

Anleitung zur Erstellung ausführbarer semantischer Standards (SemNorm)

VDE SPEC 90010 V1.0 (de)

Vorwort

Veröffentlichungsdatum dieser VDE SPEC: 30. Mai 2023.

Zur vorliegenden VDE SPEC wurde kein Entwurf veröffentlicht.

Für diese VDE SPEC ist die VDE SPEC Projektgruppe „DKE/GRP_SemNorm“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Diese VDE SPEC wurde nach dem VDE SPEC-Verfahren erarbeitet. Die Erarbeitung von VDE SPEC 90010 erfolgt in Projektgruppen und nicht zwingend unter Einbeziehung aller interessierten Kreise.

Diese VDE SPEC ist **nicht** Bestandteil des VDE-Vorschriftenwerks oder des Deutschen Normenwerks. Diese VDE SPEC ist insbesondere auch **keine** Technische Regel im Sinne von § 49 EnWG.

Verfasser dieser VDE SPEC sind:

- Damian A. Czarny, VDE e.V.
- Jan-Bernhard DeMeer, Hochschule für Technik und Wirtschaft (HTW)
- Prof. Dr. Christian Diedrich, ifak Institut für Automation und Kommunikation e.V.
- Roland Heidel, Kommunikationslösungen e.K.
- Olga Meyer, Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA
- Andreas Müller, Schaeffler AG
- Axel Rennoch, Fraunhofer FOKUS
- Daniel Stock, Capgemini Engineering

Trotz großer Anstrengungen zur Sicherstellung der Korrektheit, Verlässlichkeit und Präzision technischer und nicht-technischer Beschreibungen kann die VDE SPEC-Projektgruppe weder eine explizite noch eine implizite Gewährleistung für die Korrektheit des Dokuments übernehmen. Die Anwendung dieses Dokuments geschieht in dem Bewusstsein, dass die VDE SPEC-Projektgruppe für Schäden oder Verluste jeglicher Art nicht haftbar gemacht werden kann. Die Anwendung der vorliegenden VDE SPEC entbindet den Nutzer nicht von der Verantwortung für eigenes Handeln und geschieht damit auf eigene Gefahr.

Im Zuge der Herstellung und/oder Einführung von Produkten in den Europäischen Binnenmarkt muss der Hersteller eine Risikoanalyse durchführen, um zunächst festzustellen, welche Risiken das Produkt möglicherweise mit sich bringt. Nach Durchführung der Risikoanalyse bewertet er diese Risiken und ergreift gegebenenfalls geeignete Maßnahmen, um die Risiken wirksam zu eliminieren oder zu minimieren (Risikobewertung). Die vorliegenden VDE SPEC entbindet den Nutzer nicht von dieser Verantwortung.

Links zu den Websites Dritter stellen keine Zustimmung zu deren Inhalten seitens VDE dar. VDE ist nicht verantwortlich für die Verfügbarkeit oder den Inhalt dieser Websites. Das Herstellen einer Verbindung zu diesen Websites geschieht auf eigene Gefahr der Benutzer.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte betreffen können. VDE ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren

Über DIN-Connect: Das Programm fördert innovative Projektideen mit Potenzial für Normung und Standardisierung und richtet sich an Start-ups und KMU. DIN und DKE stellen für das Förderprogramm eigene finanzielle Mittel zur Verfügung, um Innovationen in die Normung und Standardisierung zu überführen und ihnen auf diese Weise zu einem schnelleren Marktzugang zu verhelfen.

Inhalt

Einleitung	1
1 Anwendungsbereich	3
2 Normative Verweisungen	4
3 Begriffe und Abkürzungen	4
3.1 Begriffe	4
3.2 Abkürzungen	8
4 Der SemNorm Ansatz	9
4.1 Das Semiotische Dreieck	9
4.2 Die Bedeutung von Morphismen	11
4.3 Der Algebraische Stil	12
4.4 Das SemNorm Narrativ	12
4.5 Das Ziel einer semantischen Norm	15
5 Semantische Beschreibungselemente	17
5.1 Semantische Artefakte	17
5.1.1 Ereignisse (Graph Edges)	17
5.1.2 Prozesse und Signaturen (Graph Vertices)	17
5.1.3 Schnittstellen und Umschreiberegeln (Graph Rewrite Rules)	17
5.1.4 RAMI4.0 Kontexttransformationen (Graph Rewriting)	18
5.1.5 Kontext Signaturen (Graph Interoperability)	18
5.1.6 informationelle Beziehungen (Data Graph Relation)	18
5.1.7 Anwendungsfall-Beschreibung, System Runs (Graph Trajectory)	19
5.1.8 Value Streams	19
5.1.9 Variablen, Daten, Funktionen und Prozesse	19
5.1.10 Semantische Transferfunktionen und Inzidenzen	20
5.1.11 Interoperabilität zwischen Kontexten am Beispiel der Energieversorgung einer SM-Fabrik	21
6 Semiotische Best Practices in der Normung	22
6.1 Semiotische Darstellung Cyber-Physischer Systeme	22
6.2 Semiotische Best Practices für Systeme	23
6.3 Semiotische Best Practices für Transferfunktionen (Vorgänge)	23
6.4 Semiotische Best Practices für Entscheidungsvorgänge	23
6.5 Semiotische Best Practices um Morphismen zu definieren	24
6.5.1 Morphismen für Datentypen	24
6.5.2 Morphismen für Wissen (Knowledge)	26
Anhang A SemNorm Grundlagen	28
A.1 SemNorm Grundlagen	28
A.1.1 SemNorm Angewandte Theorien	28
A.1.2 SemNorm angewandte Konzepte	29
A.1.3 SemNorm semiotische Bezüge (Morphismen)	30
Anhang B SemNorm Engineering Guidelines	33
B.1 SemNorm Engineering	33

B.1.1	Systemmodellierung	33
B.1.2	Systemvariablen	33
B.1.3	System-Attribute	34
B.1.4	Semiotisches Dreieck als Konzept für die Normung	35
B.1.5	Graph Trajektorien und Graph-Zustandsräume	36
B.1.6	Common System Logics	37
Anhang C Use Case ‚Smart Grid‘ dargestellt in ‚CSlang‘ Notation		38
C.1	Formale Spezifikation eines Smart Grid Use Case	38
C.1.1	Identifikation der Stakeholders (Graph Vertices)	38
C.1.2	Stakeholder Kontexte mit ihren kontinuierlich-veränderlichen Eigenschaften	38
C.1.3	Interoperabilitäts-Signatur zwischen den Komponenten VWC und ROT	38
C.1.4	Einfügung einer Kante zwischen zwei interaktionsfähigen Knoten	39
Anhang D Use Case Vergleich (Bild D.3) von heterogenen Inforationsmodellen ‚Heizung‘ (Bild D.1) und ‚Fahrrad‘ (Bild D.2)		40
Literaturverzeichnis		42
Bilder		
Bild 1	– Semiotisches Dreieck mit semiotischen Domänen und Morphismen	11
Bild 2	– Narrativ des SemNorm-Ansatzes	13
Bild 3	– Anwendung I4.0-Sprache in der VDE SPEC SemNorm	14
Bild 4	– Block- und Graphen Diagramm eines automatisierten regelungstechnischen-Systems (IACS)	21
Bild 5	– Beispiel Termalgebra des ‚BDSG Datengeheimnisses‘	25
Bild 6	– Beispiel ‚Copyright Agreement‘ als Event und Knowledge Graph [Source: SCI4.0]	26
Bild B.1	– Systemmodellierung mit DGLs im Zustandsraum	33
Bild B.2	– Morphismus als Transformation eines semantischen Systemmodells in ein reales System	34
Bild B.3	– Informationsmodell für Systemeigenschaften	35
Bild B.4	– Semiotisches Dreieck am Beispiel ‚Merkmal Wärmekapazität‘	36
Bild B.5	– Heizungs-Informationsmodells in formaler DL dargestellt	36
Bild D.1	– Informationsmodell ‚Heizung‘ in formaler DL dargestellt	40
Bild D.2	– Informationsmodell ‚Fahrrad‘ in formaler DL dargestellt	40
Bild D.3	– Vergleich heterogener ‚Heizung‘ – ‚Fahrrad‘ Informationsmodelle	41

Einleitung

Ein sogenannter *semantischer Standard* ‚SemNorm‘ ist ein natürlich-sprachiges Standard-Dokument, das nicht direkt maschineninterpretierbar ist, sondern Anforderungen an ein semantisches Modell, das ausgestattet mit analytischen Fähigkeiten, auch ‚Digitaler Zwilling‘ genannt, stellt. Das Modell ist nach Transformation auf eine geeignete Plattform maschinell interpretierbar, bzw. kann rechner-gestützt simuliert werden.

Eine daraus folgende wichtige Empfehlung zur Gestaltung einer semantischen Norm ist es, das Konzept eines semantischen Modells schon beim ersten Entwurf in Betracht zu ziehen. Ein Beispiel hierfür ist das Konzept ‚Smart Manufacturing‘ (siehe IEC TC65/WG23) in welchem die Zusammenwirkung (Interoperabilität) und Kooperation zwischen Menschen, Maschine, Roboter und Produktionssystem gestaltet wird.

In der Systemtechnik wurde der Begriff Semantik in mehreren Entwicklungsstadien und mit Schwerpunkt auf verschiedene Aspekte eingeführt. In [79] wo ein Überblick über den Bereich des ‚semantischen Web‘ gegeben wird, heißt es, dass sich das Gebiet der Semantik nun seit dem Jahr 2000 in ihrem dritten Einführungszustand befinden soll.

In der ersten Phase der Einführung wurde die Semantik durch Hinzufügen von vielen Metadaten "erklärt", womit sich ein Begriff ergeben hat, der von einem Objekt- oder Subjekttyp mit seinen Beziehungen zu anderen Objekt- oder Subjekttypen beschrieben wird. Eine Zusammenfassung von Begriffen wird als ‚**Ontologie**‘ bezeichnet. Es war die Zeit großer ontologischer Datenbanken, für die Zugangsmechanismen wichtig wurden. W3C gab Antworten mit RIF und RDF, d. h. dem ‚Rule Interchange Format‘ und dem ‚Resource Description Framework‘. Beide sind kompatibel mit der OWL d. h., der ‚Web Ontology Language‘.

Syntaktische, in einer Datenbeschreibungssprache, z. B. mittels ADT oder XML, etc. und semantische, in einem formalen Kalkül, z. B. FOL, Graphentheorie oder auch UML2 dargestellte Beschreibungen werden oft nicht klar voneinander getrennt. So wird an manchen Stellen ein Meta-Referenzmodell, wie z. B. in [IEC SyC SM2 DTR 63306] gefordert, um damit eine einheitliche Semantik mit anderen kontextspezifischen Semantiken gleichzusetzen. Das widerspricht der Forderungen nach einer einheitlichen ‚Common Logic (CL)‘, wie es vielfach im Rahmen der Normung, u.a. auch im ‚SemNorm‘-Ansatz gefordert wird. Jeder Kontext soll seine eigenen individuellen Signaturen, d. h. spezifische syntaktischen Konzepte, auf einer gemeinsamen Basis der semantischen Interpretation, beibehalten können.

Die zweite Phase war die Phase der Erfindungen von ‚linked Open Data‘, die als RDF-Graphen entworfen wurden. Die Verlinkung wurde jedoch durch IRIs, den ‚Internationalized Resource Identifiers‘ vorgenommen, die in mehreren Graphen in Erscheinung treten würden, womit sich ein sehr großer verteilter Graph ergeben würde.

Die dritte laufende Phase ist die Phase der Knowledge-Graph Entwicklungen, wie sie in großen Industrien, wie Google und anderen, ausgeführt werden. Der Hauptunterschied zu den beiden vorherigen Phasen besteht darin, dass die Industrie ihre Knowledge-Graphen nicht zum Download freigibt. Auf die Knowledge-Graphen kann jedoch über eine API zugegriffen werden. Google hat eine alternative Syntax zur RDF-Syntax bereitgestellt, um verlinkte Daten herunterzuladen.

Eine erste einfache Art der Kommunikation wird auf syntaktischer Ebene mit dem Austausch von Nachrichten erreicht, z. B. indem Nachrichten mit OWL, RDF, RIF oder anderen ontologischen Beschreibungstechniken beschrieben werden.

Eine weitere, ‚höhere Art‘ der Kommunikation erfolgt auf der ‚**Ebene des Verstehens**‘ einer Nachricht, die nach dem Austausch syntaktisch formatierter Nachrichten, z. B. eine Maschinenblaupause, stattfindet. Im I4.0 Kontext bedeutet **Verstehen**, dass Maschine und menschlicher Bediener das gleiche Modell mit verschiedenen Werkzeugen interpretieren, wobei der Mensch seine kognitiven Fähigkeiten und die Maschine ihre bzw. der Computer seine technisch rechnerischen Möglichkeiten einsetzt.

Eine Blaupause wird aus den empfangenen Nachrichten rekonstruiert und vom Empfänger Mensch, Maschine und Computer auf jeweils eigene Art gelesen und ‚verstanden‘. Sie können somit ein ‚vergleichbares Verständnis‘ erzeugen, weil es auf einer geteilten Darstellung, der Blaupause, beruht. Das setzt voraus, dass die kommunizierten Blaupausen in einer dem Menschen und dem Computer verständlichen Weise dargestellt werden. Dies ist mit der mathematischen Graph-Theorie gegeben. Der Mensch versteht die mathematische Darstellung und der Computer kann sie maschinell verarbeiten.

Das Semiotische Dreieck, worauf sich der SemNorm Ansatz bezieht, erklärt das erzeugte gemeinsame Verständnis, dadurch, dass seine drei semiotischen Ecken miteinander in Bezug gebracht werden.

Dieser Mechanismus wird **Morphismus** genannt und man erzeugt damit konkrete Semantik, indem Artefakte einer semiotischen Ecke, Artefakten einer der beiden anderen semiotischen Ecken zugeordnet werden.

Die drei semiotischen Ecken sind zum besseren Verständnis farbig kodiert (siehe Bild 1). Grün steht für die Symbolik und Sprache, blau steht für das Konzeptionelle und die Semantik, grau steht für die technische Lösung, den Dingen und Assets bzw. den betrachteten Phänomenen.

Für jede semiotische Ecke ist in Bild 1 jeweils ein Anwendungsbeispiel gegeben. Die analytischen Fähigkeiten der konzeptionellen blauen Ecke werden demonstriert mit einer Graphen-theoretischen Simulation, ausgeführt am DaiN Lab der TU Berlin. Der Screenshot zeigt einen Simulationszustand eines *smart grid*, bestehend aus einem *Power Management System (PMS)*, nach IEC TC65, gekoppelt mit einem Smart Manufacturing System (SMS), nach IEC TC65. Für die Kopplung der beiden Systeme wird ein weiteres System, das Übertragungsnetzwerk (Distributed Energy Resources DER) benötigt, das in der schematischen Systemdarstellung mit *System Interface*, nach TC57, bezeichnet wird.

Ergänzt wird die Darstellung im grünen Bereich, wo die Spezifikationen der Normen aus den technischen Komitees TC57 und TC65, mit informalen, semiformalen und formalen Sprachen spezifiziert werden sollen.

Das gekoppelte Gesamtsystem ($PMS \mid \mid DER \mid \mid SMS$) ist Gegenstand der Analyse in der blauen Domäne und Gegenstand der Beschreibung und Normung in der grünen Domäne. Die Semantik des Gesamtsystems entsteht durch Angabe der drei Morphismen zwischen den semiotischen Domänen. Ein Morphismus ordnet Artefakte aus verschiedenen Domänen einander zu, sodass ein bestimmtes Symbol ein bestimmtes Konzept repräsentiert, ein Konzept ein Phänomen definiert und ein Phänomen durch ein bestimmtes Zeichen symbolisiert wird. Alle drei Morphismen sind für ‚eine‘ eindeutig beschreibbare Semantik eines technischen Systems nötig. Die Ausführbarkeit und damit die Analysefähigkeit wird nicht durch eine formale Sprache gewährleistet, sondern durch eine operationalisierbare Form der eindeutigen mathematischen Semantik.

Es wird also nicht die Norm simuliert, sondern das Modell der Norm. Die Verbindung zwischen beiden ist der Morphismus zwischen Symbol und Konzept in Bild 1 und 10.

Ein weiteres Beispiel: In Bild B.4 wird das Semiotische Dreieck mit seinen Beziehungen anhand eines einfachen Beispiels einer Heizung erklärt. Das (blaue) Konzept von Heizungen stellen die thermodynamischen Differentialgleichungen dar. Die sich daraus ergebende veränderliche Wärmekapazität eines Körpers c beschreibt das Differential (vgl. Konzept der kontinuierlichen Variable) der zugeführten Wärme dQ zur Temperaturerhöhung dT . Um eine gut funktionierende (graue) Heizung zu bekommen, müssen diese nicht-linearen Differentialgleichungen nur für einen linearisierbaren Bereich mittels PID-Regler implementiert werden.

Die Anleitung zur Implementierung enthält der in Bild B.4 eingetragene ‚bezieht-sich-auf‘ Morphismus. Für die symbolische (grüne) Beschreibung würde das Wort ‚Wärmekapazität‘ reichen, was mit dem ‚steht-für‘-Morphismus erfüllt wird. Ein semantischer Standard jedoch enthält i.d.R. weitere Anforderungen an das Ding einer Heizung, die für die Formung des Begriffs unbedingt erforderlich sind. Diese begrifflichen Auslegungen und Ergänzung werden im ‚erweckt‘-Morphismus beschrieben.

Zusammenfassend kann man sagen, die drei kolorierten semiotischen Ecken werden durch semiotische Agenten bzw. durch die *Stakeholder Normer, Analyst und Ingenieur*, typischerweise repräsentiert. So definiert der (grüne) Normer alle Anforderungen, die das Ding erfüllen muss und wofür Konzepte anzuwenden sind. Der (blauer) Analyst, der u.U. mit dem Digitalen Zwilling gleichgesetzt werden kann, führt in nahezu Realzeit Validierungen des Verhaltens durch. Der Analyst alias Digitaler Zwilling kann auch im *Lebenszyklus* eines Produkts zur Verbesserung des Produkts mittels rückgeführter Evaluationsdaten, in langen Zeitkonstanten, beitragen. Der (graue) Ingenieur übernimmt die technische Realisierung der Phänomene oder die Überwachung der Herstellung von Produkten. Alle drei semiotischen Stakeholder teilen sich Konzept, Norm und Ding in einer einheitlichen semantischen Darstellung des Verhaltens und der Daten eines betrachteten Systems oder Komponente.

Die Morphismen garantieren, dass sich Beschreibung, Vorschriften und Implementierungen auf die gleichen Konzepte stützen, also eine gemeinsame Semantik haben, die auch an andere *Stakeholder* in anderen Kontexten übertragen werden können, vorausgesetzt die *Stakeholder* teilen das gleiche semiotische Dreieck mit seinen konkreten Morphismen, miteinander.

1 Anwendungsbereich

Diese VDE SPEC richtet sich an Softwarehersteller von I4.0-Systemen, welche eine Anleitung zur Erstellung von Software, aber auch zur Verarbeitung von technischer Dokumentation, benötigen. Weiterhin richtet sich die VDE SPEC an technische Redakteure bzw. Ersteller, Bearbeiter und Nutzer von technischen Informationen. Die in der VDE SPEC präsentierten Methoden und Werkzeuge sollen ein gemeinsames Verständnis für eine einheitliche Interpretation technischer Dokumentation schaffen.

ANMERKUNG: Die VDE SPEC erfüllt folgende der Kernanforderungen in ihrer Konzeption und Anwendung:

- 1) Sie beruht auf dem Konzept des semiotischen Dreiecks.

Bei einem **semiotischen Dreieck** handelt es sich um ein, in der Linguistik und Semiotik verwendetes Modell, das in erster Linie die Dreiecksbeziehung zwischen dem betrachteten Gegenstand, seinem Konzept und seiner Benennung darstellt. Durch die konsequente Anwendung der hier beschriebenen Methoden und Werkzeuge und des auf Basis des semiotischen Dreieckskonzepts abgeleiteten **Metamodells**, kann eine gemeinsame Interpretation von (Normungs-)Texten, von beobachtbaren Ereignissen und Aktionen in Systemen oder von formalen Darstellungen, vorgenommen werden. Formale Darstellungen, hier graph-theoretische Darstellungen, werden gebraucht zur Analyse von Beobachtungen oder zur Zuweisung von Semantik zu Symbolen oder Semantik zu Dingen (mittels sogenannter Morphismen).

BEISPIEL: Die in der VDE SPEC definierten semiotischen Konzepte können beispielsweise als Kontrollkomponenten in Form eines semantisch-mathematisches Modells, ggf. ausgeführt von einem Digitalen Zwilling, eingesetzt werden. Das Modell, das der Digitale Zwilling ausführt, erlaubt die Beobachtung oder die Korrektur am konkreten System, einem Asset oder auch seiner Anforderungen, mittels Asset-Feedback. Dadurch wird in Metadaten-basierten Softwaresystemen, sowie in datengetriebenen oder KI- (Künstliche Intelligenz) basierten Prozessen die durchgehende **Provenienz von Metadaten** sichergestellt und somit die Informationsqualität von Metadaten signifikant gesteigert.

HINWEIS: In der VDE SPEC wird ein ‚digitales Modell‘ in einer mathematisch operationalen Semantik konzipiert. Damit stellt es konzeptionell das ‚digitale alter-ego‘ seines technischen Gegenstücks aus der cyber-physischen Realität, Ding oder Asset genannt, dar. Während die sogenannte Realität einer Produktionsanlage aus informationsverarbeitender und operativer Technik (IT/OT), kombiniert mit Automatisierungs- und Kontrolltechnik (A/CT) besteht, besteht die ‚semantische Realität‘ eines digitalen Modells aus einer analyse-fähigen formalen symbolisch interpretierbaren Notation oder Darstellung, z. B. mit graph-theoretischen Artefakten.

- 2) Diese VDE SPEC enthält somit grundlegende Anforderungen an die schriftliche Darstellung von technischen Normungstexten, welche eine Transformation in eine Darstellung mit formaler mathematischer oder operationaler Semantik, mittels semiotischer Zuordnungen, sogenannter Morphismen, erlauben.

Die VDE SPEC definiert ferner auch semantische Anforderungen an eine Norm, indem sie für das Schreiben einer sogenannten semantischen Norm, die Existenz der drei semiotischen Morphismen fordert. Ein **Morphismus** ist eine formale Zuordnung von beschreibenden symbolischen Artefakten zu deklarativen semantischen Artefakten oder zu implementierenden technischen Artefakten.

Zum Beispiel werden Ausdrücke und Sätze einer Sprache zur Beschreibung der Charakteristika eines betrachteten Assets, Elementen aus der operationalen Graphentheorie oder anderen algebraischen Elementen zugeordnet. Weitere Beispiele von Zuordnungen sind diejenigen, die symbolische Artefakte technischen Artefakten einer Maschine oder Fabrik oder semantische Artefakte technischen Artefakten zuordnen.

- 3) In der VDE SPEC stehen die semiotischen Domänen: normative Texte, semantische ausführbare Modelle und technische Lösung untrennbar miteinander in Bezug. Man kann sagen, Semantik wird erst mit der Definition der drei Morphismen zwischen den drei semiotischen Domänen definiert.

Ein ‚**ausführbarer semantischer Standard**‘ ist somit nicht direkt ausführbar, sondern seine Ausführbarkeit wird indirekt, über die Zuordnungen von Symbolen und Texten aus der ontologischen Domäne zu operationalen semantischen Artefakten aus der semantischen Domäne, definiert.

Das operationale semantische Modell ist ausführbar, nicht der beschreibende Standard. Jedoch kann auf syntaktischer Ebene dem Text eines Standards insbesondere eine Beschreibung von Daten und statischen Strukturen in rechner-verarbeitbarer Form hinzugefügt werden. Letzteres ergänzt die Kommunikationsfähigkeit von Maschinen sehr sinnvoll, aber die Ausführung dynamischer Prozesse und Vorgänge bleibt weiterhin einem operationalen semantischen Modell vorbehalten, das in der Lage ist, aktuelle Zustände eines technischen Systems zeitgleich zu evaluieren.

Das semiotische Dreieck stellt somit eine konkrete Verfahrensbeschreibung, d. h. eine Anleitung zur Morphismenbildung zwischen semiotischen Domänen dar, welche die Grundlagen (Motivation und Theorie) für **semantische Normen** bilden. Darüber hinaus ergibt sich aus dem operationalen graph-theoretischen Ansatz, im Gegensatz zur mathematischen Darstellung eines Sachverhalts z. B. als nichtlineare Differentialgleichung (DGL), eine praktikable Lösung zur Simulation des Verhaltens eines technischen Geräts oder einer Maschine.

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

DIN EN 61000-6-2 (VDE 0839-6-2):2016-05, *Elektromagnetische Verträglichkeit (EMV) – Teil 6-2: Fachgrundnormen – Störfestigkeit für Industriebereiche (IEC 77/488/CDV:2015); Deutsche Fassung FprEN 61000-6-2:2015*

DIN EN 61508-1 (VDE 0803-1):2011-02, *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme – Teil 1: Allgemeine Anforderungen (IEC 61508-1:2010); Deutsche Fassung EN 61508-1:2010*

DIN EN ISO 12944:2017-11, *Beschichtungsstoffe – Korrosionsschutz von Stahlbauten durch Beschichtungs-systeme – Teil 1: Allgemeine Einleitung.*

IEC TR 63283-1:2022-03, *Industrial-process measurement, control and automation - Smart manufacturing - Part 1: Terms and definitions*

ISO/IEC 24707:2018, *Information technology – Common Logic (CL) – A framework for a family of logic-based languages*

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach ISO/IEC 24707:2018, *Information technology – Common Logic (CL) – A framework for a family of logic-based languages*, sowie, im Rahmen der VDE SPEC, die neuen semiotischen Begriffe¹.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- ISO Online Browsing Plattform: verfügbar unter <http://www.iso.org/obp>
- IEC Electropedia: verfügbar unter <http://www.electropedia.org>

DIN und DKE stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- DIN-TERMinologieportal: verfügbar unter <https://www.din.de/go/din-term>
- DKE-IEV: verfügbar unter <http://www.dke.de/DKE-IEV>

3.1.1

Abstract Data Type

<Dokument> mathematische Kategorie für datenbezogene semantische Artefakte wie Mengen, Variablen, Operationen, Axiome und Regeln.

Anmerkung 1 zum Begriff: Zusammenfassung der ADT-Theorie ist in Anhang A enthalten.

¹ Mit neuen ‚semiotischen Begriffen‘ werden alle Definitionen zusammengefasst, die sich auf die Grundkonzepte des Semiotischen Dreiecks beziehen. Im Wesentlichen sind das die semiotischen Domänen wie ‚Symbol – ontologische Beschreibung‘, ‚Semantische Darstellung‘ und ‚Engineering von Geräten und Maschinen (Assets)‘ und die Zuordnungsvorschriften, Morphismen genannt, zwischen den Artefakten der drei semiotischen Domänen. Erst mit den Zuordnungen entsteht die intendierte Semantik.

3.1.2

Artefakt

<Dokument> domänen-spezifisches Ausdrucksmittel, das sich in die drei semiotischen Kategorien ontologische, semantische und technische Artefakte aufteilt.

3.1.3

Deklaration, deklarativer Stil

<Dokument> vermeidet **Algorithmen**, stattdessen werden Sorten, Fakten und charakteristische Eigenschaften in Form von Regeln, Variablen und Invarianten deklariert.

Anmerkung 1 zum Begriff: Deklarativer und algorithmischer Stil sind Gegensätze. Deklariert werden Objekte, Axiome und Regeln ohne Angaben zur Ausführung ihrer Reihenfolge, also ohne Angabe eines Algorithmus. Für eine mathematische Semantik wird der axiomatische deklarative Stil bevorzugt.

Anmerkung 2 zum Begriff: ADTs werden deklariert, indem ihre Signatur Σ (formale Syntax) und ihre charakteristischen Eigenschaften (formale Semantik) in Form von Σ -Umschreiberegeln (Σ -rewrite rules) spezifiziert werden.

Anmerkung 3 zum Begriff: Prozesse und Ereignisse werden deklariert, indem *head*- und *tail*-Signaturen (gerichtetes Paar von Graph Vertices) des möglichen Ereignisses deklariert werden.

3.1.4

Daten

<Dokument> **Objekte**, die mindestens eine Sorte mit oder ohne Variablen und eine Typbeschreibung mit Axiomen haben, d. h. es gibt einfache oder zusammengesetzte Datentypen, die in ihren axiomatischen Eigenschaften nicht veränderbar, aber als Datentypdefinition transferierbar sind;

Anmerkung 1 zum Begriff: Daten und Informationen sind nicht synonym! Während Daten unabhängige statische Objekte sind, die kommuniziert werden können, entsteht Information dynamisch durch Beobachtung eines Ereignisses. Das ist kompatibel mit der Informationstheorie.

3.1.5

Typgraph

<Dokument> ungerichteter Graph, der statische Strukturen von Daten- oder Wissensbeziehungen zwischen interagierenden Agenten darstellt.

Anmerkung 1 zum Begriff: Ein Beispiel für einen angerichteten Graphen ist das *Copyright Agreement* zwischen den Agenten Autor und Verlag in Bild 6. Graphen können Relationen zwischen statischen Datenstrukturen oder zwischen dynamischen Vorgängen und Prozessen darstellen. Statische Relationen werden mit einem ungerichteten Typgraphen, dynamische Relation werden mit einem gerichteten Digraphen repräsentiert.

Anmerkung 2 zum Begriff: Die Interpretationen eines Graphen als Typ- oder Inzidenzgraph repräsentieren an dieser Stelle die beiden Anwendungsoptionen graphentheoretischer Darstellungen in den Formaten ungerichteter und gerichteter Graph. Während Datenstruktur-Relationen als ungerichtete Graphkanten dargestellt werden, werden Ereignisrelationen als Ursache und Wirkungs-Beziehungen als gerichtete Kante oder axiomatisch als Wissensimplikation in der Form ‚*cause* \rightarrow *effect*‘, dargestellt. Die Implikation ist gleichbedeutend mit der Darstellung eines Ereignisses.

3.1.6

dot-Notation (kontinuierliche Variablen)

<Dokument> diskrete Variablen, die sich zeit-kontinuierlich ($\Delta/\Delta t$) verändern, also diskrete Funktionen darstellen, die invarianten Bedingungen, z. B. Ober- und Untergrenzen, welche mittels Umschreibungsregeln überprüft werden, unterliegen, bzw. von den Prozessen eingeschränkt werden.

BEISPIEL 1 ‚dot-Notation‘ in der Anwendung. Die *dot-Notation* wird im Smart Grid verwendet für die kontinuierliche Überprüfung der Stabilitätskriterien, wie in den Gleichungen (f 9) bis (f 14) formal dargestellt.

3.1.7

Inzidenzgraph

<Dokument> Erzeugung von Ereignissen, die als gerichtete Kanten in einem Graphen mit ungerichteten Kanten markiert werden.

Anmerkung 1 zum Begriff: Ereignisse haben probabilistische Eigenschaften. Daher sind ihre Ergebnisse vor dem Eintreten des Ereignisses unbekannt und ein Inzidenz-Graph stellt einen Prozess dar, der dynamische Ereignisse (d. h. Interoperabilitäten) zwischen Systemkomponenten beschreibt.

3.1.8

Graph (gerichtet, ungerichtet)

<Dokument> Zusammengesetztes semantisches Artefakt, welches das technische System, die Komponenten, Prozesse (architekturell und prozessual) repräsentiert, wobei die architekturelle Darstellung mit dem ungerichteten Graphen (siehe Def. Anhang A) und die prozessuale Darstellung mit dem gerichteten Graphen (siehe Def. Anhang A), gegeben ist

Anmerkung 1 zum Begriff: Während der **ungerichtete Graph** auch als **Typ-Graph** (die Systemstrukturen, bzw. Systemarchitekturen repräsentierend) bezeichnet wird, wird der **gerichtete Inzidenz-Graph** als ‚**Ereignis- oder Prozess-Graph**‘ (den Systemprozesse repräsentierend) bezeichnet.

Anmerkung 2 zum Begriff: Der Typ-Graph beschreibt Relationen (Architekturen) zwischen statischen Datenobjekten, z. B. ADTs. Der Inzidenz-Graph beschreibt Relationen (Interoperabilität an Schnittstellen) zwischen dynamischen Prozessen und ihren Variablen. Während der Typ-Graph ungerichtete Kanten hat, hat der Inzidenz-Graph gerichtete Kanten. In einem kombinierten Daten-Ereignis-Graph können den Variablen in den Systemknoten Datenarchitekturen zugeordnet werden.

3.1.9

Gradient

<Dokument> **minimale Trajektorie** (minimaler Pfad) durch einen gerichteten, Prozess-Graphen, erzeugt von der Inzidenzabbildung (siehe 3.2.12)

Anmerkung 1 zum Begriff: Da ein Gradient eine Trajektorie mit bestimmtem Ziel darstellt, kann man die Ereignisse des Gradienten auch als Folge, von mit anderen Prozessen, synchronisierten Operationen, mit dem Ziel einen bestimmten Systemzustand zu erreichen, begreifen.

Anmerkung 2 zum Begriff: Der Gradient eines Graphen kann als minimales sequenzielles Programm mit einem bestimmten Ziel, verstanden werden womit sich andere Graphen, bzw. Programme aus anderen Domänen, synchronisieren können. Ein Gradient oder *Subgraph* fungiert also auch als Schnittstelle zwischen Anwendungen. (Beispiel: Wetterbedingungen kooperieren mit Solar- und Windanlagen, indem sie die Energie zwischen verschiedenen Zuständen transformieren.)

3.1.10

Information

<Dokument> erhält man durch **Beobachtung** von probabilistischen Ereignissen (in einem Graphen, erzeugt durch Inzidenz-Abbildungen) mit unbekanntem Ergebnis. Je unwahrscheinlicher das Ergebnis, desto größer die Information. Wenn sich nichts ereignet, gibt es auch keine Informationen.

Aus der Beobachtung des ‚Ereignis->Ergebnis‘ -Paars bzw. des ‚Ursache->Wirkung‘ - Paars ergeben sich Informationen, die in geeigneten Datenformate dargestellt werden können.

3.1.11

Interoperabilität

<Dokument> potenziellen Ereignisse an Schnittstellen zwischen Entitäten eines Netzwerks, die semantisch dargestellt werden als ungerichtete Paare von Graphknoten (*vertices*) mit den Signaturen: $\{\sum_{head}, \sum_{tail}\}$ und eigenen *head*- und *tail*-Variablen, deren Interoperabilität ggf. konvergiert, wenn z. B. kinetische in mechanische in elektrische Energiekontingente umgewandelt wird.

Anmerkung 1 zum Begriff: Eine ökonomische Betrachtung bewertet die Kosten der Interoperabilität bzw. die Anpassung an kontext-spezifische Variablen (**coercion**), z. B. von kinetischer in mechanische Energie oder zurück.

3.1.12

Inzidenz-Abbildung

<Dokument> Ereignisse, bestehend aus Paaren von geordneten Graphknoten (Vertices), die als gerichtete Kanten in den **Inzidenzgraphen** (siehe 3.2.5) eingetragen werden.

3.1.13

Narration (narrativer Stil)

<Dokument> Geschichte der beobachteten **Interoperabilität** zwischen kontinuierlichen Variablen (Entitäten) disjunkter Kontexte, die interagieren, sobald die tatsächlichen Umgebungsbedingungen der Variablen des einen Kontexts den Erwartungen der Variablen des anderen Kontexts entsprechen

Anmerkung 1 zum Begriff: Ein Digitaler Zwilling, der das Informationsmodell mit allen relevanten Variablen repräsentiert, kann im Modell überprüfen, wie sich Änderungen in den Anwendungskontexten, d. h. Änderungen in den Variablen, auf das Verhalten von Assets auswirken können.

3.1.14

Regelmengen (rewrite rules)

<Dokument> Definitionen für:

- 1) die Semantik eines ADT mit der Signatur Σ
- 2) die Semantik einer Inzidenzabbildung (Ereignis) mit den Signaturen einer gerichteten Graphkante ($\Sigma_{\text{head}} \rightarrow \Sigma_{\text{tail}}$)

3.1.15

Selbsterklärbarkeit

<Dokument> Fähigkeit einer Maschine oder Produktionsanlage, ihren aktuellen Zustand ‚erklären‘ zu können, indem die Maschine ihren semantischen Graphen rückwärts analysiert, also alle vorangegangenen Ereignisse (Graphknoten-Paare) mit dem aktuellen Graphknoten-Endpunkt Σ_{tail} herausucht.

Anmerkung 1 zum Begriff: Die rückwärtsgewandte Suche in einem semantischen Graphen, wird mithilfe von Provenienz-Metadaten, die während des Verlaufs eines Prozesses und der Manipulation von Daten gesammelt werden, vorgenommen.

Anmerkung 2 zum Begriff: Ähnlich kann auch die Zukunft analysiert werden, indem alle künftigen Ereignisse vom neuen Graphknoten-Startpunkt ($\Sigma_{\text{head}} = \Sigma_{\text{tail}}$) analysiert werden.

3.1.16

Signatur (synonym für Syntax)

<Dokument> Syntaktische Schnittstellenbeschreibung eines semantischen Objekts, bzw. eines Konzepts.

Anmerkung 1 zum Begriff: Beispiele sind AAS- Signatur zu einem Asset oder Signaturen von Prozessen, bzw. kontinuierlichen Variablen in ihren Kontexten.

3.1.17

Semantik

<Dokument> mathematisch algebraisches Modell mit einer Signatur Σ , welche die syntaktischen Artefakte der Algebra, bestehend aus Prozess- und Datenstrukturen, definiert.

Anmerkung 1 zum Begriff: So ist die Semantik eines Abstract Data Type (ADT) eine **mehrsortige Algebra**. Die Signatur Σ des ADT erzeugt die Σ -Termalgebra, die isomorph zu ihrer Datentyp Algebra ist. Daher ist ein Σ -Prozess isomorph zu seiner Σ - Algebra.

Anmerkung 2 zum Begriff: Die Semantik eines Systems wird oft auch mit dem Begriff ‚Konzept‘ repräsentiert, ohne auf eine mathematische Darstellung einzugehen. Ein Konzept repräsentiert informell die Bedeutung eines Dings oder Vorgangs.

3.1.18

Formales System

<Dokument> Netzwerk von Funktionsblöcken, bzw. Entitäten, das als Typ- und Inzidenzgraph mit interagierenden Variablen formalisiert wird.

Anmerkung 1 zum Begriff: Der formale Graph beinhaltet Eingangs- und Ausgangsknoten (*Vertices*). Während ein Eingangsknoten nur ausgehende Inzidenzen (*Edges*) hat, besitzt ein Ausgangsknoten nur eingehende Inzidenzen (*Edges*). Ein Graph kann Schleifen beinhalten, die bestimmte Einschränkungen erfüllen müssen, z. B. eine Schleife beginnt und endet im gleichen Knoten.

Anmerkung 2 zum Begriff: Ein Pfad durch einen Graphen ist definiert durch eine Folge von Kanten einer Trajektorie in einer Richtung. Das Netzwerk von Funktionsblöcken, auch Graphenmodell genannt, und das Zustandsdiagramm, abgeleitet aus DGLs können ineinander überführt werden. Beide Darstellungen repräsentieren das gleiche System.

BEISPIEL 2 Ein Motor kann nur in seinem linearen Verhaltensbereich betrieben werden.

Das dynamische Verhalten des Motors ist mit seiner charakteristischen Transferfunktion definiert. Da auch der Regler i.d.R. eine nicht-lineare Charakteristik hat, müssen alle Geräte auf den linearen Bereich eingestellt werden.

Der lineare Aussteuerungsbereich des Motors wird durch das Aus- Eingabeverhältnis der Laplace Transformierten $V_a(s)/V_i(s)$, eines Reglers K^2 bestimmt.

3.1.19

Technisches System

<Dokument> Netzwerk von interagierenden regelungstechnischen Funktionsblöcken, mit einem Vektor R der Referenzparameter und einem Vektor Y der Ergebnisparameter. Ausgangsvektor Y und Eingangsvektor R stehen über einen Transferfunktionsblock miteinander in Bezug. Die Transferfunktion repräsentiert das Gerät bzw. die Maschine, welche gesteuert bzw. geregelt wird.

BEISPIEL 3 Das Drehmoment Y eines Motors, wird über das Referenzsignal R angesteuert.

Diese Steuerung kann mit einer Rückführung von Y nach R ergänzt werden, sodass das gewünschte Drehmoment auch bei Belastung konstant bleibt.

3.1.20

Trajektorie

<Dokument> Pfad durch einen gerichteten Ereignis-, bzw. Inzidenz-Graphen mit sequenziell ausführbaren Kanten.

Anmerkung 1 zum Begriff: Axiome und Invarianten in allen Knoten bzw. Anwendungskontexten, geprüft und eingehalten werden können. Eine gerichtete oder ungerichtete Kante wird entweder als geordnetes oder ungeordnetes Knotenpaar notiert. Graphen mit ungerichteten Kanten stellen Optionen für Trajektorien dar, während Graphen mit gerichteten Kanten beobachtete Ereignisfolgen darstellen.

3.1.21

Wissen, ~s-Pyramide, (*knowledge*)

<Dokument> Fähigkeit zwischen **Ursache und Wirkung** zu unterscheiden, um damit einen Vorgang erklären zu können.

Anmerkung 1 zum Begriff: Vgl. Selbsterklärbarkeit von Maschinen.

3.2 Abkürzungen

Abkürzung	Bezeichnung
SemNorm	Semantisch Norm (im Sinne dieser VDE SPEC)
AAS	Asset Administration Shell, [nach IEC TC65 WG24].
ADT	Abstract Data Type (siehe Begriff ‚Term-Algebra‘ im Anhang A)
AI	Artificial Intelligence [nach ISO/IEC JTC1/SC42]
BI	Business Intelligence [nach OLAP.org]
CL	Common Logic [nach ISO/IEC 24707:2018]
CPS	Cyber-Physische Systeme
CL/CS	Common Logic (syn. Common Semantics)
DER	Distributed Energy Resources [nach IEC TC57 IEC 61850-7]
DM/DF	Digital Manufacturing, Digital Factory
DGL	Differentialgleichung

² Die Laplace Theorie bzw. ‚Laplace Transformierte‘ wird im *SemNorm*-Ansatz nur insofern bemüht, um die algebraische Berechenbarkeit hervorzuheben, wie z. B. die Transformation einer DGL in einen Zustandsgraphen.

Abkürzung	Bezeichnung
DL	Description Logic (W3C)
DT, DTw	Digital Twin, (Digitaler Zwilling)
FOL	First Order Logic
IACS	Industrial Automation and Control System, (Industrielles automatisiertes und geregeltes System) [nach IEC 62443]
IIoT	Industrial Internet of Things, [nach JTC1/SC41]
HMI	Human-Machine, Mensch-Maschine Interaktionen/Schnittstelle
KI	Künstliche Intelligenz
LC	Lebenszyklus im RAMI4.0 Modell
MMI	Maschine-Maschine, Interaktionen/Schnittstelle
ML	Machine Learning
OT/IT/CT	Operations- (Objekt-verarbeitende) Technik / Informations- (Daten-verarbeitende) Technik, Control (Signal-verarbeitende) Technik
OWL	Web Ontology Language (W3C)
PMS	Grid Power Management System [nach TC57]
SMS	Smart Manufacturing System [nach TC65]
SE	Smart Energy

4 Der SemNorm Ansatz

4.1 Das Semiotische Dreieck

Der SemNorm Ansatz ist inspiriert von der Idee des Semiotischen Dreiecks, das schon von Platon und Sokrates für die Erklärung [Wiki-Eintrag: https://de.wikipedia.org/wiki/Semiotisches_Dreieck#Geschichte] der Zusammenhänge zwischen Symbol, Vorstellung und Ding, verwendet worden ist. In dieser VDE SPEC wird das Semiotische Dreieck als Darstellung einer „**triadischen Relation**“ **im industriellen Bereich** der automatisierten Produktion zwischen Standard-Texten, der semantischen ‚Blaupause‘ und den realen Dingen, d. h. industrielle Produktionsstätten dahinter, verwendet (siehe Bild B.4), wobei die semantische Blaupause aus einem Computer simulationsfähigen Format besteht, das ‚Ding‘ die normativen Anforderungen der Standards, und die konstruktiven Anforderungen der Blaupause erfüllt. Diese triadischen Relationen heißen Morphismen.

Die semantische Blaupause, die ja den Vorstellungen über Dinge entspricht, also semantikgebend wirkt, wird auf der Grundlage der algebraischen Mathematik formalisiert bzw. computerisiert. Eine Algebra stellt das mathematische Modell von Prozessen in industriellen Automatisierungs- und Kontrollsystemen (IACS) dar. Im operationalen Modus auf Basis der algebraischen Graph- und Datentyptheorien können diese IACS-Prozesse von einem Digitalen Zwilling analysiert werden, wobei das Wissen über einen industriellen Vorgang mittels Systemvariablen und Ursache-Wirkungs-Relationen in der Blaupause dargestellt wird und daher auch maschinell interpretiert werden kann.

Im Semiotischen Dreieck (siehe Bild 1 und Bild 10) wird dieser Zusammenhang zwischen den Texten von Normungsdokumenten (in Bild 1 grün markiert), dem maschineninterpretierbaren semantischen Modell (in Bild 1 blau markiert) und der realisierten Maschine (in Bild 1 grau markiert) hergestellt,

indem folgende grundsätzliche Best Practices für eine semantische Norm, die sich an den drei Gestaltungsrollen semantische Normung (grün), validiertes Verhalten (blau) und sichere Produktion (grau) orientieren, angewendet werden:

- 1) Best Practices für den **Menschen und Normer** (in der grünen Domäne) lautet, ein beschreibendes Standard-Dokument herzustellen, dessen normative Konzepte sich auf formale mathematische Artefakte zurückführen lassen, was gleichbedeutend ist, Texte semantisch zu verstehen, um Wissen über die zu gestaltenden Dinge zu generieren und semantisch korrekt darzustellen. Diese Darstellung wird **Semantischer Standard** genannt.
- 2) Best Practices für den **sogenannten Digitale Zwilling bzw. Digital Twin** (in der blauen Domäne), z. B. von einer Smart Manufacturing Anlage lautet, eine computer-interpretierbare Blaupause (ein Modell) zu haben, dessen normative Konzepte, sich digital ausführen, d. h. semantisch analysieren lassen, um Fehlerursachen (in der Vergangenheit) zu begründen oder Vorhersagen von möglichem Verhalten (in der Zukunft) der Produktionsanlage vorzunehmen und zu bewerten.
- 3) Best Practices für **das Ding, die maschinelle Anlage** (in der grauen Domäne), lautet, einen sogenannten Semantischen Standard (siehe Punkt 1) zu haben, dessen normative Konzepte, sich über eine ‚Asset Administration Shell‘ verwalten, d. h. semantisch administrieren lassen. Zu diesem Zweck ist der Digitale Zwilling eng mit der maschinellen Anlage verbunden, d. h. er spiegelt das Ding samt seinem Verhalten, indem er eine Zustandsanalyse in Realzeit ausführt und die Ergebnisse dieser Analyse in die maschinelle Welt der Dinge zurückführt, wo sie ggf. in eine Korrektur oder in Fortschritt in einen bestätigten sicheren Zustand umgesetzt werden.

Diese drei Zuordnungen werden Morphismen genannt. Sie finden sich im semiotischen Dreieck wieder und werden dort definiert als Beziehungen zwischen Artefakten der Normungs-Domäne (Ontologie), der Semantik-Domäne (Semantik) und der Asset-Engineering-Domäne (Implementierung, Realisierung) untereinander.

In der Semiotik stellen die Morphismen eine Zuordnung zwischen Artefakten aus den drei **Gestaltungs-Domänen der Normung, der Semantik und dem Engineering** dar, wobei in der Normungs-Domäne sprachliche, beschreibend definierende Artefakte, in der semantischen Domäne deklarative axiomatische Artefakte und in der Engineering Domäne technische Artefakte, verwendet werden.

Die Semantik der Produktion von Dingen bzw. die semantischen Veränderungen im Lebenszyklus von Produkten bedeuten die vollständige Angabe der drei Morphismen zwischen textueller sprachlicher Norm, deklarativer Definition (Semantik) und technischer Realisierung mittels OT/IT³.

Der sogenannte Digitale Zwilling ist ein Modell, das aus semantischen Artefakten konfiguriert wird, wobei Teile oder u.U. das gesamte Modell eine operationale Semantik haben, um maschinell interpretiert werden zu können. Mit geeigneten Werkzeugen kann die realisierte technische Produktion, mit vorausschauender, ergebnisorientierter als auch mit rückschauender, ursachenorientierter Analyse, auch parallel zum Produktionsbetrieb, unterstützt werden.

ANMERKUNG 1 Semantisches Modell bzw. Digitaler Zwilling, Normungstext und technisches Asset stehen in semiotischer Beziehung zueinander: Sie sind ähnliche, semantisch vergleichbare Repräsentanten, Artefakte, aber aus drei verschiedenen Domänen.

ANMERKUNG 2 Die algebraische Ähnlichkeit zwischen den semantischen Repräsentationen eines Abstract Data Type (ADT) und eines Prozess-Graphen gestattet es, sich in beiden Fällen auf die erzeugten *Termalgebra* (siehe Definition Anhang A.2) zu stützen. Die Termalgebra dient als Ausdrucksmittel, d. h. als eine Sprache zur Darstellung der Interoperabilität zwischen Prozessen und den Charakteristika von Daten. Die Fähigkeiten eines Assets bzw. Dings, werden durch sprachliche Ausdrücke (im Standard) repräsentiert, wobei die Grammatik der Sprache ‚operationserzeugte‘ algebraische Eigenschaften liefert und durch die Signatur der Termalgebra syntaktisch definiert ist.

³ Cyber-Physikalische Systeme (CPS), Smart Grid z. B., werden hierbei verstanden als technische Systeme, die aus der Kombination von **Operativer industrieller Fertigungs-Technik (OT)** und **Informations-Technik (IT)** bestehen und sich ggf. zu einem Smart Manufacturing (SM) System oder auch zu einem industriellem ‚Internet der Dinge (IIoT)‘ zusammenfügen.

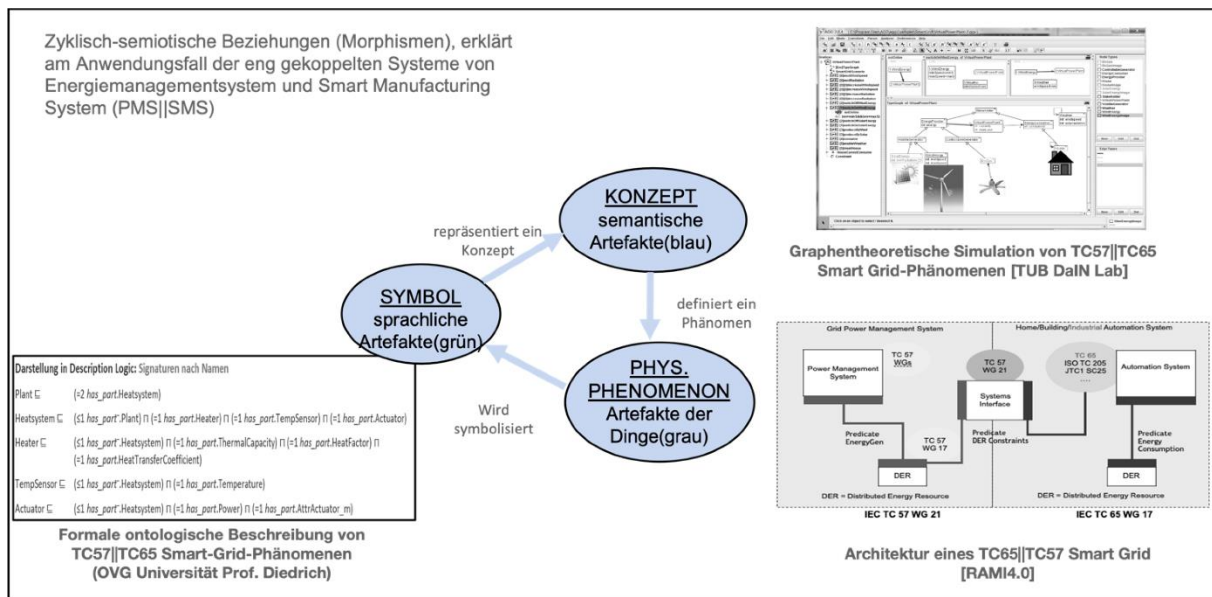


Bild 1 – Semiotisches Dreieck mit semiotischen Domänen und Morphismen

4.2 Die Bedeutung von Morphismen

Morphismen sind der wesentliche Bestandteil einer formalen Semantik, denn damit werden Artefakte einer der drei oben eingeführten semiotischen Domänen jeweils Artefakte einer der beiden anderen Domänen strukturverträglich zugeordnet.

Ein Beispiel für einen Morphismus, ist die Σ -Term Algebra, der jedem syntaktischen Term ein Element der semantischen Algebra zugeordnet ist, d. h. die intendierte semantische Algebra ist isomorph zur textuellen Σ -Algebra, wobei Σ die syntaktische Struktur der Algebra definiert. Im semiotischen Dreieck heißt das, dass die Σ -Terme, d. h. Symbole, den computer-interpretierbaren Graphen, d. h. Begriffe, und ebenso den technischen Artefakten, d. h. Dinge, zugeordnet werden.

Die Syntax Σ stellt die Struktur des gewünschten Datentyps mit algebraischer Semantik dar. Jede Datentypalgebra besteht aus drei definierenden Artefakten: Datenmengen mit Variablen, Sorten genannt, Funktionen auf den Datenmengen, Operationen genannt und Axiomen, Term-Ersetzungsregeln genannt. Mit diesen drei syntaktischen Elementen wird die Semantik, d. h. es werden die Axiome des Datentyps deklariert.

Aus semantischer Sicht wird ein technisches System als Menge von getypten Systemvariablen verstanden und deklariert. Die Systemvariablen stehen miteinander in Bezug und bilden so ein Netzwerk interagierender verteilter Prozesse. Ein Netzwerk von Variablen zur Beschreibung eines komplexen technischen Systems kann leicht in ein Graphennetzwerk, überführt und aufgrund ihrer algebraischen Eigenschaften simuliert werden.

In einem Morphismus wird ein textueller Ausdruck aus einem Standard, der ein oder mehrere Artefakte aus der beschreibenden Ontologie beinhaltet, in der semantischen Domäne mit mathematischen Artefakten implementiert, die die intendierte Semantik des ontologischen Ausdrucks (bzw. des Terms im Standard) 'möglichst adäquat', d. h. strukturverträglich, repräsentieren.

Ebenso repräsentiert der gleiche ontologische Term Artefakte aus der technisch/physischen (realen) Welt, z. B. einer Produktionsstätte oder ein Energieversorgungssystem. Die Artefakte der technischen Welt des Asset Engineerings stellen Produkte, Maschinen, Dinge und Produktionsdaten dar.

Artefakte aus den drei Domänen, textuelle Ausdrücke bzw. Terme (grün), technisch-konstruktive Elemente (grau) und computerinterpretierbare Begriffe (blau) müssen semantisch korrekt einander zugeordnet werden können, sodass das analysierte Verhalten des computerinterpretierten Modells (d. h. des Digitalen Zwillinges), dem Verhalten der zusammengeführten Geräte und Dinge als auch den Anforderungen eines Standards, genügen.

Ein komplexes Beispiel dargestellt als ein semantisches Netzwerk algebraischer Graphen, ist das elektrische Energieversorgungsnetz mit seinen Geräten und Schaltstellen und den in einer Norm geforderten elektrischen Eigenschaften und Charakteristiken der verwendeten technischen Assets.

In einem Energie-Standard z. B. werden Anforderungen an ein Verteilnetzwerk auf der Grundlage von Kenntnissen realer und semantischer Artefakte mit Ausdrücken und Termen beschrieben. In dieser beschreibenden Domäne werden alle notwendigen technischen Artefakte mit Invarianten, Variablen, Funktionen, Axiome, Regeln definiert und mit Graphen kombiniert. Die Graphen repräsentieren die Prozessdynamik

Morphismen, d. h. die Zuordnungen von Artefakten aus verschiedenen Domänen sollten daher unter Beachtung einer kontext-spezifischen Anwendungsregel entstehen, umso mehr Genauigkeit von Zuordnungen von Artefakten aus verschiedenen Domänen zu erreichen.

Zusammenfassend kann man sagen, dass die drei Morphismen zwischen *Standard/Symbol – Technik/Ding – Semantik/Begriff*, die 'eigentliche Semantik' einer maschinen-interpretierbaren Norm darstellen.

4.3 Der Algebraische Stil

Im Semiotischen Dreieck werden die triadischen Elemente und Relationen zwischen semantischen, technischen und deskriptiven Artefakten von der Klammer einer Common Logic (CL) zusammengehalten. In der Literatur wird die Common Logic, mit regulären Ausdrücken und Symbolen zu einer formalen I4.0 Sprache (wie in Bild 3 dargestellt) auf FOL-Basis transformiert.

Die SemNorm Methodik enthält demgegenüber eine formale I4.0 Sprache nur ansatzweise. Die SemNorm Methodik stützt sich begrifflich semantisch auf einen algebraisch operationalen Stil. Statt eines festen Gebildes einer I4.0 Sprache zu nutzen, werden Anwendungsregeln in Form von Morphismen mit dem Ziel angewendet, Verhalten und Architektur der Dinge mit semantischen Konzepten zu erklären und notwendige Vorschriften und Normen davon abzuleiten bzw. in Einklang zu bringen. Es ist das gleiche Ziel, wie mit der Nutzung einer formalen Sprache, doch die Wege dorthin sind verschiedenen.

Ein wichtiger Unterschied besteht darin, dass die Semantik einer formalen Sprache unabänderlich ist, die Semantik-Zuweisungen, Morphismen genannt, aber offen sind, d. h. sie sind Teil der Lösung eines Problems, z. B. wie linearisiert man einen bestimmten Anwendungsbereich einer DGL, die das Problem vollständig beschreibt?

Das gemeinsame Ziel besteht in der standardisierten Darstellung der Anforderungen in Standards, einer operationalen Semantik, die zwischen Anwendungen aus unterschiedlichen Domänen geteilt werden kann und der Anwendung von gesichertem Wissen und Fakten bei der Inbetriebnahme oder im Betrieb eines I4.0 Produktionsprozesses.

4.4 Das SemNorm Narrativ

Das Narrativ des SemNorm-Ansatzes ist in den Bildern (2) und (3) skizziert und beinhaltet die nachfolgenden sechs Anwendungsregeln. Die sequenzielle Anwendung dieser Regeln stellen eine Art formaler Sprache, welche ohne eine bestimmte Grammatik auskommt, dar.

ANMERKUNG 1 Als ergänzende Anwendungsregel könnte ggf. ein Feedback Schritt sein, um die Folgerungen aus dem gesammelten Wissen eines Produktionsprozesses im Lebenszyklus des Produkts zur Verfügung zu stellen. Feedback als siebter Schritt, ist in der Zeichnung nicht eingetragen.

ANMERKUNG 2 Die Feedback Anwendungsregel beinhaltet die Rückführung von erworbenen Kenntnissen (*knowledge*) zur Problembeschreibung und Lösungssuche und steht in Bezug zur fünften Anwendungsregel zur Generierung von Morphismen und der semantischen Modellbildung mit einer vorwärts gerichteten Lernschleife des Kenntniserwerbs über einen betrachteten Produktionsvorgang.

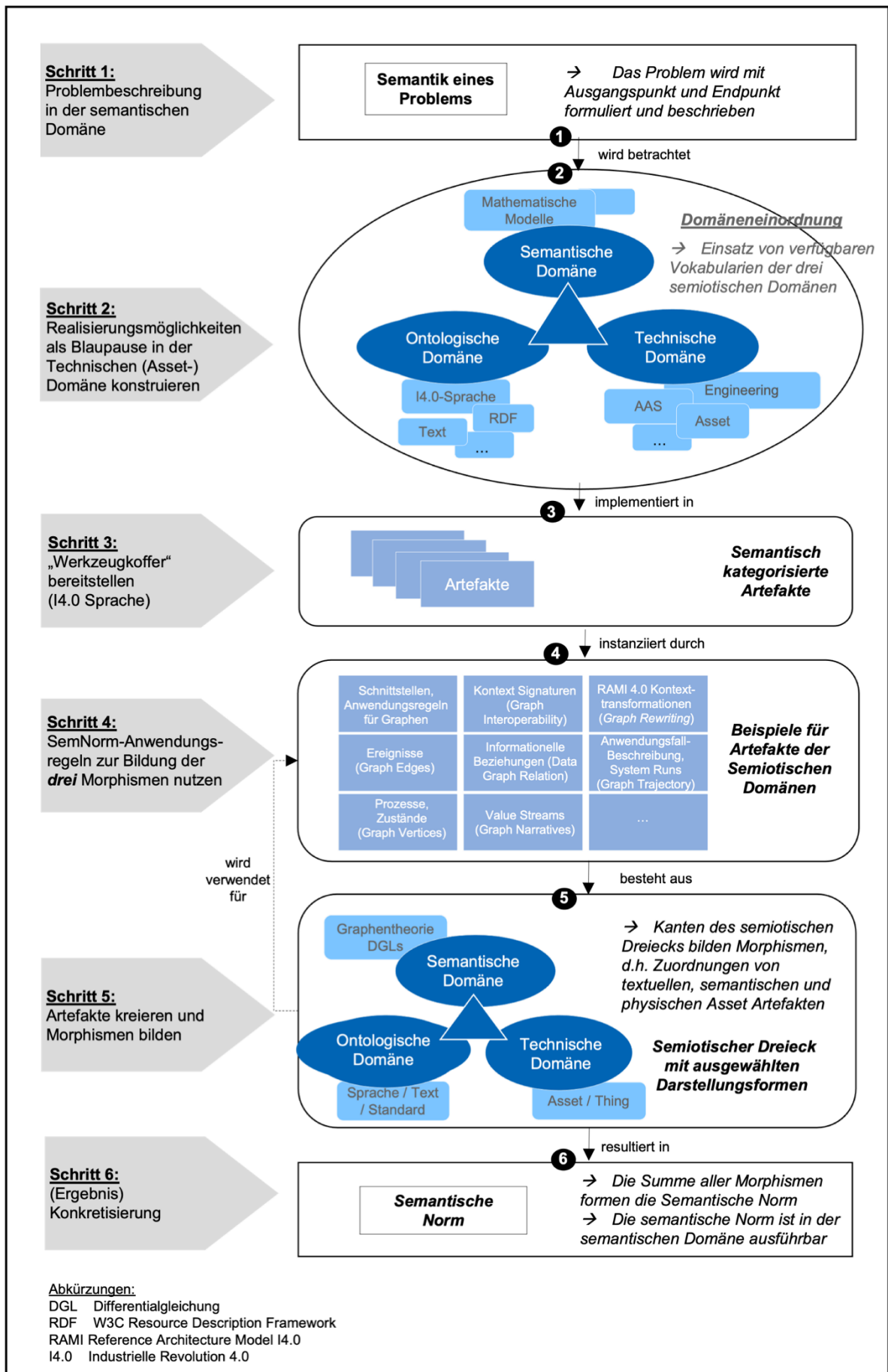


Bild 2 – Narrativ des SemNorm-Ansatzes

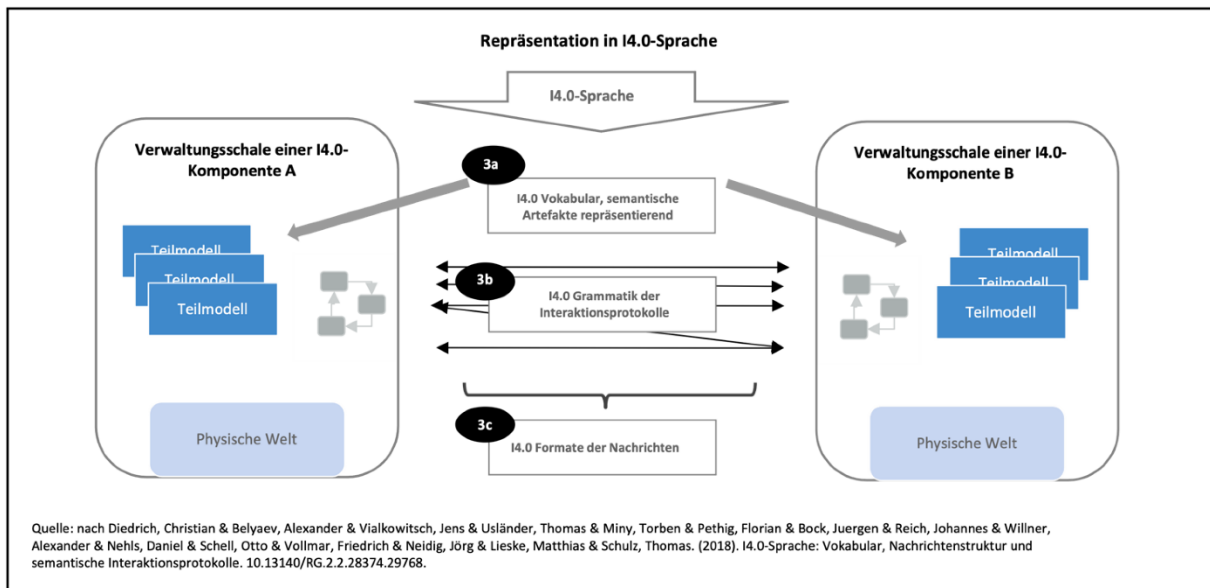


Bild 3 – Anwendung I4.0-Sprache in der VDE SPEC SemNorm

In der Bild 2 ist das SemNorm Narrative vollständig dargestellt und in Bild 3 werden Anforderungen an eine **I4.0 Sprache** formuliert. Eine feste I4.0 Sprache mit Syntax, Semantik und Grammatik hat jedoch den Nachteil, keine flexiblen Morphismen, d. h. Zuordnung von Artefakten aus den drei Domänen des semiotischen Dreiecks kreativ, d. h. kontext-sensitiv a-posteriori für jede Aufgabenstellung vornehmen zu können, sondern eine a-priori definierten Semantik einer I4.0 Sprache, d. h. mit einem kontext-unabhängigen Morphismus nutzen zu müssen.

- 1) Das Narrativ der SemNorm Methodik erzählt die Geschichte, bzw. den Ablauf eines Vorgangs, mit einem Ausgangspunkt, die intendierte Semantik einer spezifischen Aufgabenstellung und einem Endpunkt, die Lösung in einer technischen Anwendungsdomäne (siehe Bild 2) zu finden. Beispiele dafür sind die Energieversorgung über ein Smart Grid, oder die Herstellung eines Heizgeräts. Solch ein Gerät oder System besteht aus den physischen Artefakten der operativen Technologie (OT), der Informationstechnologie (IT) und der Automations- und Kontrolltechnologie (CT), einschließlich Software zur Steuerung und Kontrolle der Heizung, das Asset.

Die physischen Artefakte, die sich selbst repräsentieren, bedürfen aber einer semantischen und symbolischen Erklärung, um sie zu verstehen und Anforderungen, z. B. Safety und Security, an sie stellen zu können. Anforderungen und einheitliche Semantik können in einer I4.0 Sprache, nach Bild 2, mit unveränderlichen syntaktischen, semantischen und grammatikalischen Regeln, wie in Schritt 3 eingeführt, beschrieben werden. Ebenso ist es möglich mit veränderlichen Morphismen zu arbeiten, die jedoch für jeden Anwendungskontext neu definiert werden müssen.

- 2) Jede Anwendungsdomäne hat ihre eigenen besonderen Artefakte. Diese semiotischen Gruppen von Artefakten (z. B. Symbole, Semantik, Technik) werden für eine Anwendung, wie z. B. die Verkehrszeichenerkennung in einem autonomen Fahrzeug, zuerst definiert. Sie bilden die **verfügbaren Vokabularien** der drei semiotischen Domänen. Mit den Vokabularien können Lösungsanforderungen, z. B. technische Standards für bestimmte Produktionsumgebung oder semantische Modelle für die mathematisch simulative Analyse des Verhaltens von Maschinen oder Systeme beschrieben werden.

Eine ‚**Blaupause**‘, die ein technisches System semantisch beschreibt, stützt sich u.a. auf die Erkennung von Signaturen von Datentypen. In diesem Schritt werden die notwendigen Daten-signaturen eines bestimmten Produktionsprozesses definiert.

- 3) Mit dieser rudimentären **I4.0-Sprache der Signaturen** kann bereits die **Interoperabilität** zwischen Artefakten, die gemeinschaftlich eine Aufgabe zu erfüllen haben, beschrieben werden. Die semantische Interoperabilität kann aber erst mit der Definition der Morphismen festgelegt werden, weil damit die Korrespondenz von Artefakten aus verschiedenen Domänen bestimmt wird. Z. B. wird das Symbol ‚Heizung‘ der Semantik eines thermo-dynamischen Systems, beschrieben in einer DGL, zugeordnet (siehe auch Anhang B). Genauso wird dem Symbol

‚Heizung‘ ein Gerät zugeordnet, das den semantischen Thermodynamischen Gesetzen gehorcht. Es gibt also über die zugeordneten Artefakte eine ‚logische Interoperabilität‘ zwischen Symbolen in Standards, Semantik in der Mathematik und Lösungen (Assets) in der Technik.

- 4) Mittels **Morphismen** werden z. B. semantische Artefakte auf technische Artefakte oder symbolische Artefakte auf semantische Artefakte und symbolische Artefakte auf technische Artefakte zugeordnet.

Aufgrund dieser Zuordnungen **entsteht Semantik**, indem z. B. einem Term, bzw. einem sprachlichen Ausdruck (einer I4.0 Sprache), ein technisches Teilmodell, eine bestimmte Lösung oder ein einzelner Knowledge-, Daten- oder Prozess-Graph zugeordnet und damit ‚erklärt‘ wird.

Morphismen erklären, bildlich gesprochen, die Anwendung der Tools im ‚**Werkzeugkoffer**‘. Sie stellen eine Verbindung zwischen den semiotischen Domänen der Sprache bzw. des beschreibenden Texts zu mathematischen Modellen zur Analyse und zu technischen Lösungen (Assets) für eine bestimmte Aufgabenstellung her.

- 5) Das semiotische Dreieck repräsentiert den ‚**Begriff einer Semantischen Norm**‘, weil es die semantischen Zuordnungen zwischen den drei unterschiedlichen Darstellungen in den drei Domänen liefert: beschreibende/normative Ontologie, operationale Semantik und technisch funktionale Problemlösung. Die Verbindungen dazwischen stellen die Morphismen her.

Eine semantische Norm ist nur indirekt ausführbar, weil nicht direkt die Texte der Norm ausgeführt werden, sondern nur indirekt, über Morphismen definierte Semantik, indem Werkzeuge zur Simulation der semantischen Darstellung, z. B. von Knowledge oder Prozessgraphen, genutzt werden können.

- 6) Eine formale I4.0 Sprache, die Morphismen nutzt, vereint deklarative mit operativen Aspekten. Sie kann sowohl zur Aufgaben- und Anforderungsbeschreibung als auch zur Beschreibung der Assets/technischen Lösung verwendet werden. Sie sollte genug Ausdruckstärke haben, um Nutzung von Wissensbasen (ML Pattern Space) und Interaktionsmechanismen eines semantischen Modells verwenden zu können. Das Modell ist darauf angelegt, seine analytischen Fähigkeiten des Verhaltens einer technischen Lösung zu nutzen und die Ergebnisse zurückzuführen.

Zwischen technischer Lösung und Analyse kann zum Zweck der Ursachenforschung von Angriffen oder Fehlverhalten oder zum Zweck der Validierung von neuen Maßnahmen zur Fehlerbehebung zwischen den beiden ‚Entitäten‘ der Technik und der mathematischen Simulation im Modell hin- und hergeschaltet werden. Dieses Verfahren wohnt dem Digitalen Zwilling inne.

Eine technische Lösung mit Digitalem Zwilling kann also in zweierlei Modi gefahren werden, im operativen und im simulativen Modus. Zwischen beiden Modi kann u. U. in Realzeit umgeschaltet werden. Dieser Betriebsmodus ist vergleichbar mit den Bezügen zwischen der algebraischen Laplace Ebene und der Ebene zur Lösung von Differentialgleichungen.

In gleicher Art und Weise kann auch ein Standard validiert werden, indem die Terme eines Standards symbolisch im Simulator ausgeführt werden.

4.5 Das Ziel einer semantischen Norm

In der industriellen Praxis werden sogenannte **Blaupausen** zur Modelldarstellung technischer Anlagen verwendet. Der Begriff der Blaupause soll an dieser Stelle als Synonym für eine semantische Darstellung stehen, womit der Austausch Wissen und Informationen zwischen Systemkomponenten dargestellt werden kann. Eine überprüfbare semantische Darstellung benötigt ein mathematisches Modell, wie es z. B. die Graph-Theorie bietet. Das heißt, es handelt sich bei Graph-Modellen um Mathematik, somit werden solche Modelle für den Menschen ‚denkbar‘, für den Computer ‚berechenbar‘ und für die Maschinen ‚ausführbar‘.

In einem Gedankenexperiment sollen Denkbarkeit, Berechenbarkeit und Ausführbarkeit in Einklang gebracht werden. Der Mensch verleiht seinen Gedanken in Form von (normativen) Standards Ausdruck. Der Computer berechnet mittels operationaler Modelle ‚Risiken und Nebenwirkungen‘ von Geräten und Systemen in der Produktion, die konform zu Standards funktionieren sollen. Die Maschine führt die konform-geprüften Anweisungen einer Blaupause strikt aus und liefert das spezifizierte Produkt.

Die Beziehungen zwischen Standard, Modell und System, gleicht der Beziehung zwischen menschlicher Kreativität, dargelegt in Standards, algorithmischer Berechnungsfähigkeit des Computers und der

funktionalen Ausführung von Anweisung an eine Maschinerie im technischen Bereich. Diese Beziehungen werden mittels Morphismen, wie in Bild 1 dargestellt, beschrieben.

Je nach Bedarf können verschiedene Morphismen gebildet werden. Ein Morphismus beschreibt eine Relation zwischen standardisierten Begriffen und computer-berechenbaren Ausdrücken von Graphen, die das Verhalten eines Dings im Modell in der semantischen Domäne repräsentieren. Ein anderer Morphismus beschreibt Implementierung-Relationen auf Maschinen im technischen Bereich und ein weiterer Morphismus beschreibt Relationen zwischen dem Verhalten der technischen Maschinerie oder der semantischen Modellbildung mit Graphen und den Anforderungen aus entsprechenden Standards

In der industriellen Fertigung gibt es Produktionsprozesse, Daten, Produktionsdatenmuster und Maschinen (Assets). Im Modell wird das Verhalten eines technischen Prozesses mathematisch, hier mit Konzepten algebraischer Graphentheorie, ergänzt mit Deklarationen Abstrakten Datentypen (ADT) für die Datentypeneigenschaften, ausgedrückt. Produktionsmuster werden in Mengen ähnlicher Datenmuster zu einem Datenmodell konsolidiert. Die Maschinen implementieren technische Übertragungsfunktionen nach dem Ursache-Wirkungs-Prinzip, Feed-back-Schleifen oder Feed-forward links.

Die Semantik von Prozessen wird mit linearen Differentialgleichungen (z. B. thermo-dynamischen Gleichungen) dargestellt. Die technische Lösung gilt aber nur für einen linearen Betriebsbereich. Dort findet die Laplace Transformation Anwendung, um Steuerung oder Regelung algebraisch berechenbar zu machen.

Der Morphismus, d. h. die Transformation von den DGLs, die ein operatives interagierendes System, in Bild B.1, bestehend aus den Subsystemen S_{YS_S} und S_{YS_E} , mitsamt seinen kontinuierlichen internen und Ein-Ausgangs-Variablen $x_S(t)$, $u_S(t)$, $y_S(t)$ und $x_E(t)$, $u_E(t)$, $y_E(t)$ etc., beschreiben das veränderliche Systemverhalten. Der gesuchte Implementierungs-Morphismus weist nun diesen semantischen Konzepten aus der transformierten DGL, technisch-funktionale Prozesse, Variablen, Konstanten, Anforderungen, Komponenten etc. in einer Weise zu, dass ein funktionierendes interagierendes Gesamtsystem entsteht, wie in den Bildern 7, 8 und 10 dargestellt. Der Implementierungsmorphismus stellt die Formalisierung der gleichen Regeln dar, die auch ein guter Ingenieur berücksichtigen muss. Dasselbe gilt für den Normungs- bzw. Symbolisierungs- und dem Konzeptionalisierung-Morphismus an den anderen Seiten des Semiotischen Dreiecks.

Schließlich stützt sich die **Entscheidungsfindung** mittels computergestützter semantischer Modelle, auf die Homomorphie- bzw. Isomorphie-Prüfung von Graphen. Das heißt, dass homomorphe Graphen nachweislich strukturell gleich sein müssen. Eine weitere Handlungsempfehlung für die Normung betrifft das Maß für strukturelle Gleichheit bzw. Ungleichheit, das bedeutet, mit Unsicherheit bei der Entscheidungsfindung in der Praxis umgehen zu können, z. B. beim Erkennen von Mutationen in molekularen Graphen, Farbpixelvariationen in Computervisionen usw. Um solche kombinatorischen Strukturen erzeugen zu können, müssen graphen- und Datentypalgebren in polynomialer endlicher Zeit auf ihre Normalformen reduziert werden können⁴.

ANMERKUNG 1 Formal ausgedrückt ist Graph Homomorphie nichts weiter als eine Abbildung zwischen zwei Graphen: von einem Graphen G mit Vertexen V und Kanten E auf einen Graphen H mit Vertexen W und Kanten F , mit einer Zuordnung der Vertexe V auf H und der Kanten E auf F , wobei Ähnlichkeit und Nicht-Ähnlichkeit der Graphen G und H beibehalten werden.

ANMERKUNG 2 Im Anhang B werden diese Details eines formalen Engineerings ausführlich vorgestellt. Das kontinuierliche Systemverhalten wird systemtheoretisch als Lineare Differentialgleichung (DGL) definiert. Das ist der Ausgangspunkt des formalen System-Engineering, welches in Bild B.5 dargestellt wird. Der nächste Schritt der Variablen-Deklaration und -Initialisierung (mittels ADT) ist in Bild B.2 dargestellt. Mit Variablen, Konstanten und Attributen werden gleichzeitig die quantifizierbaren Merkmale deklariert. In Bild B.3 wird das daraus abgeleitete Informationsmodell vorgestellt, womit sich der Kreis des formalen Engineerings schließt, weil das Informationsmodell mit dem Systemmodell ‚homomorph‘ anzustreben ist.

Die Abstraktion der Systeme und ihre Eigenschaften im formalen Engineering kann verglichen werden mit linguistischen Ansätzen, wie dem semiotischen Dreieck, wie es in der Bild B.4 dargestellt worden ist. Bild B.5 enthält ein Beispiel einer dem System- und Informationsmodell kompatiblen formalen Beschreibung.

⁴ Für diese Isomorphie-Prüfverfahren gibt es auch schon öffentlich verfügbare Werkzeuge. Eines davon ist der von Babai 2015 entwickelte "Quasy-Polynomial-Time-Algorithmus" und ein weiteres ist McKays "Practical Graph Isomorphy Testing Tool" von 1981.

5 Semantische Beschreibungselemente

5.1 Semantische Artefakte

Semantische Artefakte sind Darstellungen von semantischen Inhalten, entweder in operationaler, computer-ausführbarer Form oder in Form mathematischer Ausdrücke. Von letzterem, d. h. die Lösung von DGLs, wird hier weiter kein Gebrauch gemacht, weil ein operationales Modell die Vorteile einer computer-gestützten Analyse des semantischen Modells bietet. Dementsprechend sind die semantischen Artefakte alle auch operationale Artefakte, was u.a. in graphtheoretischen Elementen seinen Ausdruck findet.

5.1.1 Ereignisse (Graph Edges)

Ereignisse werden als Kanten in Graphen dargestellt. Grundlage der Graphen-Semantik, parametrisiert mit ADTs, sind Regelmengen. Regelmengen bestimmen das Verhalten eines Systems und werden verschiedenen semantischen und realen Artefakten, wie Graphknoten (Vertices), Graphkanten (Edges), Datentypen (ADTs) oder Funktionen zugeordnet. Mit der Prüfung der Gültigkeit von Regeln in bestimmten Umständen können dynamisch sich verändernde Zustände von Anlagen (den Assets), Eigenschaften und Merkmale von Dingen oder Werkstücken etc. festgestellt, bzw. validiert werden. Daraus ergibt sich eine Folge von Ereignissen, Trajekturen sich verändernder Systemzustände. Die Menge der möglichen Ereignisstrajektorien, auch Runs genannt, bestimmen die Funktionalität eines Systems.

5.1.2 Prozesse und Signaturen (Graph Vertices)

Ein Prozess definiert sich über seine

$$\text{kontinuierlichen Variablen } \dot{X} = \Delta/\Delta t (f 1),$$

z. B. die veränderliche Windenergie oder der tägliche Bedarf an elektrischer Energie beim Endverbraucher. Alle Variablen, Axiome, Regeln und funktionalen Eigenschaften werden in abstrakten Datentypen definiert und die Kontexte (Graph Vertices) damit parametrisiert.

Damit die Prozessinstanzen der Kontexte, miteinander interagieren können, müssen ihre Prozess-Signaturen syntaktisch aneinander angepasst werden.

5.1.3 Schnittstellen und Umschreiberegeln (Graph Rewrite Rules)

Ein Beispiel für permanente Transformation sind alle Trajektorien eines energie-erzeugenden (PMS) und energie-verbrauchenden (SM) gekoppelten Systems, beginnend mit der volatilen Energieerzeugung, über die mechanisch-elektrische Transformation, des Transports durch Energienetzwerke, bis hin zum Tageszeit abhängigen Verbraucher in der Industrie oder im städtischen Bereich.

Ein weiteres einfaches Beispiel einer lernenden (*rewrite*) Umschreiberegeln ergibt sich, wenn verschiedenartige Kontexte, wie z. B. ‚Klima-UV Einstrahlung‘ und ‚persönliches-Krebsrisiko‘ semantisch miteinander in Verbindung gebracht werden, d. h. sie können ‚interagieren‘ unter Umständen. Interaktion bedeutet hier, wenn sich eine betrachtete Person der UV-Strahlung aussetzen würde. Im Graph bedeutet die Interaktion eine semantische Änderung, nämlich einen neuen Eintrag einer gerichteten Kante, d. h. die Interaktion hat stattgefunden, in den sogenannten Typ-Graph mit bisher ungerichteten Kanten. Graph-theoretisch wird die

Relation zwischen zwei Graph Vertices mit den Bezeichnungen C und P, (f 2)

dargestellt als ungeordnete Menge {C, P}, (f 3)

wird umgeschrieben in ein geordnetes Paar, dargestellt als Kante (C, P). (f 4).

In den Graphen wird ein geordnetes Vertexpaar als gerichtete Kante (Pfeil von C nach P) eingetragen.

Im Beispiel, als Folge dieses erhöhten persönlichen Risikos, nach der Kopplung bzw. Interoperation der Kontexte ‚Climate Y‘ und ‚Person X‘ können zwei Dinge gelernt werden, nämlich, dass Einwirkung auf die Objekte oder Subjekte (dargestellt als Variablen der Kontexte) genommen wird, indem sich die Person, z. B. in den Schutz eines Hauses begibt oder die klimatische Sonneneinstrahlung reduziert wird. Letzteres wird schwerlich adhoc möglich sein, also bleibt nur die Hoffnung, dass die Person aus ihrem erhöhten Risiko lernt und sich in den Schutz eines Hauses begibt, nach der

Regel ‚make(X) independentfrom(Y)‘, wobei X für das (abhängige) Risiko der Person und Y für das (autonome) Verhalten des Klimas, steht.

Es ist wichtig anzumerken, dass ‚rewriting‘, also das Umschreiben eines Graphen, Fortschritt in einer Trajektorie (Pfad, Systemverhalten, Use Case) bedeutet und somit mit dem maschinellen (Daten-) Lernen einer KI verglichen werden kann, allerdings mit dem wichtigen Unterschied, dass die gelernte Regel, nämlich ‚der UV-Strahlung aus dem Wege zu gehen‘, aus der aktuellen Belegung der Variablen X und Y der umgeschriebenen Kante, regelbasiert hervorgeht, bzw. abgeleitet werden kann.

5.1.4 RAMI4.0 Kontexttransformationen (Graph Rewriting)

Das RAMI4.0 (siehe [11] und [12]) gibt mit den ‚*Hierarchy Levels*‘ (auf der Y-Achse), als auch mit dem ‚*Life Cycle and Value Stream*‘ (auf der X-Achse), mehr als einen Fingerzeig, wie die Kontexttransformationen gehandhabt werden sollen: Die Kontextanpassungen des Value Stream gelten offensichtlich von der Produkttypentwicklung bis zur Produktinstanzentwicklung und -nutzung. Vergleichbar sieht es auf Seiten der Gerätehierarchie aus: Ein Produkt wird in ein Field Device eingebaut, das von einem Kontrollgerät auf einer Station, überwacht wird. Stationen werden in Work Centers zusammengefasst, die Teil eines Unternehmens sind. Die Unternehmen selbst interagieren mit anderen Unternehmen in einer ‚Connected World‘. Ähnlich sieht es mit den Interoperabilitätsschichten (auf der Z-Achse) aus: Ein physisches Asset wird jeweils aus verschiedenen Blickwinkeln der Interoperabilität betrachtet, bis es in der Businessschicht in einen geschäftlichen Wert transformiert worden ist.

Diese Geschichten von sequenziellen Kontexttransformationen erfordern ein semantisches Beschreibungs-Artefakt, das mit dem Narrativ einer Graph Trajektorie gegeben ist und womit die Interoperabilität zwischen sequentiellen Kontexten validiert werden kann.

5.1.5 Kontext Signaturen (Graph Interoperability)

Wenn in der Normung von ‚Interoperabilität‘ zwischen Maschine-zu-Maschine (MMI) oder Mensch-zu-Maschine (HMI) die Rede ist, ist das gleichzeitig auch ein semantisches, weniger ein syntaktisches Problem, wiewohl die Definition für IIoT, zwischen syntaktischer und semantischer Interoperabilität unterscheidet. Hier gibt es Handlungsbedarf diese Unterscheidung zugunsten einer eindeutigen semantischen Darstellung aufzuheben, denn Signaturen aus unterschiedlichen Kontexten können sich durchaus eine gemeinsame Semantik teilen, falls sie strukturverträglich angelegt sind. Man kann also **strukturverträgliche Signaturen**, d. h. alle passenden syntaktischen Kontextschnittstellen mit einer einzigen Semantik (der dazu isomorphen Algebra) erklären. Mit anderen Worten: In den Signaturen (Schnittstellen) liegen die syntaktischen Freiheitsgrade, während sie in der Semantik bedeutungslos gleich sind.

Die Signaturen der Datentypen enthalten die **Konstanten, Variablen, Invarianten, Axiome und Regeln**, welche jeden Kontext charakterisieren. Die Transformation vom Zustand des einen Kontexts in den nächsten, kommt durch Prüfung der Regelmengen in den Anfangs- (head) und den Ziel- (tail) Knoten (Paar Graph Vertices) zustande.

5.1.6 informationelle Beziehungen (Data Graph Relation)

Die **Daten einer Beziehung** zwischen Objekten oder Subjekten werden als Relationen in einem **ungerichteten Data Graph** (d. h. in einem Netzwerk⁵) dargestellt, die jeweils Paare von Knoten, welche **Akteure (Stakeholder)** oder **Daten Objekte** repräsentieren, miteinander in Bezug setzen:

BEISPIEL 3 Autoren-Netzwerk, als Knowledge Graph dargestellt.

Die Beziehung zwischen Autoren und ihren Publikationen werden als Graphen dargestellt. Knoten (Graph Vertices) repräsentieren Autorenrollen oder Publikation; Kanten (Graph Edges) stellen möglichen Beziehung zwischen Autoren und ihren Publikationen dar, z. B. die Beziehung der Autor-Ko-Autorenschaft (1) oder die Beziehung Publikation-Urheberschaft (2).

Aus der Graphentheorie (Anhang A) geht hervor, dass die Kanten eines ungerichteten Graphen⁵ als paarweise Mengen von Knoten behandelt werden. Eine Beziehung (1) besteht zwischen Paaren von Autoren. Eine andere Beziehung (2) besteht zwischen Paaren ‚Autor - Publikation‘. Der Daten Graph (bzw. das ‚Autoren‘-Netzwerk) enthält also für jeden Namen und jede Publikation die statischen Beziehungen (1) und (2).

BEISPIEL 4 Smart Manufacturing Daten bzw. Knowledge als Netzwerk von OT/IT Beziehungen,

⁵ Im technischen Sprachgebrauch wird ein Netzwerk mit einem Graph mit ungerichteten Kanten gleichgesetzt. Eine ungerichtete Kante ist eine etikettierte Knotenpaarmenge $\{\text{Etikett}_{|A, E}\}$ mit Anfangs- und Endknoten und dem Kantenlabel Etikett;

repräsentieren Assets auf den Schichten der Interoperabilität und den hierarchischen Schichten einer globalen Produktion und dem Lebenszyklus der Produktion.

Die Attribute und Eigenschaften der Akteure oder Objekte werden als Abstrakte Datentypen (ADT) spezifiziert und den jeweiligen Knoten zugeordnet. Relationen zwischen Datenobjekten, bzw. ADTs, können den Objektvariablen zugeordnet werden. In dieser Zuordnung existieren *Typgraph* und *Inzidenzgraph* nebeneinander, indem ADT Variablen Daten Graph Relationen zugeordnet werden.

5.1.7 Anwendungsfall-Beschreibung, System Runs (Graph Trajectory)

Trajektorien sind Use Cases, die auf der Basis eines System Architektur (Type) Graph validiert worden sind. Eine validierte Menge von Trajektorien definieren ein System. Die Trajektorien werden in einer Spezifikation oder Norm als Narrative dargestellt, wobei das Narrative das sequentielle Zustandekommen von gerichteten Kanten im Graph, erzählt. Zustandekommen heißt, es werden an den Schnittstellen der Kontexte, Regelmengen geprüft, wobei jeweils eine oder mehrere Regeln gültig sein oder werden müssen (**Eventualität**).

5.1.8 Value Streams

Value streams entstehen im operativen Bereich, in der Produktion, aber auch in der Energieversorgung, weil ein ‚Produktionsanfang‘ und ein ‚Produktionsende‘ auszumachen ist. Es kann also ein Narrative über den ‚Fluss der Dinge‘ durch viele Be- und Verarbeitungskontexte erzählt werden. Value Streams gibt es aber auch für Daten, wenn man konstatiert, dass der Mehrwert, der in der Produktion durch Produktveredelung, in der Datenverarbeitung durch ‚Veredelung der hierarchischen Datenstrukturen‘ entsteht. Veredelung bedeutet hierbei Metadaten, also Informationen durch Beobachtung, z. B. des Produktionsprozesses, zu generieren (vgl. Dynamisierung der *Knowledge Pyramid*).

Beispiel 5 Value stream, ausgehend von verfügbaren Energie-Ressourcen,

über die mechanische Energieerzeugung, über die Erhaltung des elektrischen Energiezustands in den sogenannten ‚Distributed Energy Resources (DER)‘, bis hin zum Endverbraucher in der Industrie oder in den Städten, nimmt die Energie viele Formen an. Diese Formen sind miteinander über die globalen Invarianten verbunden, die darin bestehen, dass ‚Ausgeglichenheit‘ im Energie Value Stream vom Erzeuger über den Transport bis zum Verbraucher herrscht.

Ein Narrativ beschreibt die Suche nach einem Gradienten im *Ereignis- bzw. Incident Graph* (Graph mit gerichteten Kanten)⁶. Ein **Gradient** ist der minimale (kanonische) Pfad durch den Incident Graph, d. h. durch die statische System-Architektur oder -Struktur.

Im nachfolgenden Abschnitt, werden zuerst die **interagierenden Kontexte** (ggf. Stakeholder) und danach die **Variablen und Prädikate** des Value Streams durch die Kontexte definiert.

5.1.9 Variablen, Daten, Funktionen und Prozesse

In der Modellwelt der algebraischen Semantik, repräsentieren die erzeugten Ausdrücke (Terme genannt) Daten, Funktionen und Prozesse.

In den Abschnitten 5.1 und 5.2 sind bereits Variablen für Datenterme eingeführt worden. Alle Terme einer bestimmten Sorte können durch eine Variable in der Signatur des Abstrakten Datentyps repräsentiert werden.

Im folgenden Fall ist es die Variable A, die einen Term aus der Menge ‚authorization‘, in der Termalgebra SORT genannt, z. B. A:= ‚firstAuthor‘ enthalten kann.

```
1. VAR A OFSORT authorization;
```

Funktionen haben Definitions- und Wertebereiche, die mittels Operatoren deklariert werden. Eine Operation wird mit ihrem Namen, ihren Definitions- und Wertebereichen, getrennt durch das Zeichen ‚→‘ deklariert, wobei Definitions- und Wertebereiche vorher deklarierte Sorten enthalten müssen. Die nachfolgende Deklaration ist dem Beispiel für die Beziehungen zwischen Koautoren einer Publikation (Bilder 5, 6) entnommen:

```
2. DefineContent: <structureA>, authorization|1st author --> document;
```

⁶ Ein Graph mit ungerichteten Kanten heißt Typ Graph, im Gegensatz zum Incident oder Event Graph, der gerichtete Kanten, die Ereignisse (Incidents) symbolisieren, hat.

Ein Prozess oder Vorgang wird, wie das nachfolgende Beispiel zeigt, mit mehreren Variablen beschrieben, u.a. mit sogenannten

3. kontinuierlichen Variablen $\dot{x} = f(t) = \partial x / \partial t$,

oder einfachen Variablen, die in der **Signatur des Prozesses** deklariert worden sein müssen. Z. B. der Konsum von Energie, gemessen über 24h, eines Wohnhauses, einer Fabrikanlage oder einer Kleinstadt etc. \dot{x} beschreibt also eine Wertemenge, was der Aufzählung von Konstanten einer Sorte eines ADT entspricht.

In **technischen Systemen** werden Funktionen von Geräten bzw. Systemkomponenten mit multiplen Eingangs- und Ausgangsvariablen realisiert, wobei die Variablen als

4. Vektoren in Matrixschreibweise $(p \times 1)$ und $(q \times 1)$

dargestellt werden.

5.1.10 Semantische Transferfunktionen und Inzidenzen

Daher werden diese Funktionen im technischen Bereich, **Transferfunktionen** genannt und in Blockdiagrammen verwendet. Eine Transferfunktion beschreibt die algebraischen⁷ Zusammenhänge zwischen **Ursache und Wirkung** einer Systemkomponente, die in ihrem linearen Bereich betrieben wird.

ANMERKUNG Diese Betrachtungsweise wurde in Kapitel 5.1 Prozess- und Typgraphen eingeführt und wird hier in die Graphenbeschreibung überführt. Ein Prozess hat in dieser algebraischen Darstellung zwei Möglichkeiten Folgen von Zustandsänderungen darzustellen bzw. zu nutzen:

- 1) Daten-orientiert, indem eine Funktion $f(t) = \dot{x}$ angegeben wird, die eine Messreihe z. B. des Energieverbrauchs einer Siedlung oder Fabrikanlage während eines Tages im Stundentakt ausgeführt, darstellt, sodass jeder Messpunkt einen Zustand repräsentiert und,
- 2) Ereignis-orientiert, indem eine Zustandsänderung mit einem Ereignis direkt verknüpft ist, das seine Ursache in einem bestimmten Zustand und seine Wirkung in einem Folgezustand des Prozesses hat. Graphentheoretisch ist ein Ereignis also eine Graphkante, die als gerichtetes Paar von Graph Vertices (bzw. von Zuständen) deklariert wird.

Zustandsfolgen können zu Trajektorien zusammengefasst werden. Eine Trajektorie kann als geprüfter Anwendungsfall (*Use Case*) eines Systems betrachtet werden. Eine Menge von Trajektorien, gesammelt nach bestimmten Kriterien beschreibt das System (under consideration). Das gilt sowohl für die Daten- also auch für die Ereignisorientierung. Während eine Messreihe einen Prüffall darstellt, womit ein System getestet wird, stellt eine Ereignisfolge eher die Auswirkungen eines Prüffalls auf das getestete System dar. Das gilt umso mehr, weil in jedem Zustand des Systems instanziierte Regeln auf Gültigkeit geprüft und angewendet werden. (Dieser Vorgang ist auch als ‚coercion‘ bekannt.)

In der operationalisierten (Laplace) Domäne können die algebraischen Grundrechenarten, Addition, Subtraktion, aber auch die nicht-lineare Multiplikation, die in der Laplace-Domäne mit dem **Operator ‚complex convolution‘**⁸ ersetzt wird, auf die transformierten Transferfunktionen angewendet werden.

Mit diesem ‚Kunstgriff‘ lassen sich komplexe Systemstrukturen, wie Rückführungsschleifen durch Anwendung des Laplace Operators *complex convolution*, algebraisch darstellen, z. B. in dem System $G(s) * H(s)$, mit den Transferfunktionen $G(s)$ im Vorwärtspfad und $H(s)$ im Rückwärtspfad, wie in der Bild 4 dargestellt.

⁷ Wenn ein System im linearen Bereich operiert, kann es mittels Laplace Transformation algebraisch behandelt werden. Die Transferfunktions-Transformierte $L(t)$ wird dann mit $L(s)$ bezeichnet. Nicht-lineares Verhalten kann nur in der Realzeit-Domäne mittels Lösen von DGLs behandelt werden.

⁸ Zur Unterscheidung wird die Multiplikation mit ‚x‘ und Complex Convolution mit ‚*‘ bezeichnet.

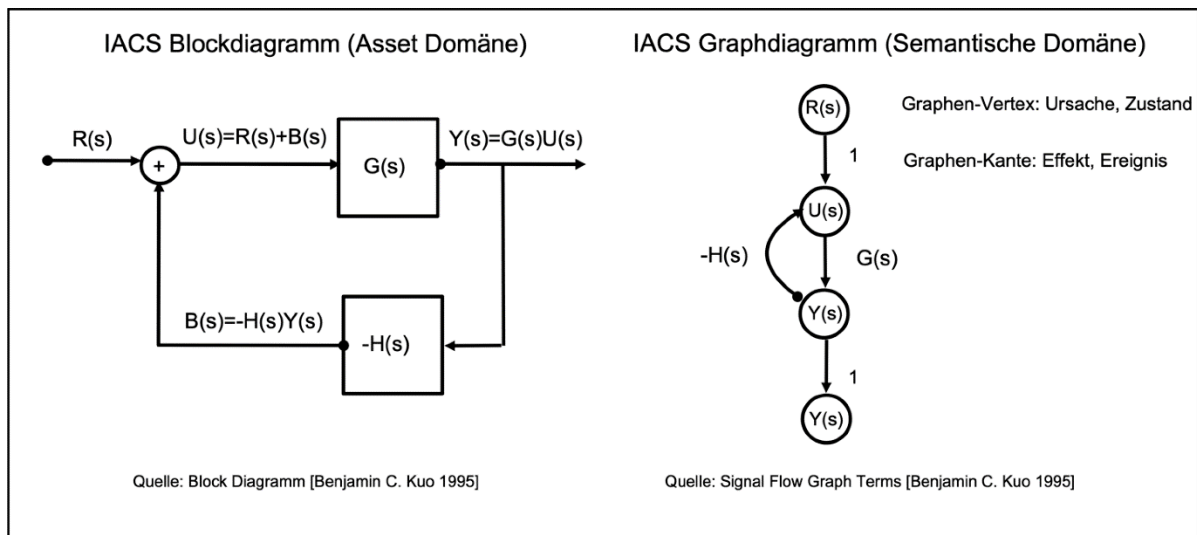


Bild 4 – Block- und Graphen Diagramm eines automatisierten regelungstechnischen-Systems (IACS)

Aus dem in Bild 4 dargestelltem Graphendiagramm lässt sich, bei Unterscheidung von Eingangs- und Ausgangsvariablen, folgendes algebraisches Gleichungssystem ableiten [67]:

$$U(s) = R(s) * 1 - H(s) * Y(s) \text{ /Rückkopplung (f 5)}$$

$$Y(s) = G(s) * U(s) \text{ /Strecke (f 6)}$$

$$Y(s) = Y(s) * 1 \text{ /Systemergebnis (output) (f 7).}$$

In der Graphen-Algebra⁹ werden Gleichungen nach dem Ursache-Wirkungs Prinzip generiert. Ursache und Wirkung werden als Vertices dargestellt, die über ein aktiviertes Ereignis (Incident) verbunden werden. Die Gleichung für diese Relation ergibt sich dann aus:

$$\text{Wirkung (Ausgangsvariablen)} == \text{Inzidenz} * \text{Ursachen (Eingangsvariablen)} \text{ (f 8)}$$

Ausgangs- und Eingangsvariablen, bzw. Wirkung und Ursache, werden im Graphendiagramm dadurch unterschieden, dass der sie repräsentierende Diagrammknoten (Graph Vertex), entweder nur eingehende Kanten oder nur ausgehende Kanten hat. Im ersten Fall ist wird eine Ausgabevariable und im zweiten Fall eine Eingabevariable dargestellt. In vielen praktischen Fällen ist diese Unterscheidung nicht möglich. In diesem Fall wird ein Vertex mit ein- und ausgehenden Kanten, mit einer Einheitskante ‚1‘ erweitert, so dass der heterogene Vertex in Eingangs- und Ausgangs-Vertices aufgeteilt werden kann.

5.1.11 Interoperabilität zwischen Kontexten am Beispiel der Energieversorgung einer SM-Fabrik

Am Beispiel der Energieerzeugung, die aus volatilen (Wind) Quellen, elektrische Energie in ein Netzwerk einspeisen und damit die Industrie und häusliche Abnehmer versorgt, soll die Verwendung von Variablen in einer Sequenz von Handlungen, bzw. Ereignissen, gezeigt werden.

Das Narrativ ‚Smart Grid‘ der Energieerzeugung aus volatilen Quellen wird in 4 Schritten erzählt. Im Beispiel für die Kommunikation zwischen den volatilen autonomen Windverhältnissen und einem Rotor zur wind-abhängigen Erzeugung von mechanisch-elektrischer Energie. (Anmerkung: Die vollständige formale Darstellung der Smart Grid Graph-Spezifikation ist im Anhang C gegeben):

- 1) Identifikation der Stakeholder Kontexte (Graph Vertices),
- 2) Spezifikation des Verhaltens der Stakeholder (kontinuierlichen Variablen in der dot-Notation),

⁹ ‚Graphalgebra‘ wird von B.C.Kuo [69] als ‚Theory of Signal Flow Graphs‘ und von Lunze [67] als ‚Strukturanalyse gekoppelter Systeme‘ beschrieben.

- 3) Spezifikation der Signaturen, d. h. der Schnittstellen zwischen den Stakeholdern (Graph Edges),
- 4) Spezifikation der Ereignisse an den Schnittstellen (Kontext Umschreibe-Regeln).

Zuerst werden alle Stakeholder Kontexte identifiziert, z. B. die volatilen Wetterbedingungen, der mechanische Rotor, der elektrische Windgenerator, die Verteilten Energie Netzwerk Ressourcen (DER), die Energieverbraucher

Danach werden die charakteristischen Eigenschaften, mittels Variablen und Invarianten der identifizierten Stakeholder definiert. Die Abläufe, bzw. die zeitlichen Veränderungen in einem Produktionsprozeß werden mit sogenannten **kontinuierlichen Variablen**, in der sogenannten *dot-Notation* VAR^{dot} , als kontextuelle Veränderungen der Energie dargestellt.

BEISPIEL 6 Modell der Energieübertragung im Smart Grid mit 5 *Stakeholder* Kontexten (f9-f14).

Definition Variablen, Konstanten: $c, A = \text{const}, ((m, v), (p, v), (M, \Omega), (.), (f(\Delta T)))$ (f 9)

beschreiben die veränderlichen Kontexte der 5 kooperierenden Stakeholder.

Sh Wetter: $c(m v^2)^{dot}$ (f 10)

stellt die veränderliche Bewegungsenergie des Windes im Wetterkontext dar,

Sh Rotor: $cA(p v^3)^{dot}$ (f 11)

stellt die gewonnene Rotorleistung im Kontext der Windradkonstruktion dar,

Sh Generator: $c(M \Omega^2)^{dot}$ (f 12)

stellt die gewonnene elektrische Energie im Kontext der Leistungsfähigkeit des Generators dar,

Sh DER: transportierte elektrischen Energie im Netzwerk (f 13)

Sh Heim: $f(\Delta T)$ (f 14)

stellt die vom Verbraucher genutzte elektrische Energie, als empirische Funktion der Tagesstunde, dar. Die verbrauchte Energie entspricht der elektrischen Generatorleistung, die mittels DER Ressourcen zum Verbraucher übertragen worden ist.

Die beiden Stakeholder ‚Wetter‘ und ‚Rotor‘ interagieren über ihre kontinuierlichen Variablen *Windmasse* $m[kp s^2/m]^{dot}$, *Windgeschwindigkeit* $v[m/s]^{dot}$ und *Rotorimpuls* $p[kp s]^{dot}$ miteinander. Diese Schnittstelle wird mit ihrer **spezifischen Signatur**, wie in Anhang C, definiert.

Graphisch betrachtet sind alle Stakeholder über Kanten miteinander verbundene Graph-Vertices (Ecken, Knoten). Eine Graph-Edge (Kante) macht aus einem ungerichteten Knotenpaar ein gerichtetes Knotenpaar mit einem Anfang (*event head*) und einem Ende (*event tail*) des Ereignisses. Im Beispiel ist die Wandlung des Knoten-Paars ‚WVC‘ und ‚ROT‘ dargestellt. Die Energie kann nur dann fließen, wenn am Ereignisanfang ‚sich der Wind bewegt‘ und am Ereignisende der ‚Rotor kontinuierlich Energie (in den Generator) liefern‘ kann.

6 Semiotische Best Practices in der Normung

6.1 Semiotische Darstellung Cyber-Physischer Systeme

Die mathematische Modellierung Cyber-Physischer Systeme (CPS), im semiotischen Dreieck Assets genannt, stützt sich auf Differentiale und Integrale der kontinuierlich veränderlichen Systemvariablen (vgl. x^{dot} Notation). Als Beispiel dafür kann ein elektrisches Netzwerk mit Ohmschen (linearem), induktiven (differenzialem), kapazitiven (integralem) Widerständen, in dem der Strom $i(t)$ eine unbekannte abhängige Variable darstellt, genannt werden.

Weil eine Folge von Zuständen eine Graph Trajektorie ist, die eine Folge von Systembedingungen in der Vergangenheit, Gegenwart und Zukunft, z. B. des Stroms im o.g. Netzwerkes, darstellt, wird ein dynamisches System als eine Menge von Zustandsgleichungen modelliert. Zustandsgleichungen werden durch Graph Knoten (Vertices) repräsentiert. Es ist wichtig, dass Systemzustände nicht mit Systemausgaben, bzw. Ergebnisse verwechselt werden, denn Systemzustände sind nicht immer direkt

beobachtbar, während Systemergebnisse, zwar von den Systemzuständen abhängig, aber immer beobachtbar sind.

Das Zustandsmodell von dynamischen Systemen ist ein Energiemodell. Z. B. speichert ein induktiver Widerstand kinetische Energie, während ein kapazitiver Widerstand potenzielle Energie speichert. Systeme, die ihre internen Zustände semantisch am Energiemodell orientieren, sind interoperativ, weil sie anhand der Zustandsgleichungen ineinander überführt werden können.

Die energetische Sichtweise funktioniert ganz analog für mechanische Geräte, indem z. B. die bewegte Masse mit der elektrischen Induktivität assoziiert wird.

Systemzustand $\frac{dx(t)}{dt}$ und Ausgang $y(t)$ eines Geräts oder Systems, wie auch in den Bildern 7 und 8, oder in der Gleichung (f 15) dargestellt, beruhen auf der Addition (Vergleich) der Zustandsvariable $x(t)$ mit der Eingangsvariable $r(t)$ und können direkt aus dem Graphenmodell unter Beachtung der nachfolgenden Best Practices abgeleitet werden.

Ein Zustandsgraph kann direkt von der System DGL abgeleitet werden und liefert direkt die Zustandsvariablen und -gleichungen eines Systems, indem zuerst die Graphknoten, d. h. die Zustände, und anschließend die Graphkanten, die Rückführungen (*loops*) aus integrierenden Knoten darstellen, identifiziert werden.

Das gesamte abgeleitete Graph-Zustandsdiagramm wird ergänzt, mit den Kanten ($s-1$), die Knoten miteinander verbinden und mit den Referenzwerten als Anfangs- ($R*1$) und den Ausgangswerten als Endbedingungen ($Y*1$) eine Folge von Zuständen, bzw. von Ereignissen bilden.

6.2 Semiotische Best Practices für Systeme

Grundlegend für die Anwendungsrichtlinien, semantische Standards zu schreiben, ist die Vorstellung, deklarativ statt algorithmisch zu denken und zu schreiben. Diese Vorgehensweise ist auch als axiomatischer Ansatz bekannt. Axiome stellen die grundlegenden Fixpunkte eines Systems dar. Um diese Fixpunkte herum werden Regeln gruppiert, die zu neuen Zuständen des Systems, abhängig von ihren Wahrheitsgehalten und Kontexten, z. B. Smart Grid, Smart Factory, HealthCare etc. führen können.

Aus den sieben, in Abschnitt 4, beschriebenen Anwendungsregeln ergeben sich folgende Konsequenzen:

Ein vollständig deklarierter und instanzierter Typ-Graph kann auf einer geeigneten Plattform als operationales Modell ausgeführt werden, sofern der *Typgraph* als Theorie, d. h. Algebra vollständig beschrieben worden ist.

Mittels Simulation kann eine Reihe gültiger Ereignisfolgen (*Trajektorien*) beobachtet bzw. analysiert werden. Da die Trajektorien Läufe des deklarierten Systems sind, sind sie gleichzeitig Läufe (Use Cases, Verhaltensmuster), die das analysierte System definieren.

6.3 Semiotische Best Practices für Transferfunktionen (Vorgänge)

Prozesse definieren sich über ihre Variablen. Variablen haben einen Typ, der eine Datenmenge, eine so-genannte Sorte, repräsentiert. Variablen können mit Transferfunktionen miteinander verbunden werden, das dem algebraischen Konzept $y=f(x)$ entspricht. Transferfunktionen beschreiben Ursache-Wirkungs-Relationen in allen Prozessen in einem technischen System.

Ein-Ausgangsverhalten kann, je nach repräsentiertem Gerät („*thing*“) linear oder nicht-linear sein, wobei lineares Verhalten Bedeutung (Semantik) im Zeitbereich hat; sie kann damit als Laplace Transformierte algebraisch manipuliert werden. Nicht-lineares Verhalten hat nur Bedeutung im Realzeitbereich und kann damit nur als lineare Differentialgleichung gelöst werden. Nicht-lineares Verhalten ist im Anwendungsbereich von *SemNorm* nicht enthalten.

Das heißt, jedes komplexe industrielle System, das in seinem linearen Betriebsbereich auf eine kanonische semantische Darstellung reduziert werden kann, kann als ausführbares Daten-Prozess-Modell zur Systemlaufzeit auch berechnet werden. Damit lassen sich künftige Systemzustände kalkulieren und evaluieren. Genauso können in dem Systemgraphen Ursache-Wirkungskausalitäten anhand der Trajektorien zurückverfolgt und berechnet werden.

6.4 Semiotische Best Practices für Entscheidungsvorgänge

Entscheidungsvorgänge können sowohl die Ursache als auch die Wirkung eines Ereignisses beeinflussen. Indem die Ursache für einen gemessenen Zustand beeinflusst werden soll, wird eine Rückführung

im Graphen benötigt. Diese Rückführung könnte z. B. die gemessene Abweichung von einem erwarteten Ergebnis sein. Wenn diese Kenntnisse in geeignetem Format am Eingang zur Verfügung stehen, könnten entsprechende Maßnahmen zum Ausgleich angewendet werden. Die Entscheidungsfindung ist hier in einem logischen Netzwerk fest verdrahtet. Der Inzidenzgraph beinhaltet an Stellen der Entscheidungsnotwendigkeit eingebaute rückwärts-orientierte Schleifen (siehe Bild 4). Schleifen stellen ein typisches Muster in der Struktur von Graphen dar und sind daher wichtig beim Strukturvergleich, ggf. in Morphismen.

Um Auswirkungen eines Ereignisses vorherzusagen, benötigt man ein Modell. Dieses Modell wird aus ausgesuchten Darstellungsmustern gelernt. Wenn ein neues unbekanntes Muster ankommt, wird dieses Muster auf Strukturvertäglichkeit (Homomorphie) geprüft. Sind die Strukturen ‚gleichwertig‘ kann das neue Muster im Modell als bekannt genug akzeptiert werden, andernfalls wird das unbekannte Muster verworfen. Insofern wird das mögliche Ergebnis mit der Akzeptanz des Musters verändert oder nicht. Dieses Charakteristikum wird graphisch als vorwärts gerichtete Kante in den Graphen eingetragen. U.U. kann die Kante den gleichen Knoten als Ursprungs- und Endknoten haben. Mit der vorwärts gerichteten Kante können auch Knoten übersprungen werden. Es bilden sich also komplexe Verhaltensstrukturen heraus.

6.5 Semiotische Best Practices um Morphismen zu definieren

Normen und Gesetze bestehen aus aufgeschriebenen Texten, die meist einen komplexen rechtlichen oder technischen Zusammenhang darstellen, der von Dritten korrekt, d. h. ohne mögliche Fehlinterpretationen, die u.U. Sanktionen nach sich ziehen, verstanden werden muss. Der semantische Zusammenhang zwischen (Normungs- oder Gesetzes-) Text und inhaltlicher Interpretation ist hier einer der sogenannten Implementierungs-Morphismen.

6.5.1 Morphismen für Datentypen

Ein Implementierungs-Morphismus bildet bestimmte sprachliche Ausdrücke, z. B. Ausdrücke eines Gesetzestextes, hier des in §5 BDSG enthaltenen sogenannten ‚Datengeheimnisses‘ auf eine bestimmte semantische Darstellung ab. Diese semantische Darstellung ist geeigneterweise eine axiomatisch-deklarative Darstellung im algebraischen Stil. Sie stützt sich auf Graphentheorie und auf die Theorie der Abstrakten Datentypen (ADT).

Das folgende Beispiel 7 illustriert in Bild 5, die anzuwendende Vorgehensweise zur Definition eines Morphismus zwischen den beiden semiotischen Domänen ‚Gesetzestext (Symbol)‘ und algebraische Interpretation (Konzept, Semantik). Dem Gesetzestext des §5 BDSG *Datengeheimnis*, ist die Abfolge der Ereignisse als Inzidenzgraph (links in der Bild 5) und als Graph-Trajektorie der Inzidenzen (rechts in der Bild 5) zugeordnet.

BEISPIEL 3 Ein Morphismus bildet den Text der Regulierung (Symbol) des Datengeheimnisses, nach §5 BSDG auf algebraische Artefakte ab.

„Datenverarbeitenden Personen (**DVP**) ist untersagt, Personen-Bezogene Daten (**PBD**) unbefugt zu erheben, zu verarbeiten, zu nutzen; DVPs ... sind auf das Datengeheimnis (**DG**) zu verpflichten; die DG-Verpflichtung (**DGV**) bleibt auch nach Beendigung der Tätigkeiten der DVP bestehen.“

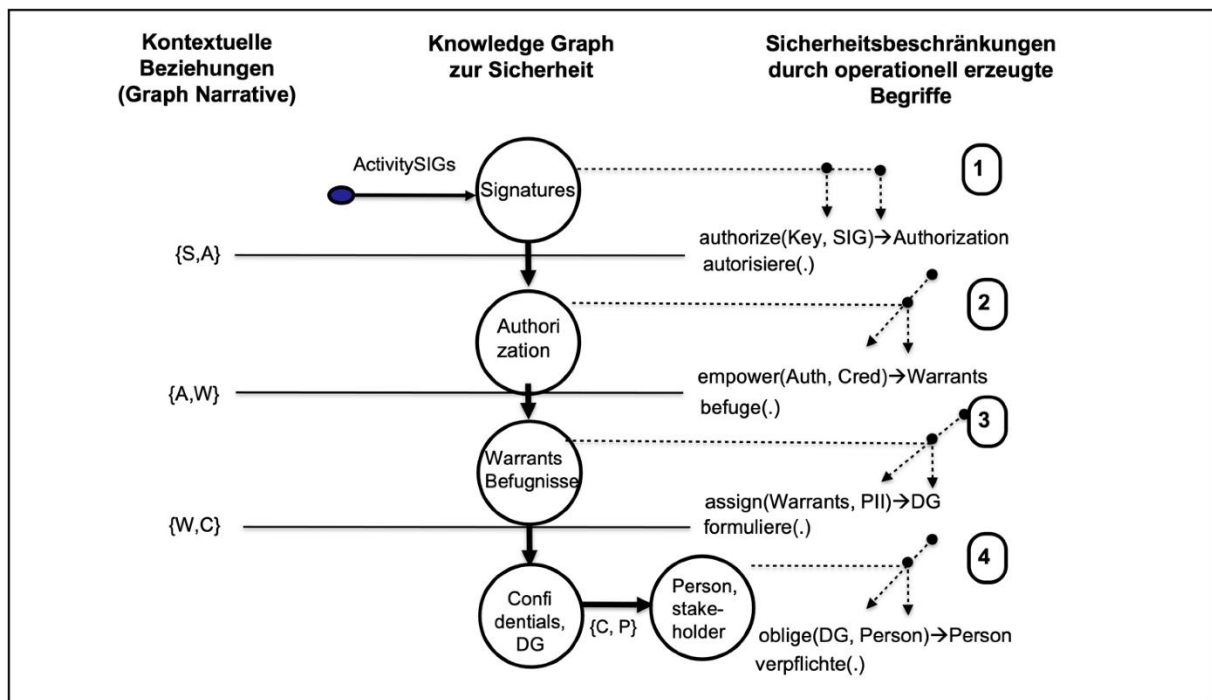


Bild 5 – Beispiel Termalgebra des ‚BDSG Datengeheimnisses‘

Im Folgenden wird nun die, den zitierten Text des Datengeheimnisses repräsentierenden Signaturen in 4 Schritten erzeugt.

Alle Variablen der Termalgebra ‚BDSG Datengeheimnis‘, werden in den Signaturen mit ihren Sorten, nach dem Schema: ‚Var_Name: Sorten_Name‘, definiert und lauten:

DG: Datengeheimnis, DGV: DG Verpflichtung, DVP: Datenverarbeitende-Person,

PBD: PersonenbezogeneDaten;

1. Schritt: Deklaration der Operationen und ihre Parametersorten zur Verarbeitung PBD im DG:

zu_erheben(Parameter Eingabe PBD) ->PBD

zu_verarbeiten(Parameter PBD) ->PBD

zu_nutzen(Parameter PBD) ->Ausgabe;

2. Schritt: Deklaration der potenziellen Ereignisse als ungerichtete Graph-Kanten (d. h. Kontextrelationen):

{Signatures, Authorizations}

{Authorizations, Warrants}

{Warrants, Confidentials}

{Confidentials, Persons};

ANMERKUNG 1 Alle anderen notwendigen Deklarationen, wie Kontextvariablen, Kontext-operationen, Invarianten und Axiome, sind an dieser Stelle der Übersicht wegen, weggelassen worden.

3. Schritt: Daten-repräsentierende Terme auf Basis ihrer Signaturen konstruieren, die ein potenzielles Ereignis zwischen den identifizierten Kontexten darstellen.

autorisiere(Key, SIG) ->Authorization

befuge(Authorization, Credentials) ->Warrants

formuliere(Warrants, PBD) ->DG

verpflichte(DG, Person) ->Person;

4. Schritt: Aktualisierung der Ereignisparameter mit geeigneten Ausdrücken aus den Kontexten, um Trajektorien zu erhalten (Bild 6 rechte Darstellung):

```
verpflichte(formuliere(befuge(autorisiere(key, SIG), credentials),
warrants), PBD), person);
```

Die Trajektorie in Schritt 4 beschreibt also in operationaler Semantik das Narrativ des Datengeheimnisses.

ANMERKUNG 2 Der operationale Ausdruck oben muss von innen nach außen interpretiert werden: Zuerst wird die Autorisierungs-Signatur erzeugt, womit die Befugnisse ‚credentials‘ definiert werden können. Mit den Befugnissen werden die Garantien deklariert, die einer ‚person‘ zugewiesen werden.

6.5.2 Morphismen für Wissen (Knowledge)

Im Beispiel von Bild 5 sind die sogenannten ‚Common Features‘ nach [ISO 27030-2:2020], als kombinierte Daten- und Inzidenzgraphen am Beispiel der Verteilung der Verantwortlichkeiten zwischen den Koautoren beim Schreiben einer gemeinsamen Publikation, dargestellt. Die Verantwortlichkeiten der Autoren sollen zum einen als gerichteter Prozessgraph (Di-Graph) und zum anderen als ungerichteter Typgraph (*Abstract Data Type*) dargestellt werden. Semantisch gesehen, stellen der ADT Graph und der *Inzidenzgraph* die gleiche intendierte Semantik dar. Der kleine Unterschied zwischen einem ADT Graph und einem Prozess-Graphen ist der Unterschied zwischen einem Term, der eine ADT-Operation darstellt und einem Ereignis, das als gerichtete Graphkante (d. h. eine Inzidenzabbildung, siehe Anhang A Graphentheorie) einem gerichteten Paar von Graph Vertices mit einem Ereignis-Startknoten und einem Ereignis-Endknoten, entspricht. Die gesuchte formale Sprache ist die sogenannte **Algebra aller Terme**, die von der **ADT-Signatur** erzeugt wird.

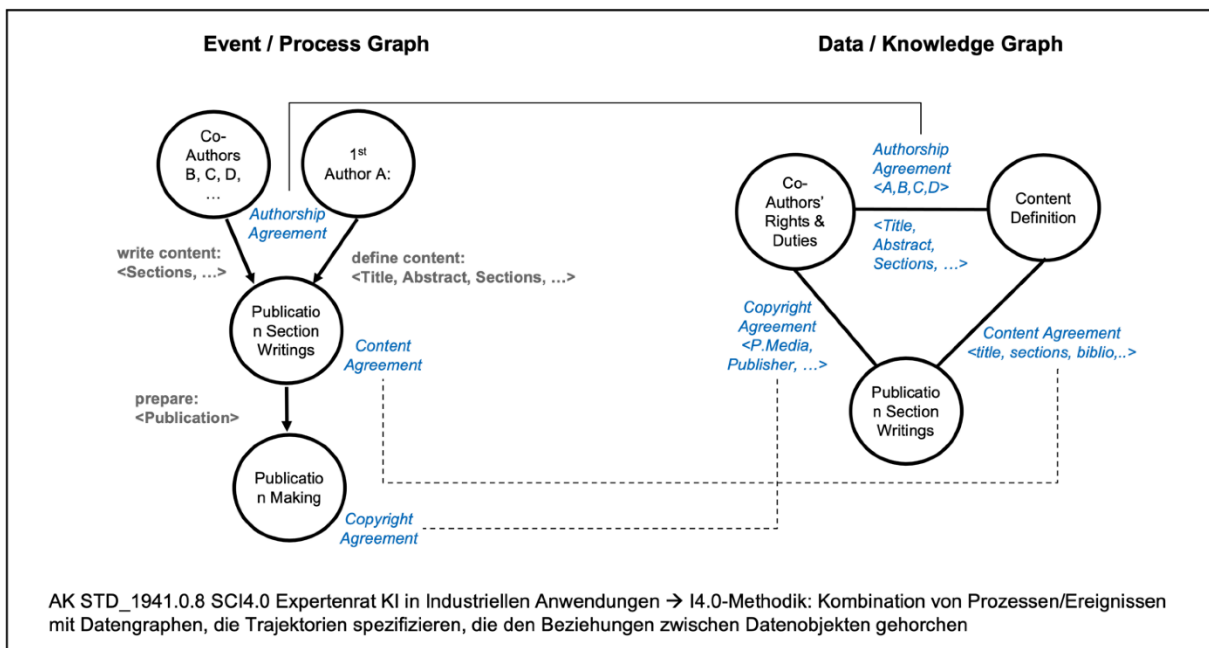


Bild 6 – Beispiel ‚Copyright Agreement‘ als Event und Knowledge Graph [Source: SCI4.0]

In Bild 6 steht links der Ereignis- bzw. Inzidenzgraph mit **gerichteten** Kanten (Di-graph) und rechts der Knowledge Typgraph mit **ungerichteten** Kanten. Gerichtete Kanten werden mit Operationen und ungerichtete Kanten mit Vereinbarungen (agreements, associations) verbunden.

BEISPIEL 8 ‚Copyright Agreement‘ als Knowledge Typgraph

In Bild 6 enthält der ungerichtete Typgraph drei Vereinbarungen. Jede Vereinbarung wird als identifizierbarer <n-tuple> ADT dargestellt:

1. Vereinbarung der Autorenschaft zwischen den Koautoren und den Inhaltsteilen der Publikation:
<authorshipA: coauthorRights&duties|1...n, contentSpec>;
2. Copyright Vereinbarung zwischen den Koautoren und dem Akt des Schreibens:
<copyrightA: pubMedia, publisher, coauthors|1...n>;

3. Vereinbarung über die Struktur des Inhalts bestehend aus den Inhaltsteilen und dem Akt des Schreibens:

```
<structureA: title, abstract, sections, bibliography, ...>
```

Diese Vereinbarungen finden sich auch in den Operationen wieder. Im Beispiel (Bild 6) enthält der Inzidenzgraph drei Operationen mit Angaben des Definitions- und Wertebereichs, sogenannte Termsorten, d. h. einfachen Datentypen. Die Definitionsbereichssorten $1, \dots, n$ stehen links und eine Wertebereichssorte steht rechts vom Pfeil \rightarrow , der das Termrewriting anzeigt.

1. Struktur des Dokuments festlegen:

```
DefineContent: <structureA>, authorization|1st author --> document;
```

2. Abschnitte mit Inhalten füllen:

```
WriteContent: <authorshipA>, content|i=coauthor document --> document;
```

3. Dokument veröffentlichen:

```
PreparePublication: <copyrightA, document> --> publication;
```

Wenn ein Sortenname, d. h. der Name eines Typs auf der linken und rechten Seite des Umschreibepfeils auftritt, wird die Sorte rekursiv definiert, um eine Zustandsänderung, d. h. eine Veränderung der Menge von Ausdrücken (Termen) dieser Sorte, zu ermöglichen. Zum Beispiel bedeutet die Operation Nr. 2 von oben, also das ‚Umschreiben‘ eines bestimmten Dokuments, indem diesem Dokument ein Inhalt hinzugefügt und aktualisiert wird.

Das Umschreiben von Termen wird bedingt durchgeführt. Die Bedingungen werden durch die Variablenbelegungen im Prozess (Graph Vertex) gesetzt.

Die deklarierten Sorten, werden durch Variablen repräsentiert, die einen beliebigen Ausdruck der gleichen Sorte beinhalten können. Die Sortendeklaration im Beispiel

```
SORT authorization
```

beinhaltet Ausdrücke zur Autorisierung von Ko-Autoren repräsentiert durch die Variable A.

```
VAR A OFSORT authorization
```

Damit kann ein ‚document‘ mit dem ersten verantwortlichen Autor autorisiert werden:

```
A:=1st_author.
```

Diese Instanziierung muss im geänderten ‚document‘, auf der rechten Seite der Umschreiberegeln, überprüfbar sein:

```
term of SORT document.
```

Anhang A

SemNorm Grundlagen

A.1 SemNorm Grundlagen

Die formale Grundlage des SemNorm Ansatzes bilden Algebren aus der Mathematik. Algebren können sehr gut verglichen werden mit Programmen aus der Informatik. Programme beinhalten i.d.R. Algorithmen zur Datenverarbeitung. Algorithmen bestehen aus Folgen von Regeln, angewendet auf Operationen. Ähnlich sind Algebren beschaffen: Sie beinhalten verschiedene Datenmengen, Sorten genannt, Deklarationen von Operationen, sortiert nach den Sorten und Regeln zur Anwendung der Operationen, Axiome genannt. Diese Axiome werden in der sogenannten Equational Logics der Termalgebren als Termgleichungen dargestellt. Diese Termgleichungen definieren, verwendet als Umschreiberegeln (rewriting) die Semantik eines Programms.

Die möglichen bedingten Abläufe eines Programms, die Prozesse, werden ebenfalls algebraisch als Graphen definiert. Dabei werden die Zustände eines Programms als Graphknoten (Vertices) und die Aktionen eines Programms als Graphkanten (Edges) dargestellt.

A.1.1 SemNorm Angewandte Theorien

A.1.1.1 Graphentheorie

Ein Graph besteht aus den Mengen: Knoten (Vertices) und davon verschiedenen Kanten (Edges).

Ein ungerichteter Graph ist ein Tripel $G = \langle V, E, i \rangle$, aus Ecken V und Kanten E und einer sogenannte ungerichteten Inzidenzabbildung $i: E \rightarrow P(V)$, mit $1 \leq |i(e)| \leq 2$;

Ein gerichteter Graph ist ein Tripel $G = \langle V, E, i^* \rangle$, aus Ecken V und Kanten E und einer sogenannte gerichteten Inzidenzabbildung $i^*: E \rightarrow V \times V$;

Jedem gerichteten Graph liegt also ein ungerichteter Graph zugrunde. Der ungerichtete Graph beschreibt die möglichen Operationen, die von einem Anfangs-Knoten A zu einem End-Knoten E ausgeführt werden können - der gerichtete Graph enthält dann die zwischen A und E zustande gekommene Operation. Interoperabilität zwischen Systemkomponenten wird in der Graphentheorie als Graph Manipulation einer ausführbaren Operation (eines ADT mit der Signatur Σ) an der Schnittstelle $\{A, E\}$ in ein beobachtbares Ereignis (A, E) modelliert [61:1974].

A.1.1.2 Term Algebra und Signatur

Σ -Term-Algebra und Signatur Σ stellen sozusagen Semantik und Syntax eines Abstrakten Datentyps [60:1985], [64:1978], [63:1985] dar. Terme einer 'Sprache' bezüglich einer Signatur Σ sind unabhängig von der Σ -Algebra, weil sie rein syntaktisch definiert sind. Die Σ -Terme bilden die Datenmenge einer Σ -Algebra, die daher Termalgebra genannt wird. Eine Termalgebra besteht aus den Datenmengen aller Σ -Terme für alle Mengentypen, Sorten genannt. In jedem Mengentyp gibt es ein ausgezeichneten (Null-) Term und Operationen mit n Parametersorten und einer Zielsorte. Die Termalgebra ist strukturverträglich, d. h. homomorph, mit der Σ -Algebra. Während ein Abstrakter Datentyp (ADT) semantisch betrachtet, eine Termalgebra ist, bestimmt die Signatur die Struktur des ADT, ist also sozusagen die syntaktische Benutzerschnittstelle des ADT (bzw. Der Σ -Algebra)!

ANMERKUNG Die Ähnlichkeit zwischen semantischer Repräsentation als Abstract Data Type (ADT) und Process Graph gestattet eine direkte Nutzung der *Termalgebra* (siehe Definition Anhang A.2). Diese dient als Ausdrucksmittel, d. h. eine Sprache, in der ADT Theorie Signatur genannt. und zur Darstellung der Interoperabilität und Charakteristika von Daten in den Verwaltungsschalen von Assets. Eine Termalgebra beinhaltet alle erzeugbaren Terme (Ausdrücke) einer zugrundeliegenden isomorphen semantischen Algebra, d. h. die Fähigkeiten eines Assets werden durch sprachliche Ausdrücke repräsentiert, wobei die Grammatik der Sprache ‚operationserzeugte‘, algebraische Eigenschaften hat und durch ihre Signatur syntaktisch definiert wird.

A.1.1.3 ADT-parametrisierter Graph: Graph(ADT)

Graphen werden mit Abstrakten Datentypen parametrisiert, um Variablen, Konstanten, Axiome, Regeln, Invarianten, Constraints usw. in den Graphen zur Validation unbekanntes Systemverhaltens (Theoreme) zur Verfügung zu haben [61:1974], [63:1985], [65:2019].

A.1.2 SemNorm angewandte Konzepte

A.1.2.1 Normungstexte

Nicht nur technische Normen, sondern auch Referenz-Architektur-Modelle, wie z. B. RAMI4.0, Smart Manufacturing Standards Mapping (SyC SM2 DTR 63306), oder Smart Energy (SyC SE TS 63268) können von der schärferen Differenzierung zwischen syntaktischen und semantischen Texten in der Normung profitieren.

Mit syntaktischen Texten (Symbol) ist ein Standard und mit semantischen Texten ist ein mathematisches Modell, das ggf. als Digitaler Zwilling eingesetzt werden kann, gemeint. Zwischen beiden Texten gibt es einen Morphismus der ‚syntaktische‘ beschreibende Artefakte (aus einem Standard), semantischen deklarativen Artefakten (aus einem graphtheoretischen Modell) zuordnet.

Im Prinzip ist es das gleiche wie ‚Syntax und Semantik einer formalen Sprache‘ zu definieren, was der Erzeugung eines Morphismus entspricht. Nur eben nicht a-priori für alle sprachlichen Elemente, sondern fallweise für jede Problemlösung einen eigenen Digitalen Zwilling, bzw. ein besonders konstruiertes Gerät, neu.

BEISPIEL 9 Das RAMI4.0 beschreibt auf seinen drei Achsen vertikale und horizontale Abhängigkeiten.

Der Life Cycle Value Stream bezieht sich auf die Wandlung eines Produkts entlang des Lebenszyklus. Diese ist abhängig vom Wertestrom, also von Bereitstellung von Ressourcen zur Fertigung, über die Inbetriebnahme und Instandhaltung bis zur Produkt-Obsoleszenz am Ende des Lebenszyklus. Das Produkt hat in seinem Lebenszyklus immer auch Bezug zur Hierarchie-Achse, vom einfachen Produkt bis zur vernetzten Welt und den Ebenen der technischen und geschäftlichen Interoperationsschichten.

In einem vorgeschlagenen Morphismus wird die Lebenszyklus-Achse der semiotischen Domäne Semantik, die Systemhierarchie der semiotischen Domäne der Assets und die Schichten der Interoperation der semiotischen Domäne der Schichten-Standards zugewiesen.

A.1.2.2 Semantische Programmierung

Die Graphentheorie (wie oben in Abschnitt A.1.1.1 beschrieben), bietet den Vorteil, Daten und Prozesse mit den gleichen Mitteln, d. h. mit den gleichen semantischen Artefakten darzustellen. Ein Graph setzt sich aus Knoten (engl. vertex/vertices) und Kanten (engl. edge/edges) zusammen. So wird in [52] der Typgraph (andernorts auch Informationsnetzwerk oder Knowledge-Graph genannt), vorgestellt, welcher Prozess-Knotentypen und Daten-Kantentypen enthält. In einen ungerichteten Prozessgraphen werden Folgen von Ereignissen zwischen Systemkomponenten, den Knoten, in einen Di-Graphen¹⁰ als Trajektorien, d. h. eine Aufeinanderfolge gerichteter Kanten, eingetragen (wie in Bild 8 dargestellt).

Analytisch betrachtet, stellen Abhängigkeiten und Hierarchien auf den Achsen des RAMI4.0 die semantischen Anpassungen dar, welchen ein Produktions- ‚Value Stream‘ beim Durchlaufen der Lebenszyklus-Kontexte, ein Gerät oder Produkt in einer ‚Connected World‘ oder Schnittstellen für schicht-spezifische Kommunikation, ausgesetzt sind.

Die semantischen Anpassungen zwischen den Kontexten werden in einem Graphen formal mittels Graph-Manipulationen, als Folgen von Ereignissen, sogenannten Trajektorien, in Di-Graphen¹¹ und semiformal als ‚Narrative der Trajektorien‘, dargestellt.

A.1.2.3 Operationale Semantik

Der Anwendungsnutzen einer formalen operationalen Semantik soll anhand der drei Anwendungsbereichen Digital Factory, Digital Twin und Kontrollschleifen zur Autonomisierung und Automatisierung erläutert werden.

¹⁰ Gerichtete (digraph) und ungerichtete Graphen unterscheiden sich durch gerichtete und gerichtete Kanten. Während gerichtete Kanten ein Ereignis, das vom Ursprungsknoten (head vertex) auf den Zielknoten (tail vertex) gerichtet ist, darstellen, stellen ungerichtete Kanten nur eine Beziehung zwischen zwei Knoten dar. Diese Beziehung kann zum Ereignis werden, sobald Regeln in beiden Knoten angewendet werden können und ein Ereignis eintritt.

¹¹ Ein Di-Graph (Abk: directed - gerichteter Graph) ist ein Graph mit gerichteten Kanten (directed edges), der aus einem sogenannten Typ-Graph mit ungerichteten Kanten, hervorgeht, indem Ereignisse zwischen 2 Knoten (graph vertices) beobachtet und in den Graph eingetragen werden können;

Der Begriff ‚**Digital Factory**‘ aus der Normung ist vergleichbar mit ‚Digital Twin‘, jedoch angewendet auf alle Phasen des Produktionslebenszyklus (Production Life Cycle) für verschiedene Produkt Ontologien [IEC 62832-1, IEC 62832-2, IEC 62832-3 entsprechend];

ANMERKUNG 1 Die Erklärung des Begriffs Digital Factory ergibt sich aus den noch nicht abgeschlossenen Bestrebungen, die Begriffe ‚Digital Factory‘ und ‚Smart Manufacturing‘ [24], [25], [26], IEC TR 63283-1 normativ zu definieren.

ANMERKUNG 2 Der Begriff Digital Factory wird gleichzeitig zur Normung als semantischer Begriff einer Common Logic verstanden, wobei der genormte dem semantischen Begriff entspricht.

Der Begriff ‚**Digital Twin**‘ aus der Normung wird in der semantischen Domäne des semiotischen Dreiecks [vgl. IEC 62443 IACS:02/2019] verstanden als semantischer, operationaler Repräsentant einer Klasse von technischen IT/OT (cyber-physischen) Systemen eines Asset Types;

ANMERKUNG 1 Die Normung des Begriffs *Digital Twin (DTw)* ist noch nicht abgeschlossen, verschiedene Gremien [02], [03], [04], [20] bemühen sich darum. Es wird jedoch angestrebt, die obige semantische Definition mit den technischen Definitionen der Normungsgremien in Einklang zu bringen, indem diese VDE SPEC (die ja die Morphismen zwischen den semiotischen Domänen Normungstexte, Maschineninterpretierbarkeit (DT) und Asset Engineering) angewendet werden.

A.1.2.4 ML Feed Forward Transferfunktion (Autonomisierung):

Machine Learnig (ML) ist eine Transferfunktion, wie ein Rechner/Computer spezifische Aufgaben, wie z. B. Zeichen zu erkennen oder Wein zu klassifizieren, ‚lernen‘ kann. ‚Lernen‘ steht hier für die Programmierung eines (Klassifizierungs-) Algorithmus im Gegensatz zu menschlicher Lernfähigkeit, z. B. es gibt eine Menge von Variablen mit Messungen der täglichen Wetterbedingungen, die als Ursache (dargestellt als Eingangs-Features) für das mögliche ‚Erlernen‘, d. h. Vorhersagen, einer bestimmten Wirkung (Repräsentanten aus Ereignisklassen), z. B: Energieerzeugung unter beobachteten Wetterbedingungen, verwendet werden kann.

Die ML Transferfunktion beinhaltet folgende Funktionalitäten:

- 1) Beobachtung (Example, Instanz) ist ein Objekt von Interesse, das beobachtet werden konnte,
- 2) Charakteristik (Feature) charakterisiert die Beobachtung und stellt die Eingangsvariable dar (siehe 3.2.18+1, System Definition mit Variablen),
- 3) Klasse der Beobachtungen (Label) wird von einem (Energie-) Experten bestimmt und den Beobachtungen zugeordnet,
- 4) Klassifikator (Classifier) ist eine Transferfunktion, welche die beobachteten Charakteristiken (der Eingangsvariablen) auf das Etikett eine Klasse abbildet.

Daraus kann gefolgert werden, dass es möglich ist, unbekannte Beispiele, zu klassifizieren, bzw. ‚zu lernen‘.

Feedback Transferfunktion (Automatisierung): Durch permanente Rückführung der Wirkung einer Maschine oder Dings zu einem Aktor wird die Anpassung der Versorgung der Maschine an ein gewünschtes Ergebnis automatisiert. Die Entscheidungsfindung, d. h. der Algorithmus, ist mit dieser Struktur fest verbunden.

Rückkopplungen werden in der graph-theoretischen Darstellung als Schleifen repräsentiert.

A.1.3 SemNorm semiotische Bezüge (Morphismen)

A.1.3.1 Semiotik in der I4.0 Normung

Wie schon das sogenannte semiotische Dreieck der Sprachwissenschaften (erfunden 1923, siehe Wikipedia) die Relationen *einer natürlichen Sprache* zwischen Symbolen (1), Dingen (2) und Begriffen (3), wie in Bild B.4, darstellt, so werden auch im Bereich von I4.0 Systemen die *Relationen eines technischen Systems*, zwischen beschreibenden Standards (1), technischem Gerät (2) und analysefähiger semantischen Darstellung, im Modell (3) explizit dargestellt.

ANMERKUNG 1 Die Relationen zwischen den drei genannten Domänen haben besondere strukturelle Eigenschaften. Daher werden sie mit einem Begriff aus der Algebra, als **Morphismen** bezeichnet, die im semiotischen Dreieck in Bild 1 und 10, zwischen Normen in Textform (1), der realen (cyberphysischen) Welt der Dinge (2) und der Informationswelt des Modells, der Welt der maschinellen Interpretation von Modellen (3), semiotische, also bedeutungsvolle Beziehungen herstellen.

ANMERKUNG 2 Durch die Verbindung von der Welt der Normen mit der digitalen und der realen physischen Welt, entsteht in Anlehnung an die sprachwissenschaftliche Semiotik, eine ‚digitale Semiotik‘, wie in Bild 1 und 10 dargestellt. Sie ist eine angepasste Darstellung der VDE SPEC *SemNorm* für die Erstellung maschinell interpretierbarer Normen: Darin werden drei unabhängige *Darstellungsformen*, semiotische Domänen genannt, zueinander in Bezug gesetzt: 1) Beschreibende Ontologien und Standards (Farbe: grün), 2) technische Lösungen von z. B. Produktions- oder Energieübertragungs-Systemen bei Verwendung von administrierten assets (AAS) (Farbe: grau) und 3) eine geeignete digital ausführbare Darstellung der Semantik (Farbe: blau).

Jede semiotische Domäne hat ihre ‚eigene Sprache‘, bestehend aus einer Menge von Bausteinen, Artefakte genannt. Mit diesen Artefakten wird in jeder Domäne ein ‚Modell‘ gebaut, das in seiner jeweiligen Ausprägung ein textueller System-Standard (1), eine technische Smart Factory (2) oder ein semantisches Modell (3) sein kann.

Die semiotischen Bezüge zwischen Domänenmodellen gelten auch für einfache Zuordnungen, wie sie in Bild B.4 zwischen ‚Begriff – Symbol – Ding‘ besteht, bzw. hergestellt werden kann, wobei Begriff für Semantik, Symbol für Standard und Ding für Technik synonym stehen.

Jede einzelne Darstellung in den drei Domänen ist jedoch abhängig von den jeweils anderen Darstellungen in den anderen Domänen. Diese Abhängigkeiten werden mittels konstruierter Zuordnungen, den Morphismen, definiert. Die gefundenen **Morphismen implementieren** sozusagen textuelle Anforderungen in einem Gerät, das diese Anforderungen erfüllt. Deshalb spricht man auch von sogenannten ‚Implementierungsmorphismen‘, weil Artefakte der einen Domäne auf Artefakte der anderen implementierenden Domäne abgebildet bzw. implementiert werden.

A.1.3.2 Darstellung von OT/IT Datenflüssen in der realen Asset Domäne

Während die operative Technik (OT) gekennzeichnet ist, durch die Kontrolle von Strömen von Kontingenten, Objekten, Werkstücken, Energie, Gas-Volumina, i.d.R. ausgetauscht werden zwischen Menschen, Dingen oder Maschinen, wie z. B. bei den Ansätzen des *Smart Manufacturing (SM)* oder *Smart Energy (SE)* etc. - ist die Informationstechnik (IT) gekennzeichnet durch die Datenverarbeitung, wie z. B. bei Ansätzen von AI/ML, IoT, Smart City etc

BEISPIEL 4 ‚ML‘ lernt ein Datenmodell, als Muster für die Klassifikation unbekannter Datenmuster.

Das IoT führt in seinen Sensornetzen Messungen aus, die es in seinen Applikationen zur Entscheidungsfindung verwendet.

BEISPIEL 5 Das Smart Manufacturing Systemautomatisierungsmodell

gewinnt aus Messdaten und einer Wissensbasis (Datenmodell) Signale zur Steuerung eines Systems.

Der Regelungsalgorithmus (z. B. PID) geniert aus dem Soll/Ist Vergleich einen Stellwert, der dann an das System zurückgegeben wird, d. h. die Abweichungen führen zu entsprechenden Steuersignalen.

Zwischen OT ‚Objekt Value-stream‘ und IT ‚Daten Value-stream‘ gibt es einen grundsätzlichen Unterschied: Während des OT Objekt Value-stream (siehe Graph Trajektorien in Abschnitt 5.1.7) werden laufend Daten gesammelt, die charakteristisch sind, für den jeweiligen Verarbeitungs- bzw. Veredelungsschritt des Objekts. Im Daten Value-stream dagegen ist das Datensammeln überflüssig, weil statt des Objekts die Daten verarbeitet werden.

Jedoch teilen sich OT und IT das Problem der semantischen Interoperabilität. Semantisch interoperabel heißt, dass sich kommunizierende Komponenten zwar eine Semantik (d. h. eine gemeinsame semantische Darstellung) teilen, sie aber eigene strukturverträgliche Signaturen haben können. Konkret heißt das, dass die Schreibweisen in verschiedenen kontextspezifischen Ontologien voneinander abweichen können, solange die semantischen Strukturen über den gegebenen semiotischen Morphismus erfüllt werden können.

ANMERKUNG 1 Die Abwesenheit einer gemeinsam benutzten semantischen Darstellung in der Normung (z. B. Graphtheorie, Logikkalküle etc.) führt geradewegs in diese Problematik, dass syntaktische Strukturen mit syntaktischen Meta-Strukturen beim ‚*Context Change*‘ syntaktisch erklärt werden sollen, statt eine gemeinsame semantische Darstellung zu wählen, worauf sich Norm mit ihren Anforderung, das Asset mit seiner funktionalen Sicherheit und das Informationsmodell mit seiner semantischen Interpretierbarkeit beziehen zu können. Alle haben andere Anforderungen, die aber mit den gleichen semantischen Mitteln ausgedrückt werden können.

Die Graph-Semantik ist genau wie die in der Automatisierungstechnik übliche Beschreibung als Differentialgleichung oder die der Zustandsraumbeschreibung eine mathematische Semantik (siehe Anhang A). Folglich kann sowohl der Mensch, der das mathematische Modell ‚versteht‘, als auch die

Maschine, die das operationale Modell ‚versteht‘¹², das betrachtete semantische Modell (z. B. einer I4.0 OT/IT Verarbeitungsanlage) in gleicher Weise logisch, berechnend oder simulativ interpretieren.

¹² Es ist klar, dass eine Maschine, im menschlichen Sinne, nichts verstehen kann. Die synonyme Verwendung des Verbs ‚verstehen‘ bezieht sich auf die unterschiedlichen Fähigkeiten des Menschen und der Maschine. Der Mensch versteht einen Text, weil er Ursache und Wirkung unterscheidet und die Maschine interpretiert einen Text, weil sie einen gegebenen Text in einen anderen, ggf. ‚kürzeren‘ Text umschreiben kann. Dieser Vorgang endet, wenn eine kanonische Darstellung des Textes gefunden werden kann.

Anhang B SemNorm Engineering Guidelines

B.1 SemNorm Engineering

Der Entwurf, die Entwicklung, die Inbetriebnahme, das Betreiben und Instandhalten von technischen Systemen sind **ingenieurtechnische Aufgaben**. Grundlage für alle Tätigkeiten sind Modelle, die bei dieser Arbeit unterstützend wirken. Die Modelle beschreiben sowohl strukturelle als auch Verhaltensaspekte der technischen Systeme.

B.1.1 Systemmodellierung

Ein sehr grundlegender Ansatz kommt aus der Systemtheorie und findet in weiten Teilen der Modellierung technischer Systeme Anwendung. Dabei steht die Modellierung des Verhaltens im Vordergrund, das durch die Reaktion von Ausgangsgrößen auf Eingangsgrößen und die Veränderung der inneren Zustände charakterisiert wird. Das Verhalten wird unter anderem durch **Differentialgleichungssysteme** beschrieben. In diesen Differentialgleichungen ist die im Modell abgebildete Bedeutung aller **Zustandsgrößen, Variablen und Parameter** sowie Ein- und Ausgangsgrößen enthalten. Bedeutung heißt, dass der Einfluss einzelner Größen auf das Verhalten des Gesamtsystems eindeutig und formal definiert ist. Das heißt auch, dass die **Semantik** dieser Größen durch deren Verwendung in den Systemfunktionen, die durch die Modelle beschrieben werden, gegeben ist.

Werden beim Systementwurf mehrere Teilsysteme miteinander gekoppelt, so sind bestimmte Ausgangsgrößen eines Systems mit Eingangsgrößen anderer Systeme zu verbinden (Bild B.1). Das ist gerade die Aufgabe, die bei der Entwicklung von technischen Systemen die ingenieurtechnische Leistung ausmacht.

Eine Notationsform der Differentialgleichungen (DGLs) ist der **Zustandsraum**, in dem die Differentialgleichung höherer Ordnung auf eine gekoppelte Reihe von Differentialgleichung 1. Ordnung umgewandelt werden (für ein näheres Verständnis, siehe z. B. Lunze [66] oder andere regelungstechnische Standardliteratur, wie B. C. Kuo [67] usw.). Dabei bedeuten, wie in Bild 7 dargestellt:

\mathbf{x} Zustandsgrößenvektor, \mathbf{u} Eingangsgrößenvektor, \mathbf{y} Ausgangsgrößenvektor (f 15).

Die Matrix A und die Koeffizientenvektoren \mathbf{b} , \mathbf{c} , \mathbf{d} enthalten die Parameter, die die Kopplung der Gleichungen 1-ter Ordnung enthalten. Damit ist ein technisches System so vollständig beschreiben, wie es für die Modellierung benötigt wird.

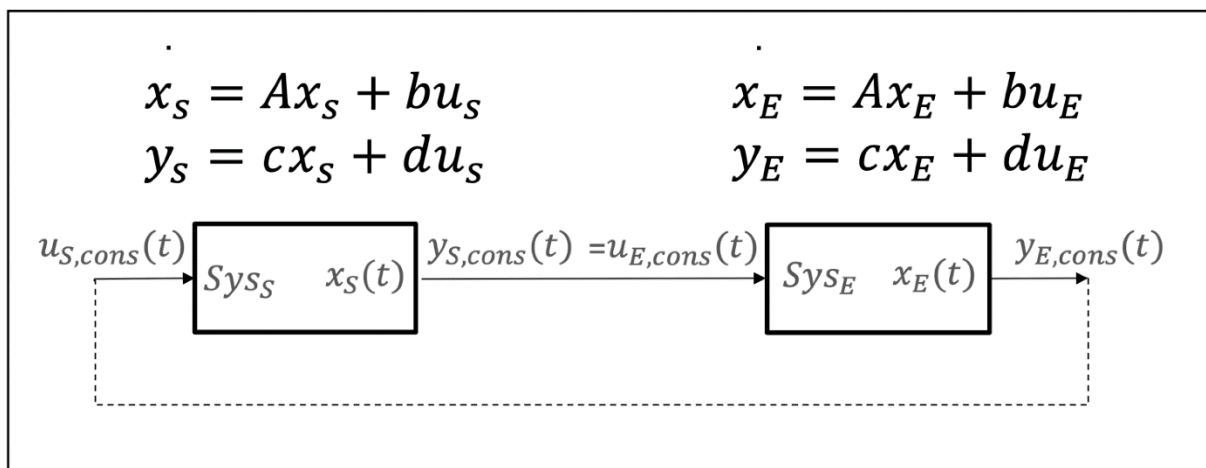


Bild B.1 – Systemmodellierung mit DGLs im Zustandsraum

Die im Modell definierten Systeme müssen in reale Systeme umgesetzt werden. Dies wird auch als **Engineering von Systemen** (vgl. [78] Absatz ‚on modelling for security engineering‘) bezeichnet und im semiotischen Dreieck (Bild 1, 10) mit ‚Morphismus‘ bezeichnet.

B.1.2 Systemvariablen

Für die im Modell beschriebenen Teilsysteme müssen reale Systeme entsprechend den intendierten Vorgaben zusammengesetzt werden. Ein großer Teil dieser Umsetzung erfolgt heutzutage durch

Softwaresystemen, die hier im weiteren Fokus stehen. Dieser Teil des Gesamtverhaltens wird durch programmierte Funktionen umgesetzt, deren funktionale Eigenschaften deklariert werden müssen und in denen die **Variablen** (in der Programmierung die Umsetzung für Ein- und Ausgänge, Parameter und Zustandsgrößen) verwendet werden.

Die Variablen sind ebenfalls zu deklarieren und mit Initialwerten zu versehen (siehe Bild B.2). Während die Funktionen den prozeduralen Anteil der Funktionalität darstellen, können die Variablen in Informationsmodellen dargestellt werden. Dies wird vor allem beim Engineering benötigt, da die Parameter und Ein- und Ausgangsgrößen jeweils die sie repräsentierenden Systemkomponenten näher beschreiben. Beim Engineering reicht oft diese Abstraktion aus.

Im linearen Verhaltensbereich können algebraische Operationen auf Variablen, wie Addition, Subtraktion und Multiplikation definiert werden. Während Addition und Subtraktion unmittelbar im Zeitbereich der Variablen angewendet werden können, braucht man für die Multiplikation eine Laplace-Transformation in den komplexen Bereich¹³, sodass die Multiplikation, die für die feedback-Schleifen eines industriellen Automations- und Kontrollsystems (IACS) benötigt werden, auch berechnet werden können.

Somit können Addition, Subtraktionen und Multiplikation in der Laplace-Ebene miteinander algebraisch kombiniert und auf alle Variablen angewendet werden. Komplexe Systeme mit mehreren Kontrollschleifen können auf eine kanonische Darstellung eines Systems mit einer einzigen Kontrollschleife, d. h. einer Multiplikation, bzw. *Konvolution*, reduziert und berechnet werden.

Es wird hier auch von sogenannten **Merkmalen** gesprochen, das sind quantifizierte Eigenschaften, die in den Informationsmodellen verwendet werden. Reicht die Repräsentation der Systeme durch die Merkmale nicht aus, um die Entwurfsentscheidungen zu treffen, werden z. B. Simulationen verwendet, die dann das Systemverhalten viel detaillierter nachbilden und zur Entscheidungsfindung beitragen.

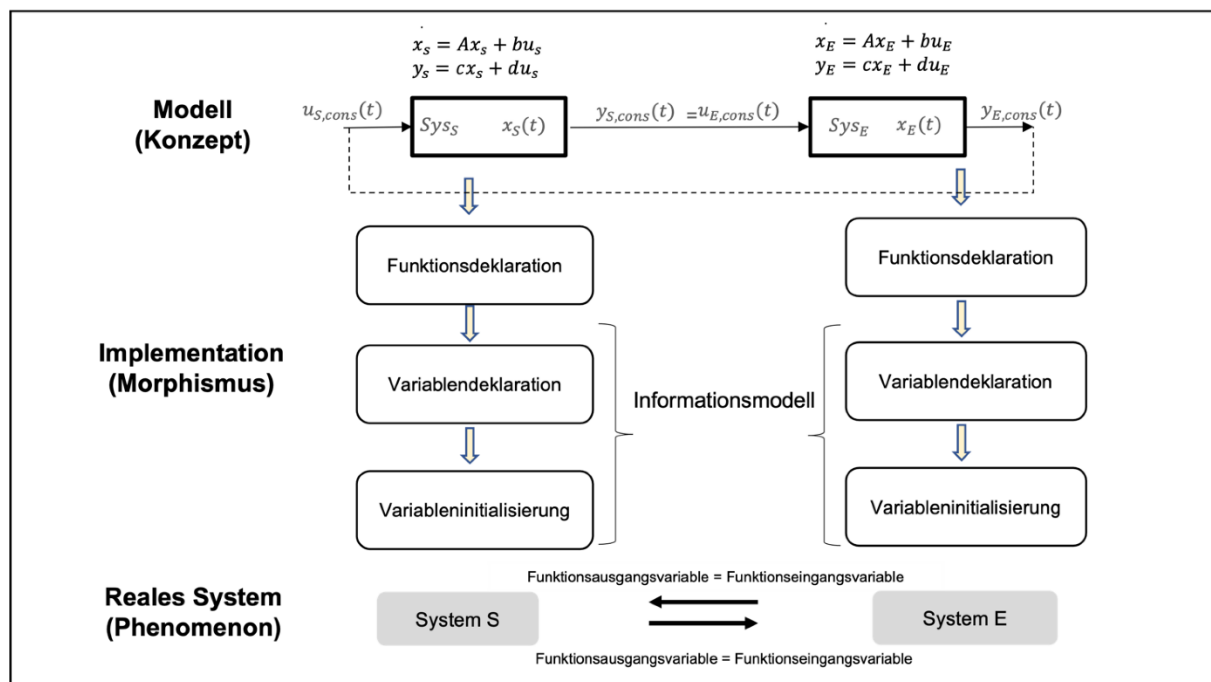


Bild B.2 – Morphismus als Transformation eines semantischen Systemmodells in ein reales System

Ein derart erweitertes **Informationsmodell** ist beispielhaft in Bild B.3 dargestellt. Durch farbliche Kennzeichnung ist der Bezug zu dem Systemmodell hergestellt.

B.1.3 System-Attribute

Bei dem, in Bild B.3 und B.5 dargestellten Beispiel, handelt es sich um ein *Heizungssystem*, das aus zwei Heizungen besteht. Jede Heizung ist mit einem Sensor ausgerüstet, der die Temperatur misst und mit einem Aktuator, der die Ansteuerung des Heizers mit der erforderlichen Energie ermöglicht.

¹³ Zur Unterscheidung wird die Multiplikation in der komplexen Ebene (komplexe) Konvolution genannt.

Die Eigenschaften der Heizung sind im Klassendiagramm des Informationsmodells als **Attribute** enthalten (z. B. Wärmekapazität). Diese Eigenschaften sind in den Koeffizienten der Matrix A bzw. des Vektors c rechnerisch enthalten. Die Temperatur des Sensors ist die Ausgangsgröße des Systems und die Leistung (Power) die Eingangsgröße. Nicht berücksichtigt im Informationssystem sind hier andere Ein- und Ausgangsgrößen, die durch den Austausch des Systems mit der Umwelt, z. B. durch Konvektion eine Rolle spielen. Alle diese Größen werden in den Funktionen verwendet. Es ist Aufgabe im Engineering z. B. die richtigen Ein- und Ausgänge miteinander zu verbinden und die Parameter mit den richtigen Werten zu versehen.

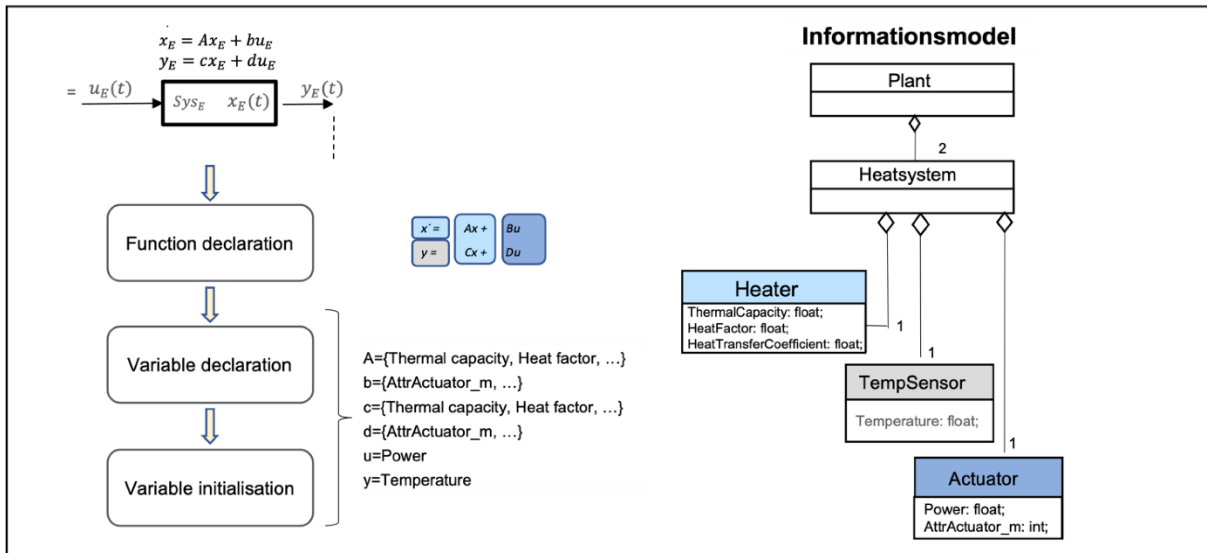


Bild B.3 – Informationsmodell für Systemeigenschaften

Die in dem Informationsmodell enthaltenen Daten sind Grundlage für das Engineering. Beispielsweise werden im Engineering, wie im Modell enthalten, 2 Heizungen verwendet. Das Engineering hat beide Heizer auseinanderzuhalten und jeweils mit den für diese zutreffenden Variablenwerte zu initialisieren. Außerdem sind die entsprechenden Sensoren und Aktoren als Ein- und Ausgangsgrößen zuzuordnen.

B.1.4 Semiotisches Dreieck als Konzept für die Normung

Die Abstraktion der Systeme und ihrer Eigenschaften im Engineering basieren heutzutage auf **linguistischen Ansätzen**. Das **Semiotische Dreieck** ist dafür ein viel verwendetes Modell. Es beschreibt die Zusammenhänge zwischen dem Objekt der Betrachtung, seinen Konzepten und symbolischen Darstellungen. In Bild B.4 wird das Objekt der Betrachtung mit ‚Ding‘ bzw. Phänomen bezeichnet, die Benennung des Objekts, wird mit ‚Symbol‘ repräsentiert und die begriffliche Vorstellung des Dings mit ‚Begriff oder Konzept‘.

Am Beispiel ist das geschriebene Wort „Wärmekapazität“ das *Symbol*, die Stoffeigenschaft das zu benennende und beschreibende *Ding* und der *Begriff* durch eine Definition und eine Gleichung gegeben. Im Begriff stecken die semantisch relevanten Aussagen. Das Symbol ist nur der Stellvertreter des Dings und verweist auf den Begriff. Wird das Symbol z. B. durch einen eindeutigen Identifier gebildet (z. B. eine IRDI oder URI) kann der Bezug zu einem standardisierten Begriffswörterbuch (*ontological dictionary*) hergestellt werden, in dem sich alle benötigten Definitionen einer Domäne, z. B. Wärmegeräte, befinden.

Die Wärmekapazität eines **Körpers** ist das Verhältnis der ihm zugeführten **Wärme** zu der damit bewirkten **Temperaturerhöhung**. Die Einheit der Wärmekapazität ist **J/K**.

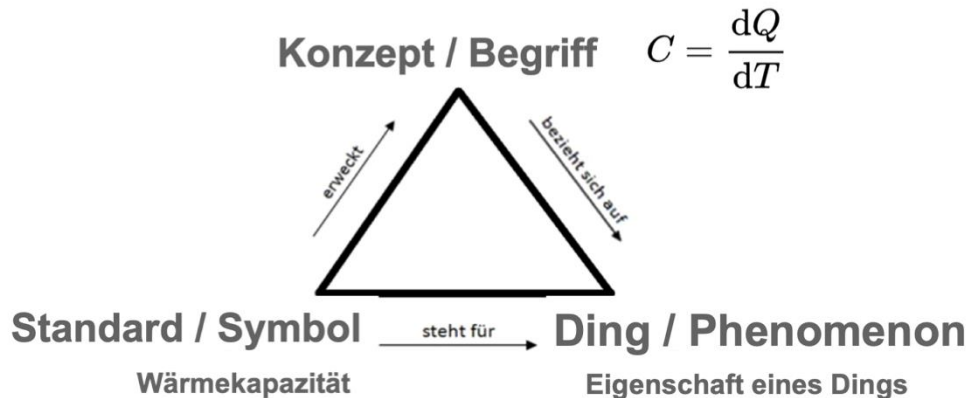


Bild B.4 – Semiotisches Dreieck am Beispiel ‚Merkmal Wärmekapazität‘

Informationsmodelle in beschreibender Notation sind semi-formal. Will man mit formalen Mitteln arbeiten, kann dieses Modell z. B. in **Description Logic (DL)** übertragen werden. Damit könnten entsprechende Analyse- und Schlussfolgerungsmethoden auf die Modelle angewendet werden, wie z. B. formale *reasoning Werkzeuge* oder Abfragesprachen (Bild B.5).

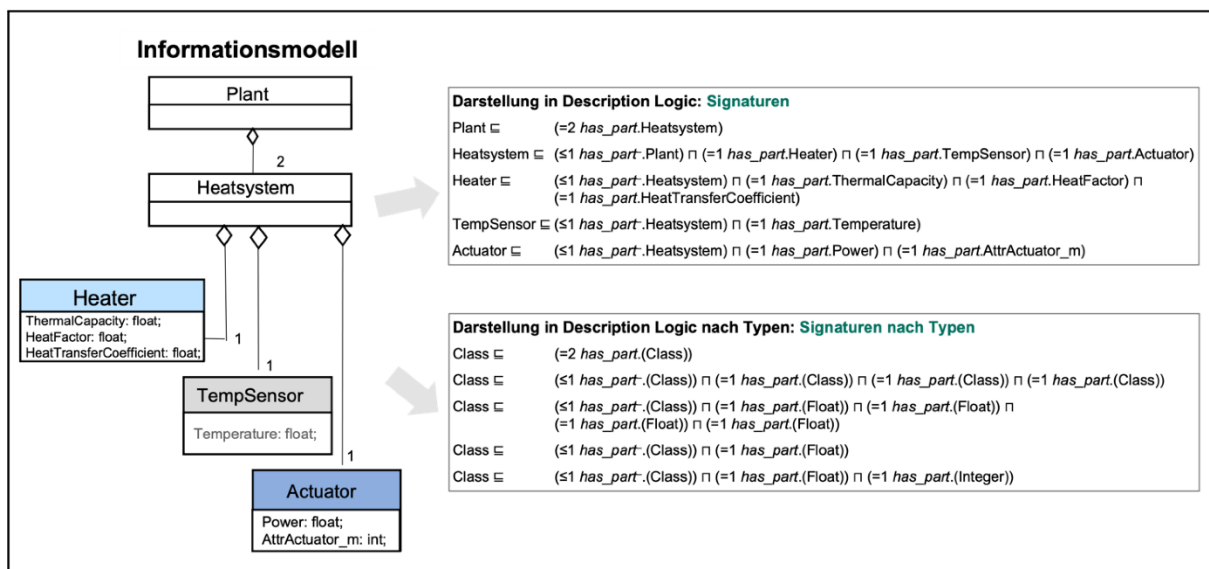


Bild B.5 – Heizungs-Informationsmodells in formaler DL¹⁴ dargestellt

Die Informationsmodelle sind von struktureller Natur. Praktisch heißt das, dass die dynamischen Verhaltensaspekte so nicht modelliert werden können. Aus semantischer Sicht ist jedoch die Verbindung zu den gewollten oder ungewollten Veränderungen der Zustandsgrößen von besonderem Interesse.

B.1.5 Graph Trajektorien und Graph-Zustandsräume

Die Änderungen der Zustandsgrößen über die Zeit sind die interessierenden **Trajektorien**, die durch die Modellgleichungen, wie hier dargestellt, im **Zustandsraum** berechenbar sind. Dazu bietet die

¹⁴ Description Logic [W3C]

klassische Regelungs- und Steuerungstheorie die benötigten Hilfsmittel. Dies gilt sowohl für das zeitkontinuierliche und zeitdiskrete Verhalten von Systemen (für ein näheres Verständnis siehe z. B. Jan Lunze in [66] und [65]).

Daraus ergibt sich die Option der Beschreibung dynamischer Vorgänge mit **Graphen** und darauf operierenden Algorithmen. Mittels **Graph Manipulationswerkzeugen** (siehe *KIT's GrGen.net* und *TUB's AGG* etc.) können Systeme und ihre Trajektorien, die auf graph- und mengentheoretischer Grundlage modelliert worden sind, in die Vergangenheit (reasoning) und in die Zukunft (prediction) simuliert und analysiert werden.

In Bild B.4 und Bild 1 besteht zwischen ‚*Begriff – Ding – Symbol*‘ der gleiche Zusammenhang wie zwischen dem Modell, dargestellt mit **operationaler Semantik** für die Simulation, der cyber-physischen Realität, ggf. mit den AAS-administrierten Assets und den beschreibenden Standards mit den normativen Anforderung an die möglichen, gewollten oder ungewollten Zustandsveränderungen im Lebenszyklus von Produktion oder der nachhaltigen Nutzung von Dingen bzgl. Sicherheit, Energieverbrauch oder Verfügbarkeit etc.

B.1.6 Common System Logics

Zusammenfassend kann festgestellt werden, dass der Ausgangspunkt aller semantischen Betrachtungen von technischen Systemen das **Systemverhalten** und somit Gegenstand der semantischen Modellierung, ist. Dies entspricht der Betrachtung der operationalen Semantik. Im Engineering wird mit Informationsmodellen gearbeitet, die die strukturellen und datentechnischen Aspekte der Systeme modellieren, soweit dieses Abstraktionsniveau ausreichend für gestellte Aufgabe ist. Dies entspricht der Betrachtung der denotationalen Semantik. Datenmodelle können auf der Basis von **Abstract Data Types (ADTs)** und dynamische Transformationsmodelle auf der Basis von Graphentheorie definiert werden.

Für die formale Darstellung der Informationsmodelle sind **Beschreibungslogiken (Description bzw. Common Logic)** geeignet. Simulation dynamischer Systemmodelle kann im Engineering ebenfalls nützlich sein. Für die Betrachtung des Verhaltens eines Systems sind die **Trajektorien der Zustandsgrößen** geeignet, die durch Graphen beschrieben werden können.

Anhang C

Use Case ‚Smart Grid‘ dargestellt in ‚CSlang‘ Notation

‚CSlang‘ ist der ETSI ISG ISI Ansatz einer ‚Common Specification Language‘, die zur Kategorie der algebraischen Notationen gehört. CSlang findet Anwendung für eine ‚Common (Specification) Logic‘ bzw. einer ‚Common Language‘ zur semantischen Darstellung von Normungstexten oder des ‚Digital Twin‘, d. h. eines operationalen (maschinen-interpretierbaren) Modells.

C.1 Formale Spezifikation eines Smart Grid Use Case

In der nachfolgenden Spezifikation sind die 5 Smart Grid (SG) Komponenten, ‚stakeholder sh‘ genannt (1): ‚Volatile Wetterbedingungen (VWC)‘, ‚Rotor (ROT)‘, ‚elektrische Windenergiegenerierung (WEG)‘, ‚Verteilte Energie Ressourcen (DER)‘ zur Energieübertragung, ‚Hausenergieverbrauch (HOME)‘ als Variablen spezifiziert worden. Die Variablen definieren die Stakeholder Kontexte, indem Transformationsregeln, gekennzeichnet mit dem Symbol ‚ \rightarrow ‘ (auch ‚rewrite rules‘ genannt) hinzugefügt werden (2).

Diese Variablen werden als Graph Vertices (Knoten) dargestellt, die mit anderen Variablen in anderen Knoten kommunizieren, dargestellt als Graph Edges (Kanten), vorausgesetzt die Interoperations-Signaturen (3) und die aktualisierten Regeln (4) versetzen die Komponenten VWC und ROT in einen Zustand, in welchem die Energietransformation von kinetischer (VWC) in mechanische Energie (ROT) möglich ist.

C.1.1 Identifikation der Stakeholders (Graph Vertices)

```
{sh| VolatileWeatherConditions,  
  Rotor,  
    WindEnergyGeneration,  
      DistributedEnergyResource,  
        HomeEnergyConsumption};
```

C.1.2 Stakeholder Kontexte mit ihren kontinuierlich-veränderlichen Eigenschaften

- a. **VWC:=** (VAR E_{WIND} OFSORT Joule [kp m],
 VAR m OFSORT mass [kp sec²/m],
 VAR v OFSORT windspeed [m /sec];
 WINDBEWEGUNG: [true] ($\frac{1}{2} (m \dot{v}^2)$) $\rightarrow E_{WIND} \dots$);
- b. **ROTOR:=** (VAR E_{ROT} OFSORT Joule,
 VAR A OFSORT plane [m²],
 VAR p OFSORT density [kp sec²/m⁴],
 Rotor**LEISTUNG:** [v isin speedWindow] ($\frac{1}{2} A (p \dot{v}^3)$) $\rightarrow P_{ROT} \dots$);
- c. **WEG:=** (VAR E_{GEN} OFSORT Joule [kWh],
 VAR M OFSORT drehmoment [kp m],
 VAR Ω OFSORT winkelgeschwindigkeit [¹/sec],
 VAR n OFSORT drehzahl [¹/sec];
 ROTATIONSENERGIE: [M, J constant] ($\frac{1}{2} M \Omega^2$);
- d. **DER:=** (VAR E_{DER} OFSORT impedanz [kWh]

C.1.3 Interoperabilitäts-Signatur zwischen den Komponenten VWC und ROT

- a. VWC-ROT SIGNATURE:= (
 VAR m OFSORT mass [kp sec²/m],
 VAR p OFSORT density [kp sec²/m⁴][kg/m³],
 VAR v OFSORT windspeed [m/sec];
 INTEROPER OFSORT r(harvestPower [kp m/sec]) WHEN (E_{WIND}) \rightarrow (P_{ROT});

C.1.4 Einfügung einer Kante zwischen zwei interaktionsfähigen Knoten

{VWC, ROT} → (VWC, ROT) bedeutet: r(RotorliefertEnergie) WHEN (Windbewegtsich → Rotordrehtsich).

Anhang D

Use Case Vergleich (Bild D.3) von heterogenen Informationsmodellen ,Heizung‘ (Bild D.1) und ,Fahrrad‘ (Bild D.2)

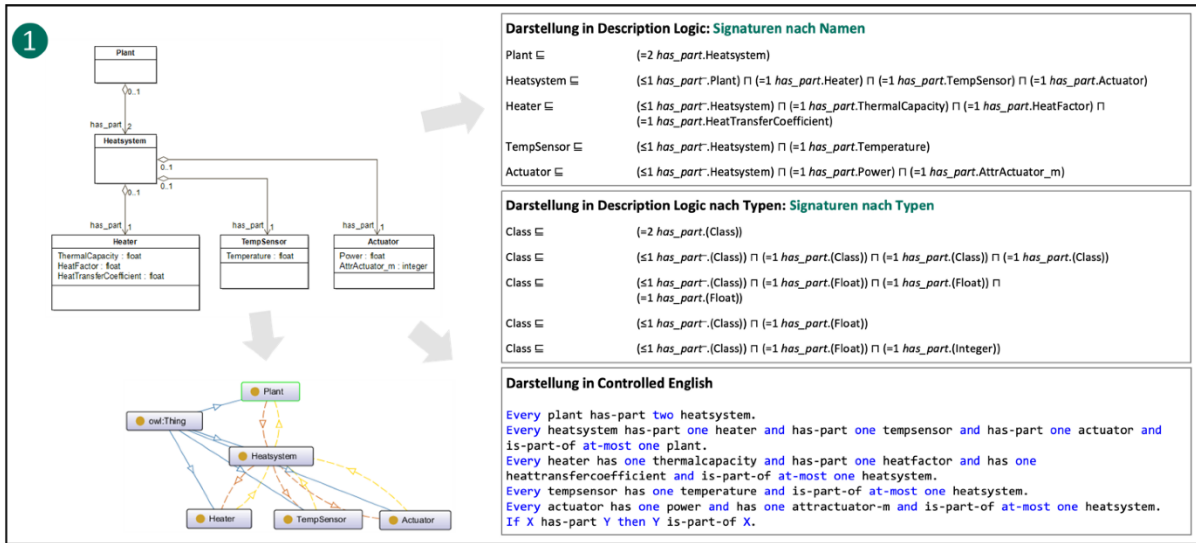


Bild D.1 – Informationsmodell ,Heizung‘ in formaler DL dargestellt

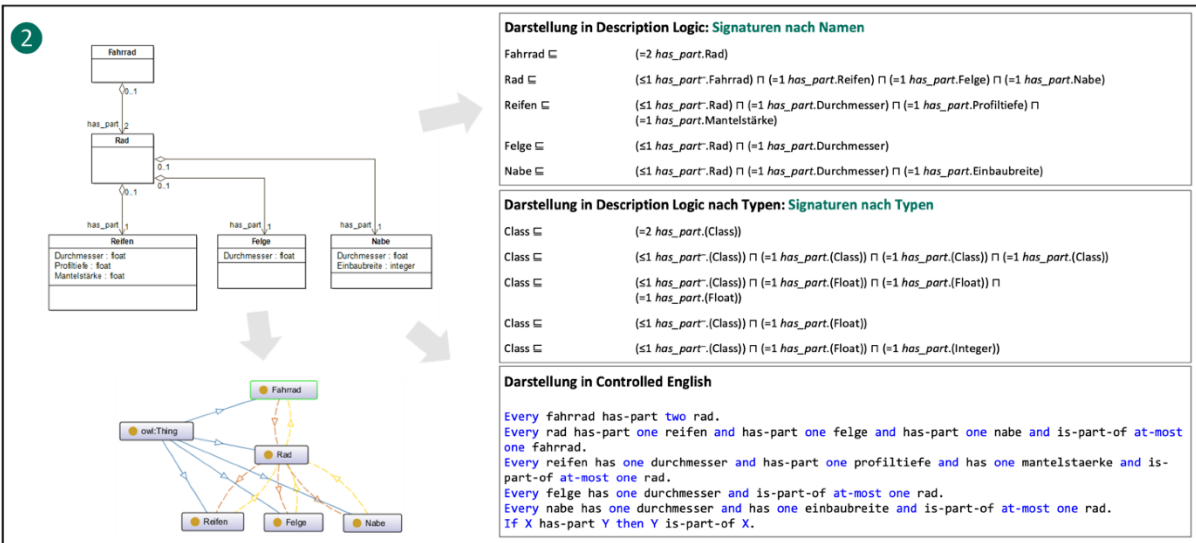


Bild D.2 – Informationsmodell ,Fahrrad‘ in formaler DL dargestellt

- ① Informationsmodell Heizung
- ② Informationsmodell Fahrrad

Vier Stufen der Heterogenität zwischen Modellen:
Syntaktisch - Terminologisch - Konzeptuell - Semiotisch

Betrachtung	Vergleich	Betrachtung der (automatisierbaren) Unterscheidbarkeit nach jeweiligem Kriterium vor dem Hintergrund "Semiotisches Dreieck"
Logische Struktur (Signatur) der Partonomie (Gesamtsicht über Klassen hinweg)	Sowohl bei ① als auch bei ② zweistufige identische Klassenpartonomie mit identischen Kardinalitäten. Nimmt man die Attribute hinzu, ergibt sich eine dreistufige Partonomie mit identischen Blattstrukturen für die Attribute auf der dritten Ebene.	Nicht unterscheidbar – Konzeptebene identisch; Modell ② könnte für Domäne ① eingesetzt werden, ohne dass der Bedeutungsunterschied erkannt werden könnte. Der Unterschied zwischen Heatsystem und Rad ist darüber nicht erkennbar.
DL-Signaturen der Konzepte (per Klasse)	Sowohl bei ① als auch bei ② identische Operatorstrukturen. Namen sind unterschiedlich.	Bei Prüfung auf Namen möglich wg. Differenzen der Symbole; gleichzeitig keine Erkennung von Bedeutungsunterschieden.
Signaturen nach Typen (und Typpräferenzen in DL ClassExpressions)	Sowohl bei ① als auch bei ② identische Typen (Ranges) bei ebenfalls identischen Operatorstrukturen.	Nicht unterscheidbar – vollständig identische Strukturen.
Signaturen nach Typen (erweitert um DL-Domain-Betrachtungen)	Werden die Domains mit hinzugezogen, ergeben sich Unterschiede zwischen ① und ② auf der Blattebene (Domain eines float-Attributs ist eine Union in ② wo bei ① nur Einzelklassen sind), da im Metamodell weniger verschiedene Attribute definiert werden müssen.	Da bei ② weniger wiederverwendbare Datentyp-Attributdefinitionen (4 statt 6) erforderlich sind, lässt sich ein Unterschied zu ① erkennen, auch wenn Namen und Partonomien identisch sind.

Bild D.3 – Vergleich heterogener ‚Heizung‘ – ‚Fahrrad‘ Informationsmodelle

Literaturverzeichnis

- [01] PLM Automation Siemens: <https://www.plm.automation.siemens.com/global/de/our-story/glossary/digital-twin/24465>
- [02a] NA 043-01-27AA N9877 – JTC1/SC27 N19734: JTC1 Hawaii Resolution#20 on JTC1 AG11 Digital Twin Call for Participation
- [02b] NA 043-01-27-05AK N4407: Establishing a JTC1 Advisory Group (AG#11) on Digital Twin, Experts to be nominated by JTC1 NBs
- [03] NA 043-01-42AA N2380: Sha Wei's Open Plenary Report from JTC1/AG11 Digital Twin Plenary Tokyo, October 2019
- [04] TC65 WG23 CS N002_v13:20-02-21 CD on Smart Manufacturing Requirements for Cyber Security, section 7.3.4 '**digital twin**', section 8.8 'resource availability' are both related to DT but in the context of CS
- [05] DINCONNECT 2020 SemNorm Project#602668: slide '**Semiotic Triangle**' presented to TC65 WG23 → see Annex A
- [06] IEC TC65 WG24 N002 **Asset Administration Shell**
- [07] ETSI GS ISI006 ISI-Measurement and Event Management Architecture and a **Common ISI Semantics Specification Language ,CSlang'**
- [08] DINSPEC 91406 on Automatic Identification of Physical Objects: <https://www.beuth.de/en/technical-rule/din-spec-91406/314564057>
- [09] IEC 62443 multipart standard on Security for Industrial Automation and Control Systems (IACS): <https://www.dke.de/de/arbeitsfelder/industry/iec-62443-cybersecurity-industrieautomatisierung>
- [10] Jan deMeer smartspacelab.eu GmbH, Karl Waedt AREVA GmbH: Security and Privacy Benchmarking based on IEC 62443-4-2, GI-Jahrestagung/DACH Security 2016 Herausgeber P.Schartner, Syssec(2016)pg.253-263
- [11] Pierre Kobes, Siemens AG, DKE member: Leitfaden Industrial Security: IEC 62443 einfach erklärt, VDE-Verlag 2016
- [12] Roland Heidel, Michael Hoffmeister et al: Industrie4.0 – The Reference Architecture Model RAMI4.0 and Industry4.0 Components, Publisher DIN-Beuth
- [13] IEC White Paper: **Semantic Interoperability**: Challenges in the Digital Transformation Age, <https://din.one/display/MNB>
<https://din.one/pages/viewpage.action?pageId=38865607>
- [14] DIN Mitteilungen August 2019: **Workshop ,Digitale Normen'** - Austausch zwischen Wirtschaft, Wissenschaft und DIN zur Arbeit mit **Normen in der Zukunft**, an der Uni Duisburg-Essen - LS Produktentstehungsprozesse und Datenmanagement, 30.8.2019
- [15] Leslie Lamport, A simple Approach to Specifying Concurrent Systems, December 25, 1986
- [16] ISO/IEC JTC1/SC41 N0676 FDIS Ballot 21823-1 IoT Framework
- [17] ISO/IEC JTC1/SC41 N1136 FDIS Ballot 21823-2 IoT Transport Interoperability
- [18] ISO/IEC JTC1/SC41 N1116 3rd CD 21823-3 IoT **Semantic Interoperability**
- [19] ISO/IEC JTC1/SC41 N1020 NP Ballot 21823-4 IoT **Syntactic Interoperability**
- [20] ISO/IEC JTC 1/SC 42 N 404 (NA 043-01-42 AA N 2380): Report of the JTC 1/SC 42 Representative to JTC 1 AG 11 – **Digital Twin**; Other Standardization Groups dealing with DT/DM → ISO TC184/JWG 21 – IEC TC65/JWG 21 Smart Manufacturing Reference Model, ISO TC184/AhG 02 **Digital Twin**, ISO TC184/SC 04/WG 15 **Digital Manufacturing**
- [21] ISO/IEC JTC1/SC42 N485 22989:2019(X) **AI Concepts and Terminology**
- [22] Bundesverband WindEnergie BWE Berlin: Anlagentechnik - **Energieumwandlung**
<https://www.wind-energie.de/themen/anlagentechnik/funktionsweise/energieumwandlung/>
- [23] NA043-01-41AA N1741 – JTC1/SC41 N1031 DTR 30166 IoT - **Industrial IoT**

- [24] IEC 62832-1:2020-10, *Industrial-process measurement, control and automation – Digital factory framework – Part 1: General principles*
- [25] IEC 62832-2:2020-10, *Industrial-process measurement, control and automation - Digital factory framework - Part 2: Model elements*
- [26] IEC 62832-3:2020-10, *Industrial-process measurement, control and automation - Digital factory framework - Part 3: Application of Digital Factory for life cycle management of production systems*
- [27] NA 42AA N2022 – JTC1/SC42/W5 N7 SG1 Report – **Computational Approaches** and AI Systems, 2019-06-28
- [28] NA 42AA N3649 – JTC1/SC42/WG2 N247(replaces N245) Process Management Framework for Big Data Analysis, 2020-04-03
- [29] NA 42AA N3574 – JTC1/SC42/WG2 N223 – US Expert Contribution on Final Version of NIST Big Data Interoperability Framework (NBDIF), 2020-03-30
- [30] NA 42AA N3575 – JTC1/SC42/WG2 N224 – US Expert Contribution on NBDIF – Volume1 Big Data Definitions
- [31] NA 42AA N3576 – JTC1/SC42/WG2 N225 – US Expert Contribution on NBDIF – Volume 2 **Big Data Taxonomies**
- [32] NA 42AA N3577 – JTC1/SC42/WG2 N226 – US Expert Contribution on NBDIF Volume 3 Big Data Use Cases and Requirements
- [33] NA 42AA N3578 – JTC1/SC42/WG2 N227 – US Expert Contribution on NBDIF Volume 4 Big Data Security and Privacy
- [34] NA 42AA N3579 – JTC1/SC42/WG2 N228 – US Expert Contribution on NBDIF Volume 6 Big Data Reference Architecture
- [35] NA 42AA N3580 – JTC1/SC42/WG2 N229 – US Expert Contribution on NBDIF Volume 7 Big Data Standards Roadmap
- [36] NA 42AA N3581 – JTC1/SC42/WG2 N230 – US Expert Contribution on NBDIF Volume 8 Reference Architecture Interface
- [37] NA043-02-03AA N906 – JTC1/WG11 N661 Smart City NP 5087-1 **City Data Model** part 1 Foundation Level Concepts
- [38] NA043-02-03AA N908 – JTC1/WG11 N662 Smart City NP 5087-2 City Data Model part 2 City Level Concepts
- [39] NA043-02-03AA N909 – JTC1/WG11 N663 Smart City NP 5087-3 City Data Model part 3 Service Level Concepts
- [40] WIPANO Vorhaben 2019 ‘DiTraNo‘
- [41] Mathias Uslar et al. OFFIS Oldenburg ‘Security by Design for the Smart Grid by Combining **SGAM and NISTIR7628**’, 2014 IEEE 38th Annual International Computers, SW, and Application Conference;
- [42] Deutsche Normungsroadmap I4.0 v3.0, <https://www.din.de/de/forschung-und-innovation/themen/industrie4-0/roadmap-industrie40-62178>
- [43] JTC1/SC27/WG4 N4001 ‘ISAGCA Quick Start Guide – An Overview of ISA/IEC 62443 Standards - Security of Industrial Automation and Control Systems
- [44] Thomas Walloschke, Jürgen Neises Fujitsu TS GmbH, John Soldatos, IoT Group AIT: Multi-level Policy Matching applied to Industry 4.0
- [45] Jürgen Neises, Thomas Walloschke Fujitsu TS GmbH, George Moldovan, Bianca Popovici Siemens RU: Trustworthiness in Supply Chains;
- [46] Günther Schuh RWTH Aachen et al: Industry 4.0 Maturity Index – Managing the digital transformation of Companies, AcaTech – the National Academy of Science and Engineering, Wolfgang Wahlster DFKI (Editor) München 2017, www.utzverlag.de
- [47] NA 42AA N3792 – JTC1/SC42/WG3 N738 Meeting Report 10-04-2020 on ISO 24029-2 Assessment of Robustness of NN - part 2: Formal Methods Methodology (Form 04)

- [48] NA 42AA N3547 – JTC1/SC42/WG3 N675 WD TR 24029 Robustness of Neural Networks - part 1 Overview v0.3
- [49] DKE 931_2020-0093, SyC SM/42/DTR:2020-04 – IEC TR 63306-1ed1 – System Committee Smart Manufacturing – Smart Manufacturing Standards Map (SM²) – part 1 Framework
- [50] DKE 901_2020-0050, SyC Smart Energy/136/DTS:2020-03 – IEC TS63268ed1 SRD: Energy and Data Interfaces of Users connected to the smart Grid with other Smart Grid Stakeholders – Standardisation Landscape
- [51] F.E.Peters Uni Bonn: Einführung in mathematische Methoden der Informatik, Reihe Informatik/15, BI Wissenschaftsverlag, Zürich 1974
- [52] Paulo Orlando, Universidade Federal de Pernambuco, Brazil et al. 'A Review on OLAP Technologies Applied to Information Networks', Article No.8 in ACM Transactions on Knowledge Discovery from Data, Vol.14, No.1, Article 8, December 2019
- [53] Online Analytical Processing Technology: What is the Definition of OLAP?
- [54] VDI/VDE 2193 **Sprache für I4.0 Komponenten** Blatt 1: Struktur von Nachrichten, https://www.vdi.de/fileadmin/pages/vdi_de/redakteure/richtlinien/inhaltsverzeichnisse/3134409.pdf
- [55] VDI/VDE 2193 **Sprache für I4.0 Komponenten** Blatt 2: Interaktionsprotokoll für Ausschreibungsverfahren, https://www.vdi.de/fileadmin/pages/vdi_de/redakteure/richtlinien/inhaltsverzeichnisse/3111375.pdf
- [56] Plattform I4.0 Diskussionspapier **I4.0 Sprache**, https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/hm-2018-sprache.pdf?__blob=publicationFile&v=6
- [57] ISO/IEC JTC1/SC27 N20699 2nd CD 27030 IT-ST – Security Techniques - Guidelines for Security and Privacy in IoT; 2020-05-28 Text for CD Ballot
- [58] Rodrigo Moreno et al. From Reliability to Resilience – Planning the Grid against the Extremes, IEEE Power&Energy Magazine, August 2020 (1540-7977/20^(c)2020IEEE)
- [59] Hyde M. Merrill et al. Nipping Blackouts in the Bud – Introducing a Novel Cascading Failure Network, IEEE Power&Energy Magazine, August 2020 (1540-7977/20^(c)2020 IEEE)
- [60] Zohar Manna StanfordU, Richard Waldinger SRI International ,The Logical Basis for Computer Programming Volume I Deductive Reasoning', Addison-Wesley 1985
- [61] Fritz Eduard Peters UBonn ,Einführung in mathematische Methoden der Informatik', BI Wissenschaftsverlag 1974
- [62] Dan Pilone ,UML2.0' 2. Auflage, deutsche Übersetzung Dorothea Heymann-Reder, O'Reilly 2006
- [63] H.Ehrig, B.Mahr TUBerlin ,Fundamentals of **Algebraic Specification** 1 – Equational and Initial Semantics', EATCS Monographs on Theoretical Computer Science, Springer Verlag 1985
- [64] H.J Kreowski: Algebra für Informatiker, Lehrveranstaltung im WS1978/79, TU Institut f. SW und Theoretische Informatik
- [65] Jan Lunze ,**Graphentheoretische Methoden** der Regelungstechnik' Edition MoRa, bookmundo.de/shop 2019
- [66] Jan Lunze, **Automatisierungstechnik**, 5.Auflage 2020 Walter de Gruyter Verlag
- [67] Benjamin C. Kuo, **Automatic Control Systems**, 7th edition 1995 Prentice Hall
- [68] John Wylie Lloyd Univ. of Melbourne 'Foundation of **Logic Programming**', Symbolic Computation Springer Verlag
- [69] Andrew W. Appel Editor of Facsimile Reproduction of Alan Turing's Princeton Dissertation "System of Logic based on Ordinals" Princeton PhD Thesis 1938; 2012 Princeton University Press
- [70] Zohar Manna, Richard Waldinger 'The Logic Basis for Computer Programming – Volume 1 **Deductive Reasoning**' 1985 Addison-Wesley Publishing Inc.

- [71] Uwe Kastens, Hans Kleine Büning 'Modellierung - Grundlagen und **formale Methoden**' 2005 Carl Hanser Verlag München Wien
- [72] Markus Krötzsch et al. University of Oxford 'A **Description Logic** Primer' v1.2
- [73] ISO/IEC JTC1/SC 41 N1060:2019 'IoT and related technologies – part 2 Transport Interoperability', revised text of CDV 21823-2' Committee Manager, jooran@ksa.or.kr
- [74] ISO/IEC JTC1/WG11 N882 'IoT Interoperability for IoT Systems – part 3 **Semantic Interoperability**', revised text of CDV 21823-3' submitted by Mark Fox
- [75] Martin Grohe, Pascal Schweitzer: The **Graph Isomorphism Problem**, CACM.ACM.ORG 11/2020 Vol.63 No.11 pgs.128-134, Review Articles
- [76] Boris Glavic, Klaus R. Dittrich 'Data Provenance: A Categorization of Existing Approaches, Proceedings 7.-9.März 2007, Aachen Conference BTW 2007, GI FB DBIS
- [77] Martin Braun et al. 'Blackouts, Restoration, and Islanding', pp.54-63, IEEE PEM, 17.06.2020
- [78] Sarah Fluchs, Admeritia GmbH www.admeritia.de 'On Modelling for Security Engineering' as a Submodel for the Digital Twin (Whitepaper V1.0, 2.01.2021)
- [79] Pascal Hitzler, A Review of the Semantic Web Field, CACM.ACM.ORG 02/2021 Vol.64 No.02 pgs.76-83 Review Articles.

VDE Verband der Elektrotechnik
Elektronik Informationstechnik e.V.

DKE Deutsche Kommission
Elektrotechnik Elektronik Informationstechnik
in DIN und VDE
Merianstraße 28
63069 Offenbach am Main

Tel. +49 69 6308-0
dke@vde.com
www.dke.de

VDE