

Modellbasierte Entwicklung cyber-physischer Systeme

Energienetze, integrierte Mobilitätslösungen oder moderne Produktionsanlagen sind Beispiele für cyber-physische Systeme. Sie sind umfangreich, heterogen, offen und dynamisch. Ihre Systemgrenzen sind unscharf, und sie sind mit ihrer physischen Umwelt intensiv verwoben, so dass sie mit dieser Smart Ecosystems bilden. Cyber-physische Systeme sind damit so komplex, dass sie kein Mensch mehr vollständig überblickt und steuernd eingreifen kann. Sie müssen daher im Wesentlichen autonom agieren. Ein Weg, um solche Systeme zu bauen, ist die Modellierung. Hierzu sind aber ganz andere Modelle nötig als die derzeit verwendeten. Diese dynamischen Modelle müssen durch die Systeme selbst zur Laufzeit ausgewertet werden können, um z.B. zu ermitteln, ob eine zu treffende Entscheidung zu einem akzeptablen Ergebnis führen wird. Die traditionelle Unterteilung einer Systementwicklung in die zuständigen beteiligten Disziplinen, die dann jeweils für sich ihren Teil des Systems entwickeln, ist zunehmend hinderlich, da für das Erreichen optimaler Lösungen für das Gesamtsystem eine ganzheitliche Sicht erforderlich ist. Informatiker, Maschinenbauer, Elektrotechniker und Fachleute weiterer Disziplinen müssen im Sinne eines ganzheitlichen Systems Engineering zusammenarbeiten.



Prof. Dr.-Ing. **Peter Liggesmeyer** (Jahrgang 1963) studierte Elektrotechnik (Schwerpunkt Datentechnik) an der Universität Paderborn und promovierte 1992 an der Fakultät für Elektrotechnik und Informationstechnik der

Ruhr-Universität Bochum. Anschließend war er bis 2000 in der Entwicklungs- und Forschungsabteilung der Siemens AG in München tätig. Nach seiner Habilitation war er von 2000 bis 2004 Inhaber des Lehrstuhls Softwaretechnik und Qualitätsmanagement an der Universität Potsdam (HPI). Seit 2004 ist er Inhaber des Lehrstuhls für Software Engineering: Dependability am Fachbereich Informatik der TU Kaiserslautern. Er ist seit 2015 Institutsleiter des Fraunhofer-Instituts für Experimentelles Software Engineering IESE in Kaiserslautern, dessen wissenschaftlicher Leiter er von 2004 bis 2014 war. Von 2014 bis 2017 war er Präsident der Gesellschaft für Informatik und von 2010 bis 2013 Sprecher der von ihm gegründeten Fraunhofer-Allianz Embedded Systems. Seit 2018 ist er Mitglied der Deutschen Akademie für Technikwissenschaften acatech.

Fraunhofer-Institut für Experimentelles Software Engineering IESE,
Fraunhofer-Platz 1, 67663 Kaiserslautern und Technische Universität
Kaiserslautern, Postfach 3049, 67653 Kaiserslautern

Professor Liggesmeyer ging in seinem Vortrag der Frage nach, wie man etwas so Komplexes wie ein cyber-physisches System so baut, dass es nicht nur funktioniert, sondern sich sogar autonom verhält und an geänderte Situationen anpassen kann.

„Das ist eine neue Herausforderung zum bisherigen Prinzip, etwas gut zu verstehen, es zu entwickeln, zu prüfen und zum Schluss zu zertifizieren, einzuschalten und es dann möglichst nicht mehr zu verändern. Das war der bisherige Status quo. Wir werden aber zunehmend neue Systeme bekommen, die anders sind und eine neue Herangehensweise bei der Entwicklung erfordern. KI, also die Fähigkeit, Systeme sich auf der Basis von Daten selbst verändern zu lassen, ist eine Möglichkeit, eine modellbasierte Entwicklung eine andere“, sagte Liggesmeyer. Wenn er sich im Folgenden auf Modellierung beschränke, so sei ihm klar, dass es wohl keinen Königsweg gibt, mit dem alles gehe.

Der bisher gebrauchte Begriff *Software Engineering*, den auch das Fraunhofer-Institut, an dem er arbeitet, im Namen führt, sei nach wie vor wichtig, aber im Konzert mit weiteren Disziplinen, denn Software steht immer in Verbindung mit anderen Artefakten. Sie ist Teil eines gesamten Systems, deshalb sei Systems Engineering treffender. Die Software ist nur ein Teil eines umfassenderen Systems, wie er auf einfache Weise verdeutlichte: Die Sicherheit von Software kann für sich allein nicht beurteilt werden. Es kommt darauf an, welche Funktion die Software erfüllt und wie kritisch ein Versagen dieser Funktion wäre. Deshalb müsse man Software stets als Teil der Umgebung und immer in ihrem Kontext betrachten. Die Trennung zwischen Informatik und Maschinenbau, Elektrotechnik und BWL ist aus seiner Sicht im Hinblick auf die zu entwickelnden Systeme nicht mehr zeitgemäß, denn der Robotiker braucht

Software Engineering vs. Systems Engineering

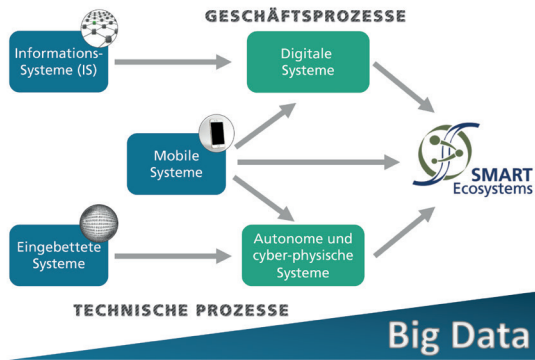


Abb. 1. Die Verschmelzung der einstmals an den Universitäten getrennt betrachteten Bereiche, die sich mit Informationsverarbeitung (Informatik), mit dem klassischen Engineering der Systemkomponenten (Maschinenbau, Elektrotechnik u. a.) sowie wirtschaftlichen Aspekten (BWL) beschäftigten, führt zu einem umfassenden *Systems Engineering*. Die Entwicklung von cyber-physischen autonomen Systemen, deren Leistungen von datenbasierten Dienstleistern jederzeit abgerufen werden können, ist nur möglich dank der massiven Nutzung von Daten (Big Data).

die Daten und die Datenbanker brauchen die Anwendungen. Und tatsächlich wachsen die ursprünglich getrennten Disziplinen immer mehr zusammen, weil sie jeweils aufeinander angewiesen sind, aber auch voneinander profitieren. Heutige Systeme sind oft so kompliziert, heterogen und autonom, dass sie ähnlich funktionieren wie natürliche Ökosysteme, allerdings mit einer Vielzahl technischer Komponenten. Man könnte von Smarten Ökosystemen (Smart Ecosystems) sprechen (Abb. 1).

„Smart Grid – das neue elektrische Energieerzeugungs- und Verteilsystem – mit Tausenden von Mitspielern im System (Energieerzeuger mit Solarzellen auf dem Dach, Energieverbraucher, Konsumenten, Lieferanten) ist ein solches Smart Ecosystem – mit vielen Abhängigkeiten und Einflussfaktoren, vielen Bedürfnissen, Engpässen oder Überangebot an Energie. Ein solch komplexes System entzieht sich weitgehend der manuellen Steuerung durch den Menschen; das muss autonom erfolgen. Dem steuernden Menschen fehlt es an Übersicht und vor allem an Reaktionsgeschwindigkeit. Das System muss selbst die Fähigkeit haben, die Zielgrößen in einem akzeptablen Bereich zu halten. Es muss autonom agieren können“, erklärte Professor Liggesmeyer. Das bedeutet eine große Herausforderung für die Entwickler solcher Systeme.

Herausforderungen

Die großen Herausforderungen hängen mit den Eigenschaften der cyber-physischen Systeme und den an sie zu stellenden Ansprüchen (vor allem in Punkto Sicherheit, Verlässlichkeit) zusammen (Abb. 2).

• **Diversität**

Die Systeme sind komplex: „Wir brauchen integrierte, systemübergreifende Entwicklungsmethoden – es nützt nichts, wenn der Maschinenbauer den Antrieb optimal entwickelt, der Elektrotechniker die Steuerung, der Informatiker die ‚Intelligenz‘ dafür – so bekommt man drei lokale Optima, aber es geht um das optimale

Zusammenspiel aller Komponenten, um das globale Optimum. Methoden dafür gibt es bislang nur ansatzweise!“

• **Selbstadaptation**

Das System ändert sich ständig und kann nicht nur einmalig zertifiziert werden, wie es etwa bei neu zugelassenen Schienenfahrzeugen Voraussetzung ist, bevor sie auf die Schiene dürfen. „Da kann nicht jedes Mal der TÜV bei jeder kleinen Adaption des Systems kommen; das muss das System selber durchführen. Es ändert sich durch Selbstadaptation also auch die Art und Weise der Zertifizierung. Neue Technologien erfordern neue Prozesse in der Zertifizierung. Zukünftige Systeme werden über die Fähigkeit verfügen müssen, sich im laufenden Betrieb selbst zu zertifizieren!“, betonte Liggesmeyer.

• **Autonomie**

Die Systeme müssen selbst wissen, wie es ihnen gerade geht und entscheiden, ob sie den Zustand ändern sollen, und sie müssen abschätzen, wie sich Änderungen auswirken. Sie müssen also die Fähigkeit zur Selbstdiagnose haben!

• **Qualitätsgarantien**

Datenbasierte Lösungen sind nur dann marktfähig, wenn sie Daten sowohl nutzen als auch schützen können. Die Nutzung der Daten ist für die Funktion unerlässlich. Der Schutz der Daten garantiert einen angemessenen Schutz der Privatsphäre. Dies führt notwendigerweise zu Konflikten (Big Data vs. Privacy). Das ist eine Herausforderung, an der gearbeitet wird und zu der auch bereits Lösungen existieren, z. B. die sogenannte Datennutzungskontrolle.

• **Umgebungseinflüsse**

Die neuen Systeme müssen weitgehend unempfindlich gegen Störeinflüsse von außen sein (Resilienz). Das war kein Problem, als Computer noch abgetrennt in verschlossenen Räumen



Abb. 2. Cyber-physische Systeme sind in eine hochvariable, nicht konstante Umwelt eingebettet (Smart Ecosystems), die sowohl Anpassungsfähigkeit als auch Stabilität verlangt.

Kein Nachdruck, keine Veröffentlichung im Intranet oder einem Internet ohne Genehmigung des Verlags und der GDNA

standen. Durch die weltweite Vernetzung in einer Welt voller Smartphones und Tablets ist dies jedoch eine zusätzliche neue Herausforderung.

• **Umgang mit Big Data**

Autonome, selbstadaptive Systeme müssen in vielen Fällen Daten in Echtzeit analysieren können. Lange Zeit war auch das kein Problem. „Denn die Rechner wurden immer leistungsfähiger. Doch mittlerweile haben wir einen Punkt erreicht, an dem die Datenmengen schneller ansteigen als die Rechenleistungen nach dem Mooreschen Gesetz! Lange Zeit hat man sich hinter der Effizienz der Hardware verstecken können – das geht jetzt nicht mehr. Das bringt ein altes Wissenschaftsgebiet aufs Tapet, das viele Informatiker tot geglaubt hatten: die effizienten Algorithmen.“

Die Herausforderungen sind also vielfältig – manche sind gelöst, bei anderen zeichnet sich eine Lösung ab.

Aktuelle Entwicklungen

„Informatiker programmieren nicht mehr nur im stillen Kämmerlein, sie beschäftigen sich immer mehr mit Modellen. In großen Projekten nimmt die reine Programmierung von Codes nur noch 10 Prozent ein“, führte Liggesmeyer weiter aus, um noch einmal den Wandel im Berufsbild des Informatikers vor Augen zu führen. Die Aufgaben sind vielfältig und anspruchsvoll, wie der Blick auf die aktuellen Entwicklungen zeigt. Drei Entwicklungen stehen im Vordergrund:

• **Modellbasierte Entwicklung**

„Speziell bei sicherheitskritischen Aspekten in Systemen, etwa bei der Entwicklung von Flugzeugen, nutzt man Modelle, um diese Systeme auf Herz und Nieren zu prüfen und das Restrisiko für einen Systemausfall quantitativ bestimmen zu können. Es geht also nicht nur darum, die Funktionalität eines Systems zu modellieren, sondern diese auch mit der Fehleranfälligkeit und der Fehleranalyse zu verknüpfen. Diese beiden Aspekte sind in sicherheitskritischen Systemen stets miteinander verbunden.“

• **Quantifizierbare statt qualitativer Qualitätsaussagen**

Statt allgemeinen Sicherheitsversprechen sind zunehmend belastbare Aussagen dazu erforderlich, dass Restrisiken unter einer konkret benannten Toleranzschwelle liegen.

• **Virtualisierung**

Getragen von dem Wunsch, dynamische Modelle zu entwickeln, gelangt man zur Simulation – d. h. zu einem Modell, das bestimmte Aspekte eines Systems oder seiner Komponenten wiedergibt. Ziel ist, neben dem realen System einen sogenannten „Digitalen Zwilling“ zu haben. Dann kann man reale Komponenten durch eine Modellkomponente austauschen, um zu prüfen, wie diese sich im realen System auswirkt oder auch umkehrt reale Systemteile im virtuellen System betrachten. Wenn die Ergebnisse beider Simulationen übereinstimmen, dann ist die Virtualisierung gelungen.

Liggesmeyer betonte, dass es keine ideale Modellierungsmethode gibt: „Die Anforderungen sind so vielfältig – allein schon in der Informatik –, dass dies nicht erreichbar ist. Sinnvoll ist es, wenn Modelle mehrere Modellierungszugänge enthalten, die z. B. aus Sicht unterschiedlicher Disziplinen erforderlich sind. Es gibt damit Modellierungsumgebungen, die aus sehr unterschiedlichen Komponenten bestehen.“

Industrie 4.0

Wie weit der Prozess modellbasierter Entwicklung bereits in der Arbeitswelt, speziell in der Optimierung von Produktionsprozessen, Einzug hält, zeigte Liggesmeyer stellvertretend am nationalen Referenzprojekt BaSys 4.0 – einer offenen Plattform für die vierte industrielle Revolution (Abb. 3).

„Industrie 4.0 ermöglicht individualisierte Produkte. Zur Zeit des Biedermeier waren Möbel oft ganz individuell auf den Kundenwunsch abgestimmt. Das Möbelstück wurde vom Tischler des Vertrauens kundenspezifisch entworfen und angefertigt. Das

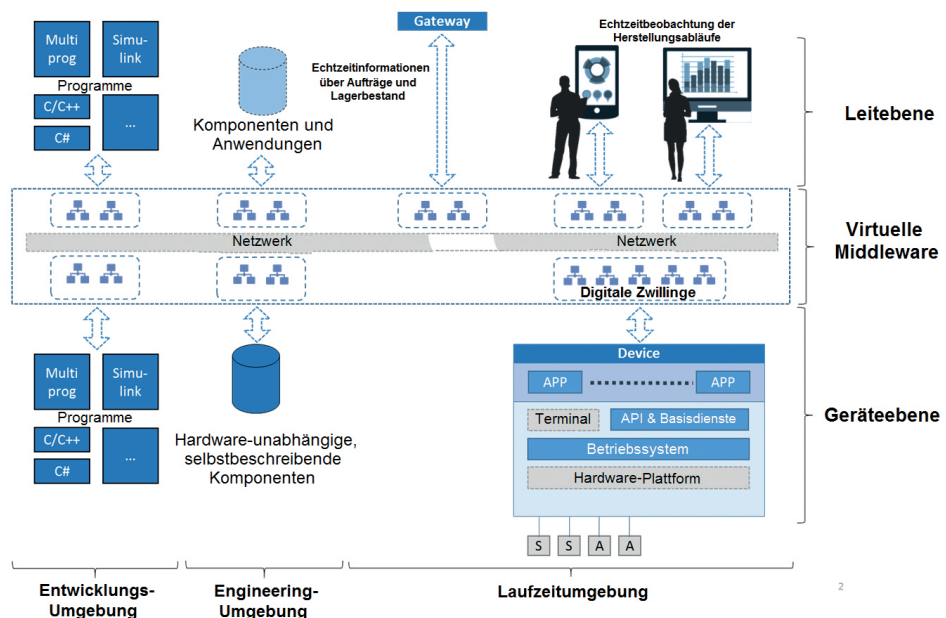


Abb. 3. Die Virtualisierung von Produktionsabläufen ist das Rückgrat der Industrie 4.0. Durch Simulationen lassen sich Entscheidungen bereits frühzeitig absichern. Im Bild ist das Deployment der BaSys-4.0-Middleware dargestellt. Die einzelnen Geräte der Fabrik sind an die Middleware angeschlossen und dadurch zu einem homogenen Kommunikationssystem integriert. Dabei dienen Digitale Zwillinge – dargestellt durch Verwaltungsschalen – als zentrale Beschreibung der für die Produktion relevanten Assets. Verschiedene Komponenten, wie zusätzliche Datenquellen oder Applikationen wie Dashboards werden analog integriert und angesprochen. Über ein Gateway, z. B. zu einem ERP-System (Enterprise Resource Planning – Geschäftsressourcenplanung), werden die Aufträge eingespeist und auf die unterschiedlichen Geräte verteilt.

Hauptziel von Industrie 4.0 ist, diese Individualisierbarkeit auf industriellem Niveau zu ermöglichen. Das betrifft sowohl die Stückzahlen als auch die Kosten. Die Produktionsumgebung muss daher sehr flexibel sein, um geänderte Anforderungen befriedigen zu können. Und diese Flexibilität muss weitgehend autonom realisiert werden.“ Geplant ist, die Lösung als Open Source anzubieten, so dass jede Firma darauf zurückgreifen kann, denn das Ziel ist, eine hohe Durchdringung in der Praxis zu erreichen (mehr unter <https://www.basys40.de>).

Vieles ist also im Werden, viele Ansätze existieren schon. Für Liggesmeyer steht fest: „Software Engineering muss zunehmend Systemanforderungen beachten. Die Entwicklungsprozesse werden agiler, Systeme autonomer, und neue Technologien müssen in

das Engineering integriert werden. Klar ist auch, dass die Eigenschaften zunehmend zur Laufzeit sichergestellt werden müssen. Das bedeutet, das Engineering von Qualität durch den Entwickler wird durch ein Engineering eines Systems ersetzt werden müssen, das die Fähigkeit besitzt, sich selbst um seine eigene Qualität zu kümmern. Das erfordert eine umfassende Nutzung von integrierten Modellen und Virtualisierungsansätzen.“ Und speziell an die Informatiker gewandt sagte er: „Wir müssen uns verabschieden von der lieb gewonnenen Vorstellung des JAVA-Programmierers mit der Pizzaschachtel im dunklen Keller. Wir brauchen Informatiker mit einer hohen kommunikativen Fähigkeit und hohen Abstraktionsfähigkeiten, die in der Lage sind, gut mit diesen Modellierungsansätzen zu arbeiten.“