

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Dieter Masak

# Digitale Ökosysteme

Serviceorientierung bei dynamisch  
vernetzten Unternehmen

Dieter Masak  
plenum Management Consulting  
Hagenauer Str. 53  
65203 Wiesbaden  
dieter.masak@plenum.de

ISBN 978-3-540-79129-4

e-ISBN 978-3-540-79130-0

DOI 10.1007/978-3-540-79130-0

Xpert.press ISSN 1439-5428

Bibliografische Information der Deutschen Nationalbibliothek  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;  
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© 2009 Springer-Verlag Berlin Heidelberg

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten waren und daher von jedermann benutzt werden dürften.

*Einbandgestaltung:* KünkelLopka, Heidelberg  
*Satz und Herstellung:* le-tex publishing services oHG, Leipzig

Gedruckt auf säurefreiem Papier

9 8 7 6 5 4 3 2 1

springer.de

*...für Christiane...*

*Dr. Dieter Masak*

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	1
<b>2</b>	<b>Grundlagen der Serviceorientierung</b>	11
2.1	Serviceorientierungsparadigma	11
2.2	Service Oriented Enterprise	18
2.3	Intelligente Organisation	28
2.4	Flexibilitätsdesign	33
2.5	Service Oriented Architecture	34
2.6	Service Oriented Platform	57
2.7	Service Oriented Computing	72
2.8	Service Oriented System Engineering	85
2.9	Sicherheit	101
<b>3</b>	<b>Mobile Computing</b>	113
3.1	Mobile Computing	115
3.2	Übertragungstechnologien	120
3.3	Mobile Services	125
3.4	Serviceentdeckung in mobilen Umgebungen	127
3.5	Präsentationsservices	134
3.6	Kontextwahrnehmung	137
3.7	Verhandlungsprotokolle	141
3.8	Serviceroaming	143
3.9	Webservices	144
<b>4</b>	<b>Pervasive Computing</b>	145
4.1	Ambiente Intelligenz	153
4.2	Smartspaces	158
4.3	Technik	159
4.4	Multitierarchitektur	163
4.5	Autonome Kommunikationsservices	165
4.6	Entwicklungsmodelle	167

4.7	Dynamische Komposition .....	170
4.8	Evolution .....	171
<b>5</b>	<b>Digitale Businessökosysteme .....</b>	<b>173</b>
5.1	Evolution der IT-Adoption .....	176
5.2	Gesundheit und Robustheit .....	178
5.3	Ecosystem Oriented Architecture .....	180
5.4	Habitate .....	188
5.5	Peeringtaktiken .....	191
5.6	Modelle .....	192
5.7	Infrastruktur .....	194
5.8	Basisservices und semantische Services .....	196
5.9	Broker .....	197
5.10	Digitale Spezies .....	197
5.11	Evolution und Komplexität .....	200
5.12	Verhandlungen .....	202
5.13	Transaktionen .....	204
5.14	Komplexität und Modelle .....	205
<b>6</b>	<b>Digitale Ökosysteme .....</b>	<b>209</b>
6.1	Netzwerke .....	214
6.2	Softwareentwicklung .....	217
6.3	Digitale Biologie .....	223
6.4	Thermodynamik und statistische Mechanik .....	224
6.5	Coevolution .....	228
6.6	Zeitliche Evolution .....	231
6.7	Durchsatz .....	233
6.8	Selbstadaption .....	233
6.9	Ultra Large Scale Systems .....	236
6.10	Überlebensfähigkeit .....	241
<b>7</b>	<b>Epilog .....</b>	<b>245</b>
	<b>Anhang .....</b>	<b>247</b>
<b>A</b>	<b>Systemtheorie .....</b>	<b>249</b>
A.1	Komplexe Systeme .....	251
A.2	Enge Koppelung .....	256
A.3	Ashby-Conant-Theorem .....	257
A.4	Strukturen .....	259
A.5	Rekursionen .....	260
A.6	Autonomie .....	261
A.7	Selbstorganisation .....	261
A.8	Autopoiesis .....	263
A.9	Unbeherrschbarkeit .....	265
A.10	Skalenfreie Netzwerke .....	266

A.11	Thermodynamik .....	274
<b>B</b>	<b>Viable System Model</b> .....	277
B.1	Viable System Service .....	286
B.2	VSM-Design .....	289
B.3	Kontrollerdesign .....	293
B.4	Adaption .....	295
B.5	CODA .....	296
<b>C</b>	<b>Metriken</b> .....	299
C.1	Messbarkeit .....	300
C.2	Rating .....	301
C.3	Komplexitätsmaße .....	301
C.4	Koppelungsmaße .....	305
<b>D</b>	<b><math>\pi</math>-Kalkül</b> .....	307
D.1	Definition .....	307
D.2	Kongruenz .....	308
D.3	Abstraktion .....	309
D.4	Reaktion .....	309
D.5	Replikation .....	310
D.6	Transaktionen .....	311
D.7	Stochastisches $\pi$ -Kalkül .....	312
	<b>Literaturverzeichnis</b> .....	313
	<b>Sachverzeichnis</b> .....	317



# Kapitel 1

## Einleitung

*Kühner als das Unbekannte zu erforschen  
kann es sein, Bekanntes zu bezweifeln.*

Alexander von Humboldt  
1769–1859

Etwas ist verkehrt mit dem heutigen Design von Systemen. Komplexität überwältigt uns und in bestimmten Bereichen hat sie uns schon überwältigt – Software ist heute so komplex geworden, dass faktisch keine verlässlichen und fehlerfreien Softwaresysteme mehr existieren. Die heutige Art und Weise der Produktentwicklung in der Softwareindustrie – kumulativ mit immer mehr Features – führt zu großen Mengen von aufgeblähtem, nicht mehr handhabbarem Code. Die Menge an Programmcode ist mittlerweile so groß geworden, dass selbst riesige Teams von gut ausgebildeten Softwareentwicklern neben der Komplexität allein das angehäuften Volumen an Programmcode nicht mehr beherrschen. Ein interessantes Beispiel für wachsende Komplexität im Softwaresektor ist, obwohl es sich hier nicht um die Komplexitätsentwicklung eines einzelnen Produktes handelt, die Größe des Betriebssystems Windows (s. Tabelle 1.1). Der Faktor an Codezeilen zwischen DOS und Windows XP liegt bei über 10 000. Zwischen Windows XP und NT 3.1 beträgt er immerhin noch 7,5! Heutige Software kann gar nicht mehr komplett und vollständig ausgeliefert werden, sie wird in Versionen, Patches und Updates mit hoher Frequenz verteilt und erhält dadurch immer mehr Fehler, welche wiederum durch nachfolgende Patches und Updates beseitigt werden sollen. . . Eine neue Art Software zu entwickeln und einzusetzen muss geschaffen werden – und wie jede neue Kunst oder Disziplin, muss diese Art der Softwareentwicklung auch radikal und einfach sein!

Im Allgemeinen sind Menschen nicht besonders gut im Entwurf komplexer Systeme.<sup>1</sup> Selbst einfache idealkinetische Systeme wie Snooker oder Poolbillard kann der Mensch nicht beherrschen, umso weniger Systeme mit komplexen Wechselwirkungen und diversen Strukturen. Eine der möglichen Lösungen für das komplexe Design ist es, Verfahren ähnlich der biologischen Evolution zu nutzen und so Systeme zu implementieren, die sich letztlich selbst designen und steuern. Zwar hat es schon immer komplexe Systeme gegeben<sup>2</sup>, aber die zunehmende Technisierung unserer Lebenswelt führt zu einer immer stärkeren Abhängigkeit der Teile dieser Welt untereinander und damit zu einer immer größeren Komplexität.

<sup>1</sup> Einer der Gründe hierfür ist, dass wir nicht mehr als 7 Chunks an Informationen gleichzeitig wahrnehmen können.

<sup>2</sup> Für eine Definition komplexer Systeme, s. Anhang A.

Neben den „individuellen“ Erlebnissen von Komplexitäten werden auch volkswirtschaftliche Strömungen die Entwicklung der nächsten Jahre stark beeinflussen. Nach der Theorie der Kondratieffzyklen<sup>3</sup> können Volkswirtschaften durch technische Basisinnovationen stark beeinflusst werden. Diese Kondratieffzyklen haben eine Dauer von etwa 40–60 Jahren und ermöglichen jeweils einen großen volkswirtschaftlichen Aufschwung. Die Auslöser dieser Zyklen waren in der Vergangenheit zum Beispiel die Dampfmaschine oder das Automobil. Die Basisinnovation der IT setzt um das Jahr 1970 ein und wird als der fünfte Kondratieffzyklus bezeichnet (s. Abb. 1.1).<sup>4</sup> Angefangen von den Großrechnern und einem niedrigen Durchdringungsgrad von Organisationen mit IT von etwa 20%, erreicht der Zyklus heute das Internetzeitalter mit einem Grad der Durchdringung von, je nach Industriesektor, 75–98%. Der Zyklus wird vermutlich zwischen 2010–2030 auslaufen und durch eine noch unbekannte Folgeinnovation<sup>5</sup> ersetzt werden. Zu diesem Zeitpunkt sollte die IT analog den anderen Basisinnovationen eine Commodity, wie zum Beispiel das heutige Telefon, geworden sein. Insofern erleben wir das Ende eines Kondratieffzyklusses direkt mit. Der sehr hohe Durchdringungsgrad unserer Lebens- und Arbeitswelt mit IT ist eines der Anzeichen für das Ende des fünften Zyklusses.

**Tabelle 1.1** Das Wachstum von Windows

„Betriebssystem“	Lines of Code
PC DOS <sup>6</sup> 1.0	4 000
NT 3.1	6 000 000
Windows 98	18 000 000
Windows 2000	35 000 000
Windows XP	45 000 000

Die verschiedenen Entwicklungsstränge der IT sind in Abb. 1.1 dargestellt. Die letzte Welle wird nicht den Computer, sondern die transportierten Inhalte in Form von Video, Musik, Sprache und Daten in den Vordergrund stellen und damit letztlich auch von dem Computer an sich abstrahieren.

Diese Form der Entwicklung wird von einigen Autoren auch als „content centric“ bezeichnet und ist einer der Schwerpunkte dieses Buches. Da sich dieses Buch auf die IT-Entwicklung und nicht auf die Entwicklung der Inhalte fokussiert, wird für die Welle der Begriff „Pervasive Computing“<sup>7</sup> genutzt. Aus dem Blickwinkel der Inhalte betrachtet kommt es zu einer immer stärkeren Konvergenz der Geräte. Techniken wie VoIP<sup>8</sup> ermöglichen eine gemeinsame Übertragung von Sprache und

<sup>3</sup> *Nikolai Dmitriyevich Kondratieff*, 1892–1938, russischer Volkswirtschaftler entdeckte, dass Wirtschaften in Zyklen ablaufen.

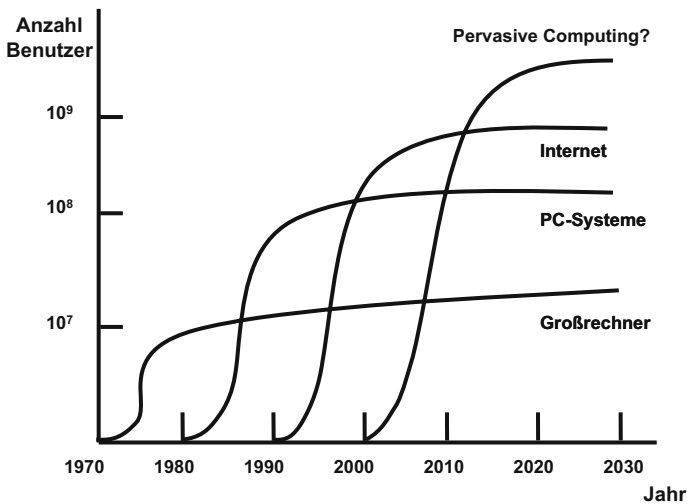
<sup>4</sup> Die Computer sind zwar älter (Charles Babbage, 1837 und Konrad Zuse, 1941), aber ein volkswirtschaftlich messbarer Anteil beginnt erst um 1970.

<sup>5</sup> Umwelttechnologie ist ein möglicher Kandidat, ein anderer die Wasserstofftechnologie.

<sup>6</sup> Im Fall von PC DOS handelt es sich um Assemblerzeilen.

<sup>7</sup> Siehe Kap. 4.

<sup>8</sup> **Voice over IP.**



**Abb. 1.1** Die verschiedenen „Wellen“ der IT als Teil des fünften Kondratieffzyklusses. Die letzte Welle ist zurzeit noch unklar, aber die Internetwelle wird ihre Sättigung deutlich vor dem Ende des Kondratieffzyklusses (circa 2020) erreicht haben

Daten, PDAs<sup>9</sup> sind eine Kombination aus Mobiltelefon, Computer und Internetbrowser. In Zukunft werden für die verschiedenen Kommunikationsformen keine dedizierten Geräte mehr existieren und diese technische Konvergenz auf Geräte- wie auch Übertragungsebenen wird die Art und Weise unseres Umgangs mit den Inhalten grundlegend verändern. Auch die in den letzten Jahren stetig zunehmende Rechenkapazität (s. Abb. 1.4) hat zur Beschleunigung dieses allgemeinen Trends beigetragen.

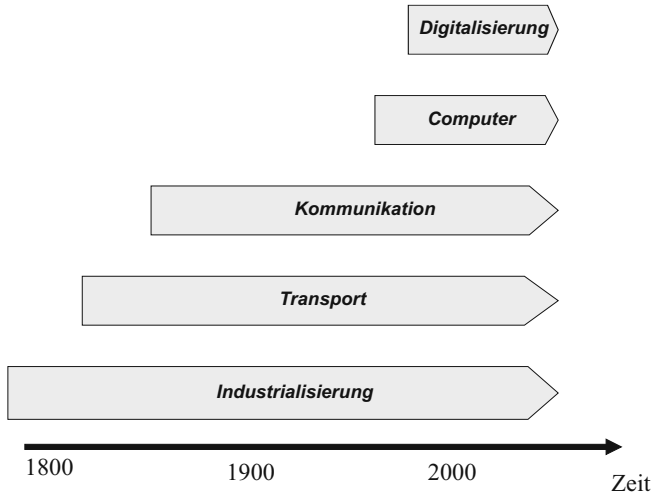
Aber nicht nur im Privat-, sondern auch im Wirtschaftsleben wird es zu Veränderungen kommen müssen, denn obwohl das Rückgrat der europäischen Wirtschaft durch kleine und mittelständische Unternehmen<sup>10</sup> gebildet wird, sind es gerade diese Unternehmen, welche im Normalfall erst sehr spät IT für ihre Zwecke einsetzen. Neue Geschäftsmodelle, B2B<sup>11</sup>-Modelle und ähnliche Marktzugänge werden primär von großen Unternehmen eingesetzt, um mit anderen großen Unternehmen zu interagieren. Die in diesem Buch vorgestellten digitalen Ökosysteme<sup>12</sup> sind ein möglicher Weg, den kleinen und mittelständischen Unternehmen den Zugang zu dem lukrativen B2B-Sektor als auch einen weltweiten Zugang zu einem elektronischen Marktplatz zu bieten. Elektronische Marktplätze sind für Käufer wie

<sup>9</sup> Personal Data Assistant, so zum Beispiel Blackberry.

<sup>10</sup> Über 99% aller Unternehmen fallen in diese Kategorie mit weniger als 250 Arbeitnehmern oder weniger als 5 Millionen € Umsatz. Diese Unternehmen sind für etwa die Hälfte des europäischen Bruttosozialprodukts verantwortlich.

<sup>11</sup> Business-to(2)-Business.

<sup>12</sup> Der Ausdruck Ökosystem wurde 1866 von *Ernst Haeckel* geprägt. Die Ökologie (aus dem Griechischen *οικος* (das Haus) und *λογος* (die Lehre)) ist als die „gesamte Wissenschaft von den Beziehungen des Organismus zur umgebenden Außenwelt“ definiert.



**Abb. 1.2** Die Entwicklung der heute genutzten Technologien

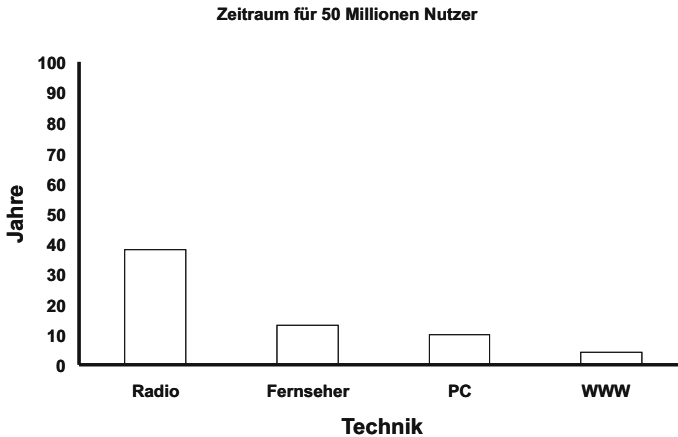
auch Verkäufer auf Grund ihrer Reichweite und auch Geschwindigkeit sehr interessant.<sup>13</sup> Der Namenspatron, das biologische Ökosystem, besteht aus einer Anzahl von Lebewesen in einer mehr oder minder definierten Umwelt; bei den digitalen Ökosystemen hingegen handelt es sich um Komponenten, Services, Applikationen und Geschäftsmodelle, welche hier die Lebewesen darstellen. Diese Teile existieren meist auf einer Art erweiterten Infrastruktur, welche das Analogon zur Umwelt der biologischen Ökosysteme darstellt und oft Umgebung genannt wird. Im Fall der digitalen Ökosysteme hat jede Umgebung zwei Bestandteile, zum einen den sozioökonomischen Kontext und zum anderen eine digitale Infrastruktur, die dann den Lebensraum für die „digitalen“ Lebewesen bildet.

Speziell die Wichtigkeit und Prägung unserer Welt durch die IT ist stark merklich. War man in den sechziger und siebziger Jahren noch der Ansicht, dass wir in einem Atomzeitalter<sup>14</sup> leben, so hat sich mittlerweile gezeigt, dass Software und Internet die Schlüsseltechnologien des neuen Jahrhunderts sind, mit noch rasanter anwachsender Verbreitung. Zwei Grundtrends sind durch die IT für den Menschen der Industrienationen in seiner allgemeinen Lebenswelt zu erkennen:

- IT wird einen immer stärkeren Einzug in das Alltagsleben haben. Mittlerweile ist mit Software nicht nur der PC oder Mainframecomputer durchsetzt, sondern alle möglichen Geräte. Immer mehr Alltagsgegenstände, Waschmaschinen, Kühlschränke, Heizungen, Kaffeemaschinen sind in Wirklichkeit schon zu Computern geworden; obwohl wir sie als solche gar nicht wahrnehmen.
- Begann die Kommunikation in der IT als eine Mensch-Mensch-Kommunikation im Sinne von Telefon oder auch E-Mail, so ist heute eine verstärkte Mensch-

<sup>13</sup> Siehe den Erfolg von *eBay*.

<sup>14</sup> Letztlich sind stromerzeugende Atomkraftwerke nichts anderes als gigantische Wasserboiler.



**Abb. 1.3** Der Zeitbedarf, um 50 Millionen Benutzer zu erhalten, sinkt drastisch

Maschine-Kommunikation – zum Beispiel: Suchmaschinen, *Amazon.de* und so weiter – zu beobachten.

- Die Zukunft wird verstärkt durch eine Maschine-Maschine-Kommunikation geprägt werden. Erste Formen hiervon existieren im elektronischen Börsenhandel XETRA<sup>15</sup>, wo Computer automatisch mit der Börse interagieren und versuchen vorgegebene Ziele<sup>16</sup> einzuhalten. Da Computer dies sehr viel schneller können als Menschen, steigt die Gefahr eines Börsencrashes an.

Die angesprochene Technikdurchdringung wird sich in den nächsten Jahren zusätzlich beschleunigen, mit der Folge, dass die einzelnen Teile sich zu immer größeren Systemen vernetzen und dadurch neue, bisher unbekannte Eigenschaften entwickeln. Das, was sich auf der Ebene des einzelnen Geräts abspielt, spiegelt sich auch auf der Ebene des Geschäftslebens wider; auch hier entstehen völlig neue Formen der Zusammenarbeit mit eigener Semantik und eigenen Modellen, sogar mit eigenständigen und vor allen Dingen völlig neuen Geschäftsideen.

Man mag sich darüber beschweren, dass unsere Welt immer technisierter wird und sich auf eine sogenannte „gute alte Zeit“ zurückbesinnen<sup>17</sup>, aber diese hat es eigentlich nicht wirklich gegeben. Unsere gesamte Umwelt, nicht nur in Städten, sondern auch in ländlichen Gebieten, ist in Europa die Folge einer jahrtausendlangen Kultivierung und das, was vom Ursprünglichen<sup>18</sup> noch übrig geblieben ist – das

<sup>15</sup> Exchange Electronic Trading.

<sup>16</sup> Bestimmte Aktien zu verkaufen, wenn sie unter einen Wert fallen.

<sup>17</sup> So versuchten Künstlerkolonien, wie zum Beispiel *Barbizon*, in der zweiten Hälfte des 19. Jahrhunderts die Industrialisierung zu negieren, indem sie das Bild einer präindustriellen Welt malten.

<sup>18</sup> In den letzten Jahrzehnten wurden Anstrengungen unternommen die Lüneburger Heide zu erhalten, da sie in Gefahr steht zuzuwuchern. Nur, diese Heide ist das Resultat einer jahrhundert alten Überweidung und sah nicht immer so aus...

Unberührte – wird in Schutzgebiete eingesperrt wie in einen Zoo.<sup>19</sup> Die Welt, in der wir leben, ist eine vom Menschen gemachte, eine künstliche Welt; de facto leben wir in einem Technotop. Insofern ist eine zunehmende Durchdringung dieses Technotops mit immer mehr Computern und Software nur die Konsequenz einer Logik, welche schon vor Jahrhunderten einsetzte und dieses Technotop steht nicht still, es verändert sich permanent.

Diese Veränderungen sowie die Konvergenzen der Technologien werden jedoch auch von Schwierigkeiten begleitet. Folgende Punkte stellen maßgebliche Hindernisse in diesem Prozess dar:

- Akzeptanz – Die meisten neuen Technologien werden primär von sehr technikaffinen Personen eingesetzt. Im Durchschnitt der Bevölkerung nimmt der Einsatz neuer Techniken mit zunehmendem Alter ab.<sup>20</sup> Die Konzentration auf die technischen Aspekte reduziert die Bereitschaft von nicht-technikaffinen Personen, neue Geräte oder Kommunikationsformen einzusetzen. Erst wenn die Gebrauchsfähigkeit, beziehungsweise Einfachheit der Nutzung in den Vordergrund tritt, kann diese Hemmschwelle beseitigt werden.<sup>21</sup>
- Medienbrüche – Besonders in der Wertschöpfungskette innerhalb einer Organisation oder auch organisationsübergreifend reduzieren eventuelle Medienbrüche die Effizienz drastisch, mit der Folge von erhöhten Kosten für die Erstellung von Produkten oder Services. Daher ist besonders die Vermeidung digitaler Brüche innerhalb von Wertschöpfungsketten eines der Anliegen von effizienten Organisationen. Die Medienbrüche gehen oft einher mit Brüchen in der Semantik. Sie sind zunächst einmal sichtbar bei der Koppelung unterschiedlicher Syntaxen, da hier an der Bruchstelle syntaktisch „übersetzt“ werden muss, was natürlich das Risiko einer semantischen Fehlinterpretation erhöht.
- Rahmenbedingungen – Neue Technologien verlangen für ihren Einsatz stets staatliche und markttechnische Rahmenbedingungen.<sup>22</sup>
- Standardisierung – Ein großes Hindernis stellt eine fehlende marktweite Standardisierung dar. Mit der Existenz keines einzigen Standards oder, noch schlimmer, mehrerer inkompatibler Standards, scheuen viele Benutzer und Organisationen vor entsprechenden Investitionsleistungen zurück. Umgekehrt versuchen Hersteller geradezu neue Standards zu schaffen und diese als erste exklusiv zu vermarkten, da ihnen dies einen immensen Vorsprung vor der Konkurrenz liefert.

Parallel dazu leiden wir schon heute unter einer drastisch zunehmenden Informationsmenge. Schon in den siebziger Jahren wurden die Folgen dieser Informationszunahme vorhergesagt:

<sup>19</sup> Meist noch versehen mit einem Schild, welches uns sagt, was wir erleben sollen.

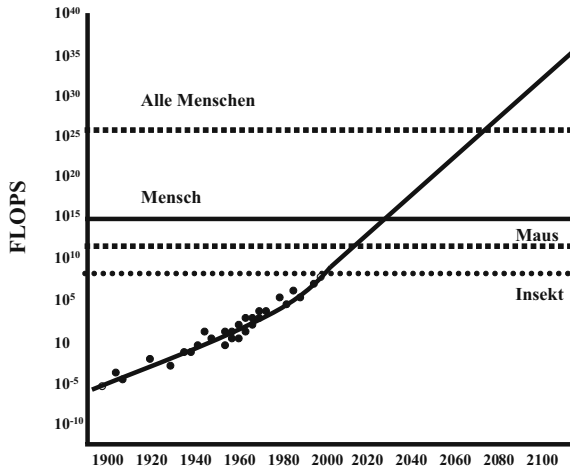
<sup>20</sup> Für das Marketing neuer Produkte und Services ist die Altersgruppe der unter 25-Jährigen besonders interessant, da hier die größte Veränderungsbereitschaft herrscht.

<sup>21</sup> Das rasante Wachstum der Mobiltelefonie und die starke SMS-Nutzung sind Beispiele in dieser Richtung. Die meisten SMS-Nutzer sind in der Altersgruppe unter 40.

<sup>22</sup> Voice over IP macht wenig Sinn, wenn die Internetverbindung zu langsam ist.

*What information consumes is rather obvious: it consumes attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it.*<sup>23</sup>

Daher muss die zunehmende Informationsmenge durch immer weniger Aufmerksamkeit bearbeitet werden, mit der Folge, dass Information zunehmend automatisiert verarbeitet wird oder der einzelne Mensch in Informationen ertrinkt<sup>24, 25</sup>.



**Abb. 1.4** Die Entwicklung der Rechenkapazität in FLOPS (FLoating Point Operations Per Second) im Vergleich zu biologischen Systemen

Eine der Herausforderungen im modernen Einsatz von Geräten und Software ist es, Systeme zu entwickeln, welche komplexe dynamische Probleme in einer skalierbaren und effizienten Art und Weise lösen. Eine Lösungsmöglichkeit hierfür ist der Aufbau digitaler Ökosysteme, welche das Verhalten biologisch komplexer Systeme – und die entsprechenden Metaphern – imitieren, um somit ein dynamisch adaptives Gesamtsystem aufzubauen. Auf Grund der Tatsache, dass die Softwareentwicklung heute auf eine Komplexitätsbarriere trifft, wurden unterschiedlichste Methoden entwickelt, um diese Barriere methodisch zu überwinden, so zum Beispiel MDA<sup>26</sup>, SOA<sup>27</sup> oder auch eXtreme Programming und viele andere mehr. Allerdings fehlt noch immer der Nachweis, dass diese Verfahren bei sehr großen Systemen wirksam sind. Insofern ist eine Modellierung durch Services in gewisser Weise der Versuch, die Komplexitätsbarriere durch Abstraktion zu überwinden.

<sup>23</sup> Simon, H.: 1971, Designing Organizations for an Information-rich World, The John Hopkins Press.

<sup>24</sup> So zum Beispiel durch Spam-Mails. Der Ausdruck Spam geht auf einen *Monty Python* Sketch zurück.

<sup>25</sup> Nicht immer handelt es sich hierbei um Spam-Mails. Der Autor erhält zurzeit etwa 1200 E-Mails pro Monat, von denen aber nur etwa 30% unter den Begriff Spam fallen.

<sup>26</sup> Model Driven Architecture.

<sup>27</sup> Service Oriented Architecture, s. Kap. 2.

**Tabelle 1.2** Evolution der Architekturen

Gebiet	1960–70	1980–90	1990+	1995+	2000+	2010+
<b>Markt- imperativ</b>	Marktanteile	Effektivität	Dezentra- lisierung	Kunden- bindung	Real Time Enterprise	Service Oriented Enterprise
<b>Architektur</b>	Mainframe	Module	Client- Server	Applika- tionsserver	SOA <sup>28</sup>	EOA <sup>29</sup> /SOA
<b>Ziel- vorstellung</b>	Skalen- ökonomie	Business Process Reengin- eering	Business Applika- tionen	Kunden- bindung	Entflechtung	Service- ökonomie
<b>Treiber</b>	Status Quo, keine Skalierung	Sinkende CPU- Kosten	PC und Netzwerke	WWW	Webservices	Semantische Services

Eine mögliche Form der Abstraktion von Hard- und Software ist es, Services zu formulieren. So einfach dieses Postulat erscheint, so mächtig ist seine Anwendung. Was ist ein Service? Ein Service ist eine Dienstleistung, welche einem Kunden (genannt Consumer) vom Leistungserbringer (genannt Provider) zur Verfügung gestellt wird. Wie jede Form von Dienstleistung haben Services als Charakteristika ein hohes Maß an Kundenbeteiligung in ihrer Definition und Weiterentwicklung sowie die Schwierigkeit der Standardisierung. Die Schwierigkeit hinter der Standardisierung liegt in dem Wunsch der Kunden begründet, ein hohes Maß an Individualisierung in den Services zu haben. Dieses hohe Maß an Individualisierung schafft umgekehrt Probleme für die Standardisierung, dem genauen Gegenteil einer Individualisierung. Der Provider hingegen möchte ein möglichst hohes Maß an Standardisierung erreichen, da dies seine Kosten senken kann.

Einer der Gründe für die öffentliche Aufmerksamkeit für Serviceorientierung und speziell für SOA ist die Tatsache, dass die Serviceorientierung eine hervorragende Metapher für nicht-technisch-orientierte Menschen ist. Diese können nun auch den Wert einer Architektur verstehen und den Herausforderungen der Veränderung und Anpassung in Organisationen und Technik begegnen. Der wirkliche Mehrwert einer SOA ist, dass es die erste Architektur ist, die Software und Organisation transzendiert. Langfristig gesehen muss die Serviceorientierung sowohl Auswirkungen auf die Betriebssysteme, als auch auf die zur Verfügung stehende Hardware haben. Heutige Ansätze eines Hardware Abstraction Layers<sup>30</sup> bauen auf der Idee einer generischen Hardware auf; es ist zurzeit noch unklar, wie eine serviceorientierte Hardware oder ein serviceorientiertes Betriebssystem konzipiert sein könnten, aber beides wird kommen. Die gleiche Idee der Hardwareabstraktionsschicht findet sich heute in Haushaltsgeräten wieder, insofern ähneln diese immer mehr Computern und sind auch mit denselben Vorzügen „gesegnet“, beziehungsweise Problemen

<sup>28</sup> Service Oriented Architecture, s. Abschn. 2.5.

<sup>29</sup> Ecosystem Oriented Architecture, s. Abschn. 5.3.

<sup>30</sup> So zum Beispiel in *Windows* und *Linux*.



geplant. Die Geschäftswelt hat auf ihrem Weg zur Dienstleistungsgesellschaft<sup>31</sup> ein gewisses Maß an Erfahrung über Services, deren Nutzung, Verwendung und Einsatz aufgebaut. Die IT-Welt und hierbei speziell die Softwarestruktur in Form von Services, steht jedoch noch am Anfang einer solchen Entwicklung. Die Services in einer Software müssen dem Anwender eine wohldefinierte Funktionalität in einem veränderbaren Kontext zu verifizierbaren Qualitätskriterien und ab initio festgelegten Preisen bieten können. Eine interessante Eigenschaft von Services aus der Geschäftswelt wird sehr oft bei der Einführung von Services in der Software übersehen: Erfolgreiche Services werden stets aus der Sicht des Consumers definiert und nicht aus Sicht des Providers! Im Gegensatz dazu entstehen die meisten heutigen Services in der Software aus Sicht des Providers, der sein bestehendes System in Services zerlegt und diese anbietet. Diese providerzentrische Form der Serviceentwicklung führt zu großen Hindernissen bei der Nutzung und Akzeptanz.

Technologien und Softwareparadigmen schlagen sich stets auch in der Architektur nieder (s. Tabelle 1.2 und Tabelle 1.3). Die aus dem Serviceorientierungsparadigma entstehende Architektur wird als **Service Oriented Architecture (SOA)** bezeichnet. Eine solche SOA ist die Folge und simultan auch die notwendige Voraussetzung für die Zerlegung bestehender und den Aufbau neuer Applikationen aus Services.

**Tabelle 1.3** Paradigmenwechsel

<b>Zeitraum</b>	<b>Revolution</b>	<b>Computing Paradigma</b>	<b>Architektur</b>
1970–80	Mainframe	Monolithisch	Single Tier
1980–90	Midrange	Abteilungsorientiert	Single Tier
1990–95	Client/Server	Power für den Desktop	2 Tier
1995–2000	Web	Portale und Backendsysteme	3 Tier
2000+	SOA	Servicebasiert	Servicebasiert
2010+	EOA	Digitales Ökosystem	Servicebasiert

Nicht nur bei der Software fand ein Paradigmenwechsel statt, sondern auch in der Hardware. Nach den ersten Anfängen in der Analogtechnik mit Röhrengeräten setzten sich in den siebziger Jahren des letzten Jahrhunderts digitale Systeme durch. Diese sind heute die dominante Form von Geräten. Stand Anfangs noch der einzelne Chip im Vordergrund, so verschwindet dieser heute immer stärker und das Gerät mit seinen direkt nutzbaren Fähigkeiten rückt in das Bewusstsein des Benutzers.<sup>32</sup> Diese neuartigen Systeme führen dazu, dass wir die Kommunikations- und Computingparadigmen, die wir heute nutzen, überdenken müssen. Die traditionellen Paradigmen der Softwareentwicklung wurden für eine Umgebung eingeführt, welche heute obsolet ist, nämlich primär für kabelgebundene, sehr homogene Netzwerke mit Großcomputern und leistungsfähigen PCs. In solchen Umgebungen sind Verbindungsabbrüche und Ausfall von Komponenten die großen Ausnahmen und

<sup>31</sup> Aus soziologischer Sicht die Phase nach der Industrialisierung.

<sup>32</sup> Der PC verkauft sich allerdings auch heute noch mit Angaben über CPU-Typ, -Frequenz...

nicht die Regel und es wird angenommen, dass Netzwerk- oder Systemmanager die Umgebung kontinuierlich rekonfigurieren, so dass ein Optimum an Leistung oder Durchsatz erzielt werden kann. Aber heutige Netzwerke sind sehr viel heterogener, bestehen aus vielen „fehleranfälligen“ Teilen und interagieren mit einer Vielzahl von unterschiedlichen drahtlosen Kanälen. Hier sind Unterbrechungen und Ausfälle die Regel und nicht mehr die Ausnahme; außerdem können die Netzwerk- oder Systemmanager auf Grund der Komplexität des Gesamtsystems nur noch sehr bedingt eingreifen.

Aber nicht nur die Komplexität der Systeme hat in den letzten Jahren drastisch zugenommen, parallel dazu ist auch die Größe der Systeme sehr stark angewachsen. Dieses rapide Größenwachstum hat zwei Ursachen: Zum einen eine stetig zunehmende Vernetzung – primär getrieben durch das Internet und seine Möglichkeiten – und zum anderen die stark erhöhte Durchdringung aller Lebensbereiche der Industrienationen mit Services, beziehungsweise Geräten, die diese Services ermöglichen. Auch in den Entwicklungsländern zeigen sich analoge Tendenzen, hier allerdings mit dem Unterschied, dass die Mobiltechnik starke Impulse setzt, da ihre Nutzung oft keine hohen Anforderungen an die Infrastruktur in Form von Kabeln oder Stromversorgung stellt, die in solchen Ländern oft nur gering entwickelt ist.

Für den Menschen, der in solchen Systemen lebt, stellt sich die Frage, inwieweit er noch in der Lage ist, das System, in dem er lebt, zu steuern, da sich die so entstehenden Systeme oft völlig nicht-intuitiv verhalten. Außerdem lassen sie sich nicht auf die technologischen Aspekte allein reduzieren, sondern es bilden sich riesige soziotechnische Systeme heraus, in denen sich die Menschen und die Technik gegenseitig bedingen. Technologie im Allgemeinen und IT im Besonderen sind untrennbar verknüpft mit sozialen Aspekten und den Praktiken, wie sie entstehen und genutzt werden. Dies kann sogar so weit gehen, dass ein solches System sich selbstständig verändert und in gewisser Weise selbstständig erschafft (s. S. 263).

Eine Möglichkeit mit solchen Systemen umzugehen ist es, überhaupt nicht zu versuchen, die Systeme direkt zu kontrollieren, sondern sie durch die Veränderung der Umgebung indirekt zu beeinflussen. Neben dem Versuch, die Rahmenbedingungen des Systems zu verändern, ist eine andere – und oft viel bessere – Methode, das entstehende System als solches zu akzeptieren und zu versuchen, seine emergenten Eigenschaften (s. S. 253) zu fördern. Wir werden insgesamt gesehen solchen sehr großen Systemen in Zukunft immer stärker ausgesetzt sein, zum Teil sind wir, in Form des Internets oder sehr großer Konzernnetze, einem solchen Ultra Large Scale System (s. Abschn. 6.9) schon heute ausgesetzt.

Die Entwicklung auf der Ebene des einzelnen Benutzers (Consumer) wird sich durch den zunehmenden Einsatz von autonomen Services auf den einzelnen physischen Geräten komplettieren. Hier haben die Hersteller das Problem, dass diese Geräte sich auf diverse – nicht a priori definierbare – Umgebungen und Services einstellen können müssen. Eine solche Randbedingung verlangt von den entstehenden Services ein sehr hohes Maß an Anpassbarkeit, was wiederum die Komplexität der einzelnen Serviceimplementierung erhöht und damit massiv zur Steigerung der Gesamtkomplexität beiträgt.

# Kapitel 2

## Grundlagen der Serviceorientierung

*I have travelled the length  
and breadth of this country  
and talked with the best people  
and I can assure you  
that data processing is a fad  
that won't last out the year.*

Editor Business-Books  
Prentice-Hall  
1957

Warum überhaupt Services und die Serviceorientierung? Heute bestehende, aber vor allen Dingen sämtliche zukünftige Systeme dürften von diversen Lieferanten zusammengestellt werden, aus Teilen, welche alle miteinander interagieren und so ein neues großes System aus Software erzeugen.<sup>1</sup> In einem solchen Umfeld ist es leicht, sich im Detail des einzelnen Geräts, Übertragungsprotokolls oder der Schnittstellendefinition zu verlieren.<sup>2</sup> Die Abstraktion von Geräten und Softwarebestandteilen in diesen Geräten in Form von Services ermöglicht es uns, solche Systeme auf Elementebene zu beschreiben, beziehungsweise die Beschreibung erst möglich zu machen. Daher eignen sich Services in einem solchen Gebiet als Beschreibungs- und Modellierungssprache. Die aus der Abstraktion auf Services entstehenden Vergrößerungen, als auch die Tatsache, dass die Elemente des beschreibenden Systems keine „echten“ Services sind, erzeugen zwar ein Maß an Unschärfe, können aber im Sinne des Verständnisses für das Gesamtsystem hingenommen werden.

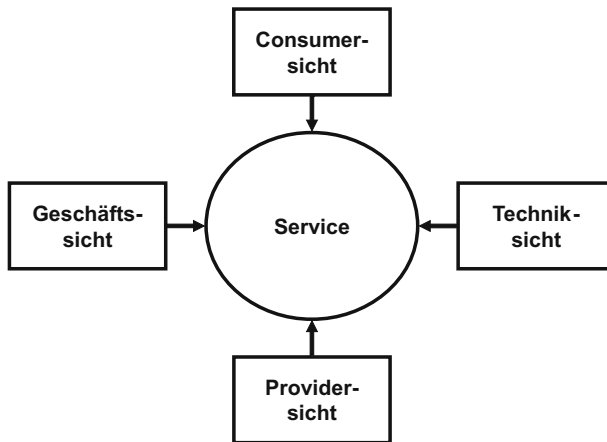
### 2.1 Serviceorientierungsparadigma

Den Begriff des Service zu definieren ist nicht einfach, obwohl er zunächst intuitiv klar erscheint. Etymologisch betrachtet geht das Wort Service auf das Lateinische *servitium* mit der Bedeutung von Sklaverei und Knechtschaft zurück. Ein Service ist nicht deckungsgleich mit einer Komponente, wie man sie aus der komponentenbasierten Softwareentwicklung her kennt, wenngleich viele Konzepte auch auf die serviceorientierte Softwareentwicklung übertragen werden können. Eine andere

---

<sup>1</sup> Die heutigen Softwarehersteller erwecken zwar gerne den Eindruck, dass die Software, die sie liefern, vollständig von ihnen selbst stammt, aber meist ist dies nicht der Fall. Aus Marketinggründen scheint das Alles-Aus-Einer-Hand-Argument jedoch erfolgreich zu sein.

<sup>2</sup> Unsichere Menschen flüchten sich gerne in Details, da ihnen das subjektive Sicherheit bietet. Außerdem können sie hier ihr „Fachwissen“ beweisen. . .



**Abb. 2.1** Die unterschiedlichen Sichten auf einen Service

Vorstellung ist, dass ein Service von einer Komponente implementiert wird und der Service damit nur das Interface der Komponente darstellt. Diese Betrachtungsweise ist etwas zu eng. Ein Service beinhaltet zwar auch ein Interface, zusätzlich dazu existiert aber ein Vertrag<sup>3</sup>, der eine Beschreibung von Eigenschaften enthält, welche die Semantik betreffen und die nicht über ein Interface definiert werden können. Dazu gehören Qualitätseigenschaften wie Verfügbarkeit oder Performance, die auch im Sinne eines juristischen Vertrages vereinbart werden können. Dafür reicht dann die Beschreibung eines Interface nicht mehr aus, sondern es bedarf eines komplexen **Service Level Agreements (SLA)**, in dem Rechte und Pflichten von Provider und Consumer klar geregelt sind.

Die Nutzung unterschiedlicher Perspektiven ist wichtig für das Verständnis eines Service. Hier gibt es zwei wesentliche Sichtweisen:

- **Geschäftssicht** – Innerhalb der Geschäftssicht wird ein Service als Teil einer Geschäftstransaktion betrachtet, die in einem Vertrag beschrieben und die durch die Geschäftsinfrastruktur durchgeführt wird. Was ein Service leistet, ist eng verknüpft mit den Erfahrungen des Geschäftsbereichs. Einen Service auf dieser Seite bezeichnet man mit dem Begriff **Geschäftsservice** (genau wie eine Dienstleistung). Typische Eigenschaften eines solchen Geschäftsservice sind:
  - **Geschäftsvisibilität** – Ein Service muss etwas sein, wofür Consumer (Kunden)<sup>4</sup> bereit sind, etwas zu bezahlen. Die Consumer ihrerseits erhalten etwas, das für sie von Wert oder von Nutzen ist. In diesem Zusammenhang muss der Begriff „Kunde“ weiter gefasst werden als im Bereich von Produkten. Als

<sup>3</sup> Dieser wird meist bi- oder unilateral geschlossen.

<sup>4</sup> In diesem Buch wird der Begriff Consumer für jeden beliebigen Abnehmer eines Service genutzt, dabei kann es sich um einen Menschen, ein anderes System, einen anderen Service, einen anderen Computer, denselben Service und so weiter handeln. Der Nutzer eines Interface wird stets als Consumer bezeichnet.

Consumer werden nicht nur externe, sondern auch interne Nutzer des Service bezeichnet.

- Technologie – Hierbei steht das, was implementiert werden soll im Fokus, nicht wie etwas implementiert werden soll.
  - Kontext – Die Services werden neben anderen Eigenschaften auch durch ihren jeweiligen fachlichen Kontext definiert.
- Techniksicht – Ein Service stellt eine mehr oder minder geschlossene Funktionalität bereit, wobei die Semantik eines Service in Form eines Interface beschrieben wird. Eigenschaften technischer Services sind:
    - Technologie – Bei technischen Services steht die Technologie, also wie etwas implementiert werden soll, im Vordergrund.
    - Kontext – Ein Service ist eine vom Kontext gekapselte und abstrahierte Funktionalität.

Es existieren einige Unterschiede zwischen den Sichten und damit auch zwischen den daraus jeweils abgeleiteten unterschiedlichen Servicedefinitionen. Der wesentliche Unterschied beider Sichten liegt in der Betrachtung des Kontexts. Für Geschäftsservices ist der Kontext von großer Wichtigkeit; alle Organisationen arbeiten schließlich in einem dynamischen Marktumfeld. Werden Marktchancen ergriffen, so wird von dieser Seite erwartet, dass neue Services erstellt oder bestehende Services verändert werden. Die technische Seite ist bestrebt, den Kontext möglichst statisch zu halten, da nur so die Forderung nach Effizienz und Wiederverwendbarkeit erreicht werden kann. Diese Differenzen lassen sich nicht wirklich überbrücken. Die negativen Effekte lassen sich aber abmildern, indem kommuniziert wird, welche Bedeutung ein Service etwa auf organisatorischer Ebene hat oder wie wichtig Wiederverwendung für einen implementierten Service ist.

Zusätzlich zu den beiden Geschäfts- und Technikperspektiven existiert eine Consumer- und eine Providerperspektive. Die Implementierung ist Teil der Providersicht und muss für den Consumer von geringem Interesse sein. Die Providersicht überschneidet sich mit der Geschäftssicht und der technischen Sicht. Die Consumersicht ist mehr auf die Geschäftssicht fokussiert. Wenn ein Consumer einen Service in Anspruch nimmt, muss für ihn der Wert des angebotenen Service schon vorab erkennbar sein. Ist dies der Fall, stellt der Service eine Schnittstelle zwischen Geschäftswert und Implementierung dar. In einer sehr technisierten Umgebung verschwindet oft die Fähigkeit einen Service zu wählen, da solche Umgebungen auf die Anwesenheit oder Nichtanwesenheit von Services automatisch reagieren und dies für den Menschen nicht erkennbar ist.

Die Verwendung von Services in Geräten, Organisationen und in der Software folgen einem gemeinsamen Grundsatz, dem „Paradigma<sup>5</sup> der Serviceorientierung“:<sup>6</sup>

<sup>5</sup> Paradigma aus dem Griechischen *παράδειγμα* aus *παρα* (neben) und *δειγναι* (zeigen). Spötter behaupten ein Paradigma sei die Summe aller Vorurteile, die über etwas existiere.

<sup>6</sup> Siehe auch Servicedefinition, S. 15.

**Alle Funktionen in einem realen System, seien es Abläufe in Organisationen, Prozesse, Aktivitäten, Funktionen in Softwaresystemen, Applikationen, Teile von Applikationen oder Softwarefunktionen, lassen sich als Services darstellen und aus Services aufbauen!**

Oder kürzer formuliert:

**Alles, was aus- oder durchgeführt werden kann, ist ein Service!**

Hierbei hat jeder Service mindestens einen Provider (den Lieferanten) und einen Consumer (den Kunden oder Benutzer). Außerdem ist jeder Service in seiner Funktionalität und seinen Randbedingungen vorab definiert und diese Bedingungen sind sowohl dem Provider als auch dem Consumer bekannt.

Die Idee des Service lässt sich von kleinsten Geräten – individuellen Chips – bis hin zu riesigen Organisationen einsetzen, eine Tatsache, welche von einem hohen Maß an Universalität des Serviceparadigmas zeugt. Der technische Fortschritt, speziell bei der Miniaturisierung führt dazu, dass unsere Umgebung immer stärker mit Computern durchsetzt ist. Diese Umgebung weist dadurch ständig mehr Funktionalität auf. Die heutige Standardsicht auf solche Umgebungen ist sehr technikzentriert – sie stellt das einzelne Gerät oder Protokoll in den Vordergrund – und wird oft auch auf das einzelne Gerät reduziert; dabei ist es viel erfolversprechender diese Funktionalitäten als Services zu modellieren und unsere technische Umgebung als ein „System of Systems“ oder ein System von Services zu verstehen.

Der Ursprung der Idee des Service liegt in den Dienstleistungen, dies hat zur Folge, dass es Unterschiede zwischen Produkten und Services gibt. Unabhängig von einer exakten Definition des Begriffs Service teilen sich in der Praxis alle Services eine Reihe von Eigenschaften:

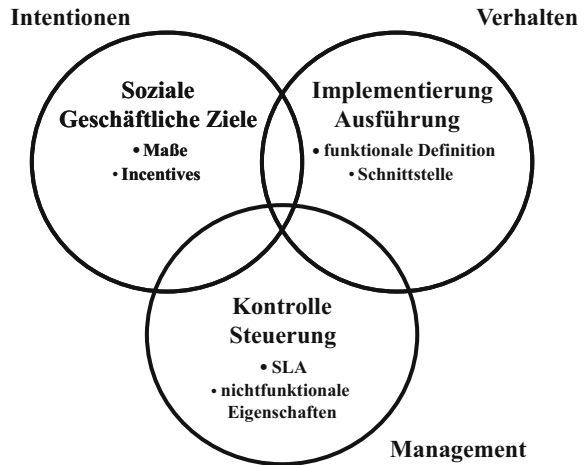
- Services sind sofort nutzbar.
- Services sind modellierbar.
- Services haben ein wohldefiniertes Verhalten.
- Services haben definierte Ein- und Ausgaben.
- Services stellen Fähigkeiten oder Funktionen zur Verfügung.
- Services werden „gemanagt“, um nichtfunktionale Ziele zu erfüllen.
- Services sind zusammenbaubar<sup>7</sup>, um damit neue Services zu erzeugen.

Die Serviceidee unterscheidet sich von der klassischen Produktidee durch folgende Merkmale:

- Services sind nicht direkt greifbar. Ihr Wert ist oft nur schwer quantifizierbar, wobei die Nichtgreifbarkeit sich in zwei Dimensionen vollzieht. Zum einen sind Services physisch ungreifbar, das heißt sie können nicht berührt oder betastet werden, zum anderen sind sie mental ungreifbar, das heißt sie können nicht mental als ein Gegenstand<sup>8</sup> verstanden werden.

<sup>7</sup> Servicekomposition, s. Abschn. 2.7.4.

<sup>8</sup> Ein Konto auf der Bank oder ein Lebensversicherungsvertrag ist zwar nicht physisch greifbar, beide stellen jedoch mentale Gegenstände dar und sind somit Produkte.



**Abb. 2.2** Unterschiedliche Sichten auf Services

- Services sind heterogener und vielgestaltiger als Produkte, da sie oft von Menschen direkt erzeugt werden.
- Services werden fast gleichzeitig produziert und verbraucht, im Gegensatz zu Produkten, welche meistens eine Lagerhaltung haben. Zentralisierung und Massenproduktion sind für Services problematisch.
- Services sind „verderblich“, das heißt Services können nicht gespeichert werden, aber aus Sicht des Consumers kann das Ergebnis des Service durchaus unverderblich sein.
- Die Qualität eines Service hängt von vielen sehr schwer kontrollierbaren Faktoren ab. Da ein Schwerpunkt der Services im direkten Kundenkontakt besteht, hat dies zur Konsequenz, dass die Qualität des Service von der Fähigkeit des Kunden determiniert wird, sich zu artikulieren, beziehungsweise der Provider in der Lage ist, dem Kunden zuzuhören.
- On-demand-Delivery – Die Services werden meistens direkt durch die Nachfrage bezüglich einer zu erbringenden Leistung ausgelöst.

Damit diesen unterschiedlichen Charakteristika Rechnung getragen werden kann, muss eine umfassende Servicedefinition sehr abstrakt sein. Die hier gewählte Definition des Service ist daher:

**Ein Service ist die Summe des beobachtbaren Verhaltens eines Systems (genannt Provider), gegeben durch die Menge aller möglichen Interaktionen und deren Relationen zwischen dem System und seiner Umgebung.**

Diese Definition ermöglicht es, Services als systemtheoretische Gebilde (s. Anhang A) zu beschreiben. Dieses Servicekonzept ist das Resultat der Trennung zwischen dem internen (Implementation) und externen (Interface) Verhalten eines Systems. Für den Consumer (als Teil der Umgebung) ist nur der externe Teil interessant. Implizit lassen sich aus dieser Definition eine Reihe von Eigenschaften für die Services ableiten:

- Ein Service hat einen Grad an Autonomie, da ohne Autonomie ein System überhaupt nicht identifizierbar ist.
- Services besitzen ein Interface (eine Schnittstelle), dieses wird durch die Grenze zwischen System und Umgebung gebildet. Da der Consumer ein Teil der Umgebung ist<sup>9</sup>, bildet seine Umgebungsteilmenge das für ihn wahrnehmbare Interface. Der andere Teil der Grenze zur Umgebung bildet für die Serviceimplementierung den Kontext.
- Da Systeme zu größeren Systemen zusammengesetzt werden können, gibt es auch die Möglichkeit zur Servicekomposition.
- Die funktionalen Eigenschaften sind die erwarteten Verhaltensmuster des Systems und ergeben sich aus dem Interface.
- Die nichtfunktionalen Eigenschaften sind die durch den Kontext (ohne den Consumer) ausgelösten Verhaltensmuster des Systems.

Ein Service muss hinreichend gut beschrieben werden. Speziell dann, wenn der Consumer den Provider nicht kennt, ist eine semantisch reichhaltige und strukturell gute Beschreibung notwendig, damit ein Service auch entdeckt und nachfolgend eingesetzt werden kann.

Die funktionalen Eigenschaften eines Service beschreiben genau die fachlich gewünschten Funktionen, die der Service erfüllen soll; das, was er tatsächlich für seinen Kunden durchführt.<sup>10</sup> Die funktionalen Eigenschaften beschreiben die Wirkung, nicht die Implementierung des Service, folglich wird hier die Grenze zur Umgebung und nicht die Substruktur des Systems beschrieben. Neben der Festlegung von dem, was durchgeführt werden soll, wird auch der Informationsfluss für den Service beschrieben. Services, die durch Menschen durchgeführt werden, wie zum Beispiel Reinigung, Outsourcing oder Reparaturen, werden im Allgemeinen auch als manuelle Services oder Dienstleistungen bezeichnet. Bei manuellen Services gestaltet sich der Informationsfluss in Form eines Dialogs zwischen dem Kunden (Consumer) und dem Dienstleister (Provider). Die starke menschliche Verflechtung und das iterative Verhalten aller Beteiligten ermöglichen es, dass sich diese Services auf diverse Kontexte einstellen können.<sup>11</sup> Im Fall von Software werden die funktionalen Eigenschaften von Services durch Interfaces, sowie die Vor- und Nachbedingungen spezifiziert. Services in der Software erzeugen ein eindeutiges syntaktisches Verhalten auf Grund der Tatsache, dass die genutzten Programmiersprachen Anforderungen an den Typ der Information in Form von Datentypen stellen – im Gegensatz zu manuellen Services.

Eine Servicebeschreibung besteht neben der Funktionalität des Service auch aus der Beschreibung der nichtfunktionalen Eigenschaften. Speziell das Fehlen von Beschreibungen nichtfunktionaler Eigenschaften verhindert eine „vernünftige“ Suche und Entdeckung von Services. Sinnvolle nichtfunktionale Eigenschaften für einen Service sind:

---

<sup>9</sup> Aus Sicht der Implementierung.

<sup>10</sup> Allerdings nur aus Sicht des Kunden.

<sup>11</sup> Viable System Services sind der Versuch, diese Flexibilität auch auf Software zu übertragen, s. Anhang B.1.



- Name des Providers – Provider müssen eine eindeutige und verifizierbare<sup>12</sup> Identität haben. Die logische Folge einer Identität ist die Existenz eines eindeutigen Namens und die Zugehörigkeit zu einer Organisation.
- Zeit – Angaben über die Servicezeiten oder auch Wochentage sind wichtige Größen für den Consumer.<sup>13</sup>
- Sprache –
- Qualität des Service –
- Sicherheit –
- Ort – Obwohl es in den meisten Fällen transparent sein sollte, wo der Service tatsächlich ausgeführt wird, kann der Ausführungsort bezüglich rechtlicher Belange oder Sicherheitsaspekte durchaus relevant sein. Bei nicht-IT-basierten Services kann es durchaus sein, dass der Service nur an einem bestimmten Ort durchgeführt werden kann oder darf. Neben den klassischen Formen der Ortsangabe können Telefonnummern oder URI-Adressen bei SLA-Verletzungen Kontakte und Möglichkeiten zum Ausweichen anbieten.
- Verfügbarkeit – Unter Verfügbarkeit versteht man die Kombination aus den zeitlichen und örtlichen Aspekten der Services.
- Obligation – Stellt die Verpflichtungen von Provider und Consumer dar.
- Preis – Neben dem Preis pro Serviceaufruf sind auch andere Formen von Preismodellen möglich, so zum Beispiel Flat, Bulk oder Prime Rate.
- Zahlungsmodalitäten – Analog zum Preis der Services kann auch die Art und Weise der Zahlung relevant sein.
- Strafen – Werden Zahlungsmodalitäten oder Obligationen verletzt, so treten die entsprechenden Strafen ein.

Für die Consumer sind die funktionalen und nichtfunktionalen Eigenschaften von Services zum Teil nicht voneinander unterscheidbar, insofern ist eine solche Unterteilung bis zu einem gewissen Grad willkürlich. Je bekannter und standardisierter die funktionalen Eigenschaften von Services sind, desto stärker rücken die nichtfunktionalen Eigenschaften bei der Entscheidung des Kunden für einen spezifischen Provider in den Vordergrund.

Der Ansatz, eine Organisation mit allen Prozessen und der gesamten Menge an Aktivitäten und Software durch ein Netz von miteinander wechselwirkenden Services zu modellieren, führt dazu, dass Organisationen als ein Graph von Services dargestellt werden können. Diese haben auf abstrakter Ebene ähnliche Charakteristika und bilden auf Grund ihrer Autonomie in sich geschlossene Systeme. Daher kann die Organisation als ein System von Systemen betrachtet werden. Auf ein solches Konstrukt lassen sich dann auch die systemtheoretischen Methodiken anwenden. Aber nicht nur Organisationen, auch unsere Umgebung lässt sich als ein solches „System of Systems“ modellieren und durch die entsprechenden systemtheoretischen Instrumente steuern.

Die gesamte technologische Entwicklung der letzten Jahre strebt in diese Richtung; durch die zunehmende Bandbreite und Geschwindigkeit des Informations-

<sup>12</sup> Die Verifikation kann auch über Dritte, so zum Beispiel Trustcenter, erfolgen.

<sup>13</sup> Der Service Personentransport der Deutschen Bahn kennt Wochen- und Feiertagsfahrpläne.

austauschs, als auch der stark gestiegenen Automatisierung, sind heutige Systeme deutlich stärker miteinander vernetzt, als dies je zuvor der Fall war. Der Endpunkt dieser Entwicklung ist dann ein Ultra Large Scale System (s. Abschn. 6.9). Mittlerweile sind Phänomene wie Emergenz und nichtlineare Skalierbarkeit in großen Organisationen und in technisierten Umgebungen merklich; diese Phänomene lassen sich nicht durch die Betrachtung einzelner Geräte, Applikationen oder Teilsysteme erklären, hier bedarf es des Einsatzes der Systemtheorie. Mit zunehmender Abhängigkeit und Wachstum der IT-Systeme treten Fragestellungen in Richtung der Überlebensfähigkeit solcher Konstrukte in den Vordergrund, womit große serviceorientierte Systeme immer stärker natürlichen oder sozialen Systemen ähneln und Fragestellungen nach Selbsterhalt, Selbstorganisation und Selbstreproduktion des jeweiligen Systems immer interessanter werden.

## 2.2 Service Oriented Enterprise

Eine Organisation lässt sich am besten analysieren, wenn nicht die formale Struktur gesehen, sondern die Kommunikationswege und -stile betrachtet werden. Dies liegt darin begründet, dass aus systemtheoretischer Sicht alle Teile einer Organisation – seien es Menschen oder ganze Suborganisationen – folgende notwendigen Bedürfnisse haben:

- Die Elemente müssen ihre Tätigkeiten synchronisieren. Schließlich verkörpern Organisationen auch Prozesse, beziehungsweise dienen dazu bestimmte Prozesse zu leisten, dies geht nicht ohne Formen der Koordination mit den Basistypen: Orchestration und Choreographie (s. S. 46).
- Die Elemente führen Diskussionen über Ziele, beziehungsweise kommen überein bestimmte Ziele zu verfolgen.
- Es müssen Fehlfunktionen entdeckt und kommuniziert werden, damit Reaktionen auf diese Fehlfunktionen stattfinden können.

Folglich entsteht eine Organisation erst dann, wenn sie kommuniziert! Umgekehrt betrachtet kann man aber auch eine Organisation als Ergebnis von Kommunikation betrachten und folglich die Kommunikationsformen und -wege als ein Architektur-bild der Organisation nutzen.

Die unterschiedlichen Stadien einer Organisation in ihrer chronologischen Erscheinung sind in Abb. 2.3 dargestellt. Die Evolution von Organisationen ist in aller Regel problemgetrieben, das heißt die Organisation existiert so lange in einem Stadium, bis die nächste Krise eintritt. Die Krisenlösungen sind immer kontrollgetrieben: Zunächst werden Kontroll- und Steuerungsmechanismen ausgebaut, gefolgt von den operationalen Aspekten. Nicht alle Organisationen passen exakt in dieses Schema, oft beinhalten die Organisationen Mischformen oder haben etwas andere Ausprägungen der verschiedenen Stadien, aber es ist hilfreich Stereotypen zu betrachten, damit der organisatorische Kontext überhaupt klar wird. Neben den unterschiedlichen Stadien ist bei den meisten Organisationen – langfristig gesehen –

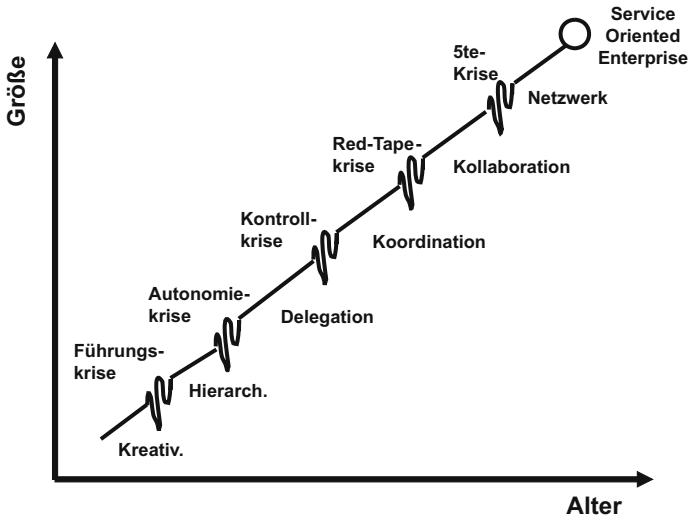


Abb. 2.3 Die Evolutionsstufen einer Organisation

ein Größenwachstum zu beobachten. Üblicherweise korrelieren die Größe mit dem Alter und der Führungsstil mit der Größe. Letzteres resultiert aus der Notwendigkeit einer effektiven Kommunikation und Steuerung von großen Organisationen. Im Laufe ihrer Entwicklung durchlaufen die meisten Organisationen verschiedene Stadien (wenn sie lange genug überleben):

- **Kreativitätsstadium** – Das erste Stadium einer Organisation ist das Kreativitätsstadium. Dieses ist dadurch charakterisiert, dass es eine kleine Zahl von Mitarbeitern gibt, welche in aller Regel sehr informell miteinander kommunizieren.
- **Hierarchiestadium** – Die nächste Stufe ist die der funktionalen Organisation. Es bilden sich immer mehr Suborganisationen heraus, welche jeweils spezifische Aufgaben erledigen. Im Laufe der Zeit werden explizite Hierarchien<sup>14</sup> geformt. Die Suborganisationen beginnen oft als Abteilungen und werden im Laufe zunehmenden Wachstums erst als Profitcenter, dann als Einheiten ausgegründet. Charakteristisch ist ein hohes Maß an Machtasymmetrie innerhalb der Organisation. Es werden definierte formale Kommunikationsstrukturen herausgeformt, Buchhaltung und Verwaltungseinheiten zeigen sich als Substrukturen. Auf Grund der Hierarchie werden alle Entscheidungen zentral gefällt und der Fokus der Führung ist auf Effizienz der Operationen, sowie auf einen hohen Grad an Professionalität innerhalb der Organisation gerichtet.
- **Delegationsstadium** – Das Hierarchiestadium wird durch die Delegation von Macht an die einzelnen Organisationsteile aufgelöst. Diese verselbstständigen sich und werden zu sogenannten „Divisionen“. Jede einzelne Division ist alleine

<sup>14</sup> Aus dem Griechischen *τεραρχία*, setzt sich zusammen aus *Hieros* = heilig und *Archos* = herrschen. Ursprünglich bedeutete es *Herrschaft durch den Priester*. Der Begriff wurde auf alle Organisationen verallgemeinert.

für alle Operationen und Geschäftsergebnisse im Markt verantwortlich (analog zum Kreativitätsstadium). Die Aufgabe des Topmanagements in einer solchen Organisation ist es primär, die langfristige Strategie sowie aktive Akquisitionen von neuen Divisionen zu betreiben. Intern sind die Divisionen meistens hierarchisch oder sogar in Subdivisionen organisiert. Jede einzelne Division ist innerhalb ihres Marktes auf Expansionskurs und versucht ein lokales Optimum bezüglich Profit und Marktanteilen zu erreichen, was in einigen Fällen zu einer innerorganisatorischen Konkurrenz führen kann.

- Koordinationsstadium – Dies ist eine Organisationsform, welche dezentral den Divisionen die Durchführung und die Verantwortung überträgt, aber zentral plant und standardisiert. Trotz organisationsweiter Standards und formalisierter Mechanismen bleibt die Divisionsstruktur faktisch vorhanden. Der Fokus des Topmanagements ist es, die einzelnen Divisionen zu Kooperationen zu bewegen und damit eine neue und wiederum „kohärente“ Organisation zu erschaffen.
- Kollaborationsstadium – In diesem Stadium der Entwicklung wird die Zentrale entmachtet und die Gesamtorganisation erhält eine Matrixstruktur. Der Fokus des Topmanagements liegt auf dem Lösen von Problemen und der Einführung von Innovationen in der Gesamtorganisation. Es kommt zu Teambildungen, welche funktionsübergreifend organisiert werden.
- Netzwerkstadium – Meist treten verschiedene Organisationen als Konkurrenten um einen gemeinsamen Markt auf. Konkurrenz ist aber nur eine mögliche Beziehung zwischen zwei Organisationen, eine andere Form ist die Kooperation. Wenn ein Netzwerk wächst, erreicht es irgendwann einmal eine kritische Masse. Dann zieht es neue Teilnehmer an, erzeugt einen hohen Mehrwert für alle Beteiligten und ist anschließend in der Lage, kleinere Netzwerke zu verdrängen. Innerhalb des Netzwerkstadiums besteht die Gesamtorganisation aus einem Netz von hochgradig autonomen Organisationen, welche eindeutige Services und Produkte mit einer klar umrissenen Funktionalität anbieten. Die Kooperation dieser einzelnen Teile geschieht auf einer Ad-hoc-Basis, um gemeinsam ein Ziel zu erreichen. Eine besondere Form eines solchen Netzwerks ist unter dem Begriff „virtuelles Enterprise“ bekannt. Eine der großen Stärken einer solchen Organisationsform liegt darin, dass verteilte Organisationen einen Teil der vertikalen Bürokratie innerhalb einer einzelnen Organisation ersetzen.

Die Beurteilung einer Organisation geschieht meist in Form eines Graphen, bei dem  $N_{\text{Knoten}}$ -Knoten über  $N_{\text{Kanten}}$  miteinander verbunden sind. Hierbei beträgt die minimale Zahl von Kanten:  $N_{\text{Kanten}} = N_{\text{Knoten}} - 1$ , während die längste Kette in solchen streng hierarchischen Bäumen durch:

$$L \sim \ln N_{\text{Knoten}}$$

gegeben ist. Eine solche Organisationsform ist jedoch nur sinnvoll, wenn sich die Prozesse und Funktionen der Organisation in Teilfunktionen und -prozesse zerlegen lassen und diese mehr oder minder unabhängig von den Suborganisationen ausgeführt werden können. Dies widerspricht in den meisten Fällen jeder Realität! Heutige Organisationen müssen Aufgaben erledigen, welche nicht klar definiert sind,

welche oft die gleiche Zeitskala haben wie die Produktion<sup>15</sup>, welche ihr Wissen erst durch den Produktionsprozess selbst aufbauen. Bei solchen Problemstellungen ist eine Zerlegung faktisch nicht mehr möglich, die Arbeit wird hier sehr stark kooperativ erledigt.

Moderne Organisationen haben daher neben der Effizienz auch das Ziel eine Form zu finden, welche der Gesamtorganisation ein hohes Maß an Robustheit gibt. Denn der einzelne Knoten im Netzwerk muss vor Überlastung durch Arbeit, nicht-antizipierten Kollaborationsmustern oder Reorganisation geschützt werden. Gleichzeitig muss aber auch die Gesamtorganisation vor einem Zerbrechen bewahrt werden. In komplexen Netzwerken werden die Informationen zur Koordinierung oft über Mittler weitergegeben, welche dadurch zusätzliche Informationslasten haben. In sehr großen Netzen, wie zum Beispiel dem Internet, ist die Last durch die Koordinierungsnachrichten immens hoch (s. Abschn. 6.7). Solche Netze haben das Risiko einer „Verstopfung“, auf das aktiv reagiert werden muss. Neben der Robustheit gegen Überlastung muss ein stabiles Netzwerk auch robust gegen den Wegfall einer Kante oder eines Knotens sein. Netzwerke, die in beiden Fällen robust sind, werden als ultrarobust bezeichnet.

In dem Modell von Dodds<sup>16</sup> wird ein allgemeines Netzwerk nach zwei Parametern charakterisiert:  $\lambda$  und  $\zeta$  (s. Abb. 2.4). In diesem Modell existieren zwei für den einzelnen Knoten interessante Größen: Zum einen der Abstand zwischen zwei Knoten  $i$  und  $k$  mit:

$$x_{ik} = \sqrt{d_i^2 + d_k^2 - 2}, \quad (2.1)$$

und die Tiefe  $D_{ik}$  des kleinsten gemeinsamen Vorgängers  $a_{ik}$ . Durch diese Betrachtungsweise entstehen zwei charakteristische Skalen: die Tiefe  $\lambda$  und die Breite  $\zeta$ . Kanten zwischen zwei Knoten entstehen nach einer Wahrscheinlichkeitsverteilung durch:

$$p(i, k) \sim e^{-\frac{1}{\lambda} D_{ik}} e^{-\frac{1}{\zeta} x_{ik}}. \quad (2.2)$$

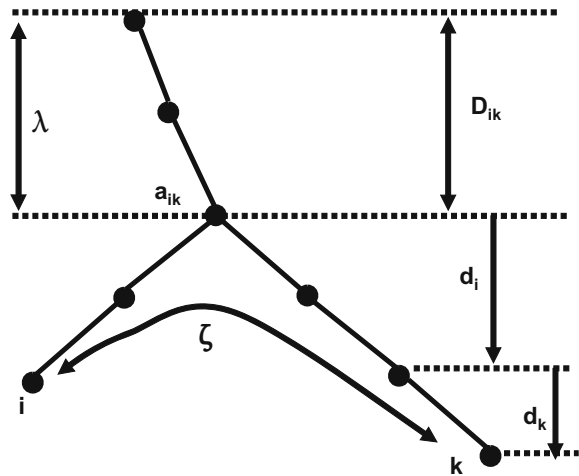
Die entstehenden Graphen der Grenzfälle lassen sich in mehrere Kategorien, je nach Parameterbereich  $(\lambda, \zeta)$ , einteilen:

- $(\lambda, \zeta) \mapsto (\infty, \infty)$  – Der Zufallsgraph, hier sind alle Kanten rein zufällig und gleichmäßig verteilt.
- $(\lambda, \zeta) \mapsto (\infty, 0)$  – Das lokale Team, Kanten sind nur bei direkten Nachbarn und mit direkten Eltern vorhanden.
- $(\lambda, \zeta) \mapsto (0, \infty)$  – Die Divisionsorganisation, Kanten sind nur innerhalb einer Division erlaubt.
- $(\lambda, \zeta) \mapsto (0, 0)$  – Nur die Wurzel kann sich direkt mit einem weiteren Knoten verbinden, eine Art Zentralknoten in Form eines Sterns entsteht.

<sup>15</sup> Typisch für Projekte, oft unterscheidet sich die Dauer des Aufbaus der Projektorganisation nur wenig von der Dauer des Projekts.

<sup>16</sup> Dodds, Information exchange and the robustness of organizational networks, 2003.

Wirklich stabile Netzwerke entstehen in der Nähe von  $(\lambda, \zeta) \approx (0.5, 0.5)$ , welche auch ultrarobust sind. Dieses einfache Modell zeigt, dass sich nur bestimmte Netzwerkformen auf Dauer als sinnvoll herausstellen.



**Abb. 2.4** Allgemeiner Graph eines hierarchischen Netzwerks nach *Dodds*

Ein virtuelles Enterprise ist strenggenommen eine Organisationsform, welche unabhängige Partner vereint, um einen einmaligen Auftrag zu erfüllen und danach wieder aufgelöst wird. Ein virtuelles Enterprise wird versuchen, für jede Teilaufgabe den bestmöglichen Provider einzusetzen, dessen Kernkompetenz identisch mit der Aufgabe sein sollte. Zu den daraus resultierenden Problemen zählen effiziente und durchaus verletzbare Verbindungen zwischen den einzelnen Organisationen. Diese Verbindungen müssen in Bezug auf Flexibilität und Versatilität ein ungleich höheres Maß an Anforderungen erfüllen als das sonst, im Rahmen einer normalen Organisation, notwendig ist. Als Organisationsform zerfallen die virtuellen Enterprises in first- und second-level Organisationen. Zu den typischen first-level virtuellen Enterprises zählen Projekte innerhalb eines Konzerns (s. Abb. 2.5), während eine echte virtuelle Organisation zum second-level Typ gehört. Die zeitliche Begrenzung des virtuellen Enterprises ist ein Schlüsselement zur Unterscheidung der beiden Typen.

Ein virtuelles Enterprise existiert mit zwei Basistopologien:

- Mit einem Provider, der den Beteiligten alle Infrastrukturfunktionen zur Verfügung stellt – so zum Beispiel für eine Open Source Entwicklung bei Sourceforge.
- Mit einem Provider, der eine erweiterte Plattform zur Koordination der individuellen Organisationen, die sich am virtuellen Enterprise beteiligen, zur Verfügung stellt.

Da das virtuelle Enterprise auf der freiwilligen Kooperation unabhängiger Partner basiert, fehlen im Vergleich zu klassischen Organisationsformen einige Elemente. Welche im Detail fehlen, kann von Fall zu Fall variieren, was dazu führt, dass