

Anhang

A

Metriken

Im wissenschaftlichen Umgang mit Systemen und Strukturen ist es notwendig, quantifizierbare Größen bestimmen zu können. Quantifizierbare Größen bieten die Möglichkeit, Steuerungsgrößen direkt oder indirekt zu berechnen.

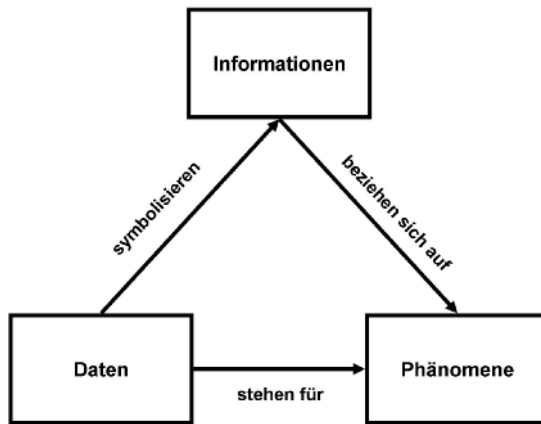


Abb. A.1: Messung, Daten und Phänomene

Erst die Messbarkeit von Eigenschaften macht ein System vergleich- und bewertbar. Mit Hilfe einer Messung wird versucht, die in der realen Welt beobachtbaren Phänomene durch Daten zu beschreiben (s. Abb. A.1). Die gemessenen Daten repräsentieren das Phänomen, aber nur indirekt über die Information, da die Realität erst im Rahmen eines mentalen Konzepts wahr-

genommen werden kann. Diese Interpretation ist nicht frei vom Kontext und der Erfahrung des Beobachters.

A.1 Messbarkeit

Eine Metrik (s. Abb. A.2) braucht immer folgende fünf Größen:

- 1 Messvorschrift – Gibt vor, was und wie gemessen werden soll.
- 2 Modell – Ein Modell, das Parameter kennt, um aus dem Modell, den Parametern und der Messung eine Vorhersage zu erzeugen. Mathematische Modelle sind die wissenschaftliche Form von Analogien. Wie jede Analogie haben sie auch ihre Limitierungen, d.h. Aspekte, die das Modell nicht beschreibt, welche aber der Phänomenologie zugänglich (beobachtbar) sind.
- 3 Parameter – Einen Satz von Parametern, damit das generische Modell konkretisiert werden kann.
- 4 Spezifikation – Eine Spezifikation der Bedeutung und Interpretation der Metrik.
- 5 Referenzwert – Den Vergleich der Vorhersage des Modells mit der realen Welt. Ohne eine solche, prinzipiell existente, Vergleichsmöglichkeit bleibt die Metrik mehr oder minder metaphysisch.

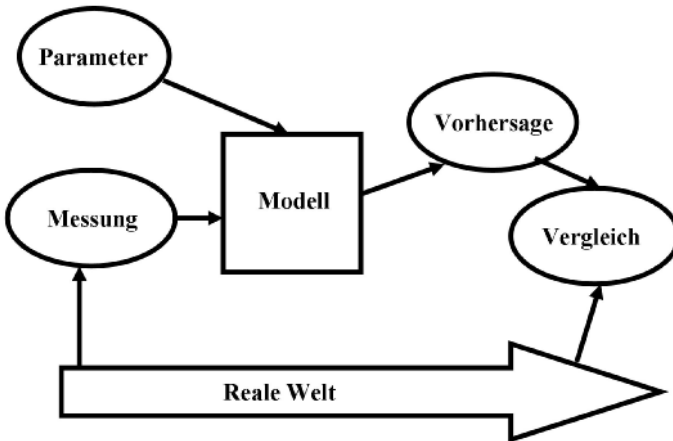


Abb. A.2: Metriken brauchen Modelle und Messvorschriften

Jede Messung besteht aus einer Messvorschrift und einer zu messenden Einheit und stellt den Versuch dar, eine Abbildung aus der realen, physischen

Welt in eine mathematische Modellwelt vorzunehmen. Die durchgeführte Messung liefert stets eine Momentaufnahme des Messgegenstandes. Die Messergebnisse können genutzt werden, um Aussagen über die Systeme zu gewinnen.

A.2 Rating

Der Begriff Ratingverfahren umfasst allgemein eine Reihe von Bewertungsverfahren, deren Ergebnis in Form einer singulären Zensur, dem sogenannten Rating, formuliert wird. Anhand vorher festgelegter Merkmale werden die untersuchten Eigenschaften dabei auf einer Skala eingeordnet. Aufgrund von Vergleichsdaten werden die betrachteten Objekte möglichst sachlich beurteilt. Zum Vergleich werden dabei in der Regel andere Unternehmen derselben Branche herangezogen. Dementsprechend kann ein Ratingverfahren als eine Funktion angesehen werden, welche das zu beurteilende Objekt gemäss seiner Charakteristika auf eine diskrete Anzahl von Kategorien abbildet. Bei den heute im Einsatz befindlichen Ratingsystemen kommt dabei meist die Scoringmethode zum Einsatz, die auf einer vorab definierten Menge von Kriterien basiert, welche separat evaluiert werden. Die einzelnen Benotungen der Kriterien werden gewichtet und zu einem Gesamtscore zusammengefasst, welcher dann wiederum in eine Kategorie übersetzt wird. Ein Scoring ist ein Bewertungsverfahren, welches als Ergebnis eine Schätzung in Form einer positiven reellen Zahl enthält. Insofern ist das Scoring eine Abbildung der Form $\mathcal{F}_{\text{Score}} \mathfrak{R}^{(n)} \mapsto \mathfrak{R}_0^+$. Üblicherweise ist das Scoring eine Summe über gewichtete Klassifikationen der einzelnen Dimensionen des $\mathfrak{R}^{(n)}$ in der Form:

$$\mathcal{F}_{\text{Score}}(x) = \sum_{i=1}^n \Omega_i \Theta_i(x_i). \quad (\text{A.1})$$

Die Gewichte Ω_i geben an, wie stark das einzelne Attribut zum Score $\mathcal{F}_{\text{Score}}(x)$ beiträgt. Die Funktionen Θ_i bilden pro Attribut eine Reihe von disjunkten Kategorien, wobei jeder Kategorie ein reeller, meist positiver Wert zugeordnet wird. Ratingagenturen liefern der Industrie oft solche Werte, wobei das einzelne Modell dann meist in den Gewichten Ω_i modifiziert wird.

A.3 Netzwerkmaße

Zu den Maßen, die zur Bestimmung der Qualität eines Netzwerkes dienen, gehören eine Reihe von Maßen, die der Graphentheorie entlehnt sind. Die einfachste Metrik ist hierbei die mittlerer Weglänge \bar{l} , welche als das Mittel über alle kürzesten Pfade zwischen je zwei Knoten in einem Graphen definiert ist:

$$\bar{l} = \frac{1}{N^2} \sum_{i,j} \min d(i, j). \quad (\text{A.2})$$

Der Wert von \bar{l} wird oft auch als Netzwerkdurchmesser bezeichnet, da er die lineare Größe in Form eines mittleren Abstands zweier Netzwerknoten darstellt. Ein anderes Maß ist der Average Node Degree N_D , welcher die mittlere Zahl der Kanten pro Knoten misst und sich durch:

$$N_D = 2 \frac{n_{\text{Kanten}}}{n_{\text{Knoten}}}$$

berechnen lässt. Ein weiteres Maß ist der Clusteringkoeffizient \bar{c} , der sich als Mittel über die einzelnen Clusterkoeffizienten c_i ergibt:

$$\bar{c} = \frac{1}{n_{\text{Knoten}}} \sum_i c_i = \frac{1}{n_{\text{Knoten}}} \sum_i \frac{2K_i}{N_i(N_i - 1)}, \quad (\text{A.3})$$

hierbei ist K die Zahl der tatsächlichen Kanten des Knoten i und N die Zahl der Knoten im Cluster (hier werden nur die verbundenen Knoten gezählt).

A.4 Komplexitätsmaße

Eine der großen Schwierigkeiten beim Umgang mit komplexen Systemen ist die Tatsache, dass es kein allgemeingültiges Maß für die Komplexität gibt. Am gängigsten ist die Verwendung der Intuition, d.h., wenn das System sich nichtintuitiv verhält, ist es komplex (s. Abb. 11.1). Im Sinne einer Metrik ist dies äußerst unbefriedigend. Bessere Metriken sind hier die Entropie und die Vielfältigkeit.

A.4.1 Vielfältigkeit

Unter dem Begriff Vielfältigkeit¹ wird ein Maß für die möglichen Zustände eines Systems verstanden. Eine Möglichkeit, die Vielfältigkeit zu definieren, ist über die Summe der möglichen Zustände²: $N = \sum_{\psi \in S} 1$. In den meisten Fällen wird jedoch der Logarithmus zur Bestimmung genutzt:

$$V = \log_2 \left(\sum_{\psi \in S} 1 \right). \quad (\text{A.4})$$

Der Logarithmus hat den großen Vorteil, dass bei disjunkten Systemen die Vielfältigkeit additiv ist:

¹ Variety

² In diesem Fall hätte ein Lichtschalter die Vielfältigkeit 2 und eine Ziffer die Vielfältigkeit 10, eine zehnstellige Zahl schon $10^{10} = 10.000.000.000$.

$$\begin{aligned}
 V(S_1 \times S_2) &= \log_2 \left(\sum_{\psi \in S_1 \cup S_2} 1 \right) = \log_2 \left(\sum_{\psi_1 \in S_1 \times \psi_2 \in S_2} 1 \right) \\
 &= \log_2(N_1 \times N_2) = \log_2 N_1 + \log_2 N_2 \\
 &= V(S_1) + V(S_2).
 \end{aligned}$$

Die meisten Systeme aus der realen Welt besitzen eine Vielfältigkeit die effektiv unendlich groß³ ist, daher haben Organisationen wie auch der Mensch eine Reihe von Filtern entwickelt, um die Vielfältigkeit der Beobachtungen auf ein erträgliches Maß⁴ zu reduzieren. Das Gegenteil davon ist ein Vielfältigkeitsverstärker. Einfache Lebensformen nutzen ihre hohen Reproduktionsraten, um die Vielfältigkeit zu verstärken, höhere Lebensformen setzen hierfür Nervensysteme und Wahrnehmungsorgane ein.⁵

A.4.2 Entropie

Die Entropie⁶ stellt ein Maß für die Unordnung in einem System dar. Ursprünglich im Rahmen der klassischen Thermodynamik definiert, wurde der Begriff auf die statistische Mechanik ausgedehnt. Die Entropiedefinition der statistischen Mechanik kann auf die Informationstheorie übertragen werden. Die Entropie eines Systems ist definiert als:

$$S = - \sum_{j=1}^N p_j \log_2 p_j, \quad (\text{A.5})$$

wobei p_j die Wahrscheinlichkeit ist, dass der Zustand j angenommen wird. Wenn es nur einen Zustand gibt, so gilt $p_j = 1$ und damit folgt: $S = 0$ wenn $p = 1$. Sind in einem System alle Zustände unterschiedlich und alle gleichwahrscheinlich ($p_j = \frac{1}{N}$), resultiert die Entropie zu:

$$S = - \sum_{j=1}^N \frac{1}{N} \log_2 \frac{1}{N} = \log_2 N.$$

Daher gilt für jedes System: $0 \leq S \leq \log_2 N$. Da diese obere Schranke auch die Vielfältigkeit (s. Gl. A.4) ist, folgt:

$$S \leq V, \quad (\text{A.6})$$

und im Grenzfall, dass alle Zustände des Systems gleichwahrscheinlich sind: $S = V$.

³ Dadurch wird ein solches System leicht unbeherrschbar (s. Abschn. 11.8).

⁴ Wir sehen nur das, was wir sehen wollen.

⁵ Der Mensch nutzt seine Intelligenz zur Verstärkung der Vielfältigkeit.

⁶ Der Physiker Helmholtz entlehnte den Begriff Entropie aus dem griechischen Wort $\epsilon\nu\tau\rho\epsilon\pi\iota\nu$ mit der Bedeutung von „umkehren“ oder „umwenden“.

Die Vielfältigkeit (s. Gl.A.4) und die Entropie (s. Gl. A.5) haben einen engen Zusammenhang. Wird nämlich jeder mögliche Zustand, der in der Messung der Vielfältigkeit auftaucht (bei der Vielfältigkeit mit dem Faktor 1), mit der Wahrscheinlichkeit multipliziert, mit der er in dem System angenommen werden kann, so ergibt sich, bis auf triviale Vorfaktoren und Konstanten, die Entropie.

Wie verändert sich die Entropie, wenn einem System ein neuer Zustand hinzugefügt wird? Zwar lässt sich diese Frage im Einzelfall nur durch eine exakte Berechnung mit Hilfe der Wahrscheinlichkeiten beantworten, für sehr große Systeme mit $N \gg 1$ gilt aber näherungsweise

$$\begin{aligned} \Delta S &= S - S_0 \approx \log_2(N+1) - \log_2(N), \\ &\approx \frac{1}{\ln 2} \left(1 - \frac{1}{N}\right). \end{aligned}$$

Wenn zwei Systeme A und B zusammengefügt werden, so ergibt sich die gemeinsame Entropie näherungsweise zu:

$$S(A \cup B) \approx S(A) + S(B) + \frac{1}{N} \log_2 N. \quad (\text{A.7})$$

Bei sehr großen Systemen ist die Zahl N so groß, dass sich durch das Hinzufügen eines einzelnen Zustandes die Entropie faktisch um eine Konstante erhöht: $\Delta S \approx \frac{1}{\ln 2}$. Die so gewählte Definition über das System berücksichtigt jedoch nicht die innere Entropie der Elemente (Subsysteme). Wenn die innere Entropie der Elemente mit ins Kalkül gezogen wird, ergibt sich die Entropie zu:

$$S = S(\text{System}) + \sum_{\text{Zustand}} S(\text{Elemente}). \quad (\text{A.8})$$

Für den Fall, dass mehrere Systeme miteinander verglichen werden müssen, empfiehlt es sich, die Entropie zu normalisieren:

$$S^\dagger = \frac{1}{S_{\max}} S = -\frac{1}{\log_2 N} \sum_{i=1}^N p_i \log_2 p_i, \quad (\text{A.9})$$

mit der Folge, dass für die normalisierte Entropie gilt: $0 \leq S^\dagger \leq 1$. Es empfiehlt sich, die normalisierte Entropie S^\dagger bei der Betrachtung von Entropieänderungen $\dot{S}^\dagger = \frac{\partial S^\dagger}{\partial t}$ zu nutzen.

Es existiert auch ein Zusammenhang zwischen der Entropie (Gl. A.5) und der Temperatur. Wenn Information zerstört wird, ändert sich Gl. A.5 um ΔS , mit der Folge, dass die physische Entropie des Systems sich auch ändert:

$$\Delta S_{\text{phys}} = -k_B \ln 2 \Delta S. \quad (\text{A.10})$$

Dies wiederum führt zu einer Änderung der Energie im System:

$$\Delta Q = T \Delta S_{\text{phys}} = -k_B T \ln 2 \Delta S. \quad (\text{A.11})$$

Gl. A.11 wird auch als das Landauer Prinzip bezeichnet: Das Löschen von einem Bit an Information produziert mindestens $k_B T \ln 2$ Wärmeenergie. Hierbei sollte beachtet werden, dass es einen Unterschied zwischen der Entropie der Information und der „regulären“ thermodynamischen Entropie gibt. Die Informationsentropie bezieht sich auf Systeme, welche eine informationsbasierte Wechselwirkung haben und keine energetische, d.h. beide Systeme sind energetisch entkoppelt und keine strukturellen Veränderungen werden durch den Informationsaustausch vorgenommen. Außerdem ist die informationstechnischen Entropie durch $S = -\sum_j p_j \log_2 p_j$ definiert, während die

physikalische Entropie durch $S_{\text{phys}} = -k_B \sum_i p_i \ln p_i = -\frac{k_B}{\log_2 e} \sum_j p_j \log_2 p_j$,

wobei die Boltzmannkonstante k_B den Wert $1.38065 \times 10^{-23} \frac{\text{J}}{\text{K}}$ hat, festgelegt ist. Die physikalische Entropie erhöht sich nach dem Landauer Prinzip, wenn die Menge an Information abnimmt. Daher lässt sich die Informationsmenge auch alternativ über die physikalische Entropie messen:

$$M_{\text{Information}}(\mathcal{Z}) = \max_{\forall \mathcal{Z}} (S_{\text{phys}}) - S_{\text{phys}}(\mathcal{Z}). \quad (\text{A.12})$$

Speziell bei der Betrachtung von Adaption ist die Nutzung von Entropie mit Vorsicht zu genießen, da es viele Phasenübergänge zwischen Systemen gibt, welche in der Entropie kaum merklich sind. Die Größe der Entropie gibt an, ob ein System strukturiert ist oder nicht, aber nicht wie es strukturiert ist. Daher ist es möglich, identische Entropie bei völlig unterschiedlicher interner Struktur zu haben.

A.5 Koppelungsmaße

Zwar sind in der Theorie alle Services zustandslos, in der Praxis wird diese Forderung jedoch oft durchbrochen. Einer der Gründe für die Verletzung dieser Forderung ist die Unterstützung langanhaltender oder transaktionaler Services oder Prozesse. Daher ist es sinnvoll, im Serviceportfolio Maße für die tatsächliche Koppelung zu besitzen. Für die Messung des Grads an „Zustandsbehaftetheit“ eines Services, wird der **Degree of State Dependency** (DSD) genutzt, welcher durch:

$$\eta_{\text{DSD}} = \frac{1}{N_{\text{Services}}} \sum_k c_k \quad (\text{A.13})$$

definiert ist, wobei c_k die Werte 0 (zustandslos) oder 1 (zustandsbehaftet) annehmen kann. Im Idealfall sollte $\eta_{\text{DSD}} \mapsto 0$ gelten. Für den Grenzfall $\eta_{\text{DSD}} \mapsto 1$ degradiert das Serviceportfolio zu einer Kollektion von Komponenten. Neben der direkten Abhängigkeit von Zuständen der Services untereinander η_{DSD} , können Services auch indirekt über persistente Daten zustandsbehaftet sein, in dem die Services sich der Daten aus einer gemeinsamen Datenbank bedienen.

Für diesen Fall lässt sich der Grad der persistenten „Zustandsbehaftetheit“ η_{DPD} definieren⁷:

$$\eta_{\text{DPD}} = \frac{1}{N_{\text{Services}}(N_{\text{Services}} - 1)} \sum_{i,j} p_{ij}, \quad (\text{A.14})$$

mit p_{ij} als Maß dafür, ob der Service i zusammen mit dem Service j ein gemeinsames persistentes Element besitzt. Der Grenzfall $\eta_{\text{DPD}} \mapsto 0$ ist der Fall der losen Koppelung und im Fall $\eta_{\text{DPD}} \mapsto 1$ ist die indirekte Koppelung so stark, dass kein serviceorientiertes System vorliegt. Neben den Zustandsmetriken η_{DPD} und η_{DSD} lassen sich auch zwei Maße für die Abhängigkeit, im Sinne einer Nutzung, η_{ARSD} und η_{ASIC} festlegen. Der mittlere Grad an Servicenutzung⁸ in komponierten Services ergibt sich zu:

$$\eta_{\text{ARSD}} = \frac{1}{N_{\text{Services}}} \sum_k r_k, \quad (\text{A.15})$$

wobei r_k die Anzahl der Services angibt, die der Service k zu seinen Operationen benötigt. Hier wird auch rekursiv gezählt, so dass der gesamte Servicebaum aufgelöst werden muss um die Zahl r_k zu ermitteln. Niedrige Werte von η_{ARSD} sind ein Indiz für lose Koppelung. Hohe Werte von η_{ARSD} hingegen (im Grenzfall gilt: $\lim \eta_{\text{ARSD}} \approx N_{\text{Services}}$) zeugen von einem so hohen Grad an Verwobenheit der Services untereinander, dass jede Änderung eines Services Auswirkungen auf die Gesamtheit hat. $\eta_{\text{ARSD}} \approx N_{\text{Services}}$ führt das Ziel der Autonomie von Services ad absurdum. Im Fall der asynchronen Koppelung zur Laufzeit lässt sich η_{ARSD} nicht gut verwenden, sondern sollte auf η_{ASIC} , die mittlere Aufrufkoppelung⁹, erweitert werden:

$$\eta_{\text{ASIC}} = \frac{1}{N_{\text{Services}}} \sum_k \left(w_s n_k^{\text{synchron}} + w_a n_k^{\text{asynchron}} \right). \quad (\text{A.16})$$

Für die Gewichte w gilt $w_s + w_a = 1$ und n_k^{synchron} misst die Zahl der synchronen und $n_k^{\text{asynchron}}$ die der asynchronen Aufrufe.

A.6 Semantische Ähnlichkeit

Eine semantische Ähnlichkeit wird über die Intention der jeweiligen Definition des Begriffs vorgenommen. Eine intentionale Definition hat immer die Form: $\{x : x \text{ hat die Eigenschaft } E\}$. Es existieren sechs Ähnlichkeitsrelationen, zu ihrer Bestimmung werden zwei Ausdrücke P, Q betrachtet und mit der jeweiligen Intention $\iota(P), \iota(Q)$ verglichen.

⁷ Degree of Persistent Dependency (DPD)

⁸ Average Required Service Dependency (ARSD)

⁹ Average Service Invocation Coupling (ASIC)

- Äquivalenz – Zwei Ausdrücke sind genau dann gleich, wenn sie identisch intendierte Definitionen haben:

$$\{P = Q\} \Leftrightarrow \{\iota(P) = \iota(Q)\}$$

- Spezialisierung – Der Ausdruck P ist eine Spezialisierung von Q , wenn die intentionale Definition von P in der intentionalen Bedeutung von Q enthalten ist:

$$\{P \leq Q\} \Leftrightarrow \{\iota(P) \wedge \iota(Q) \equiv \iota(P)\}$$

- Generalisierung – Der Ausdruck P generalisiert einen Ausdruck Q , wenn die Union der beiden intentionalen Definitionen gleich der intentionalen Definition von P ist:

$$\{P \geq Q\} \Leftrightarrow \{\iota(P) \vee \iota(Q) \equiv \iota(P)\}$$

- Überlappung – Ein Ausdruck P überlappt mit dem Ausdruck Q , wenn beide in einem Teilbereich übereinstimmen, oder anders ausgedrückt, wenn die Konjunktion der beiden intentionalen Definitionen nicht leer ist:

$$\{P \sim Q\} \Leftrightarrow \{\{\iota(P) \wedge \iota(Q) \equiv \iota(R)\} \wedge \{\iota(R) \neq \emptyset\}\}$$

- Disjunktion – Zwei Ausdrücke werden als disjunkt bezeichnet, wenn die Konjunktion ihrer intentionalen Definition leer ist:

$$\{P \neq Q\} \Leftrightarrow \{\iota(P) \wedge \iota(Q) = \emptyset\}$$

- Inverse – P ist invers zu Q , wenn seine intentionale Definition invers zu der von Q ist:

$$\{P = Q^{-1}\} \Leftrightarrow \{\iota(P) = \iota(Q^{-1})\}$$

Solche semantischen Ähnlichkeiten können genutzt werden, um Services automatisch miteinander zu vergleichen. Dazu wird eine zusätzliche Relation $\text{Sim}(Q, P)$ definiert, mit der Eigenschaft:

$$\text{Sim}(Q, P) = \begin{cases} 1 & \Rightarrow \{Q = P\} \vee \{Q < P\} \\ 0 & \Rightarrow \{Q \sim P\} \vee \{Q > P\} \\ -1 & \Rightarrow \{Q \neq P\} \\ -2 & \Rightarrow \{Q = P^{-1}\}. \end{cases} \quad (\text{A.17})$$

B

π -Kalkül

Bei dem π -Kalkül handelt es sich um eine Prozessalgebra. Eine Prozessalgebra ist eine Menge von Prozessen, die bezüglich gewisser Operationen abgeschlossen ist. Dabei kann man rein syntaktische Terme der Prozessausdrücke als Prozesse betrachten, und als Operationen zunächst nur Regeln auffassen, nach denen Prozessausdrücke gebildet werden. Auf einer höheren Ebene werden Kongruenzen hinzugefügt, die Terme vereinheitlichen, welche sich gleich verhalten. Diese Kongruenzen kann man zunächst einfach als Operationen dem Kalkül hinzufügen. Andererseits ist es nun möglich, mit ihrer Hilfe von den Termen zu abstrahieren und als Prozesse die jeweiligen Kongruenzklassen zu betrachten. Auf diesen Klassen oder auch auf den Termen selbst lassen sich Übergangsrelationen definieren, die das Verhalten der Prozesse festlegen und selbst wieder als die Operationen einer Algebra aufgefasst werden können.

B.1 Definition

Die Essenz des π -Kalküls ist die Kommunikation. Daher werden mit π die sogenannten Aktionspräfixe bezeichnet. Ein Aktionspräfix stellt das Senden oder Empfangen einer Botschaft (genauer: eines Namens) oder einen stillen Übergang dar. Die Syntax ist:

$$\pi ::= \begin{cases} x(y) & \text{sende } x \text{ entlang } y, \\ \bar{x}(y) & \text{empfange } x \text{ entlang } y, \\ \tau & \text{nicht sichtbar.} \end{cases} \quad (\text{B.1})$$

Die Menge \mathfrak{P}^π der π -Kalkül-Prozessausdrücke ist wie folgt definiert (i läuft über die Indexmenge \mathcal{I}):

$$\mathfrak{P} = \begin{cases} \sum_{i \in \mathcal{I}} \pi_i \cdot \mathfrak{P}_i & \text{Summation,} \\ \mathfrak{P}_1 | \mathfrak{P}_2 & \text{parallele Komposition,} \\ \text{new } a \mathfrak{P} & \text{Restriktion,} \\ !\mathfrak{P} & \text{Replikation.} \end{cases} \quad (\text{B.2})$$

Die Summation $x(a).\mathfrak{P} + y(b).\Omega$ beschreibt einen Prozess, der den Parameter a entlang des Kanals x empfängt und dann in \mathfrak{P} übergeht oder b entlang Kanal y erhält und in den Prozess Ω übergeht. $\mathfrak{P}_1|\mathfrak{P}_2$ zeigt, dass die beiden Prozesse parallel ausführbar sind. Die Restriktion $\text{new } a\mathfrak{P}$ bindet a an \mathfrak{P} , so dass a nur in \mathfrak{P} sichtbar ist. Die Replikation $!\mathfrak{P}$ ist das Symbol für beliebig viele Instanzen von \mathfrak{P} , welche alle parallel ablaufen.

B.2 Kongruenz

Prozesse können nicht ohne einen Kontext existieren, daher wird im Rahmen des π -Kalküls auch ein Kontext \mathcal{C} definiert:

$$\mathcal{C} ::= [] \mid \pi.\mathcal{C} + \mathfrak{M} \mid \text{new } x\mathcal{C} \mid \mathcal{C}|\mathfrak{P} \mid \mathfrak{P}|\mathcal{C} \mid !\mathcal{C}. \tag{B.3}$$

Somit ist ein Kontext ein Ausdruck, welcher eine Lücke $[]$ aufweist, die Lücke $[]$ ist wiederum genau der Prozess, der im Kontext eingebettet ist. \mathfrak{M} stellt eine Summe und \mathfrak{P} einen Prozess dar. $\mathcal{C}[\Omega]$ bezeichnet das Resultat der Ersetzung der Lücke $[]$ im Kontext \mathcal{C} durch den Prozessausdruck Ω . Die elementaren Kontexte sind $\pi.[] + M$, $\text{new } a[]$, $[]|\mathfrak{P}$, $\mathfrak{P}|[]$ und $![]$. Die Prozesskongruenz wird im π -Kalkül mit Hilfe einer Äquivalenzrelation abgebildet. Eine Äquivalenzrelation \cong ist genau dann eine Prozesskongruenz, wenn gilt:

$$\forall \mathfrak{P}, \Omega \in \mathfrak{P}^\pi \forall \mathcal{C} : \mathfrak{P} \cong \mathcal{C} \Rightarrow \mathcal{C}[\mathfrak{P}] \cong \mathcal{C}[\Omega]. \tag{B.4}$$

Wenn zwei Ausdrücke kongruent sind, dann kann man sie in beliebige Kontexte einsetzen und erhält stets äquivalente Ausdrücke. Die strukturelle Kongruenz \equiv ist eine Prozesskongruenz (Gl. B.4), welche folgende Eigenschaften besitzt:

- Die Namen der gebundenen Variablen können beliebig verändert werden. Unter Bindung versteht man die Zuordnung einer Variablen zu einem Kanal. Bei $x(a).\mathfrak{P}$ wird die Variable an den Kanal x gebunden.
- Beliebige Permutation der Indexmenge:

$$\sum_{i \in \mathcal{I}} \pi_i.\mathfrak{P}_i \equiv \sum_{j \in \mathcal{J}} \pi_j.\mathfrak{P}_j,$$

solange \mathcal{I} und \mathcal{J} die gleichen Mengen darstellen.

- Die Reihenfolge der parallelen Komposition darf keine Rolle spielen:

$$\mathfrak{P}|\Omega \equiv \Omega|\mathfrak{P}, \tag{B.5}$$

$$(\mathfrak{P}|\Omega)|\mathfrak{R} \equiv \mathfrak{P}|(\Omega|\mathfrak{R}). \tag{B.6}$$

- Beim Zusammenführen von Prozessen kann es zu Namenskollisionen kommen, daher muss für die Äquivalenzrelation gelten:

$$\text{new } x(\mathfrak{P}|\Omega) \equiv \mathfrak{P}|\text{new } x\Omega \text{ falls } x \notin \mathfrak{P}.$$

- Leere Prozesse können entsorgt werden:

$$\begin{aligned} \mathfrak{P}|\emptyset &\equiv \mathfrak{P}, \\ \text{new } x\emptyset &\equiv \emptyset, \\ \text{new } xy\mathfrak{P} &\equiv \text{new } yx\mathfrak{P}. \end{aligned}$$

B.3 Abstraktion

Eine n -stellige Abstraktion ist ein parametrischer Prozess in n Variablen, z.B. $(x_1 \dots x_n).\mathfrak{P}$. Eine Abstraktion $(x_1 \dots x_n).\mathfrak{P}$ repräsentiert alle möglichen Pfade, die \mathfrak{P} nehmen kann, abhängig von der Botschaft z , die für \hat{x} in \mathfrak{P} substituiert wird. Die möglichen Abstraktionen F, G, \dots werden als Agenten bezeichnet. Eine Konkretisierung ist in gewisser Weise der Gegensatz zu einer Abstraktion, hier werden Namen für die Variablen der Abstraktion bereitgestellt. Eine solche Konkretisierung ist etwa $\text{new } \hat{x}(\hat{y}).\Omega$, wobei die Namen aus \hat{x} eine Teilmenge der Namen in $\langle \hat{y} \rangle$ sind ($\hat{x} \subseteq \hat{y}$). Die Konkretisierungen haben im Gegensatz zu Abstraktionen eine Besonderheit: Der Bindungsbereich der Namen kann auf die Botschaft \hat{y} und den Fortsetzungsprozess Ω beschränkt werden.

- **Abstraktion:** Eine n -stellige Abstraktion hat die Form $(x_1 \dots x_n).\mathfrak{P}$, wobei $|\langle \hat{x} \rangle| = n$. Zwei Abstraktionen sind strukturell kongruent (Gl. B.4), wenn ihre gebundenen Namen $(x_1 \dots x_n)$ bis auf Umbenennungen übereinstimmen und ihre Prozessteile \mathfrak{P} strukturell kongruent sind.
- **Konkretisierung:** Eine n -stellige Konkretisierung ist $\text{new } \hat{x}(\hat{y}).\mathfrak{P}$, wobei $|\langle \hat{y} \rangle| = n$ und $\hat{x} \subseteq \hat{y}$. Zwei Konkretisierungen sind strukturell kongruent, wenn ihre Präfixe $\text{new } \hat{x}(\hat{y})$ bis auf Umbenennungen und Umordnung der Namen übereinstimmen und ihre Prozessteile \mathfrak{P} strukturell kongruent sind.
- **Agent:** Ein Agent ist entweder Abstraktion oder eine Konkretisierung. \mathbb{A}^π bezeichnet die Menge der Agenten. Prozesse sind Agenten. Ein Prozess ist sowohl eine Abstraktion wie eine Konkretisierung der Stelligkeit 0.
- **Applikation:** Die Applikation $\mathfrak{F}@\mathcal{C}$ einer gleichstelligen Abstraktion und Konkretisierung ist definiert, vorausgesetzt \hat{z} ist nicht frei, in $(\hat{x}).\mathfrak{P}$ als:

$$(\hat{x}).\mathfrak{P}@\text{new } \hat{z}(\hat{y}).\Omega \mapsto \text{new } \hat{z}(\{\hat{y}/\hat{x}\} \mathfrak{P}|\Omega). \quad (\text{B.7})$$

B.4 Reaktion

Eine Reaktion im π -Kalkül ist ein Prozessübergang, welcher ohne Beeinflussung von außen erfolgt. Eine Beeinflussung von außen ist die Einwirkung eines anderen Prozesses durch das Senden oder Empfangen von Messages. Eine Reaktionsrelation ist durch folgende Definitionen festgelegt:

$$\begin{array}{l}
\text{REACT} \quad (x(y).\mathfrak{P} + M) | (\bar{x}(m).\Omega + N) \mapsto \{m/y\} \mathfrak{P} | \Omega, \quad (\text{B.8}) \\
\text{STRUCT} \quad \frac{\Omega \equiv \mathfrak{P} \ \mathfrak{P} \mapsto \mathfrak{P}' \ \mathfrak{P}' \equiv \Omega'}{\Omega \mapsto \Omega'}, \quad (\text{B.9}) \\
\text{TAU} \quad \tau.\mathfrak{P} + M \mapsto \mathfrak{P}, \quad (\text{B.10}) \\
\text{PAR} \quad \frac{\mathfrak{P} \mapsto \mathfrak{P}'}{\mathfrak{P} | \Omega \mapsto \mathfrak{P}' | \Omega'}, \quad (\text{B.11}) \\
\text{RES} \quad \frac{\hat{P} \mapsto \hat{P}'}{\text{new } x\mathfrak{P} \mapsto \text{new } x\mathfrak{P}'}. \quad (\text{B.12})
\end{array}$$

REACT ist die zentrale Regel, welche die Informationsübertragung modelliert. Nur dann, wenn zwei Summanden parallel komponierter Summen über einen gemeinsamen Kanal verfügen, kann Kommunikation stattfinden. REACT verwirft alle weiteren Alternativen und ersetzt im Prozess \mathfrak{P} den abstrakten Namen y durch die Message m . Ein Aufruf bewirkt also nicht nur, dass sich die Prozesse synchronisieren, sondern auch, dass sich der Zustand des Empfängerprozesses verändert. Häufig müssen die Prozessausdrücke zuerst umgeformt werden, bevor eine REACT ausgenutzt kann, daher die Erweiterung um die Regel STRUCT, welche die strukturelle Kongruenz \equiv mit der Reaktionsrelation \mapsto verbindet. Die Regel TAU erlaubt es Prozessen ohne Einwirkung von außen zu agieren. Die Regeln PAR und RES wiederum ermöglichen, innerhalb der Konstrukte Reaktionen durchzuführen.

B.5 Replikation

Rekursive Definitionen werden oft zur Modellierung von Systemen eingesetzt. Da im π -Kalkül Namen übertragen werden, ist es möglich, den Formalismus der Replikation durch folgende Ergänzung zu nutzen:

$$\mathfrak{P} ::= \dots | !\mathfrak{P}, \quad (\text{B.13})$$

$$!\mathfrak{P} \equiv \mathfrak{P} | !\mathfrak{P}, \quad (\text{B.14})$$

$$\mathcal{C} ::= \dots | !\mathcal{C}. \quad (\text{B.15})$$

Die Replikation ist ein Prozessausdruck, vor den ein Ausrufungszeichen gesetzt wird. Die Semantik wird über die strukturelle Kongruenz definiert. $!\mathfrak{P}$ kann den Prozessausdruck \mathfrak{P} beliebig oft replizieren, wobei die einzelnen Instanzen parallel komponiert werden.

B.6 Transaktionen

Eine Erweiterung des π -Kalküls ist das πt -Kalkül. Ein Prozess im πt -Kalkül besteht aus der folgenden Syntax:

$$\mathfrak{P} ::= \begin{cases} \text{done,} \\ \text{abort,} \\ \bar{x}\tilde{u} & \text{Output,} \\ x(\tilde{u}).\mathfrak{P} & \text{Input,} \\ \mathfrak{P}|\mathfrak{P} & \text{Parallelität,} \\ \mathfrak{P};\mathfrak{P} & \text{Sequentialität,} \\ (x)\mathfrak{P} & \text{neuer Prozess,} \\ K(\tilde{u}) & \text{Aufruf,} \\ t(\mathfrak{P}, \mathfrak{P}, \mathfrak{P}, \mathfrak{P}) & \text{Transaktion.} \end{cases} \quad (\text{B.16})$$

Der Prozess $t(\mathfrak{P}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C})$ spiegelt die Transaktion wider, er besteht aus den Teilen:

- \mathfrak{P} – Dies ist der eigentliche Prozess, welcher im Rahmen einer Transaktion durchgeführt werden soll.
- \mathfrak{F} – Der Fehlermanager¹, welcher auf Ausnahmen reagiert.
- \mathfrak{B} – Die Fehlersammlung², welche die notwendigen Kompensationen sammelt, um sie durch \mathfrak{F} im Fehlerfall abarbeiten zu lassen.
- \mathfrak{C} – Die Kompensation, welche im Fehlerfall ausgeführt wird.

Analog zu den Prozessen \mathcal{P} werden im π t-Kalkül auch die Kontexte \mathcal{C} definiert:

$$\mathcal{C}[\] ::= \begin{cases} [\], \\ \mathcal{C}[\]|\mathfrak{P}, \\ \mathcal{C}[\];\mathfrak{P}, \\ (x)\mathcal{C}[\], \\ t(\mathcal{C}[\], \mathfrak{P}, \mathfrak{P}, \mathfrak{P}). \end{cases} \quad (\text{B.17})$$

Zusätzlich zur Kongruenz \equiv (s. Gl. B.4) des π -Kalküls erweitert sich im π t-Kalkül die Kongruenz zu:

$$\text{done}|\mathfrak{P} \equiv \mathfrak{P}, \quad (\text{B.18})$$

$$\text{done};\mathfrak{P} \equiv \mathfrak{P}, \quad (\text{B.19})$$

$$\text{abort}|\text{abort} \equiv \text{abort}, \quad (\text{B.20})$$

$$\text{abort}; P = \text{abort}, \quad (\text{B.21})$$

$$t((x)\mathfrak{P}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C}) \equiv (x)t(\mathfrak{P}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C}), \quad (\text{B.22})$$

$$t(\bar{x}\tilde{u}|\mathfrak{P}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C}) \equiv \bar{x}\tilde{u}|t(\mathfrak{P}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C}), \quad (\text{B.23})$$

$$(t(\text{done}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C})|\mathfrak{P}); \mathfrak{P}' \equiv t(\text{done}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C})|(\mathfrak{P}; \mathfrak{P}'). \quad (\text{B.24})$$

Die beiden ersten Gleichungen (B.18, B.19) implizieren, dass *done* das Element der Parallelität und Sequentialität ist. Die Gleichung B.20 besagt,

¹ Failure Manager

² Failure Bag

dass ein Prozess dann als abgebrochen gilt, wenn alle Teile abgebrochen werden und Gl. B.21, dass ein abgebrochener Prozess gleichgültig jeglicher Fortsetzung abgebrochen ist. Gl. B.23 erlaubt es, Output aus einer Transaktion herauszuholen bzw. in eine Transaktion aufzunehmen. Die letzte Gleichung B.24 ermöglicht, abgeschlossene Transaktionen frei zu verschieben, da diese auch Teil einer größeren Transaktion sein können, gelten sie nicht als *done*. Für das π t-Kalkül ergibt sich, analog den Reaktionsregeln (B.8–B.12) die Erweiterung:

$$\begin{array}{ll} t(t(\text{done}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C})|\mathfrak{P}, \mathfrak{F}', \mathfrak{B}', \mathfrak{C}') \mapsto t(\mathfrak{P}, \mathfrak{F}', \mathfrak{B}'|\mathfrak{C}, \mathfrak{C}') & \text{T-DONE,} \\ t(\text{abort}, \mathfrak{F}, \mathfrak{B}, \mathfrak{C}) \mapsto \mathfrak{B}; \mathfrak{F} & \text{T-ABORT.} \end{array} \quad (\text{B.25})$$

Die Regel T-DONE gibt an, dass auch nach erfolgreicher Beendigung die Kompensation für eine darüberliegende Transaktion aufgehoben werden muss. Die letzte Regel T-ABORT zwingt dazu, den Fehlermanager im Anschluss an eine Kompensation auszuführen.

Literaturverzeichnis

- [1] Albert, I.: 2005, Qualitätsmerkmale von Kontextinformationen, Dissertation, LMU, München.
- [2] Alonso, G., et al.: 2003, Web Services, Springer.
- [3] Apperly, H., et al.: 2003, Service-And Component-Based Development, Addison-Wesley.
- [4] Arsanjani, A.: 2004, Service-oriented modeling and architecture, www.ibm.com.
- [5] Beer, S.: 1994, Beyond Dispute, Wiley.
- [6] Berners-Lee, T. et al.: 2001, The Semantic Web, Scientific American, May 2001.
- [7] Bieberstein, N.: 2005, Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals, IBM Systems Journal, Vol. 44, S. 691-708
- [8] Birman, K.: 2006, The untrustworthy web services revolution, IEEE Computer February 2006 (Vol. 39, No. 2).
- [9] Birman, K. P.: 2005, Reliable Distributed System, Springer.
- [10] Brunner, R. J., et al.: 2002, Java Web Services Unleashed, Sams Publishing.
- [11] Bucchiarone, A., Gnesi, S.: 2006, A Survey of Service Composition Languages and Models, in: International Workshop on Web Services Modeling and Testing 2006.
- [12] Chappell, D., Jewell, T.: 2003, Java Web Services, O'Reilly.
- [13] Chappell, D.: 2004, Enterprise Service Bus, O'Reilly.
- [14] Chatterjee, S., Webber, J., Bunnell, D.: 2003, Developing Enterprise Web Services, Prentice Hall PTR.
- [15] Cherbakov, L. et al.: 2005, Impact of service orientation at the business level, IBM Systems Journal, Vol. 44, S. 653-668.
- [16] Clements, P. et al.: 2003, Documenting Software Architectures: Views and Beyond, Addison-Wesley.
- [17] CORBA: 2006, Common Object Request Broker Architecture, www.omg.com.

- [18] Damm, D.: 2003, Eine IS-Plattform zur Unterstützung kooperativer interorganisatorischer Netzwerke, Dissertation, Zürich.
- [19] Donohoe, P.: 1999, Software Architecture, Springer.
- [20] Dustdar, S., Gall, H., Hauswirth, M.: 2003, Software-Architekturen für verteilte Systeme, Springer.
- [21] Eberhart, A.: 2004, Ontology-based Infrastructure for Intelligent Applications, Dissertation, Universität Saarbrücken.
- [22] ebXML: 2003, electronic business XML, www.ebxml.eu.org.
- [23] Eichhorn, F.: 2002, Evaluation von Webservice-Techniken für den Einsatz zur Business-to-Business Integration (B2BI), Diplomarbeit, Universität Erlangen.
- [24] EJB: 2005, Enterprise JavaBeans Technology, java.sun.com.
- [25] Erl, T.: 2005, Service-Oriented Architecture. Concepts, Technology, and Design, Prentice-Hall.
- [26] Fischer, O., Wenzel, B.: 2004, Prozessorientierte Dienstleistungsunterstützung, Diplomarbeit, Universität Hamburg.
- [27] Fisher, D. A.: 2006, An Emergent Perspective on Interoperation in Systems of Systems, SEI, Carnegie Mellon.
- [28] Flenner, R., et al.: 2002, Java P2P Unleashed, Sams Publishing.
- [29] Flood, R. L., Jackson, M. C.: 1994, Creative Problem Solving – Total Systems Intervention, Wiley.
- [30] Gersdorf, J.: 2001, ebXML – Seminar WWW und Datenbanken, Universität Frankfurt.
- [31] Gioldasis, N., et al.: 2003, Service Oriented Architecture for Managing Operational Strategies, ICWS-Europe 2003, LNCS 2853, Springer.
- [32] Grossmann, M., Koschek, H.: 2005, Unternehmensportale, Springer.
- [33] He, H.: 2003, What is service-oriented architecture?, www.xml.com.
- [34] Hegering, H.-G., Abeck, S., Neumair, B.: 1999, Integrated Management of Networked Systems, Morgan Kaufman.
- [35] Hein, M., Zeller, H.: 2005, Java Web Services, Entwicklung plattformübergreifender Dienste mit XML und SOAP, Addison-Wesley.
- [36] Herring, C., Kaplan, S.: 2000, The Viable System Model for Software, in: 4th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida
- [37] IETF: 2006, Internet Engineering Task Force, www.ietf.org.
- [38] Jablonski, S. et al.: 2004, Guide To Web Application And Platform Architectures, Springer.
- [39] Jeckle, M.: 2002, Webservicearchitekturen.
- [40] Kaye, D.: 2003, Loosely Coupled: The Missing Pieces of Web Services, RDS Press
- [41] Klein, M.: 2006, Automatisierung dienstorientierten Rechnens durch semantische Dienstbeschreibungen, Universitätsverlag Karlsruhe.

- [42] Kneer, G., Nassehi, A.: 1994, Niklas Luhmanns Theorie sozialer Systeme, Fink.
- [43] Krähenbühl, A.: 2006, Soability, A Model for the Strategic Evaluation of an IT Environment's Ability to Support Service-Oriented Architecture, Diplomarbeit, Universität Zürich.
- [44] Kreger, H., Williamson, L., Harold, W. K.: 2002, Java and JMX: Building Manageable Systems, Addison-Wesley.
- [45] Lankhorst, M.: 2005, Enterprise Architecture at Work, Springer.
- [46] Lehner, W.: 2005, Data Management in a Connected World, Springer.
- [47] Liu, K.: 2000, Semiotics in Information Systems Engineering, Cambridge University Press.
- [48] Liu, K. et al.: 2001, Information, Organisation and Technology: Studies in Organisational Semiotics, Kluwer Academic.
- [49] Luckham, D.: 2002, The Power of Events, Addison-Wesley.
- [50] Manes, A. T.: 2003, Web Services, Addison-Wesley.
- [51] Masak, D.: 2005, Moderne Enterprise Architekturen, Springer.
- [52] Masak, D.: 2005, Legacysoftware, Springer.
- [53] Masak, D.: 2006, IT-Alignment, Springer.
- [54] McGovern, J. et al.: 2003, A Practical Guide to Enterprise Architecture, Prentice Hall PTR.
- [55] McGovern, J., et al.: 2003, Java Web Services Architecture, Morgan Kaufmann.
- [56] McGovern, J., et al.: 2006, Enterprise Service Oriented Architectures, Springer.
- [57] Melnik, S.: 2004, Generic Model Management, Springer.
- [58] Mettler, D.: 2004, Government Application Integration, Diplomarbeit, Universität Zürich.
- [59] Microsoft: 2006, Microsoft Biztalk Server, www.microsoft.com.
- [60] Mitschang, B. et al.: 2005, Data Management in a Connected World, Springer.
- [61] Monson-Haefel, R.: 2003, J2EE Web Services, Addison-Wesley.
- [62] Northrop, L.: 2006, Ultra-Large-Scale Systems, The Software Challenge of the Future, SEI, Carnegie Mellon.
- [63] Oaks, P.: 2005, Enabling ad hoc interaction with electronic services, PhD-Thesis, Queensland University of Technology, Australien.
- [64] Peuser, S., Zimmermann, O., Tomlinson, M. R.: 2003, Perspective on Web Services, Springer.
- [65] Polgar, J., Bram, R. M., Polgar, A.: 2006, Building And Managing Enterprise-Wide Portals, Idea Group Inc (IGI).
- [66] Puhlmann, F.: 2006, Why do we actually need the Pi-Calculus for Business Process Management? In: Abramowicz, W., Mayr, H. (Hrsg.): 9th International Conference on Business Information Systems.
- [67] Sahai, A., Graupner, S.: 2005, Web Services in the Enterprise, Springer.

- [68] Scheer, A.-W., et al.: 2003, Business Process Change Management, Springer.
- [69] Schäffer, S., Schilder, W.: 2002, Enterprise Java mit IBM WebSphere, Pearson Education Deutschland.
- [70] Schopp, B.: 2002, Logische Architektur integrierbarer Wissensmedien am Beispiel einer virtuellen Akademie, Dissertation, St. Gallen.
- [71] Schwinn, A.: 2005, Entwicklung einer Methode zur Gestaltung von Integrationsarchitekturen für Informationssysteme, Dissertation, St. Gallen.
- [72] Simon, H.: 1971, Designing Organizations for an Information-rich World, The John Hopkins Press.
- [73] Singh, M. P., Huhns, M. N.: 2005, Service-Oriented Computing, Wiley.
- [74] Shirky, C., et al.: 2001, P2P Networking Overview, The Emergent P2P Platform of Presence, Identity and Edge Resources, O'Reilly.
- [75] SOA: 2006, Reference model for service oriented architecture, www.oasis-open.org.
- [76] Sonic: 2006, Sonic ESB, www.sonicsoftware.com.
- [77] Sprott, D., Wilkes, L.: 2003, Understanding SOA, CBDI-Journal.
- [78] Stojanovic, Z., Dahanayake, A.: 2005, Service Oriented Software System Engineering, Idea Group Inc (IGI).
- [79] Strang, T.: 2003, Service-Interoperabilität in Ubiquitous Computing Umgebungen, Dissertation, LMU, München.
- [80] Tan, Y.-S. et al.: 2004, Service Domains, IBM Systems Journal, Vol. 34, S. 734-755.
- [81] Tidwell, D., Snell, J., Kulchenko, P.: 2001, Programming Web Services with SOAP, O'Reilly.
- [82] van Zyl, J.: 2002, A perspective on service based architecture, Proceedings of SAICSIT, 2002.
- [83] Vetere, G., Lenzerini, M.: 2005, Models for semantic interoperability in service-oriented architectures, IBM Systems Journal, Vol. 44, S. 887 – 903
- [84] W3C: 2003, Web Services Architecture, W3C working draft, www.w3.org.
- [85] Welte, S.: 2005, Entwurf serviceorientierter Architekturen, Diplomarbeit, Universität Karlsruhe.
- [86] Woods, D.: 2003, Enterprise Services Architecture, O'Reilly.
- [87] WSI: 2006, Web Service Interoperability Organisation, ws-i.org.
- [88] Wutka, M.: 2002, J2EE. Java 2 Enterprise Edition., Pearson Education Deutschland.
- [89] XML: 2006, Extensible markup language, www.w3.org.
- [90] Zaplata, S.: 2005, Prozessintegration in Middleware für mobile Systeme, Diplomarbeit, Universität Hamburg.
- [91] Zimmermann, O.: 2003, Perspectives on Web Services, Springer.

Sachverzeichnis

- λ -Kalkül 295, 296
- π -Kalkül 181, 256, 294–296, 367–371
- $\pi\tau$ -Kalkül 370–372
- 3-Tier 88, 115
- 4-Tier 88

- Abort 293
- Absolutheitsanspruch 307
- Abstraktion 29, 55, 72, 88, 92, 93, 96, 104, 121, 138, 195, 203, 213, 231, 232, 237, 271, 273–275, 280, 304, 346, 348, 369
- Abstraktionsniveau 275, 281
- ACID 178, 181, 182, 184
- Adapter 94, 114, 154, 237
- Adapterservice 114
- Administration 140, 141, 146, 165, 170, 235, 347
- Adoption 73, 74, 133
- Affective Computing 124
- Agent 54, 60, 116, 117, 369
- Akzeptanz 6, 120, 228
- Alberti 87
- Algebra 367
- Algorithmus 113, 221, 226, 228, 232, 233, 241, 303, 341
- Alignment 11, 21, 98, 133, 136, 185
- Alkoholiker 324
- Alleinstellungsmerkmal 64
- Allokation 144, 145
- Analogie 22, 86, 298, 307, 312, 358
- Analyse 73, 77, 138, 142, 173, 224, 258, 269–271, 273, 276, 346
- Anpassung 12, 43, 96, 144, 208, 223, 280, 281, 316
- Anreiz 12, 38, 60, 128
- Antioszillationskreislauf 343
- Anwender 5, 22, 120, 122, 193, 208, 221, 226, 229, 263
- Applicationbuilder 156, 171, 195, 230, 233, 281
- Applikation 4–8, 13, 16, 49, 88–159, 192–212, 230, 240, 252, 257–260, 270–299, 343, 345, 369
- Applikationsarchitektur 21, 136
- Applikationslayer 89
- Applikationsschicht 124–126
- Applikationsserver 91, 94, 100, 133, 139, 140, 165, 167–169
- Architekt 135
- Architektur 6–12, 20–30, 87–117, 126, 130–143, 153–156, 192–214, 229–237, 265, 275, 280, 294, 302, 304, 306, 348
- Architekturframework 20, 27
- Architekturgovernance 134
- Architekturprinzip 26, 27
- Architekturzyklus 28
- Archiv 51, 88, 263, 319
- ARSD 364
- Ashby 49, 202, 315–317, 324, 336
- ASIC 364
- ASP 13
- Assembler 231
- Assimilation 304
- Assoziation 156, 195

- Auditing 167
- Aufbauprinzipien 336, 338
- Aufrufbaum 215
- Aufrufzeitpunkt 290
- Aufsichtsrat 58, 343
- Aufwand 4, 46, 93, 95, 135, 163, 199, 203, 279–281
- Ausbildung 82, 228, 306
- Ausnahmebehandlung 108, 179, 180, 184, 278, 287
- Ausnahmezeitpunkt 283
- Ausphasung 196
- Austauschbarkeit 91, 130, 232, 237, 243, 282
- Authentisierung 159, 217, 349
- Autofahrer 279
- Automatisierung 30, 54, 158, 160, 190, 226, 262, 286, 290
- Automobilindustrie 48
- Autonomie 18, 30, 35, 36, 54, 85, 92, 100, 117, 148, 151, 191, 205, 212, 232–234, 272, 294, 299, 333, 336, 337, 342, 343, 349, 364
- Autonomiekrise 35
- Autopoiesis 322, 323, 348
- Autorisierung 111, 159
- Awareness 116
- Axiom 221, 222, 227, 340

- Balance 317, 340
- Bandbreite 30, 163, 165, 234, 298, 332
- Bank 17, 48
- Basislayer 94
- Basisservice 94, 112, 113, 121, 186, 187
- Batchsteuerung 343
- Bedarfsanalyse 281
- Beer 332
- Befehlskette 36
- Begriffsbarrieren 263
- Beherrschbarkeit 304
- Benutzer 5, 69, 91, 94, 119, 121–123, 200, 208, 209, 221, 223, 269, 277, 282, 304, 343
- Benutzerinteraktion 88, 182, 210
- Benutzerinterface 88, 89, 123, 159
- Benutzerzufriedenheit 50
- Bepreisung 142
- Beratung 59
- Berichtsweg 36
- Berufsbezeichnung 19
- Besitzer 27, 235
- Best Practice 75, 180
- Betriebssystem 12, 13, 46, 100, 119, 120, 218, 237, 310, 343, 345, 347
- Bewertung 63, 76, 108, 126, 202, 348
- Bildung 42, 268, 307, 347
- Binding 50, 57, 64, 100, 104, 127, 156, 195, 213, 225, 233, 252, 272, 282–284, 287, 368
- Biologie 316, 321
- Biomechanik 221
- Bitebene 21
- Blaupause 216
- Blickwinkel 1, 4, 63, 67, 71, 73, 135, 187, 208, 216, 274, 282, 323, 353
- Blueprintdokumentation 264
- Blutdruck 343
- Blutzuckerspiegel 316
- BOINC 165
- Boom 49
- Bootstrapping 197
- Bottleneck 143
- Bottom-Up 51, 131, 137, 206, 216, 229, 230, 280
- Boundary 305
- BOV 264
- Boyd 208
- BPEL 50, 105–107, 127, 132, 209, 212, 236, 247–254, 256, 260, 262, 276
- BPM 180
- BPMI 256
- BPML 105–107, 236, 256, 257, 276
- BPMN 256
- BPO 48, 302
- BPSS 236, 276
- BQPL 256
- Brainstorming 86
- Branche 39, 176, 359
- Bremsrakete 219
- Bricolage 206–208, 229, 230, 325, 345
- Broker 26, 49, 54, 66, 72, 73, 92, 143–145, 170, 171, 233
- Brokerarchitektur 143, 145
- Brokerorganisation 64, 71, 72
- Browser 13, 120, 121
- Bubblesort 228
- Bugfixes 196
- Bullshit 35

- Bundestagswahlen 71
- Business-Object-Layer 89
- Businessanalyst 232
- Businesseselement 204
- Businessframeworks 113
- Businessrequirements 271
- Businessservices 86, 130, 131
- Bussystem 41, 153, 154

- Caching 121, 155, 301
- Caesar 31, 231
- Call 116, 285
- Cancel 291
- Capturing 258
- Cardinality 226
- CASE 8, 193
- Caterer 319
- Champollion 262
- Changeovernance 135
- Changemanagement 30, 50, 54, 55, 61
- Chaos 306
- Chaplin 301
- Charakteristik 5, 18, 30, 34, 35, 42, 44–47, 51, 54, 68, 98, 124, 127, 134, 140, 146, 148, 177, 193, 211, 222, 234, 236, 243, 271, 299, 307, 308, 310, 323, 333, 359
- Choreographie 55, 86, 106–108, 113, 181, 214, 236, 252, 254, 256–260, 266, 267, 269, 295
- Cicero 308
- CIO 10, 133
- Citrix 115
- Client 101, 103, 170, 240, 257
- Client-Server-Architektur 88
- Client-Server-Interfaces 163
- Closed-Loop 349
- CLP 341
- Clustering 217, 329
- Clusterkoeffizient 327, 329, 360
- CMDB 172
- CMMI 44, 280
- COBOL 232, 274, 287
- CODAYSL 222
- Code 49, 90, 95, 134, 137, 149, 162, 173, 179, 195, 214, 228, 230, 232, 241
- Collaboration 266–268
- Collection 159
- COM 147
- Commit 157, 182, 291, 293
- Commitment 103, 223
- Compiler 218, 283
- Compilezeitpunkt 50, 283
- Completionprotokoll 293
- Concurrency 157
- Consistency 182
- Consultingunternehmen 2, 5, 10, 59, 74
- Consumer 15–20, 53–56, 62–66, 71–92, 100–125, 141–168, 203–218, 227–291, 303, 318, 343, 346, 351
- Consumerorganisation 61, 62, 68, 72
- Consumerperspektive 108, 206
- Container 123, 165, 167, 169–171, 344
- Content 146, 168
- Controller 324
- Conway 85, 193, 216, 301
- CORBA 6, 7, 91, 92, 95, 101, 117, 118, 120, 150, 153, 155, 158, 160, 162–164, 191, 234, 242, 275, 278, 287, 288
- CORBA-Repository 163
- COS 155
- COTS 216, 281, 283, 300, 302
- Coverage 186, 189
- CPA 265, 267
- CPP 267
- CPU 298
- CRUD 136, 138, 276
- Cryptography 169
- Curietemperatur 312
- Cutoff 328
- Cyber War 341

- Daimler-Chrysler 35
- DAML-S 222, 224
- Darwinismus 279
- Datamining 116
- Datenbank 23, 87, 89, 112, 113, 130, 132, 136, 137, 139, 152, 180, 206, 210, 219, 221, 283, 363
- Datenbankschema 219, 221
- Datenintegration 131, 132
- Datenlayer 94
- Datenmodell 23, 113, 121, 138, 221, 222, 228, 272
- Datenorientierung 86
- Datenredundanz 155
- Datenschutz 49

- Datensemantik 261
- Datenservices 86, 136
- Datensicherheit 154
- Datenzugriffslayer 89, 113
- DCOM 6, 92, 150, 275
- Debugger 286
- Definitionssprache 268
- Defizit 2, 124, 192, 214
- Dekomposition 49, 353
- Delegierung 34, 36, 91, 148, 234
- Delegationsstadium 36
- Demotisch 262
- Dengeln 19
- Denglisch 1
- Denkfaulheit 118
- Denkschemata 310
- Deployment 90, 100, 137, 140, 148, 164, 167, 283, 303
- Designer 27, 208
- Designmaxime 200, 272
- Designmodell 99
- Designphase 218
- Designrichtlinie 89
- Designtimeevolution 191
- Designtimegovernance 134
- Designzeitpunkt 50, 191, 219, 235, 283, 286, 290, 300, 301
- Desillusionierung 32
- Desintegration 42, 52, 55, 211
- Desktop 9, 159
- Deutschpuristen 2
- Dezentralisierung 59, 300, 302
- DHT 164
- Dialog 19, 84, 115, 209
- Dienstleistung 3–5, 14, 16, 19, 118, 184, 192
- Dienstleistungsgesellschaft 4, 5
- Differenz 32, 68, 69, 179, 197, 200, 290
- Differenzierung 32, 67, 76, 79, 108, 306
- DII 118, 162, 163
- Dilemma 144, 235, 294, 306
- Discovery 100, 127, 166, 210
- Diskettenaustausch 7
- Diskurs 222
- Distanz 82, 164, 311
- Disziplinierung 41
- Diversifikation 76, 237, 349
- Division 36, 37
- DoD 129
- Dokumentenfluss 266, 267
- Dokumentenorientierung 51
- Dominanz 3, 236, 320
- DOS 235
- DotCom-Unternehmen 49
- DPD 363, 364
- Drang 118
- Drogenkartelle 32, 38
- DSD 363, 364
- DSI 162
- DTD 222, 239
- Durability 152, 182, 294
- Durchdringung 3, 81, 148, 191, 192, 281
- Durchmesser 329
- Durchsatz 158, 330
- E-Mail 7, 47, 150, 241, 328
- EAI 4, 41, 105, 117, 131, 132, 136, 137, 146, 153, 180, 236, 253, 287
- Effizienz 15, 35, 46, 50, 59, 85, 147, 172, 180, 185, 250, 312
- Eigendynamik 199, 297, 302, 353
- Eigeninteresse 145
- Eigenleben 33, 231, 311
- Eigentum 129
- Einkauf 38, 44, 302
- EJB 133, 163, 169
- Elementarteilchen 221, 222
- Elsterprogramm 299
- Elternservices 287
- Emergenz 30, 208, 300, 301, 310–312, 314, 315, 320, 325, 334
- Endbenutzer 104, 119, 196, 200
- Endknoten 327
- Endpunkt 30, 94, 146, 148, 170, 249
- Energie 33, 279, 311, 322, 323, 362
- Enterprise-SOA 150
- Enterpriseagent 54
- Enterpriselayers 94
- Entkoppelung 46, 71, 85, 101, 140, 150, 163, 165, 170, 309, 318
- Entlassungswellen 50
- Entropie 179, 197, 198, 201, 202, 211, 308, 311, 312, 360–363
- Entropiedefinition 361
- Entscheidungsprozess 35, 58, 76, 83
- Entwickler 195, 279, 286

- Entwicklung 3–10, 21–52, 85–89,
105–155, 191–232, 261, 276–303,
319, 334–339
- Entwicklungsaufwand 68, 95, 280
- Entwicklungsmethodik 134, 191, 203,
314
- Entwicklungsprozess 33, 172, 206, 281
- Entwicklungsumgebung 172, 279, 281
- Entwurfsmuster 121
- Epistem 231
- Equilibrium 199, 313
- Erdbeben 328
- Ereignisorientierung 96
- Erkenntnistheorie 231
- Erl 9
- ERP 10, 26, 112
- Erreichbarkeit 259, 284, 302
- Ersatzservice 289
- ESB 11, 41, 60, 118, 130, 132, 136, 137,
139, 141, 142, 146–150, 153–155,
163–171, 205, 227, 253, 283, 287,
288, 299, 314, 330, 338, 343, 349,
350
- ESB-Implementierung 149, 150, 163,
237
- ESB-Standard 165
- Eskalationsverfahren 71
- Essenz 367
- Ethernet 23
- ETL 155
- Euphemismus 137
- Euro 219
- EVA 296
- Eventarchitektur 96, 97
- Eventhandling 120, 251
- Eventorientierung 253
- Evolution 8, 11, 32, 33, 48, 88, 96, 134,
194, 197, 199, 200, 211, 232, 279,
298, 300, 301, 303, 318, 331, 342,
345
- Evolutionsgesetze 200
- Exception 97, 242, 251, 258, 269, 278,
344
- Exceptionhandling 184, 249, 251, 253,
255, 278
- Externalisierung 214
- eXtreme Programing 206
- Extremwerte 188
- Fabrik 341
- Fachbereich 4, 27, 49, 136, 137
- Fachgebiet 177
- Fachlogik 275
- Fachpresse 133
- Fachwissen 224
- Fahndungsseiten 299
- Fakturierung 160
- Fallback-Optionen 30
- Fanatiker 231
- Faustformel 3
- Feature 107, 196, 204, 214, 215, 271
- Featureanalysis 214
- Feedback 310, 341, 349, 351
- Feedbackloop 70, 339, 343, 348
- Fehlerbehandlung 166, 183, 251, 266,
292–294
- Fehlerfall 78, 255, 283, 286, 371
- Fehlerfreiheit 303
- Fehlermanagement 135
- Fehlermanager 371, 372
- Fehlerminimierung 314
- Fehlerquellen 189, 194, 302
- Fehlertoleranz 301
- Filmproduktion 46
- Fingerprint 245
- Fliessgleichgewicht 316
- FLOP 164
- Fluktuation 217
- Flussdiagramm 264, 295
- Formalismus 178, 370
- FORTRAN 232
- Foucault 231
- Framework 22, 23, 56, 91, 99, 137, 155,
193, 229, 235, 264, 281
- Franchising 38
- Frequenz 63
- FSV 264
- FTP 7
- Fuhrparkverwaltung 61
- Funktionssemantik 261
- Fusion 33
- Gartner Group 9, 91
- Gasttheorie 307
- Gatewayservice 113
- Gehaltsabrechnungslauf 71
- Geiz-ist-Geil 302
- Geldtransfer 160

- Generalisierung 204, 274, 275, 365
 Generatoren 12, 137, 142
 Geschwindigkeit 4, 30, 110
 Gimp 328
 Gioldasis 10
 Gleichgewicht 134, 313, 316, 335, 338,
 339, 341, 344
 Globalisierung 43
 Gluecode 146
 Gnutella 163, 164
 Governance 12, 28, 37, 74, 81, 83, 84,
 96, 128–130, 134, 136, 165, 172,
 205, 234, 300, 321, 322, 324, 325
 Gremien 38
 Grimms Märchen 323, 324
 Grounding 225
 GTK 328
- Haarschnitt 184
 Halstead 179
 Hamiltonfunktion 312
 Hamlet 191, 331, 353
 Handwerker 19
 Hardware 12, 119, 218, 237, 299
 Harmonisierung 333
 Header 150, 266
 Header-Container 266
 Helmholtz 361
 Hermeneutik 208
 Hersteller 100, 146, 163
 Heuristik 347
 HGB 219
 HIC 123, 124, 301
 Hierarchie 23, 34, 35, 49, 59, 83, 162,
 222, 287, 310, 319, 320, 324, 339,
 343
 Hierarchiesgesetz 317
 Hierarchiestadium 34–36, 52
 Hieroglyphen 262
 High-Level-Debugger 142
 Hilfesystem 159
 Himmel 297
 Historyfunktionen 159
 Hochzeitsladen 19
 Hollywoodschauspieler 328
 Homepage 44
 Homo Hierachicus 84
 Homo Oeconomicus 84
 Homo Politicus 84
- Homogenisierung 65
 Homonyme 219, 242
 HSB 60
 HTML 96, 115, 120, 122, 238
 HTTP 115, 119, 135, 169, 209, 218,
 240, 244
 Hub&Spoke 153
 Humboldt 3
 Hundebesitzer 73
 Hydrodynamik 340
 Hyperlinks 296
- IAS 219
 IBM 13, 96, 247
 ICQ 163
 Idealismus 220
 IDL 118, 162, 242, 288
 IFV 264
 IIS 13
 Illusion 302
 Implementierung 19–29, 49, 61, 68,
 74–107, 120–138, 162–168, 200,
 203–237, 245, 265–294, 309, 337,
 342, 348
 Implementierungsform 118, 134, 213,
 246
 Implementierungsgovernance 29, 134
 Implementierungsmodell 202, 203
 In-band-Broker 145
 Individuum 54, 60, 308
 Industrieproduktion 65
 Ineffizienz 84
 Information 12, 153, 199, 363
 Information-Bottle-Necks 320
 Informationsbroker 143, 145
 Informationsarchitektur 11, 21, 28, 29,
 136, 202, 206, 265
 Informationsaustausch 21, 47, 63, 115,
 159, 219, 300, 363
 Informationsfluss 19, 42, 132, 154, 178,
 300
 Infrastruktur 10, 46, 48, 71, 84, 85,
 91, 92, 96, 97, 108, 113, 120, 124,
 126, 127, 130, 131, 133, 139–141,
 144–146, 148, 155, 172, 193, 224,
 241, 273, 281, 298, 299, 323
 Innovation 238
 Insourcing 205

- Instanz 177, 189, 249, 252, 253, 290, 304, 336
 Instanziierung 123, 155, 256
 Institutionalisierung 42, 59
 Intasking 52
 Integrationskosten 64
 Integrationsmodell 181
 Integrationservices 137
 Intelligenz 335, 361
 Interface 5–29, 49, 56, 61, 74–173, 212–278, 283–293, 307, 309, 318, 319, 342–346
 Internetseite 13
 Internetzeit 32
 Invarianten 343
 Inventarisierung 172
 Isolation 22, 177, 178, 182, 184, 271
 Isomorphismus 73
 IT-Governance 12, 128
 IT-Infrastruktur 46, 48, 139, 140
 IT-Sicht 86, 136
 IT-System 123, 299
 IT-Umgebung 124, 139
 Iteration 208, 255
 ITIL 14, 139
 Itinerary 168

 JAAS 169
 Java 8, 97, 162, 168–170, 193, 232, 234, 287
 JAXB 169
 JBI 169
 JBoss 91
 JCA 169, 170
 JDK 328
 JMS 169, 170
 JMX 170
 JTA 169
 JXTA 165

 Kant 297
 Kerberos 236
 Kernkompetenz 40
 Kernkomponente 245
 Kernprozess 180
 Kernservice 67
 Kernsystem 134
 KI-Planung 286
 Kindprozess 257

 Know-how-Aufbau 11
 Kollaboration 45, 53, 107, 126, 128, 129, 233, 303
 Kollaborationsservices 130
 Kollaborationsstadium 37
 Kollektiv 118
 Kommunikationsmuster 114, 153, 271
 Kommunikationspfade 301
 Kommunikationsprofil 126
 Kommunikationsstruktur 85, 193
 Kommunikationstheorie 208
 Kompensation 107, 110, 183–185, 276, 287, 292–294, 371, 372
 Komplexitätsreduktion 305
 Komponentenarchitektur 90, 94
 Komponentenbauweise 92, 122, 132, 137, 138, 191, 193, 279
 Komponentensoftware 193, 269, 272
 Komposition 55, 62, 104–106, 109, 127–129, 135, 204, 209, 210, 212, 224, 229, 247, 249, 269, 280–287, 303, 307, 328, 343, 345, 367, 368
 Kompositionsmodell 105, 106, 140, 284–286
 Kompositionszeitpunkt 282
 Kompositservice 106, 286, 294, 328
 Konfiguration 42, 81, 100, 135, 146, 163, 166, 233, 265, 325, 338, 351
 Konfigurierbarkeit 100
 Kongruenz 368, 370, 371
 Konjugation 365
 Konjunkturflaute 59
 Konsistenz 49, 90, 172, 182, 258
 Konsolidierung 273
 Konstruktionsprozess 210
 Kontext 15, 18, 33, 73–78, 110–114, 138, 166–177, 204, 206, 217, 238, 244, 255, 271–289, 295, 306–311, 337, 344, 358, 368
 Kontexttransfer 177–180
 Kontinuumansatz 328
 Kontrollalgorithmus 342
 Kontrolldesign 349
 Kontrollierer 272, 343, 345, 346, 348–350
 Kontrollerdesign 346, 348
 Kontrollfluss 178, 182, 183, 252–255, 262, 264
 Kontrollinformationen 266
 Kontrollinstanz 232, 294, 295, 345

- Kontrollkrise 36
 Kontrolllogik 345
 Kontrollmechanismus 333
 Kontrollsystem 35, 315–317, 324, 345
 Kontrollverlust 343
 Konvergenz 217
 Konvertierung 161
 Konzentration 35, 130, 228
 Konzeptionalisierung 220, 223
 Kooperation 37–40, 42–45, 77, 233, 315, 334
 Kooperationspartner 42, 62, 108
 Kooperationsprozess 80
 Kooperationsstadium 52
 Koordinationsinstanz 182
 Koordinationsprotokoll 292
 Koordinator 78, 106, 107
 Koordinierungsmechanismus 326
 Koppelung 364
 Korrelationsmenge 253
 Korrelationsschema 252
 Kosmologie 223
 Kostentreiber 193, 199
 Kultur 2, 31, 44, 128, 129, 219, 318
 Kulturkreis 290
 Kundenbeteiligung 5
 Kundenkontakt 17, 35
 Kundenorientierung 5, 51
 Kundenzufriedenheit 5, 51
 Kybernetik 331
- Landauer 363
 LAP 208–210
 Lastverteilung 120, 165
 Latenz 56, 103, 155, 234, 235, 250
 Laufzeit 77, 90, 100, 118, 141, 147, 162, 195, 211, 215, 224, 251, 256, 257, 262, 286, 287, 301, 303, 364
 Laufzeitbibliotheken 195
 Laufzeitumgebung 94, 114, 122, 146, 155, 170
 Law of Requisite Hierachy 317
 Law of Requisite Variety 316, 326
 Layer 12, 88, 89, 92–94, 96–98, 106, 112, 113, 149, 209, 222, 264, 320, 348
 Layerarchitektur 88, 93, 132, 135
 LDAP 111, 236
 Lebensdauer 164, 171, 199, 211, 300
 Lebenszyklus 54, 55, 57, 63, 71, 141, 172, 194, 196, 220, 223
 Lebenszyklusmodell 194–197, 252
 Legacysoftware 4, 136, 137, 168–171, 201, 215
 Legitimation 175, 177
 Legoprinzip 4, 195
 Lehman 198, 201
 Leistungserstellung 42
 Leistungsumfang 27, 202
 Lenkungsstrukturen 336
 Lernkurve 130
 Lieferant 11, 64
 Lifecycle 122, 134, 157, 167
 Linux 12, 46, 328
 Lizenzierung 158
 Lizenzierung 158
 Locke 305
- Macbeth 175
 Machbarkeit 105, 136
 Machtverteilung 34, 35, 39
 Mafia 38
 Magnetismus 312
 Mainframe 102
 Maintenance 26, 70, 137, 193, 196, 211, 223, 278, 282
 Makler 72, 143
 Management 35–37, 44, 54, 58, 68, 83, 84, 124, 127–129, 140, 155, 156, 158, 159, 170, 173, 223, 240, 280, 304, 340, 343
 Managementumgebung 123
 Manager 1, 59, 347, 371
 Manipulation 221
 Mannheim 298
 Mapping 160, 170
 Marktbezug 35
 Marktchancen 15, 43
 Marktdruck 217
 Markterfolg 59
 Marktkontrolle 235
 Marktwissen 35
 Marshaling 160, 287
 Maschinencode 231
 Matching 283
 Matchmaking 286
 Materialismus 220
 Matrixorganisation 37

- Matrixstruktur 37
- Maturity 129, 131, 132
- Maurer 19
- Maxime 200, 232, 272
- McCabe 179
- MDA 8
- Mechanik 309, 361
- Mediation 73, 227
- Medulla 343
- Meet-In-The-Middle 217
- Mehrwert 12, 26, 33, 37, 46, 49, 55, 85, 199, 230
- Mensch-Maschine-Interaktion 229
- Mensch-Maschinen-Grenze 301
- Menschenbild 47, 84
- Menschwerdung 223
- Meritokratie 128
- Messageaustausch 107, 108, 119, 212, 225, 257, 258, 265
- Messagebroker 153
- Messagefilterung 349
- Messageformat 242
- Messagekanal 154
- Messagekorrelation 253
- Messageorientierung 6
- Messagestruktur 170, 225, 263
- Messagetransport 263
- Messaginginfrastruktur 154
- Messagingservice 265
- Messbarkeit 357, 358
- Messung 189, 311, 357–359, 362, 363
- Metadaten 172, 283
- Metafeedbackloop 351
- Metaframe 115
- Metamessage 168
- Metamodell 171, 172, 226, 227
- Metaontologie 224
- Metaprinzip 338
- Metaservice 112, 166
- Metasprache 238
- Metasystem 345–347
- Metasysteminterfaces 346
- Metcalf 329
- Metrik 61, 126, 188, 189, 358–360
- MFC 237
- Microsoft 13, 64, 91, 96, 235, 247
- Middleware 119, 132, 150, 263, 287, 292
- Migration 26, 29, 58, 74, 114, 134, 214
- Mikro-SOA 98
- Mikrophon 159
- MIME 266
- Minimierung 237
- Mischkalkulation 65
- Mischlast 71
- Misstrauen 37, 40, 58
- Mittlerrolle 202
- Mode 10, 45
- Modellierungswerkzeuge 142
- Modularisierung 29, 214, 237
- MOM 150–154, 164, 168, 287
- Monitoring 60, 77–79, 109, 140, 165, 166, 170, 211, 334, 349
- Monolith 87, 88
- Monopol 238
- Morphogenese 324
- Mozilla 328
- Multi-Tier-Architektur 113
- Multiscale Law of Requisite Variety 326
- Multiusersystem 310
- MVC 122, 346
- Namensraum 56, 156, 157, 254, 257, 268
- Namingservice 156
- Napster 163
- Naturgeschichte 297
- Navigationssystem 233
- Nervensystem 332
- Netzwerk 36–39, 46, 47, 49, 62, 95, 101, 102, 135, 151, 160, 163–165, 222, 322, 327, 328, 330
- Netzwerkdurchmesser 360
- Netzwerkstadium 37, 38, 48, 51
- Netzwerktyp 328
- Newton 309
- NMS 169
- Non-Profit-Organisation 256
- Nutzungsbaum 187
- Nutzungshandbuch 278
- Nutzungsphase 134
- Objektlayer 89
- Objektmodell 89, 118
- Objektorientierung 5–8, 130, 138, 158, 160, 161, 191, 193, 222, 232, 233, 274, 275, 279, 280, 290, 341

- Obligation 20
Observable 312
Ontologie 49, 121, 134, 173, 202, 209,
211, 218–228, 261, 269, 273, 274,
286, 290
Open Source 46
Operationseinheit 346
Operationalkontrolle 344
Opportunismus 44
Optimierung 36, 45, 77, 82, 83, 94, 98,
131, 144, 172, 215, 256, 311, 332,
334
ORB 118, 155, 156, 158, 160, 162
Orchestration 57, 106, 107, 113, 181,
292–295
Organisationsform 5, 35, 37, 40, 42, 44,
73, 85, 148
Organisationsfunktion 99
Organisationsidentität 46
Organisationskultur 43, 59, 128
Organisationsleistung 176
Organisationsprofil 126
Organisations-schicht 127
Organisationsstruktur 35, 36, 48, 85,
216
Organisationsumgebung 53
Organismus 316, 318
Ortstransparenz 234
Othello 139
Out-of-band-Broker 145
Outsourcing 4, 19, 44, 51, 52, 61, 82,
133, 187, 295
Outtasking 52, 53
Overhead 93, 126, 144
Overnet 163
OWL-S 107, 209, 224–228, 278

P2P 96, 106, 126, 146, 163
Paketierung 214, 265
Paradigma 4, 6, 9, 16, 24, 102, 202, 229,
232, 303
Paradigmenwechsel 9, 232
Paradoxon 235
Parnas 353
Parser 217
Partitionierung 94, 95, 113
Patches 195, 196
Pattern 7, 209, 346
Payload 150, 151, 168, 169
Payload-Container 266
PBC 169
PDF 247
Peer 164
Performanz 71, 110, 121, 126, 127, 137,
142, 164, 275, 283
Permutation 368
Persistenzlayer 89
Personalabrechnung 71
Pervasive Computing 148
Petri-netz 296
Philosophie 220, 229, 265
Phishing 135
Physik 221, 222, 312, 321, 361
Physiognomie 73
PIP 262, 263, 265
PKI 236
Plant 341, 344, 345
Planungsphase 55
Platon 220
Plattform 11, 41, 60, 99, 168, 180, 287
Plautus 308
plenum 10
Poissonverteilung 327
Policy 54, 111, 112, 135, 158, 205, 338,
344
Politiker 175
Pons 343
Portale 122
Portfolio 75–77, 273
Portierbarkeit 193
Portlets 121
Postmatch 284
Potenzgesetz 198, 312
Powerpointfolien 59
Praxistauglichkeit 228
Priester 34
Primat 28, 132, 175, 176
Problemstau 57
Produktbundeling 65
Produktlebenszyklen 11
Produktzyklen 11
Profiling 233
Profitcenter 34
Prognose 70, 137, 307
Programmiersprache 103, 162, 229,
230, 232, 233, 247, 259, 288

- Protokoll 57, 92, 101, 106, 116, 119, 126, 166, 169, 209, 236, 238–241, 247, 265, 283, 288–290, 292–294
- Provider 4, 13–20, 40–75, 82–114, 122, 132–170, 187–218, 230–245, 278–290, 302
- Providerorganisation 35, 61, 62, 64, 68, 71
- Provisioning 148
- Proxy 103, 113
- Prozessablauf 49
- Prozessalgebra 367
- Prozessautomatisierung 158
- Prozessdesign 49
- Prozessebene 78, 212, 227
- Prozessfluss 89
- Prozessinstanz 252–254
- Prozessintegration 131, 133
- Prozesskongruenz 368
- Prozesskontrolle 115
- Prozesskoordination 77–79
- Prozesslayer 88, 89
- Prozesslogik 107, 115
- Prozessmanagement 140
- Prozessmodell 74, 78, 171, 225, 252
- Prozessorientierung 86
- Prozessservice 114, 115
- Prozessspezifikation 81, 286
- Pseudoadoption 73
- Psychologie 3, 58
- Pufferung 237
- Pulsschlag 316
- Puristen 1
- Pyramide 35, 36

- QoS 108, 109, 111, 136, 141, 154, 158, 163, 167, 173, 184, 206, 211, 238, 261, 274, 278, 290, 343, 348, 349

- Ranking 56, 109, 211
- Rating 62, 63, 211, 359
- Ratingverfahren 359
- RDF 222
- REACT 370
- Reaktionszeit 44
- Real-Time-Integration 155
- Realignment 202
- Rechercheprovider 319
- Red-Tape 37
- Redswoosh 163
- Reduktion 11, 98, 134, 203, 204, 315, 324, 339
- Redundanz 48, 77, 98, 137, 273, 345, 349
- Reed 329
- Reengineering 59, 213, 263, 303
- Refaktoring 138, 216
- Referenz 118, 246, 256, 278, 358
- Reflektion 313
- Regelwerk 27, 54, 298, 333
- Registrationszeitpunkt 282
- Registry 95, 101, 103, 104, 141, 157, 163, 217, 234, 243, 244, 265, 282
- Regulation 284
- Regulator 315–317, 343, 346, 347
- Reichweite 141, 148
- Reifegrad 81, 82, 120, 133
- Reifegradmodell 81, 129, 131
- Reinstanziierung 89
- Reinterpretation 2, 208
- Reiseunternehmen 72
- Rekomposition 195
- Rekonfiguration 230
- Rekursion 319, 332–334, 336, 338–340, 343–346, 348
- Relaxationszeit 339
- Release 200
- Rendering 159
- Reorganisation 60, 321, 351
- Replikation 80, 349, 367, 368, 370
- Reportingverfahren 71
- Repository 118, 130, 141, 145, 162, 171, 230, 265, 309
- Reproduktion 310, 322, 323
- Reputation 55, 56, 62, 63, 109, 284
- Requestor 9, 99, 116, 218
- Requests 109, 143, 145, 162, 240
- RES 370
- Responsezeit 164
- Ressource 62, 84, 144, 157, 163, 164, 205, 206, 277, 334
- Restrukturierung 48, 58, 59, 67, 331
- Reuse 280
- Reuters 351
- Revolution 9, 11, 33, 35, 36, 306
- Richterskala 328
- Risiko 29, 51, 73, 75, 85, 143, 202, 235, 313, 314, 324, 339

- RMI 6, 150
- RN-PIP 236
- RNBD 263
- RNIF 263
- RNTD 263
- Robustheit 89, 322
- RogueWave 237
- ROI 11
- Rollback 182, 184, 185, 291, 293
- Rollen 1, 107, 264, 268
- Rosetta 262
- RosettaNet 236, 262–265
- Routing 127, 140, 146, 149, 168
- RPC 6, 119, 147, 150, 153, 234
- RQL 222
- Runtimegovernance 134
- RUP 204, 206, 269

- SABRE-System 208
- SAML 132, 236
- Sanktionen 38
- SAP 64
- Schicht 32, 93, 124, 343
- Schwarm 301
- Scoring 359
- Sedimentation 119, 120, 140
- Seinskomponenten 220
- Seiteneffekte 78, 277, 297, 311
- Selbstorganisation 30, 83, 310, 320, 321
- Selbstreferenz 343
- Selbstregulierung 199
- Selbstreproduktion 30
- Selektion 51, 62, 211, 302, 336, 337
- Semantik 14–21, 49, 51, 77, 107, 112, 157, 202, 217–244, 277–290, 370
- Sender 150, 151, 208
- Separation 21, 29, 176, 272, 274, 343
- Serialisierbarkeit 182
- Serialisierung 257
- Servercontainer 167, 168
- Serverframework 163
- Service 101, 184, 195, 275
- Serviceadresse 102
- Serviceaggregation 145
- Serviceanalyse 270, 271
- Serviceanfrage 103
- Servicearten 112, 114, 158
- Serviceaufruf 20, 63
- Servicebaum 364
- Servicebeschreibung 19, 145
- Servicebroker 218, 230
- Servicecharakter 85, 119, 185
- Servicecommitment 103
- Serviceconsumer 14, 19, 20, 98, 99, 101–104, 148, 243, 244
- Servicecontainer 123, 165–170, 272
- ServiceDatenbank 349
- ServiceDefinition 18, 71, 92, 112, 168
- ServiceDesign 270–272
- ServiceDiscovery 210, 262
- Serviceebene 55, 133, 171, 203, 212, 288, 320
- Serviceentkoppelung 64, 71
- Serviceentwickler 75, 171
- Serviceentwicklung 6, 11, 74, 204, 269
- Servicefokus 143, 145
- Servicegedanken 155
- Servicegrenzen 272
- Servicehandling 128, 129
- Serviceidentifikation 213, 214, 273
- Serviceimplementierung 100, 101, 142, 167, 224, 289
- Serviceindustrie 65
- Serviceinfrastruktur 139, 234
- Serviceintegrator 286
- Serviceinterface 72, 242, 274, 290
- Servicekandidat 125, 126
- Serviceklassifikation 224
- Servicekomposition 16, 18, 104, 142, 170, 192, 212, 262, 282
- Servicekonsolidierung 273
- Servicekoordination 82, 83
- ServiceLayer 92
- Servicelease 103, 104
- ServiceLinie 66–68
- ServiceModell 101, 202, 203, 205
- ServiceNutzung 61, 64, 142, 143, 226, 227, 237, 262, 364
- Serviceorientierung 1–6, 8, 10–12, 16, 20, 23, 24, 26–29, 35, 61, 82–85, 114, 121, 124, 127–129, 137, 139, 140, 172, 173, 192, 193, 199, 200, 202, 204, 206, 210, 212, 216, 232, 233, 269, 271, 273–275, 279, 281, 308, 348, 353
- Serviceorientierungsparadigma 4, 6–8, 16, 18, 24–26, 41, 59, 84, 85, 92, 135, 192, 211, 213, 230, 234

- Servicepaketierung 214
- Serviceportfolio 208, 363
- Serviceprovider 4, 14, 18–20, 27, 49, 52, 54, 64, 72, 98, 99, 101–104, 163, 169, 170, 230, 238, 243, 318, 319
- Serviceproxy 103
- Servicerationalisierung 273
- Serviceregistry 101–103, 141, 143, 145
- Servicerekursion 320
- Servicerepository 130, 141, 281
- Servicevertrag 95, 102, 103, 271
- Servicevirtualizer 143, 145
- Servicingphase 197
- Sessionhandler 291
- SETI 164
- Shakespeare 1, 31, 139, 175, 191, 231, 331, 353
- Sicherheitsprofil 126
- Siloarchitektur 276
- SIM 82, 154, 171–173, 281, 283, 287, 323
- SIMM 130, 132
- Simulation 173, 346
- Sinnentleerung 177
- Skalierbarkeit 8, 30, 95, 125–127, 134, 143, 154, 241, 299
- SLA 14, 20, 62, 70, 127, 129, 205, 210, 211, 260
- Smalltalk 115, 232
- SMTP 170
- SOA 2, 7–12, 26, 28, 91–157, 180, 192, 198, 215, 216, 244, 302–309, 320–326, 353
- SOA-Definition 92
- SOA-Governance 74, 133, 321
- SOA-Hype 2, 10, 133
- SOA-Migration 124
- SOA-Paradigma 102
- SOAbility 124–126, 128
- SOAMM 130–132
- SOAP 101, 117–120, 126, 132, 135, 169, 209, 235, 239–241, 243, 244, 263, 265, 266, 291, 292
- SOC 6, 7, 10, 26, 192, 229, 232, 234, 235, 291
- SODA 192
- SOE 10, 25, 48–55, 57, 60, 61, 72, 73, 75, 79, 81, 82, 92, 192
- Softwarearchitektur 92, 229, 230, 280
- Softwareattacken 341
- Softwarebibliotheken 237
- Softwaredarwinismus 172, 279, 280, 344
- Softwareentwickler 1, 113–115, 128, 129, 136, 138, 142, 163, 196, 231, 233, 235, 277, 279–283, 286, 290, 335, 343
- Softwareentwicklung 5, 13, 29, 32, 92, 120, 124, 126, 129, 173, 176, 190–193, 204, 209, 219, 231, 232, 234, 235, 277, 280, 298
- Softwareentwicklungsparadigmen 232
- Softwareevolution 191, 197, 211, 230
- Softwarefactory 5
- Softwarehersteller 2, 8, 65, 136, 146, 169, 238
- Softwareparadigma 8, 195
- Softwaresalvaging 11
- Softwarestruktur 5, 124
- Softwaresystem 54, 90, 102, 112, 208, 308, 335, 337, 348
- Softwareupdates 219
- SOI 133
- SOP 10, 132, 135, 140–143, 170, 287, 343
- SOS 192, 194, 195, 197–201, 206, 210, 211, 229, 230, 270
- SOSE 6, 10, 192, 208, 229, 272
- Sozialisation 44
- Spezifikation 29, 70, 71, 77, 79, 80, 203, 220, 223, 232, 233, 236, 241, 243, 245, 248, 256, 264, 267, 271, 280, 291–293, 298, 346, 349, 358
- Spoofing 135
- Sprachakt 208, 209
- SQL 222
- SSL 236
- Stagnation 35, 58
- Stakeholder 128
- Standard 12, 37, 64, 73, 79, 82, 96, 105, 111, 127, 130, 132, 135, 147, 148, 150, 165, 169, 226, 228, 233, 235–237, 262, 265
- Standardinterfaces 153, 230, 238
- Standardisierung 5, 48, 64, 96, 130, 134, 148, 167, 238, 247, 262
- Status 32, 77, 295
- Stereotyp 45

- Steuerungsaxiome 340
- Steuerungseinheit 335
- Steuerungskreis 336
- Steuerungslayer 94
- Steuerungsmechanismus 317
- Steuerungssystem 219
- STRUCT 370
- Student 229
- Stufenmodell 81, 196
- Subcontracting 285
- Sublayer 94
- Suborganisation 49, 60, 84, 205
- Subscription 152, 153, 244
- Suchalgorithmus 226
- Suchraum 286
- Superklasse 290
- Superprinzip 337
- Supervision 347
- Supportphase 223
- Supportquotient 188, 189
- Sympathikus 343
- Synchronisation 79
- Synergie 334
- Synthese 62
- Systemadministrator 343
- Systemarchitektur 29, 92, 94
- Systemausfall 121
- Systembildung 305
- Systemdenken 306
- Systemfehler 313
- Systemgrenzen 40, 43, 200, 272, 282, 318
- Systemidentifikation 345
- Systemintegrator 281
- Systemkonfiguration 232
- Systemkoppelung 79
- Systemsicht 298, 319
- Systemsprache 338
- Systemstruktur 324, 325
- Systemtheorie 1, 30, 54, 305–307, 315, 316, 318
- Systemumgebung 78
- Systemzustand 121

- Tapetransfer 7
- Tarantella 115
- TAU 370
- Taxonomie 202, 217, 222, 243, 244, 261
- Taylorismus 305

- TCP 135
- Team 37, 46, 60, 128, 195
- Technologiearchitektur 21, 29, 136, 202
- Technologiegateway 113, 114
- Teilprozesse 39, 180
- Temperatur 312, 362
- Terrorismus 38
- Thermodynamik 197, 361
- Thrashing 310
- Timeouts 184, 255, 269
- TLS 236
- TOGAF 27, 28
- Top-Down 51, 193, 230, 260, 280
- Topologie 40, 134, 234, 272
- Tourismussektor 48
- TP-Monitor 210, 291
- Trader 158
- Transaktion 40–42, 87, 100, 107, 110, 127, 154, 157, 160, 169, 180–185, 209, 210, 255, 257, 267–269, 291–293, 371, 372
- Transformation 140, 170, 338
- Transparenz 62, 77, 84, 85
- Transport 102, 119, 148, 150, 155, 214, 235
- Trustcenter 19
- Tschernobylkatastrophe 314
- TTL 164

- UBL 220
- UDDI 101, 118, 119, 141, 163, 209, 218, 241–247, 256
- ULS 165, 171, 298–304, 321, 345, 351, 353
- Umgebungsbegriff 312
- Umgebungsverhalten 312
- UML 8, 296
- Umsatz 133
- Umweltdruck 336
- Unbeherrschbarkeit 304, 323, 324, 347
- Unberechenbarkeit 318
- Unicode 118
- Universum 208, 222, 223, 314, 323, 324
- Unix 100, 285
- Unmarshaling 160, 287
- Unternehmensberater 59
- Unternehmenskultur 82
- Unternehmensplanung 75
- Unversehrtheit 266

- Unzufriedenheit 68
- Update 237
- URI 118, 141, 145, 164, 168
- URL 118, 243, 245, 258
- Ursache-Wirkungs-Beziehungen 302, 306
- US-GAAP 219
- Use Case 204
- Utility 94, 113
- UUID 244

- V-Modell 206
- Variationspunkt 68
- Verbreitung 138, 237
- Vererbung 195, 217, 221, 232
- Verhaltensparameter 125
- Verifikation 19, 59, 126, 233
- Verteidigungsministerium 129
- Vertrag 14, 27, 70, 86, 102–104, 108, 205, 269
- Vertrauen 37, 45, 47, 49, 55, 56, 61–63, 72, 98, 109, 128, 335
- Verwaltung 37, 64, 68, 76, 142, 153, 172, 294
- virtuelles Enterprise 40, 42–48, 53, 72, 175, 302, 318, 329
- Virus 116
- Visualisierung 76
- Volumen 200, 237
- Vorgehensmodell 206, 230
- Vorhersagekraft 176
- VPN 236
- VSM 51, 54, 60, 304, 331–333, 336–339, 341–345, 348, 349
- VSS 19, 29, 233, 304, 341, 343, 344, 348, 349, 351

- W3C 9, 239
- Wachstum 4, 7, 30, 33, 35, 87, 95, 137, 198, 200, 298, 302, 321, 328–330
- Wachstumsmodell 198, 201
- Wachstumsperioden 33
- Wahrnehmungsorgane 361
- Wandlung 48, 61, 172
- Wasserfallmodell 269
- Webserver 13, 94, 120
- Webservice 82, 101, 105, 106, 115–120, 122, 123, 126, 135–137, 150, 225, 244, 245, 247–249, 252, 253, 259, 264, 291, 292
- Webservedefinition 119
- Wechselwirkungssymmetrie 312
- Werkzeug 171, 173
- Werkzeughersteller 137, 180
- Wettbewerb 45, 58, 233, 235, 298
- Wiederverwendung 11, 15, 27, 29, 51, 65–68, 85, 89, 90, 95, 106, 114, 117, 126, 128–130, 134, 136, 137, 172, 185, 188, 193, 195, 216, 224, 229, 232, 270, 275, 279–281
- Wiederverwendungsmetrik 188
- Wissensakquisition 223
- Wissensprofil 126
- WML 122
- Workflow 79, 80, 89, 94, 127, 158, 209, 210, 294
- WS-AtomicTransaction 291–294
- WS-BPEL 247
- WS-BusinessActivity 292–294
- WS-CDL 106, 107, 209, 257–260
- WS-Coordination 291–293
- WS-Inspection 246
- WS-Policy 111
- WS-RM 236
- WS-Security 111, 132, 236
- WS-Suite 132
- WS-Transaction-Familie 291
- WSCl 107, 236, 253–256, 276
- WSCL 236
- WSDL 101, 118, 119, 123, 132, 209, 224, 225, 228, 235, 241–243, 246, 247, 250, 253, 254, 257, 260, 261, 265, 276, 278, 290, 343
- WSFL 123, 256
- WSIL 246
- WSMO 226, 227
- WSUI 122
- WSXL 122, 123
- WWW 13, 115, 261

- X-Windows-System 120
- XACML 111, 236
- XAML 111, 209, 236, 291
- XLANG 236, 247, 256
- XML 111, 115, 117–122, 126, 151, 218, 219, 222, 238–244, 246, 247, 257, 263, 265, 266, 287

XtremWeb 165

Yahoopops 328

Zachman-Framework 21–27

Zeitskala 32, 84, 197

Zentralisierung 17

Zielerreichungsmetrik 189

Zielerreichungsquotient 189

Zivilisation 318

Zukunftsorientierung 335

Zustandsinformation 6, 77, 78, 104,
178, 363

Zustandslosigkeit 121, 176, 209, 241

Zwangsbedingung 326

Zwischenhirn 343

Zyklus 27

Zyl 9