

White Paper

Advanced Systems Engineering

Die Systeme der Zukunft

fortiss

Advanced Systems Engineering – Die Systeme der Zukunft

Autoren

Prof. Dr. Dr. h. c. Manfred Broy

*Technische Universität München,
Fakultät für Informatik,
80290 München*

broy@in.tum.de

Dr. Wolfgang Böhm

*Technische Universität München,
Fakultät für Informatik,
80290 München*

boehmw@in.tum.de

Prof. Dr. Bernhard Rumpe

*RWTH Aachen University,
Fakultät für Informatik, Software Engineering
52074 Aachen*

rumpe@se-rwth.de

Inhalt

Einleitung	4
Advanced Systems Engineering	5
MBSE als wesentliche Basis	6
Der durchgängige Ansatz von SPES/CrEST	9
Methodische Erweiterungen von SPES hin zu ASE	11
Zusammenfassung	13
Weiterführende Arbeiten	13
Impressum	14

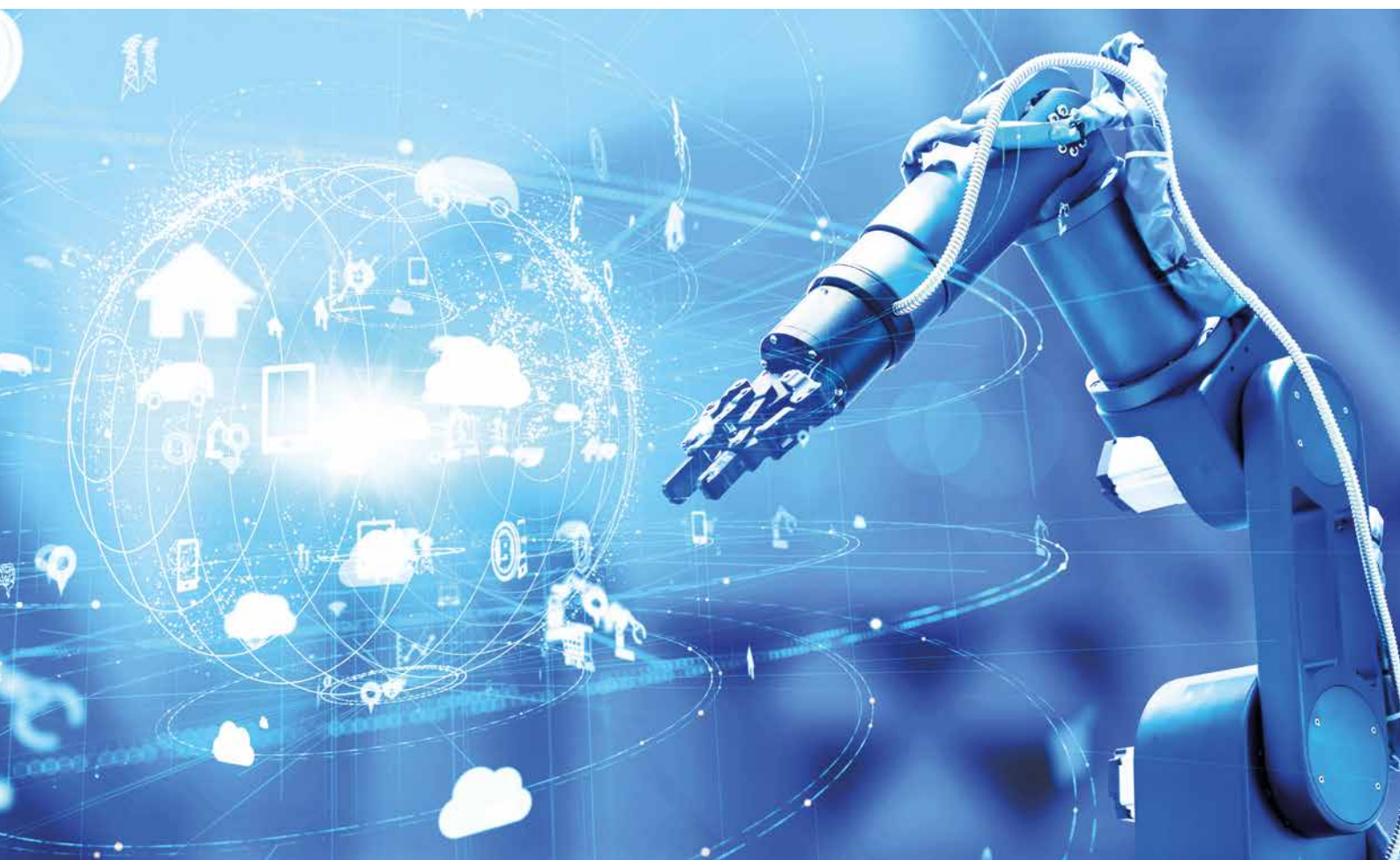
Einleitung

Die heutigen und auch die in Zukunft zu entwickelnden Systeme und technischen Produkte sind überwiegend cyber-physische Systeme mit Hochleistungsfunktionalität, die typischerweise nicht „stand alone“ sind, sondern über digitale Netze, wie das Internet, mit Nutzern und mit anderen Systemen verbunden sind und deren Dienste wechselseitig nutzen und ergänzen (s. Seite 13, [11]). Erkennbar ist, dass in den Systemen in gewissen Umfängen Teilsysteme verbaut werden, die durch heuristische Verfahren, etwa durch „Lernverfahren“, geschaffen werden.

Typisch für diese Systeme ist, dass durch die hohen Softwareanteile leistungsstarke Funktionalitäten für diese Systeme ermöglicht werden, die dramatisch über das hinausgehen, was in der Vergangenheit für eher isoliert arbeitende mechatronische Systeme möglich war. Der hohe Anteil der Software führt auf einen umfangreichen Entwurfsraum, in dem die unterschiedlichsten Anforderungen zu identifizieren sind. Somit kommt der Identifikation eines Anforderungskonzepts besondere große Bedeutung zu. Dies schafft auch umfassende Potenziale für Innovationen, sowohl bei der rein logischen Funktionalität, aber auch sehr stark in der menschenzentrierten Mensch-Maschine-Interaktion und der Automatisierung bis hin zur Autonomie.

Cyber-physische Systeme zeichnen sich dadurch aus, dass sie in der Regel über mechatronische Anteile verfügen, insbesondere aber über Sensoren und Aktuatoren, um die Interaktion zwischen den physischen Anteilen und den Softwareanteilen sowie eine Interaktion der Systeme mit ihrer Systemumgebung zu ermöglichen. Der hohe Anteil an Software schafft Funktionalitäten durch den Einsatz fortgeschrittener Softwaretechnik unter Einschluss von Verfahren der Künstlichen Intelligenz und ermöglicht menschenzentrierte Nutzerschnittstellen für diese Systeme.

In diesem Zusammenhang ist es besonders bemerkenswert, dass heutige Systeme aus guten Gründen einen übergroßen Softwareanteil enthalten, da dadurch Funktionalitäten möglich sind, die vor einigen Jahrzehnten noch völlig undenkbar waren. Durch die starke Vernetzung ist es naheliegend, Systeme mit ganz unterschiedlichen Aufgaben und Funktionalitäten zu verbinden: um Funktionalität aus anderen Systemen zu nutzen, aber auch Funktionalität für andere Systeme bereitzustellen und damit den Automatisierungs- und Optimierungsgrad zu erhöhen.



Advanced Systems Engineering

Die Systeme der Zukunft zeichnen sich entsprechend durch folgende Merkmale aus:

- Umfangreiche Softwareanteile und Funktionalität, die maßgeblich von den Softwareanteilen bestimmt wird.
- Hoher Vernetzungsgrad mit anderen Systemen zur wechselseitigen Nutzung von Datendiensten.
- Starke Integration von Software mit mechanischen und elektrotechnischen Bestandteilen.
- Umfassende, dedizierte Nutzerschnittstellen.
- Hohe Automatisierung bis hin zur Autonomie.
- Beständige Weiterentwicklung – auch während des Betriebs.
- Komplexe Integration mit Business Software.

Diese Merkmale schlagen sich auch auf die dafür benötigten Entwicklungsansätze nieder und bestimmen die Charakteristika eines Advanced Systems Engineering (ASE). ASE wird demnach durch folgende Charakteristika geprägt:

- Hohe Anteile von Modellierungstechniken zur Sicherstellung der Korrektheit komplexer Funktionalität und umfassender Werkzeugunterstützung, Frontloading.
- Starke Verzahnung der Entwicklungsprozesse der beteiligten Ingenieurdisziplinen (Informatik, Maschinenbau, Elektrotechnik) sowie der dort verwendeten Werkzeuge. Herkömmliche Prozesse wie beispielsweise sequenzielle, disziplinspezifische Entwicklung werden den neuen Anforderungen nicht mehr gerecht.
- Strikter systemzentrischer Ansatz zur holistischen Integration der erforderlichen multidisziplinären Designansätze.

- Durchgängigkeit der Entwicklung über die Familie von Modellen mit klaren semantischen Grundlagen, präzise definierten Beziehungen zwischen den Modellen und Entwicklungsschritten, die aus erarbeiteten Modellen systematisch weitere Modelle entwickeln, bis hin zur Generierung von Code und Testfällen auf Basis wohlverstandener semantischer Kohärenz.
- Gleichermaßen Unterstützung des Top-down und des Bottom-up-Ansatzes über die Durchgängigkeit der Übergänge zwischen den Modellen.
- Enge Verzahnung des datengetriebenen und modellgetriebenen Vorgehens durch Vereinheitlichung des Komponenten- und Schnittstellenbegriffs.
- Hohe Anteile von Softwarewerkzeugen für alle Phasen der Entwicklung, konsequente Artefektorientierung, Entwicklung virtuell durch Erstellung geeigneter digitaler Artefakte, Automatisierung des Entwicklungsprozesses durch Simulation, Generierung sowie automatisierte Deduktion und Qualitätsüberprüfung.
- Verschmelzung von Entwicklung und Betrieb (Continuous Development and Delivery, DevOps, Agilität).
- Nutzung der Modelle der Entwicklung in der Weiterentwicklung und im Betrieb (vom Systemmodell zum digitalen Zwilling).
- Neuartige Kostenstrukturen, höhere Entwicklungs- im Verhältnis zu niedrigeren Produktionskosten durch die oft dramatisch höhere Variabilität.
- Der hohe Softwareanteil und die dadurch erzielte Möglichkeit, neue und geänderte Funktionalität auch während der Laufzeit der Systeme hinzuzufügen, führt zu neuartigen Geschäftsmodellen.

Hier wird deutlich, dass die Punkte miteinander interagieren, sich gegenseitig ergänzen und verstärken.

MBSE als wesentliche Basis

Der Ansatz, der durch Model-based Systems Engineering (MBSE) verfolgt wird, grenzt sich von dem heute noch weit verbreiteten dokumenten-zentrierten, manuellen Vorgehen deutlich ab. Ziele, Funktionen, Komponenten, Schnittstellen oder Qualitätseigenschaften, die ein System erfüllt oder bereitstellt, werden durch explizite Modellelemente beschrieben, die sich auf wohldefinierte und gut verstandene Konzepte der Domäne stützen. In der modellbasierten Entwicklung wird mit einer Zahl von Modellierungskonzepten gearbeitet, die so ausgewählt werden, dass sie die wesentlichen Systemeigenschaften anschaulich und gleichzeitig präzise erfassen. Für jedes dieser Modellierungskonzepte kann eine eigene Theorie angegeben werden. Das Gleiche gilt auch in der Beschreibung des Zusammenhangs zwischen den unterschiedlichen Modellierungskonzepten, die zum Einsatz kommen. Dies hat den Vorteil, dass der in dem Ansatz geschulte Anwender (ähnlich wie bei Programmiersprachen) mit den Konzepten vertraut ist und weiß, welche Modelle welche Arten von Fragen beantworten. Der Modellierer kennt so auch die grundsätzlichen Problemstellungen, die er abhandeln muss, um die Modelle zielgerichtet zu erstellen, und in Systemanalyse oder -synthese einzusetzen. Modellbasierung ist viel mehr, als nur einfach Modelle zu zeichnen bzw. aufzustellen, sondern beinhaltet die Fähigkeit, einen ausgearbeiteten Modellierungsansatz umfassend zu nutzen.

Vorgefertigte Modelltypen, die sich auf wissenschaftliche Grundlagenarbeiten stützen, garantieren Eigenschaften wie Kompositionalität, wodurch die Integration von Teilsystemen, die durch Modelle beschrieben sind, klar definiert ist (wie die Kommunikation über eine Schnittstelle), und Wiederverwendung. Wesentlich ist dabei, dass die Modelle inhaltlich aufeinander abgestimmt sind und man genau versteht, wie die verschiedenen Modellierungsansätze ineinandergreifen. Ein wichtiger Punkt ist die semantische Kohärenz über Modellgrenzen und Grenzen von Modellierungssprachen hinweg, die sicherstellt, dass ein umfassendes Modell des Systems entsteht. Eine Systembeschreibung ist dann nicht mehr dieses vage informelle Gebilde von Dokumenten, sondern ein verflochtenes Netz von normierten Modellen, die ein gemeinsames Ganzes bilden – eine Instanz des Systemmodells, das in dieser Form dann auch konsequent in einem zentralen Modell-Repository („single point of truth“) verwaltet wird. Auf dieses zentrale Systemmodell haben Stakeholder unterschiedliche Sichten, die

exakt auf ihre jeweilige Rolle im Produktlebenszyklus (etwa auf Funktionsentwickler, Architekt, Service) zugeschnitten sind. Das vermeidet unentdeckte Inkonsistenzen, vereinfacht insbesondere die Änderbarkeit der Modelle und reduziert so einen wesentlichen Kostentreiber.

Der Übergang von textueller Beschreibung in natürlicher Sprache zu Modellen bietet darüber hinaus den Vorteil, Mehrdeutigkeiten zu reduzieren, Konsistenz und Vollständigkeit überprüfbar zu machen und die Kommunikation zwischen den Stakeholdern zu verbessern. Je formaler das verwendete Modell, desto weniger Mehrdeutigkeit gibt es in der Beschreibung. Wichtiger noch: Formale Modelle ermöglichen automatische Analysen, etwa um das Zusammenspiel der einzelnen Komponenten zu prüfen, und erlauben auch den Einsatz von Generatoren, um Teile weiterer Modelle und Artefakte (etwa Code oder auch Testfälle) aus ausgearbeiteten Systemmodellen zu erzeugen.

Dies zeigt, dass modellbasierte Entwicklung einen, vielleicht den Schlüssel zum Advanced Systems Engineering mit hoher Werkzeugunterstützung darstellt.

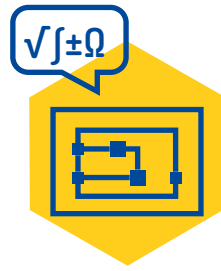
Ein wesentlicher Punkt dabei ist auch die Möglichkeit der werkzeuggestützten Entwicklung von Systemen. Hier ist die Software in zweierlei Hinsicht von besonderem Interesse. Zum einen wird die Entwicklung der Systeme in ihrer unvermeidlichen Komplexität in einer Art und Weise unterstützt werden, wie das unabdingbar ist, um solche Systeme überhaupt grundsätzlich voranzubringen. Zum anderen ergibt sich dazu ein wesentlicher Querbezug. So erfordert die werkzeuggestützte Entwicklung eine umfassende, systematische Modellierung und damit eine virtuelle Erfassung der Systeme. Dadurch können diese Modelle im Wesentlichen gleichzeitig auch als digitaler Zwilling beim Betrieb der Systeme verwendet werden und somit noch mal erweiterte Funktionalitäten sicherstellen. Die Software optimiert sich in der Entwicklung sozusagen selbst.

Drei Aspekte von MBSE sind getrennt voneinander zu betrachten, müssen jedoch geschickt aufeinander abgestimmt werden:



Methodik

- Definition der relevanten Modelltypen und ihrer Beziehungen
- Sichten (Views) auf das Systemmodell
- Möglichkeiten zur Analyse und Generierung für die spezifischen Modelle



Modellierungssprache

- Syntax: textuelle oder grafische Notationen
- Semantik: Bedeutung dieser Notationen



Werkzeuge

- Eingesetzte Werkzeugketten sollten kompatibel zueinander sein
- Methodik und Modellierungssprache werden mit hohem Automatisierungsgrad unterstützt

Methodik

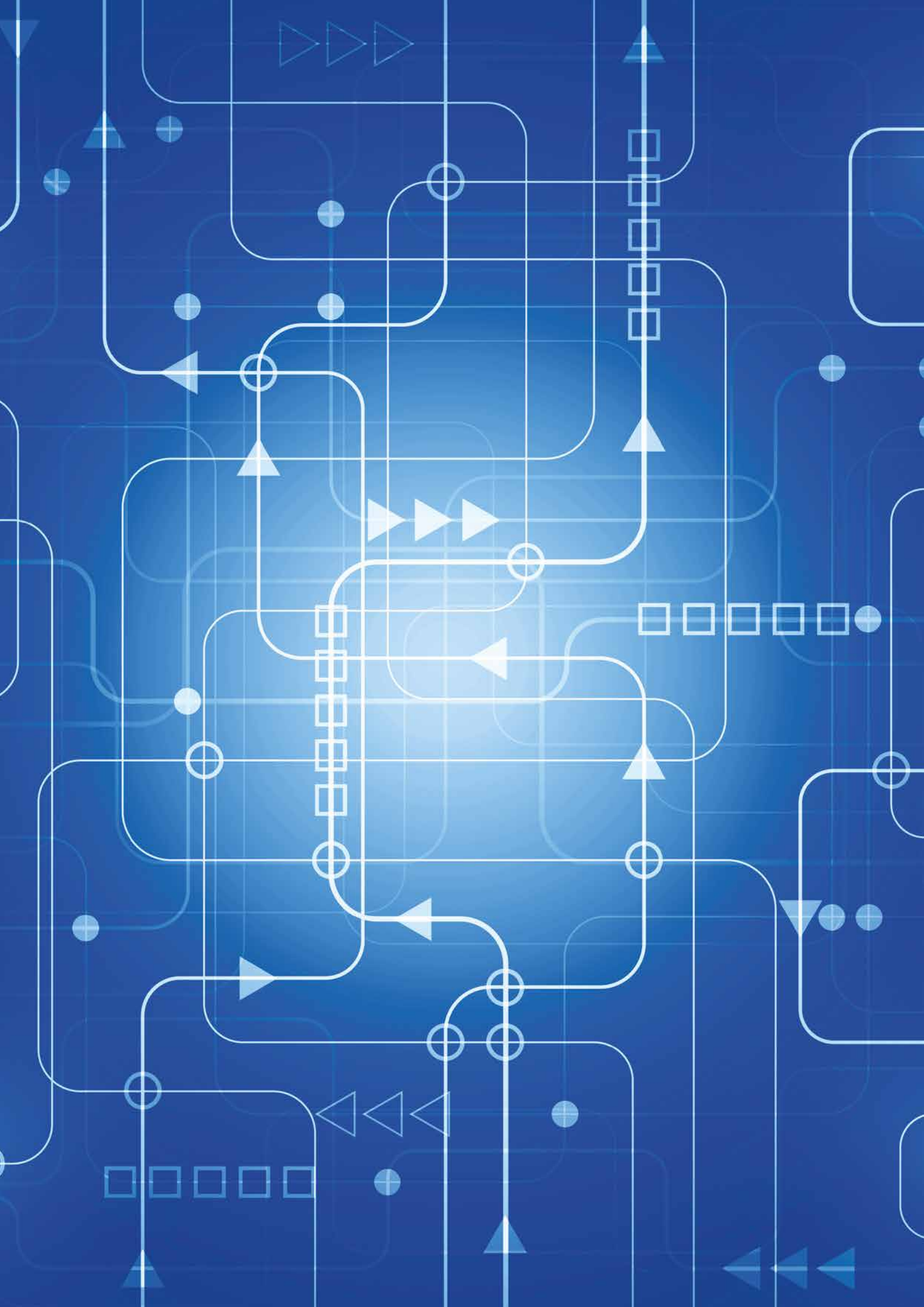
Ein Systemmodell, das seinerseits aus einer Vielzahl von Modelltypen besteht, ist selbst ein komplexes Artefakt, das ohne eine dahinterliegende wissenschaftlich fundierte Methodik nicht effektiv und effizient erstellt werden kann. Zu einer MBSE-Methodik gehört demnach die Definition der relevanten Modelltypen und ihrer Beziehungen. Ferner definiert eine Methodik Sichten (Views) auf das Systemmodell, die das komplexe Gesamtmodell in mehrere weniger komplexe und der gegebenen Entwicklungssituation angepasste Modelle strukturieren. Beispiele für Sichten sind Funktionssicht und logische bzw. technische Architektursicht. Eine MBSE-Methodik beschreibt weiterhin Möglichkeiten zur Analyse und Generierung für die spezifischen Modelle. Der Formalisierungsgrad der im Systemmodell definierten Modelle bestimmt die Automatisierbarkeit von Analysen und Modellgenerierung. Ein hoher Automatisierungsgrad, der natürlich auch von den verwendeten Werkzeugen unterstützt werden muss, erlaubt wiederum, wie beispielsweise in der reinen Softwareentwicklung, einen iterativen und agilen Entwicklungsprozess.

Modellierungssprache

Die Modellierungssprache definiert die Syntax und die Semantik, mit der die Modelle der Methodik konkret beschrieben werden, etwa welche textuellen oder grafischen Notationen erlaubt sind (Syntax). Die Semantik einer Modellierungssprache legt fest, welche Bedeutung diese Notationen haben. Problematisch hierbei ist, dass viele der gängigen Modellierungssprachen (etwa SysML) bestenfalls eine lose definierte Semantik haben. Dadurch treten Probleme ähnlich denen natürlichsprachlicher Beschreibungen auf.

Werkzeuge

Die Methodik und die Sprache müssen von passenden Werkzeugen unterstützt werden, um effizient von den Möglichkeiten, die Modelle bieten, Gebrauch machen zu können. Entscheidend ist zudem, dass die eingesetzten Werkzeugketten kompatibel miteinander sind und die genutzten Werkzeuge die gewählte Methodik und die Modellierungssprache sowohl syntaktisch als auch semantisch und mit hohem Automatisierungsgrad unterstützen.



Der durchgängige Ansatz von SPES/CrEst

In den BMBF-Projekten SPES2020 (s. Seite 13 [3]) und den Nachfolgeprojekten SPES_XT (s. Seite 13 [4]) und CrEst (s. Seite 13 [5]) wurden eine Methodik und ein Rahmenwerk für MBSE entwickelt, die eine effiziente, modellbasierte Entwicklung eingebetteter Systeme erlauben. Das SPES-Rahmenwerk liefert eine umfassende, werkzeug- und modellierungssprachenunabhängige Methodik für MBSE und bietet einen umfangreichen Satz konkreter Modellierungstechniken und -aktivitäten, die auf einem formalen, mathematischen Fundament aufbauen. Das SPES-Rahmenwerk stützt sich auf vier Prinzipien von herausragender Bedeutung:

- 1 Funktionale sowie auch nichtfunktionale Anforderungen werden auf Systemebene unter der Verwendung angemessener Abstraktionen (Sichten) vollständig modelliert.
- 2 Konsistente Betrachtung von Schnittstellen auf jeder Ebene.
- 3 Dekomposition des Schnittstellenverhaltens und die Beschreibung von Systemen über Teilsysteme und Komponenten in verschiedenen Granularitätsebenen.
- 4 Definition von Modellen, die auf den genannten Prinzipien aufbauen, für eine Vielzahl von Querschnittsthemen (Variabilität, Safety etc.) und Analyse-möglichkeiten.

Ein Systemmodell im SPES-Ansatz ist ein konzeptuelles („generisches“) Modell zur Beschreibung von Systemen und deren Eigenschaften, bestehend aus:

- Modellen für den operationellen Kontext, der das System zur Laufzeit beeinflusst bzw. von diesem beeinflusst wird,
- Modellen der Schnittstelle, die das System eindeutig von seinem operationellen Kontext abgrenzt,
- einem Verhalten des Systems, das an der Schnittstelle beobachtbar ist, und
- Modellen der inneren Struktur des Systems durch Zustandsmaschinen oder aus zueinander in Beziehung stehenden und kommunizierenden Teilsystemen (Architektur), auf die das SPES-Rahmenwerk rekursiv angewendet werden kann.

Den Kern der Methodik bildet das universelle Schnittstellenkonzept, das für alle Elemente Schnittstellen definiert, die jeweils aus der Schnittstellensyntax und einer Beschreibung des an der Elementgrenze beobachtbaren Verhaltens bestehen. Anforderungen, Funktionen, logische oder technische Komponenten werden so über die Schnittstelle beschrieben und stehen über ihre Schnittstellen miteinander in Verbindung. Das Schnittstellenkonzept liefert die Basis für Dekomposition und Modularität.

Sichten (Views) im SPES-Rahmenwerk sind Anforderungen (Requirements View), funktionale Architektur (Functional View) sowie logische Architektur (Logical View) und technische Architektur (Technical View), die das System in die beteiligten logischen oder technischen Komponenten zerlegen. Übergreifende Themen (sogenannte crosscutting topics) ergänzen die Modelle der Viewpoints entsprechend. Das erlaubt beispielsweise, Aspekte der funktionalen Sicherheit von Systemen zu beschreiben und zu analysieren. Das SPES-Rahmenwerk ist offen für die Erweiterung um neue Sichten wie etwa eine geometrische Sicht.

Um die Komplexität des Systems und des zugehörigen Entwicklungsprozesses beherrschbar zu machen, werden relevante Architekturkomponenten nach dem Prinzip „Teile und herrsche“ als eigenständige (Sub-)Systeme betrachtet, für die wiederum Modelle und Views nach dem SPES-Ansatz erstellt werden. Dadurch entstehen vordefinierte Sichten für das System und dessen Subsysteme mit darauf abgestimmten Granularitätsebenen. Die Modellierung des Systems auf den verschiedenen Granularitätsebenen bestimmt jeweils das Subjekt des

Diskurses (Scope) und ist ein wichtiges Mittel in der modellbasierten Entwicklung, um die Systemkomplexität zu reduzieren und den Entwicklungsprozess insgesamt handhabbar zu gestalten.

Die SPES-MBSE-Methodik verfolgt dabei einen strikten systemzentrischen Ansatz, der ein System auf unterschiedlichen Granularitätsebenen betrachtet. Auf der obersten Granularitätsebene befinden sich immer die Modelle, die das betrachtete System als Ganzes abbilden. Auf (unterschiedlich vielen) weiteren Granularitätsebenen werden – sukzessive – immer feinere Teilsysteme betrachtet und weitere Details modelliert. Obwohl vom Grundsatz her „Top-down“, werden auch iterative und agile Vorgehensmodelle unterstützt. Das mathematische Modell FOCUS, auf dem das SPES-Rahmenwerk basiert, stellt die Konsistenz der Modelle von System und Subsystemen sicher. Granularitätsebenen helfen dabei,

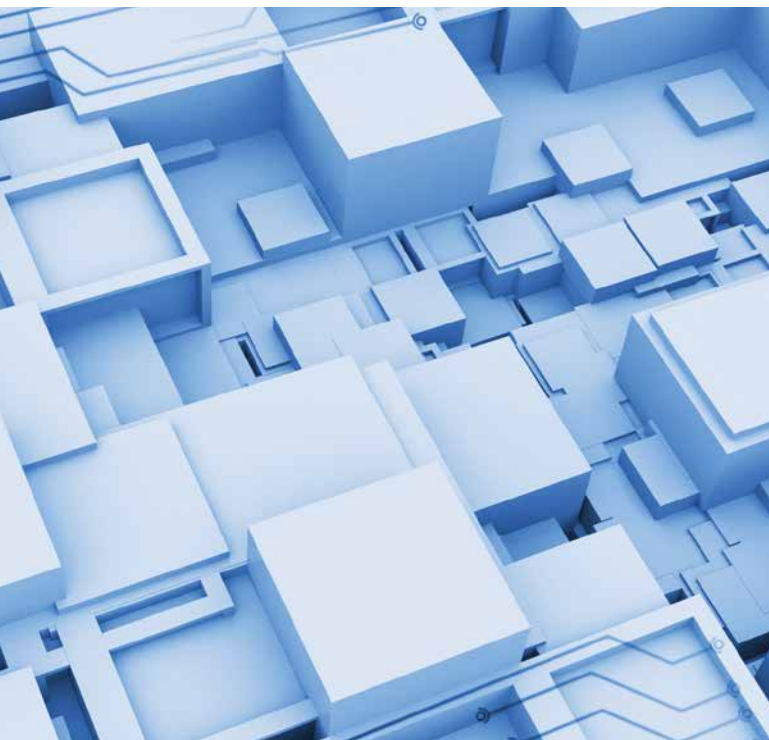
- 1 die Komplexität des betrachteten Systems zu beherrschen,
- 2 Prüfungen am System auf verschiedenen Ebenen der Komplexität durchzuführen,
- 3 Entwicklungsaufgaben, beispielsweise an Zulieferer, zu verteilen und
- 4 einzelne Modelle mehrfach wiederzuverwenden.

Da das Prinzip der Granularitätsebenen nur auf dem Schnittstellenkonzept beruht, erlaubt der Mechanismus die Integration der verschiedenen Engineering-Disziplinen (Maschinenbau, Elektrotechnik, Softwaretechnik). Solange das Schnittstellenkonzept realisiert wird, ist es unerheblich, mit welchen Methoden, Prozessen oder Werkzeugen die Subsysteme auf den unteren Granularitätsebenen entwickelt werden.

Neben Abstraktion und Granularität ist Konsistenz ein wichtiges Merkmal der Modelle im SPES-Rahmenwerk. Wir unterscheiden zwischen horizontaler und vertikaler Konsistenz. Zwei Modelle sind horizontal konsistent, wenn sie zu unterschiedlichen Views auf dasselbe System (das heißt innerhalb einer Granularitätsebene) gehören und keine widersprüchlichen Eigenschaften des betrachteten Systems repräsentieren. Zwei Modelle sind vertikal konsistent, wenn sie auf unterschiedlichen Granularitätsebenen zu einer View gehören und dabei keine widersprüchlichen Eigenschaften des betrachteten Systems in Bezug auf die spezifische View repräsentieren.

Das SPES-Rahmenwerk macht keinerlei Vorgaben über die Reihenfolge, in der die verschiedenen Modelle zu den Views erstellt werden sollten. Somit lassen sich mit der SPES-Methode Top-down ebenso wie Bottom-up-Vorgehensweisen und auch iteratives oder inkrementelles Entwickeln implementieren. Wie oben bereits erwähnt, erlaubt der Mechanismus der Granularitätsebenen die Integration verschiedenartiger Vorgehensweisen und Entwicklungswerkzeuge, wie sie beispielsweise bei mechatronischen Systemen typischerweise notwendig sind. Da die formale Basis der SPES-Methodik auch Unterspezifikation unterstützt, ist es möglich, die Modelle iterativ Schritt für Schritt zu erweitern und sukzessive zu verfeinern. Das bedeutet insbesondere, dass der modellbasierte Ansatz den Grundprinzipien der agilen Entwicklung nicht widerspricht. Auch Techniken wie beispielsweise „continuous integration“ können gezielt eingesetzt werden. Zu betonen ist dabei, dass diese Form des agilen Vorgehens dann eben nicht codezentriert, sondern modellzentriert ist.

Im gerade beendeten Projekt CrEst wurde das SPES-Rahmenwerk erweitert, um auch Kollaboration und Dynamik (Bildung von Systemverbänden zur Laufzeit) bei Systemen zu unterstützen. Die bestehende Viewpoint-Struktur wurde im Wesentlichen beibehalten, aber die enthaltenen Modelle wurden um zusätzliche Modelltypen und Informationen ergänzt.



Methodische Erweiterungen von SPES hin zu ASE

Wie in den vorangegangenen Abschnitten ausgeführt, bietet die SPES-Methodik als vollständig modellbasierter Ansatz eine ausgezeichnete Basis für ASE. Insbesondere umfasst das:

- Durchgängige Modelle, die den gesamten Produktentwicklungsprozess abdecken.
- Eine Vielzahl von Modellierungstechniken zur Sicherstellung und Analyse der Korrektheit komplexer Funktionalität.
- Modularität und Dekomposition erlauben die Wiederverwendung der Modellelemente auf allen Ebenen.
- Konsistente Architektursichten und ausführbare Modellelemente erlauben funktionale Prototypen und automatisierte Analysen bereits in frühen Phasen des Entwicklungsprozesses (Frontloading).
- Strikte systemzentrische Vorgehensweise zur Unterstützung der notwendigen multidisziplinären Designansätze.
- Verzahnung der Entwicklungsprozesse der beteiligten Ingenieurdisziplinen (Informatik, Maschinenbau, Elektrotechnik) sowie der dort verwendeten Werkzeuge über das Konzept der Granularitätsebenen.
- Umfangreiche Modelle speziell für das Software Engineering und starke Integration von Software mit mechanischen und elektrotechnischen Bestandteilen.
- Im gerade zu Ende gegangenen Projekt CrEst wurde das SPES-Rahmenwerk um Aspekte wie dynamische Vernetzung und Kollaboration von Systemen zur Laufzeit erweitert. Dazu wurde eine Reihe von zusätzlichen Querschnittsthemen definiert und die Modelle der bestehenden Viewpoints entsprechend ergänzt.

Neue Methodik- und Querschnittsthemen wären etwa:

- Erweiterung der vorherrschend diskreten Modelle hin zu analogen Modellen; Integration von regelungstechnischen Ansätzen – Stichwort interdisziplinäre Modellierung.
- Einbindung neuartiger Verfahren zur Erzeugung von Teilsystemen und deren Verhalten durch Big Data und maschinelles Lernen.
- Integration von Sicherheitsmodellen für Safety und Security in die modellgetriebene Entwicklung mit Schwerpunkt auf Zertifizierung.
- Betrachtung von digitalen Zwillingen (Digital Twins) als Teil des zu entwickelnden Gesamtsystems.
- Qualitätssicherung zur Laufzeit.
- Systemqualifizierung und Zertifizierung.
- Dedizierte Nutzerschnittstellen.

Der Fokus bei der Entwicklung des SPES-Rahmenwerks lag bislang ausschließlich auf dem Produktentstehungsprozess. Gleichzeitig bietet SPES die Möglichkeit, neue Viewpoints zu den bereits bestehenden Viewpoints hinzuzufügen bzw. die bestehenden Viewpoints über zusätzliche Querschnittsthemen zu erweitern und in das Rahmenwerk zu integrieren. Eine Weiterentwicklung in Richtung ASE sollte Erweiterungen hinsichtlich des gesamten Product-Life-Cycles einschließlich Modellen und Erweiterungen für Markt- und Geschäftsmodelle sowie System Operation und Servicemodelle berücksichtigen.

Die in den Vorhaben SPES, SPES_XT und CrEst entwickelten Modelle, Methodiken und Techniken wurden bewusst unabhängig von einer konkreten Modellierungssprache verfasst, um eine größtmögliche Einsatzbreite zu gewährleisten. In der industriellen Praxis, insbesondere bei kleinen und mittelständischen Unternehmen, hat sich gezeigt, dass nahezu alle MBSE-Einführungsprojekte auf SysML als Modellierungssprache setzen, trotz aller damit verbundenen offenen Fragen und Unzulänglichkeiten. Grund dafür ist sowohl die Verbreitung von SysML in den Unternehmen als auch die Unterstützung von SysML in vielen in der Praxis verfügbaren MBSE-Werkzeugen. Aus der Verbreitung und Akzeptanz von SysML ergibt sich die Notwendigkeit, mit der SPES-Methodik SysML als Modellierungssprache sowohl syntaktisch als auch semantisch explizit zu unterstützen. Ein Forschungsprojekt, das auf Basis

des SPES-Rahmenwerkes die heute bestehenden Barrieren zur industriellen Einführung von MBSE abbaut und somit den Weg für eine breite Erschließung der SPES-Methodik in der Industrie auf Basis verbreiteter Sprachsyntax und pragmatischer Werkzeuge ebnet, ist in Planung.

Teile zukünftiger Systeme werden durch die Nutzung von Techniken wie maschinelles Lernen (Machine Learning – ML) oder allgemeiner: Künstliche Intelligenz (KI) bestimmt. Durch die Integration von KI-Komponenten in eingebettete Systeme werden die erheblichen Potenziale aktueller und zukünftiger KI-Technologien in eingebetteten Systemen nutzbar gemacht. Deren Nutzung ermöglicht zukünftigen eingebetteten Systemen, die durch die Digitalisierung stetig wachsende Anzahl an Informationen geeignet zu verarbeiten und sich an sich verändernde Bedingungen sowie aus den Daten gewonnene Erkenntnisse zur Laufzeit geeignet anzupassen. Um auch solche Systeme effizient ent-

wickeln zu können, sind die verfügbaren expliziten Modellierungsmethoden entsprechend um implizit gelernte Modellierungstechniken zu erweitern. Grundsätzlich ist der hier vorgestellte Ansatz bereits für Systeme geeignet, die über KI-Komponenten verfügen. Das universelle Schnittstellenkonzept des SPES-Rahmenwerkes bietet hierfür eine tragfähige Basis. Zu berücksichtigen ist allerdings, dass bei solchen Systemen das Verhalten während der Laufzeit einer gewissen Variabilität unterliegt, wenn die Komponente z.B. während der Laufzeit weiterlernt.

Eine zentrale Herausforderung für die Integration von KI-Methoden in eingebettete Systeme ist also die Gewährleistung (Nachweisbarkeit) der essenziellen Funktionalitäts- und Qualitätseigenschaften der Systeme – und dies trotz nicht vollständig spezifizierbarer, oft nichtdeterministischer oder sich sogar dynamisch adaptierender Systemkomponenten aufgrund von zur Entwicklungszeit nicht vorhersehbaren Anpassungen der Systeme während der Laufzeit.



Zusammenfassung

ASE benötigt eine saubere wissenschaftliche Fundierung und eine konsequente Integration von Methoden der Softwareentwicklung in der Systementwicklung bei dem Entwurf von softwareintensiven Systemen. Zentral für ein Advanced Systems Engineering sind die Nutzung von digitalen Techniken sowohl im Produkt wie auch im Entwicklungsprozess und die Erschließung der Synergien dazwischen. Die Vorarbeiten im Bereich modellbasierte Entwicklung softwareintensiver Systeme bieten einen idealen Einstieg. Zu bewältigen ist nicht weniger als ein Paradigmenwechsel vom Engineering mechanischer Maschinen zum Engineering vernetzter, informationszentrierter Systeme.

Weiterführende Arbeiten

- [1] Broy, Stolen: „Specification and development of interactive systems: focus on streams, interfaces, and refinement“, Springer 2001.
- [2] Broy: „A logical basis for component-oriented software and systems engineering“, in: The Computer Journal, vol. 53, no. 10, pp. 1758-1782, Dec. 2010.
- [3] Pohl, Hönninger, Achatz, Broy: „Model-Based Engineering of Embedded Systems“, Springer 2012.
- [4] Pohl, Broy, Daembkes, Hönninger: „Advanced Model-Based Engineering of Embedded Systems Extensions of the SPES 2020 Methodology“, Springer 2016.
- [5] Pohl, Broy, Böhm, Klein, Rumpe, Schröck: „Model-Based Engineering of Collaborating Embedded Systems“, Springer verfügbar ab Herbst 2020.
- [6] „IEEE Standard Glossary of Software Engineering Terminology“, in IEEE Std 610.12-1990, Vol., no., pp. 1–84, 1990.
- [7] „ISO/IEC/IEEE 42010:2011 – Systems and software engineering – Architecture description“, International Organization for Standardization, 2011.
- [8] Broy, Böhm, Junker, Vogelsang, Voss: „Praxisnahe Einführung von MBSE – Vorgehen und Lessons Learnt“, White Paper, fortiss GmbH, 2020.
- [9] Broy, Rumpe: „Modulare hierarchische Modellierung als Grundlage der Software- und Systementwicklung“, in: Informatik-Spektrum. Springer Verlag, Band 30, Heft 1, Februar 2007.
- [10] Broy, Crane, Dingel, Hartmann, Rumpe, Selic: „UML 2 Semantics Symposium: Formal Semantics for UML“, in: Models in Software Engineering. Workshops and Symposia at Models 2006. Genoa, LNCS 4364, Springer, January 2007.
- [11] Geisberger, Broy: „agendaCPS Integrierte Forschungsagenda Cyber-Physical Systems“, acatech – Deutsche Akademie der Technikwissenschaften, März 2012.

Impressum

Herausgeberin

fortiss gemeinnützige GmbH
Guerickestraße 25, 80805 München
E-Mail: info@fortiss.org
www.fortiss.org

Autoren

Prof. Dr. Dr. h. c. Manfred Broy
Dr. Wolfgang Böhm
Prof. Dr. Bernhard Rumpe

Lektorat

Lektorat Süd, München

Gestaltung

Sonja Taut

Druck

Cewe Print GmbH
Enderstraße 92c, 01277 Dresden

ISSN Print

2699-1217

ISSN Online

2700-2977

1. Auflage, Juli 2020

Bildnachweise

Titel: shutterstock ©Sylverarts Vectors
Seite 4: shutterstock ©metamorworks
Seite 8: shutterstock ©bestfoto77
Seite 10: shutterstock ©alleachday
Seite 12: shutterstock ©vs148
Seite 14: fortissGmbH ©Kathrin Kahle



fortiss ist das Forschungsinstitut des Freistaats Bayern für softwareintensive Systeme und Services mit Sitz in München. Die WissenschaftlerInnen am Institut arbeiten in Forschungs-, Entwicklungs- und Transferprojekten mit Universitäten und Technologiefirmen in Bayern, Deutschland und Europa zusammen. Schwerpunkte sind die Erforschung modernster Methoden, Techniken und Werkzeuge der Softwareentwicklung, des Systems- & Service-Engineering und deren Anwendung auf kognitive cyberphysische Systeme wie das Internet of Things (IoT).

fortiss ist in der Rechtsform einer gemeinnützigen GmbH organisiert. Gesellschafter sind der Freistaat Bayern (Mehrheitsgesellschafter) und die Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.

Alle Angaben in diesem White Paper wurden mit größter Sorgfalt zusammengestellt. Trotzdem sind Fehler nicht ausgeschlossen. Es wird weder eine Garantie noch eine juristische Verantwortung oder jegliche Haftung für Folgen, die auf fehlerhafte Informationen zurückzuführen sind, übernommen.

fortiss GmbH

Guerickestraße 25

80805 München

Deutschland

www.fortiss.org

Tel.: +49 89 3603522 0

E-Mail: info@fortiss.org



fortiss