer.
- 9. Cleland-Huang, J., Zisman, A., Gotel, O. (2012): Software and systems traceability.
- Clements, P., Northrop, L. M. (2002): Software product lines practices and patterns. SEI series in software engineering. Reading: Addison-Wesley.
- Czarnecki, K., Grünbacher, P., Rabiser, R., Schmid, K., Wasowski, A. (2012): Cool features and tough decisions: a comparison of variability modeling approaches. In 6th int. workshop on variability modelling of software-intensive systems, Leipzig, Germany (S. 173–182). New York: ACM.
- Dhungana, D., Grünbacher, P., Rabiser, R. (2011): The DOPLER meta-tool for decisionoriented variability modeling: a multiple case study. Autom. Softw. Eng., 18(1), 77–114
- Egyed, A. (2006): Instant consistency checking for the UML. In L. J. Osterweil, H. D. Rombach, M. L. Soffa (Hrsg.), 28th int. conf. On software engineering (ICSE 2006) Shanghai, China, May 20-28, 2006 (S. 381–390). New York: ACM.
- Fadhlillah, H. S., Feichtinger, K., Sonnleithner, L., Rabiser, R., Zoitl, A. (2021): Towards heterogeneous multi-dimensional variability modeling in cyber-physical production systems. In Proc. of the 25th ACM international systems and software product line conference (Bd. 2). Leicester: ACM.
- Fang, M. (2019): Model-based software derivation for industrial automation management systems. Ph.D. thesis, Kaiserslautern University of Technology, Germany.
- Fay, A., Biffl, S., Winkler, D., Drath, R., Barth, M. (2013): A method to evaluate the openness of automation tools for increased interoperability. In IECON 2013 - 39th annual conf. of the IEEE industrial electronics society, Vienna, Austria (S. 6844–6849). New York. IEEE Press.
- Feichtinger, K., Meixner, K., Rabiser, R., Biffl, S. (2020): Variability transformation from industrial engineering artifacts: an example in the cyber-physical production systems domain. In SPLC'20: 24th ACM int. systems and software product line conf., Montreal, Quebec, Canada, October 19-23, 2020 (Bd. B, S. 65–73). New York: ACM. 2020.

- Feiler, P. H., Gluch, D. P. (2012): Model-based engineering with AADL an introduction to the SAE architecture analysis and design language. Reading: Addison-Wesley.
- Galindo, J. A., Dhungana, D., Rabiser, R., Benavides, D., Botterweck, G., Grünbacher, P. (2015): Supporting distributed product configuration by integrating heterogeneous variability modeling approaches. Inf. Softw. Technol., 62, 78–100.
- Gerostathopoulos, I., Bures, T., Hnetynka, P., Keznikl, J., Kit, M., Plasil, F., Plouzeau, N. (2016): Self-adaptation in software-intensive cyber-physical systems: from system goals to architecture configurations. J. Syst. Softw., 122, 378–397.
- Harrison, R., Vera, D. A., Ahmad, B. (2016): Engineering methods and tools for cyberphysical automation systems. Proc. IEEE, 104(5), 973–985.
- Hästbacka, D., Zoitl, A. (2016): Towards semantic self-description of industrial devices and control system interfaces. In 2016 IEEE int. conf. on industrial technology (ICIT) (S. 879–884).
- Heider, W., Grünbacher, P., Rabiser, R., Lehofer, M. (2012): Evolution-Driven Trace Acquisition in Eclipse-Based Product Line Workspaces (S. 195–213).
- Jahn, M., Rabiser, R., Grünbacher, P., Löberbauer, M., Wolfinger, R., Mössenböck, H. (2012): Supporting model maintenance in component-based product lines. In 2012 joint working IEEE/IFIP conf. on software architecture and European conf. on software architecture (S. 21–30).
- Jasperneite, J., Sauter, T., Wollschlaeger, M. (2020): Why we need automation models: handling complexity in industry 4.0 and the Internet of things. IEEE Ind. Electron. Mag., 14(1), 29–40. https://doi.org/10.1109/MIE.2019.2947119.
- Kharlamov, E., Grau, B. C., Jiménez-Ruiz, E., Lamparter, S., Mehdi, G., Ringsquandl, M., Nenov, Y., Grimm, S., Roshchin, M., Horrocks, I. (2016): Capturing industrial information models with ontologies and constraints. In 15th int. semantic web conf. ISWC. Lecture notes in computer science (S. 325–343). Japan: Kobe.
- Krüger, J., Nielebock, S., Krieter, S., Diedrich, C., Leich, T., Saake, G., Zug, S., Ortmeier, F. (2017): Beyond software product lines: variability modeling in cyber-physical systems. In Proc. of the 21st int. systems and software product line conf., SPLC 2017, Sevilla, Spain, September 25–29, 2017 (Bd. A, S. 237–241). New York: ACM.
- Krüger, J., Mukelabai, M., Gu, W., Shen, H., Hebig, R., Berger, T. (2019): Where is my feature and what is it about? A case study on recovering feature facets. J. Syst. Softw., 152. 239–253.
- Linsbauer, L., Lopez-Herrejon, R., Egyed, A. (2014): Feature model synthesis with genetic programming. In Int. symp. on search based software engineering (S. 153–167).
- Martinez, J., Ziadi, T., Bissyandé, T. F., Klein, J., Traon, Y. L. (2017): Bottom-up technologies for reuse: automated extractive adoption of software product lines. In Proc. of the 39th int. conf. on software engineering companion, ICSE-C'17 (S. 67–70). New York: IEEE Press.
- Meixner, K., Rabiser, R., Biffl, S. (2020): Feature identification for engineering model variants in cyber-physical production systems engineering. In 14th int. working conf. on variability modelling of software-intensive systems, Magdeburg Germany (S. 18:1–18:5). New York: ACM.
- Melik-Merkumians, M., Wenger, M., Hametner, R., Zoitl, A. (2010): Increasing portability and reuseability of distributed control programs by i/o access abstraction. In 2010 IEEE 15th conf. on emerging technologies factory automation (ETFA 2010) (S. 1–4).
- Mordinyi, R., Winkler, D., Ekaputra, F. J., Wimmer, M., Biffl, S. (2016): Investigating
 model slicing capabilities on integrated plant models with automationml. In 21st int.
 conf. on emerging technologies and factory automation Berlin, Germany: (S. 1–8).
 New York: IEEE Press.
- Moser, T., Mordinyi, R., Winkler, D., Melik-Merkumians, M., Biffl, S. (2011): Efficient automation systems engineering process support based on semantic integration of engineering knowledge. In IEEE 16th conf. on emerging technologies & factory automation, ETFA 2011, Toulouse, France: September 5-9, 2011 (S. 1–8). New York: IEEE Press. 2011.
- Nadi, S., Berger, T., Kästner, C., Czarnecki, K. (2014): Mining configuration constraints: static analyses and empirical results. In Proc. of the 36th int. conf. on software engineering, ICSE 2014 (S. 140–151). New York: ACM.
- Nielsen, C. B., Larsen, P. G., Fitzgerald, J. S., Woodcock, J., Peleska, J. (2015): Systems of systems engineering: basic concepts, model-based techniques, and research directions. ACM Comput. Surv., 48(2), 18:1–18:41.
- Novák, P., Melik-Merkumians, M., Steinegger, M., Moser, T., Šindelář, R., Zoitl, A. (2013): Extraction of automation system engineering knowledge for mapping plant and simulation interfaces (S. 247–261). Berlin: Springer.
- 38. Pohl, K., Böckle, G., van der Linden, F. (2005): Software product line engineering: foundations, principles, and techniques. Berlin: Springer.
- Raatikainen, M., Tiihonen, J., Männistö, T. (2019): Software product lines and variability modeling: a tertiary study. J. Syst. Softw., 149, 485–510.
- Rabiser, R., Dhungana, D., Grünbacher, P., Burgstaller, B. (2008): Value-based elicitation of product line variability: an experience report. In 2nd int. workshop on variability modelling of software-intensive systems, Duisburg-Essen, Germany (S. 73–79).
- Rabiser, R., Grünbacher, P., Dhungana, D. (2010): Requirements for product derivation support: results from a systematic literature review and an expert survey. Inf. Softw. Technol., 52(3), 324–346.
- 42. Rabiser, R., O'Leary, P., Richardson, I. (2011): Key activities for product derivation in software product lines. J. Syst. Softw., 84(2), 285–300.

- Rabiser, R., Grünbacher, P., Lehofer, M. (2012): A qualitative study on user guidance capabilities in product configuration tools. In IEEE/ACM int. conf. on automated software engineering, ASE'12, Essen, Germany, September 3–7, 2012 (S. 110–119). New York: ACM.
- Rabiser, R., Vierhauser, M., Grünbacher, P., Dhungana, D., Schreiner, H., Lehofer, M. (2014): Supporting multiplicity and hierarchy in model-based configuration: experiences and lessons learned. In Model-driven engineering languages and systems - 17th int. conf., MODELS 2014, Valencia, Spain (S. 320–336). Berlin: Springer.
- Rabiser, R., Vierhauser, M., Grünbacher, P. (2016): Assessing the usefulness of a requirements monitoring tool: a study involving industrial software engineers. In 2016 IEEE/ACM 38th int. conf. on software engineering companion (ICSE-C) (S. 122–131).
- Rabiser, R., Schmid, K., Becker, M., Botterweck, G., Galster, M., Groher, I., Weyns, D. (2018): A study and comparison of industrial vs. academic software product line research published at SPLC. In Proceedings of the 22nd int. systems and software product line conf., SPLC 2018, Gothenburg, Sweden (Bd. 1, S. 14–24). New York: ACM
- 47. Rubin, J., Chechik, M. (2013): A survey of feature location techniques. In Domain engineering (S. 29–58). Berlin: Springer.
- Safdar, S. A., Yue, T., Ali, S., Lu, H. (2016): Evaluating variability modeling techniques for supporting cyber-physical system product line engineering. In System analysis and modeling. Technology-specific aspects of models - 9th int. conf., proc., SAM 2016, Saint-Melo, France, October 3-4, 2016 (S. 1–19). Berlin: Springer.
- She, S., Ryssel, U., Andersen, N., Wasowski, A., Czarnecki, K. (2014): Efficient synthesis of feature models. Inf. Softw. Technol., 56(9), 1122–1143. Special sections from "Asia-Pacific software engineering conf. (APSEC), 2012" and "Software product line conf. (SPLC), 2012".
- Steinegger, M., Zoitl, A. (2012): Automated code generation for programmable logic controllers based on knowledge acquisition from engineering artifacts: concept and case study. In 17th int. conf. on emerging technologies factory automation (S. 1–8).
- Steinegger, M., Zoitl, A., Fein, M., Schitter, G. (2013): Design patterns for separating fault handling from control code in discrete manufacturing systems. In IECON 2013 -39th annual conf. of the IEEE industrial electronics society (S. 4368–4373).
- Steinegger, M., Melik-Merkumians, M., Schitter, G. (2017): Ontology-based framework for the generation of interlock code with redundancy elimination. In 22nd IEEE int. conf. on emerging technologies and factory automation, ETFA 2017, Limassol, Cyprus, September 12-15, 2017 (S. 1–5). New York: IEEE Press.

- Strasser, T., Müller, I., Schüpany, M., Ebenhofer, G., Mungenast, R., Sünder, C., Zoitl, A., Hummer, O., Thomas, S., Steininger, H. (2006): An advanced engineering environment for distributed & reconfigurable industrial automation & control systems based on IEC 61499. In D. Pham, E. Eldukhri, A. Soroka (Hrsg.), Intelligent production machines and systems (S. 493–498). Oxford: Elsevier.
- Sünder, C., Zoitl, A., Dutzler, C. (2006): Functional structure-based modelling of automation systems. Int. J. Manuf. Res., 1, 405–420.
- Temple, P., Galindo, J. A., Acher, M., Jézéquel, J. M. (2016): Using machine learning to infer constraints for product lines. In Proc. of the 20th int. systems and software product line conf., SPLC'16 (S. 209–218). New York: ACM.
- Vierhauser, M., Grünbacher, P., Egyed, A., Rabiser, R., Heider, W. (2010): Flexible and scalable consistency checking on product line variability models. In 25th IEEE/ACM int. conf. on automated software engineering, Antwerp, Belgium (S. 63–72). New York:
- Vogel-Heuser, B., Diedrich, C., Fay, A., Jeschke, S., Kowalewski, S., Wollschlaeger, M., et al. (2014): Challenges for software engineering in automation. J. Softw. Eng. Appl., 7, 46096
- Waltersdorfer, F., Moser, T., Zoitl, A., Biffl, S. (2010): Version management and conflict detection across heterogeneous engineering data models. In 2010 8th IEEE int. conf. on industrial informatics (S. 928–935). New York: IEEE Press.
- Wenger, M., Melik-Merkumians, M., Hegny, I., Hametner, R., Zoitl, A. (2011): Utilizing IEC 61499 in an MDA control application development approach. In IEEE conf. on automation science and engineering, Trieste, Italy (S. 495–500). New York: IEEE Press.
- Yue, T., Ali, S., Selic, B. (2015): Cyber-physical system product line engineering: comprehensive domain analysis and experience report. In Proc. of the 19th int. conf. on software product line, SPLC 2015, Nashville, TN, USA (S. 338–347). New York: ACM.
- 61. Zoitl, A., Lewis, R. (2014): Modelling control systems using IEC 61499. 2. Aufl.
- Zoitl, A., Prähofer, H. (2013): Guidelines and patterns for building hierarchical automation solutions in the IEC 61499 modeling language. IEEE Trans. Ind. Inform., 9(4), 2387–2396.
- Zoitl, A., Vyatkin, V. (2009): IEC 61499 architecture for distributed automation: the glass half full' view. IEEE Ind. Electron. Mag., 3(4), 7–23.
- Zou, M., Vogel-Heuser, B., Sollfrank, M., Fischer, J. (2020): A cross-disciplinary modelbased systems engineering workflow of automated production systems leveraging socio-technical aspects. In 2020 IEEE int. conf. on industrial engineering and engineering management (IEEM) (S. 133–140).

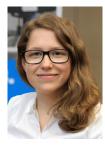
Autoren



Antonio Gutierrez

erhielt 2018 seinen Doktortitel in Software Engineering von der Universität Sevilla. Er ist wissenschaftlicher Mitarbeiter am Christian Doppler Labor Mastering Variability in Software-intensive Cyber-Physical Production Systems (VaSiCS) an der Johannes Kepler Universität Linz. Er arbeitete als Ingenieur für mehrere Unternehmen, bevor er sich der Wissenschaft anschloss. Seine Forschungsinter-

essen beziehen sich auf service-orientiertes Computing, Geschäftsprozessmanagement und Variabilitätsmodellierung.



Lisa Sonnleithner

ist wissenschaftliche Mitarbeiterin am LIT Cyber-Physical Systems (CPS) Lab als auch am Christian Doppler Labor Mastering Variability in Software-intensive Cyber-Physical Production Systems (VaSiCS) an der Johannes Kepler Universität Linz. Sie hat ihren Master an der Johannes Kepler Universität Linz in Mechatronik erhalten. Ihre Forschungsschwerpunkte sind Software Metriken und Software Architekturen für Steuerungssoftware.



Alois Zoitl

ist Universitätsprofessor für Cyber-Physical Systems for Engineering and Production am Linz Institute of Technology (LIT) und stellvertrender Leiter des LIT Cyber-Physical Systems (CPS) Lab als auch Leiter des Christian Doppler Labors Mastering Variability in Software- intensive Cyber-Physical Production Systems (VaSiCS) an der Johannes Kepler Universität Linz. Er promovierte 2007 im

Fachgebiet Automatisierungstechnik an der Technischen Universität Wien zum Thema dynamisch rekonfigurierbarer Steuerungsprogramme. Seine Forschungsthemen beinhalten adaptive Produktionssysteme, verteilte Steuerungsarchitekturen, dynamische Rekonfiguration von Steuerungsprogrammen sowie Softwareentwicklungsund Software- Qualitätssicherungsmethoden für die Industrielle Automatisierungstechnik. Er ist Gründungsmitglied und Projektleiter des Open-Source Projekts Eclipse 4diac.

ORIGINAL ARBEIT

329



Rick Rabiser

ist Universitätsprofessor für Software Engineering in Cyber- Physical Systems am Linz Institute of Technology (LIT) und Leiter des LIT Cyber-Physical Systems (CPS) Lab als auch Leiter des Christian Doppler Labors Mastering Variability in Software-intensive Cyber-Physical Production Systems (VaSiCS) an der Johannes Kepler Universität Linz. Er hat an der JKU Linz in Wirtschaftsinformatik zum

Thema Produktkonfiguration in Softwareproduktlinien promoviert und in Praktischer Informatik zum Thema Software Variabilität habilitiert. Gemeinsam mit seinem Team forscht er in Kooperation mit nationalen und internationalen akademischen und industriellen Partnern an der Beherrschung der Software-Variantenvielfalt in (Cyber-physikalischen Produktions-)Systemen. Seine weiteren Forschungsschwerpunkte sind: Softwareproduktlinien, Softwarewartung und -evolution und Usability von Software Engineering Werkzeugen.