

Emergent Computing

Eine kritische Bestandsaufnahme

eingereicht von:
Klaus A. Brunner

DIPLOMARBEIT

zur Erlangung des akademischen Grades
Magister rerum socialium oeconomicarumque
Magister der Sozial- und Wirtschaftswissenschaften
(Mag. rer. soc. oec.)

**Fakultät für Wirtschaftswissenschaften und Informatik,
Universität Wien**

**Fakultät für Technische Naturwissenschaften und Informatik,
Technische Universität Wien**

Studienrichtung: Wirtschaftsinformatik

Begutachter: Ao. Univ. Prof. Dr. Wolfgang Hofkirchner

Wien, im Juni 2002

Inhaltsverzeichnis

1	Einleitung	1
2	Emergenz	3
2.1	Ursprung und Positionierung des Emergentismus	5
2.1.1	Reduktionismus	5
2.1.2	Holismus	7
2.1.3	Dualismus	8
2.2	Elemente emergentistischer Theorien	9
2.2.1	Hierarchie	9
2.2.2	Downward Causation	11
2.2.3	Supervenienz	12
2.2.4	(In-)Determinismus	15
2.2.5	Kohäsion statt Irreduzibilität und Unvorhersagbarkeit?	17
2.2.6	Metaphysik der Prozesse als Lösung des Emergenzproblems?	20
2.2.7	Emergenzbegriffe nach Stephan	21
2.3	Zusammenfassung	24
3	Emergent Computing	26

3.1	Zum Begriff und Selbstverständnis des Emergent Computing	26
3.1.1	Clarks Emergent Explanation	29
3.1.2	Simulationsrelative Emergenz	31
3.1.3	Carianis modellrelative Emergenz	34
3.1.4	Zusammenfassung	36
3.2	Beispiel: Konnektionismus	38
3.2.1	Grundlagen	38
3.2.2	Emergenz?	45
3.3	Beispiel: Evolutionäre Verfahren	49
3.3.1	Genetische Algorithmen	49
3.3.2	Classifier Systems	51
3.3.3	Emergenz?	54
3.4	Beispiel: Zelluläre Automaten	56
3.4.1	Grundlagen	56
3.4.2	Emergenz?	59
3.5	Artificial Life: die neue Heimat des Emergent Computing	64
4	Zusammenfassung und Diskussion	68
4.1	Emergent Computing: angewandter Reduktionismus?	68
4.1.1	Kann es „starkes“ Emergent Computing geben?	71
4.2	Schwache Emergenz: stark genug für wichtige Fragen	74
4.2.1	Emergent Computing im Software Engineering	74
4.3	Schlussbemerkung	80
	Literaturverzeichnis	82

Kapitel 1

Einleitung

„Emergent Computing“ (EC) hat sich in den letzten Jahren zu einem beliebten Schlagwort in der Informatik, ganz besonders in den Disziplinen Artificial Intelligence (AI) und Artificial Life (A-Life) entwickelt. Emergent Computing bezeichnet die Umkehr klassischer Ansätze der angewandten Informatik: statt rigide im „top-down“-Verfahren geplante Systeme zu bauen, die zentral kontrolliert werden und neben ihrer erwünschten, klar definierten Funktionalität möglichst keine unvorhersehbaren Effekte zeigen, sollen im EC genau solche – zumindest scheinbar – unvorhersehbaren Effekte provoziert, analysiert und genutzt werden. Im Emergent Computing wird die Informatik somit zur experimentellen Wissenschaft; denn die erzeugten Softwaresysteme sind nicht die präzise entworfene Implementation bestimmter Funktionen, sondern lediglich das Substrat, auf dem *computing* innerhalb des Systems selbst emergieren soll. Proponenten des EC versprechen eine Fülle von Anwendungen: von der Lösung aufwendiger Optimierungsprobleme (Genetische Algorithmen) über leistungsfähige Mustererkennung (Neuronale Netze) bis hin zum Verständnis oder gar zur Erschaffung von Leben (A-Life) gehen die Ziele. Auch außerhalb des Wissenschaftsbetriebs, in populären Sachbüchern¹, taucht der Begriff Emergent Computing neuerdings häufiger auf.

¹Einen sehr oberflächlichen, breitgefächerten Überblick bietet z.B. Johnson [Joh01]; wesentlich fundierter und auch etwas kritischer schreibt Emmeche [Emm94], der sich jedoch primär auf A-Life (siehe 3.5) bezieht.

Die Ansprüche und Erwartungen sind groß – es stellt sich zum einen die Frage, wo die Grenzen des Emergent Computing liegen. Wieviel „Neues“ und „Unvorhersehbares“ (wenn überhaupt) kann aus Software hervorgehen, die letztlich immer auf deterministischer, formaler Logik basiert? Weiter noch, könnten Intelligenz oder Leben aus Software entstehen? Diese Kernfragen von AI und A-Life haben sehr viel mit Fragen der Philosophie zu tun, wie etwa dem Körper-Geist-Problem, und damit auch der Debatte um den physikalischen Reduktionismus und seine Konkurrenten, besonders den Emergentismus. Die vergleichsweise junge Disziplin der Informatik betritt hier Terrain, das bereits seit langer Zeit von anderen Wissenschaften (wie etwa der Biologie) und der Philosophie bearbeitet wird; eine Situation, in der interdisziplinäres Arbeiten nicht nur sinnvoll, sondern notwendig ist. Hier stellt sich dann die andere Frage: was hat „Emergent Computing“ mit philosophischen Theorien der Emergenz zu tun, die in den 1920er Jahren erstmals genauer formuliert wurden, und in den letzten Jahrzehnten wieder (wenn auch nicht unverändert) in den Systemwissenschaften auftauchten? Diesen Fragen widmet sich die vorliegende Diplomarbeit.

- Es wird zunächst der Begriff Emergenz genauer untersucht: hier werden die Fragen „was ist Emergenz, und was nicht?“, „welche Spielarten von Emergenz gibt es?“ behandelt.
- In der Folge wird ein Überblick über Selbstverständnis und Diskurs des Emergent Computing in der Informatik gegeben („was ist Emergent Computing?“, „was erwartet man in der Informatik von Emergent Computing?“). Dabei wird ein Überblick über die wesentlichsten Techniken des Emergent Computing gegeben.
- Abschließend werden die gewonnenen Erkenntnisse zusammengefasst und neue Fragen erörtert („was können wir daraus lernen?“).

Kapitel 2

Emergenz

Dieses Kapitel widmet sich dem Begriff „Emergenz“ – oder vielmehr den *Begriffen* von Emergenz, die durchaus zahlreich und vielfältig sind. Ausgangspunkt ist dabei die Emergenzphilosophie des frühen 20. Jahrhunderts, die Emergenz erstmals zum zentralen Element theoretischer Überlegungen machte. Die neueren Emergenzbegriffe, die einige Jahrzehnte später in der Systemtheorie, und in den letzten Jahren in der Informatik auftauchten, werden diesen Ursprüngen gegenüber gestellt. Ein Resultat dieses Kapitels soll eine Liste von Kriterien sein, nach denen Theorien auf ihren Bezug zum Emergentismus beurteilt werden können.

Die Notwendigkeit einer differenzierten Begriffsdefinition ergibt sich aus der Praxis. „Emergenz“ ist, ganz besonders im Umfeld der heute so beliebten Komplexitäts- und Systemtheorien, ein zwar relativ oft verwendetes, aber relativ selten klar definiertes Wort. Was bedeutet es, wenn Dinge oder Vorgänge als „emergent“ bezeichnet werden? Wie unterscheiden sie sich von ihren nicht-emergenten Pendanten?

In diesem Sinne weist Corning [Cor02] auf ein Problem hin, das sich spätestens bei den ersten Literaturrecherchen zu diesem Thema stellt: der Begriff wird nicht nur innerhalb wissenschaftlicher bzw. philosophischer Kreise durchaus verschieden gedeutet, sondern auch in der (vor allem englischen) Alltagssprache verwendet. Dort jedoch synonym für die wesentlich unspezifischeren Begriffe „Auftauchen“, „Erscheinen“ oder „Wachsen“. Eine Publikation mit dem Titel „The Emer-

gence of Democracy in Russia“ muss daher nichts mit Emergenztheorien im philosophischen Sinne zu tun haben.

Der engere Emergenzbegriff, der im Brennpunkt dieser Arbeit liegt, sei vorerst nur grob umrissen. Er ist zu allererst als Teil von *Emergenztheorien* zu begreifen, im Gegensatz zu der oben erwähnten umgangssprachlich-phänomenologischen Bedeutung, die keine spezifische Weltanschauung impliziert. Emergenztheorien machen Aussagen über die Welt, nämlich über deren Sein und Werden. Anders gesagt: es gibt einen synchronen und einen diachronen Aspekt.

Der *synchrone* Aspekt betrifft das *Sein* der Welt. Synchrone Emergenz basiert auf der Annahme, dass ein Ganzes genuin andersartige Eigenschaften als seine Teile haben kann.¹ Typische Beispiele: Kochsalz besitzt nicht einfach die Summe der Eigenschaften von Chlor und Natrium; Gehirne denken, aber einzelne Neuronen nicht. Wesentlich ist die Annahme, dass zumindest manche Eigenschaften des Ganzen *nicht* auf die Eigenschaften der einzelnen Teile zurückgeführt werden können.

Der *diachrone* Aspekt beschäftigt sich mit dem *Werden* der Welt. Diachrone Emergenz ist eng mit dem Begriff „Evolution“ verflochten² und besagt, dass im Laufe der Zeit aus Bestehendem genuin Neuartiges entsteht. So werden etwa aus Urschleim Zellen, aus Zellen Organismen, einige Organismen entwickeln Gruppenverhalten und sogar Bewusstsein. Die Beispiele zeigen, dass unter „genuin Neuartigem“ grundsätzlich *qualitative* und eher sprunghafte Entwicklungen verstanden werden – eine Entwicklung von „wenig Urschleim“ zu „viel Urschleim“ würde nicht als emergent bezeichnet werden.

Für beide Emergenzvarianten spielt somit der Unterschied zwischen System (oder „Ganzem“) und Bestandteil eine wichtige Rolle.

¹Sehr häufig findet sich auch die Kurzformel „das Ganze ist mehr als die Summe seiner Teile“, die jedoch einen weiten Interpretationsspielraum zulässt und eher dem Holismus zuzurechnen ist (vgl. [Ste99], Kap. 13)

²Hier im allgemeinen Sinn von Entwicklung und Wandel, nicht auf biologische Evolutionstheorien im (neo-)darwinistischen Sinn beschränkt zu verstehen (vgl. [Fut90], Kap. 1)

2.1 Ursprung und Positionierung des Emergentismus

Bei vielen philosophischen Ideen lässt sich kein eindeutiger Zeitpunkt oder eine einzelne Person festmachen, auf die sie zurückzuführen sind; so auch beim Emergentismus. Es kann jedoch zumindest ein Zeitraum bzw. Personenkreis identifiziert werden, in dem Theorien der Emergenz in wissenschaftlichen Kreisen erstmals intensiv dokumentiert und diskutiert wurden: der sogenannte britische Emergentismus der 1920er Jahre, die „Blütezeit des Emergentismus“ mit ihren wichtigsten Vertretern Conwy Lloyd Morgan, Samuel Alexander und Charles Dunbar Broad.³ Emergenztheorien konkurrieren mit anderen Theorien, die über Sein und Werden der Welt Aussagen treffen. Die folgende Kurzbeschreibung der wichtigsten konkurrierenden Ansätze ist nicht nur ideengeschichtlich interessant, sondern auch notwendig, um den Emergentismus genauer abzugrenzen: bevor genauer ausgeführt wird, was Emergenz ist, soll verdeutlicht werden, was Emergenz *nicht* ist.

2.1.1 Reduktionismus

Der Begriff „Reduktionismus“ bezeichnet ein Spektrum philosophischer Ideen, das durch eine in der Wissenschaft sehr populäre Grundhaltung charakterisiert ist: der Teil ist wichtiger als das Ganze (die Teile erklären das Ganze, das Ganze resultiert aus den Teilen, usw.). Der Reduktionismus ist die (meistens unausgesprochene) metaphysische Theorie, die hinter der Methodologie der *Analyse* steht – nur wenn sich das Ganze aus den Teilen erklären lässt, macht es Sinn, diese Teile voneinander zu trennen und genauer zu erforschen.

Michael Ruse [Rus95] unterscheidet grundsätzlich ontologischen, methodologischen und Theorien-Reduktionismus. Für die Emergenzdiskussion sind vor allem die

³Detaillierte geschichtliche Recherchen sind bei Blitz [Bli92] und Stephan [Ste99] zu finden. Nach Blitz und Stephan können die Wurzeln des Emergentismus zumindest bis zu John Stuart Mill (1806–1873) zurückverfolgt werden; auch im Kontinentaleuropa des 19. Jahrhunderts gab es frühe emergentistische Ansätze (Johann Christian Reil, Gustav Theodor Fechner als bekanntere Beispiele). Einige Autoren orten emergentistische Ideen bereits bei Empedokles, Hippokrates, Galen und anderen griechischen Philosophen (vgl. [Ste99], S. 247f.)

ersten beiden Aspekte relevant, und zwar primär für den sogenannten reduktiven Physikalismus. Reduktionismus per se muss nicht vollständig sein: auch die Reduktion einzelner Ebenen auf jeweils untergeordnete (wie z.B. die Gesellschaft auf die Psyche, ohne diese weiter auf die Biologie zu reduzieren) ist möglich. Da Emergenztheorien jedoch typischerweise nicht nur einzelne Teilbereiche der Welt beschreiben, sondern sich mit allen qualitativen Unterschieden in der Welt befassen, ist der reduktive Physikalismus ihr wesentlichster Konkurrent.

Dieser ist in ontologischer Hinsicht eine Form von Monismus. Er basiert auf der Annahme, dass sich die Welt aus *einer* Art von Substanz, nämlich Materie, zusammensetzt. Atome, Moleküle, Zellen und komplexe Organismen von Qualle bis Mensch setzen sich aus Materie zusammen; eine andere Art von Substanz (wie etwa Seele) gibt es nicht.

Methodologisch wird im reduktiven Physikalismus alles als Ausprägung und Resultat der Physik begriffen: weil letztlich jedes Ding aus physischen Elementarteilchen besteht, lässt sich zumindest im Prinzip jedes Ding hinsichtlich seiner Eigenschaften auf die Eigenschaften solcher Teilchen und der zwischen ihnen wirkenden Kräfte reduzieren. Chemie, Biologie, Psychologie und Soziologie sind daher prinzipiell auf die physikalischen Naturgesetze zurückführbar (obwohl dies in der Praxis allein schon durch die zu bewältigenden Datenmengen unrealistisch sein mag). Ein Organismus setzt sich nicht nur aus Teilchen zusammen, sondern resultiert auch aus diesen Teilchen: es ist daher prinzipiell möglich, einen lebenden Menschen mit all seinen Eigenschaften analytisch in immer kleinere Elemente und ihre Eigenschaften zu zerteilen, bis man auf der Ebene der Elementarteilchen angelangt ist. Auf dem Weg von der Makro- bis zur Mikroebene geht dabei in Summe nichts an Erklärungskraft verloren; die Grenzen zwischen psychischer, biologischer, chemischer und physikalischer Domäne sind keine Hindernisse, weil sie „in Wirklichkeit“ nicht existieren. Setzt man nun gedanklich die Teilchen wieder genauso zusammen, wird am Ende wieder der selbe Mensch dastehen. Bei jedem Schritt des Zusammensetzens (z.B. im Schritt „Moleküle zu Zellen“ oder „Neuronen zu Gehirn“) ist prinzipiell das Resultat vorher genau abzusehen, weil die Eigenschaften des zusammengesetzten Teils aus den Eigenschaften von dessen Teilen resultieren. Es gibt keine Überraschungen bei der Synthese, weil das Wissen um die Teile nicht nur notwendig, sondern auch ausreichend ist, um das Ganze zu verstehen.

Für die (effiziente) Kausalität bedeutet dies, dass diese letzten Endes nur in der Interaktion von Elementarteilchen zu finden ist („proximate causality“, wie es Salthe nennt [Sal85]). Der Laplacesche Dämon, der sowohl die Zustände aller Elementarteilchen als auch ihre Kausalzusammenhänge kennt, kann mit diesem Wissen prinzipiell die Zukunft vorhersagen.

2.1.2 Holismus

Der Holismus kann als Spiegelbild des Reduktionismus gesehen werden: während dem radikalen Reduktionisten das Wissen um die Teile genügt, um das Ganze zu verstehen (das Ganze resultiert aus den Teilen, die Teile sind nicht vom Ganzen abhängig), ist für den radikalen Holisten allein das Ganze relevant. Um das Ganze zu verstehen, ist das Wissen um die Teile (deren Existenz nicht bestritten wird) weder hinreichend noch notwendig; die Analyse der Teile verspricht somit keinen Erkenntnisgewinn über das Ganze.

Eines der bekanntesten Beispiele für holistische Theorien findet sich im Buch „Holism and Evolution“ von J. C. Smuts, einem Zeitgenossen Lloyd Morgans und Broads. Obwohl Smuts' Theorie einige emergentistische Ideen enthält, geht sie doch über den Emergentismus deutlich hinaus:

Wholes are not mere artificial constructions of thought; they actually exist; they point to something real in the universe, and Holism is a real operative factor, a *vera causa*. There is behind Evolution no mere vague creative impulse or *Èlan vital* ... This is a universe of whole-making. ([Smu27], Kap. V, Hervorhebung im Original)

Für Smuts sind die „wholes“ die treibende, kreative Kraft des Universums; die Evolution ist ein Prozess, der immer neue „wholes“ hervorbringt, die diesen Prozess weiterführen. In der zweiten Auflage seines Buches erklärte er selbst in einer Fußnote die Unterschiede zum Emergentismus Lloyd Morgans⁴:

⁴Lloyd Morgan kannte seinerseits Smuts' Werk und verfasste eine Rezension, in der er die Unterschiede zu seiner emergentistischen Theorie aufzeigte (siehe dazu [Ste99], S. 4 und 153).

To him emergence of the new in evolution of the universe is the essential fact; to me there is something more fundamental—the character of wholeness, the tendency to wholes, ever more intensive and effective wholes, which is basic to the universe, and of which emergence or creativeness is but one feature . . . ([Smu27], Kap. XII)

Ein von holistischen Ideen geprägter Bereich der Wissenschaft ist die psychologische Gestalttheorie: auch hier geht es um Ganze, die nicht sinnvoll analytisch zerlegt werden können [Ham95].

2.1.3 Dualismus

Für den von R. Descartes im 17. Jahrhundert geprägten Dualismus besteht die gesamte Welt aus zwei fundamental verschiedenen Bereichen: dem Bereich der Materie (*res extensa* oder Körperwelt) und dem Bereich des Geistes (*res cogitans*). Die Körperwelt lässt sich im wesentlichen mit jener des reduktiven Physikalismus gleichsetzen: alles ist durch Naturgesetze determiniert, Chemie und Biologie werden durch die Physik subsummiert, jedes Ding bis hin zum lebenden Wesen ist Maschine. Die einzige Ausnahme findet sich beim Menschen: dieser ist nicht ausschließlich Maschine oder Automat, sondern das einzige Wesen, in dem auch Geist zu finden ist.

Im lebenden Menschen findet sich somit die einzige Schnittstelle zwischen den ansonsten unabhängig existierenden Bereichen *res extensa* und *res cogitans*; hier gibt es die einzige Lücke in der Kausalität der sonst völlig mechanisch-deterministischen Körperwelt, mittels derer der Geist in den Vorgängen der Körperwelt interveniert.⁵

Eine Variante des Dualismus ist jene von Leibniz: für ihn gibt es genauso Geist und (völlig mechanistische) Körperwelt, die jedoch in keinem kausalen Zusammenhang stehen. Es besteht somit kein Bedarf nach einer Schnittstelle zwischen den Welten, beide sind kausal in sich geschlossen, stehen jedoch trotzdem in einer Parallelbeziehung zueinander. Jedem Zustand (und Zustandsübergang) entspricht ein

⁵Descartes glaubte diese Stelle in der Zirbeldrüse lokalisiert, was aber für den Dualismus als Philosophie nicht wesentlich ist – entscheidend ist, dass es zumindest eine solche Schnittstelle überhaupt gibt.

Zustand bzw. Übergang in der jeweils anderen Welt, trotzdem ist es nicht so, dass ein Zustand in der einen Welt den korrespondierenden Zustand in der anderen Welt *verursacht*.⁶

2.2 Elemente emergentistischer Theorien

2.2.1 Hierarchie

Emergence presupposes a notion of levels. [BC00]

Ein typisches Merkmal emergentistischer Auffassungen ist die Annahme, dass die Welt in verschiedene Ebenen strukturiert ist, wobei von „höheren“ und „niedrigeren“ Ebenen gesprochen wird. Diese Ebenen können je nach Auffassung absolut (für die gesamte Welt gleich) oder relativ bzw. kontextabhängig sein; sie können eine listenförmige oder eine baumförmige Relation darstellen; die Grenzen zwischen den Ebenen können fließend oder scharf unterscheidbar sein. Häufig wird von einer physikalischen Ebene gesprochen, einer darüber liegenden biologischen, und einer wiederum darüber liegenden psychischen Ebene.⁷

Diese Annahme für sich ist jedoch nicht einzigartig für emergentistische Ansätze. Auch radikal reduktionistische Wissenschaftler bedienen sich einer solchen Hierarchie zur Beschreibung (tatsächlich macht der Begriff des Reduktionismus nur dann Sinn, wenn überhaupt höhere Ebenen vorausgesetzt werden, die epistemisch reduziert werden können). Die wesentlichen Unterschiede zwischen Reduktionismus, Emergentismus und Holismus liegen – wie bereits erwähnt – in der ontischen Bedeutung, die den Ebenen zugesprochen wird, sowie der Frage, welche Kausalzusammenhänge zwischen und innerhalb den Ebenen herrschen. Für den radikalen Reduktionisten ist die unterste Ebene einer solchen Hierarchie (z.B. die Ebene

⁶Eine ausführliche Darstellung und Kritik des Dualismus in den Varianten von Descartes und Leibniz gibt Schleichert [Sch92]. Der Dualismus mit all seinen Konsequenzen wird heute nicht mehr ernsthaft vertreten, Teilaspekte finden jedoch immer wieder den Weg in neuere Theorien.

⁷Einige recht verschiedene Beispiele für Hierarchietheorien: Popper [PE82], Kapitel P1; Pattee et al. [Pat73]; sehr detailliert Salthe [Sal85], sehr leicht lesbar Ahl und Allen [AA96]. Siehe auch Stephan [Ste99] 3.3.

der Elementarteilchen) die einzig kausal wirksame. Was etwa auf der biologischen Ebene passiert, ist lediglich epiphänomenal. Die „biologische Ebene“ ist in diesem Sinn ausschliesslich eine komprimierte *Beschreibung* unzähliger mikroskopischer Vorgänge, die tatsächlich nur auf einer (der untersten) Ebene in der Interaktion kleinster Dinge stattfinden.

Emergentistische Theorien zeichnen sich dagegen durch eine mehr oder minder starke „Emanzipation“ höherer gegenüber den darunter liegenden Ebenen aus. Um eine emergierte Eigenschaft auf einer höheren Ebene zu verstehen, ist das Wissen um die Elemente der unteren Ebene notwendig (hierin besteht die Verbindung zum Reduktionismus), aber nicht hinreichend. Emmeche et al. bezeichnen diese emergentistische Kopplung zwischen Ebenen als Inklusivitätsrelation: „Levels are inclusive in that respect, i.e. the psychological level is built upon the biological and the physical, the biological upon the physical: phenomena on one level cannot be reduced to the lower level, but on the other hand they can never change the laws of the lower level.“ [EKS97]

Damit sind emergentistische Ansätze deutlich vom Holismus (der die untergeordneten Ebenen irrelevant macht) und dem Vitalismus mit seinem übernatürlichen *élan vital* abzugrenzen: der Emergentismus basiert auf einer materialistischen Ontologie, in übergeordneten Ebenen tauchen daher keine nicht-materiellen Entitäten auf. Unter anderem aus diesem Grund ist der bereits erwähnte Dualismus in der Tradition von Descartes oder Leibniz, der die Welt in zwei (fast) völlig getrennte Ebenen teilt, nicht dem Emergentismus zuzurechnen.⁸ Diese Position zwischen Reduktionismus und Holismus/Dualismus wird von Blitz treffend auf den Punkt gebracht: „Emergent properties are anchored in structures, and do not exist independently of them, though they are not reducible to them“ ([Bli92], S. 179).

⁸Allerdings ähneln die Probleme, die sich aus der behaupteten Interaktion zwischen Seele und menschlichem Körper ergeben, ganz deutlich jenen Problemen, die auch in der Emergenzdiskussion behandelt werden (vgl. Schleichert [Sch92], S. 18 f.) – besonders dann, wenn von Downward Causation, also einer Verursachung von oberen auf untere Ebenen, die Rede ist (vgl. das „Pepper-Kim-Dilemma“ [Ste99], Kap. 16)

2.2.2 Downward Causation

If talk of emergent properties is to have any meaning, emergent properties must be causally effective properties, with powers to affect the course of events in which they are involved. [Kim00]

Downward Causation (DC) bezeichnet einen kausalen Einfluss höherer auf niedrigere Ebenen in einem hierarchischen System: die Idee ist, dass die Eigenschaften eines Systems kausalen Einfluss auf seine Bestandteile oder deren Eigenschaften nehmen können. Obwohl nicht immer explizit darauf hingewiesen wird, ist die Existenz von DC ein wesentliches Merkmal emergentistischer Theorien⁹: eine Eigenschaft, die keinerlei kausale Wirksamkeit hat, ist epiphänomenal.

Die Debatte um Downward Causation ist fast genauso umfangreich wie die Debatte um Emergenz im allgemeinen; eine der Schwierigkeiten besteht dabei in der Definition von Kausalität bzw. den im Emergenzdiskurs beliebten vier Kausalitätsvarianten von Aristoteles (vgl. [Kla02], [ACF00]).

Emmeche et al. [EKS00] schlagen eine Klassifikation des DC-Begriffs in eine starke, mittlere und schwache Variante vor:

Starke DC Entitäten oder Vorgänge einer höheren Ebene verursachen Änderungen in Entitäten oder Vorgängen einer niedrigeren Ebene. Diese Variante der Downward Causation erfordert eine strikte Trennung zwischen den Ebenen und einen *konstitutiven Irreduktionismus*: Entitäten höherer Ebenen bestehen nicht nur aus Entitäten niedrigerer Ebenen. Sie erfordert weiters einen Bruch mit der Annahme, dass Gesetze der niedrigeren Ebene nicht durch die höhere Ebene verändert oder gar aufgehoben werden können (was z.B. bedeuten würde, dass die physikalischen Naturgesetze in biologischen Systemen verletzt würden, die physikalische Welt also nicht kausal geschlossen ist). Emmeche et al. sehen diese radikale Spielart des DC als äquivalent zum unwissenschaftlichen Vitalismus an; Blitz [Bli92] ordnet sie dem Holismus zu. Auch

⁹Die Bezeichnung „downward causation“ wurde allerdings erst in den 1970er Jahren von Donald T. Campbell [Cam74] oder auch Roger W. Sperry (laut [Cor02]) geprägt. Gelegentlich wird auch das Synonym „Makrodetermination“ gebraucht.

der cartesische Dualismus kann nur mit einer Form starker Downward Causation funktionieren.

Mittlere DC Entitäten oder Vorgänge einer höheren Ebene sind Rahmenbedingungen für die (emergenten) Vorgänge auf niedrigerer Ebene. Die höhere Ebene gibt somit die *boundary conditions* oder *constraints*, die aus den vorhandenen Möglichkeiten auf niedrigerer Ebene selektieren. Beispiele dieser Variante von DC finden sich recht häufig, so etwa in den Hierarchietheorien Salthes [Sal85] oder auch Ahl/Allens [AA96].

Schwache DC In dieser Variante wird keine effiziente Kausalität von oben nach unten angenommen; die höhere Ebene *strukturiert* die Vorgänge und Entitäten auf der unteren Ebene. Das Leitbild sind Attraktoren in Zustandsräumen, die als Entitäten höherer Ebene Potentialitäten für die untere Ebene darstellen. So kann etwa „Wasser“ als Attraktor für Sauerstoff- und Wasserstoff-Atome gelten; jedes mögliche Molekül aus diesen Atomen stellt einen solchen Attraktor dar, unabhängig davon, ob es bereits realisiert wurde.¹⁰

2.2.3 Supervenienz

Das Konzept der Supervenienz ist zumindest artverwandt mit dem Problem der Downward Causation – für einige Autoren sind die beiden Begriffe sogar synonym (z.B. Corning [Cor02] und Blitz [Bli92]). Die Supervenienzdebatte im engeren Sinn spielt sich heute jedoch um das Geist-Körper-Problem (alias „psychophysisches Problem“) ab. Die Grundfragen lauten dabei: welchen Status haben mentale Vorgänge in der physikalischen Welt? Wenn alle physikalischen Vorgänge eine physikalische Ursache haben, sind dann trotzdem auch mentale Ursachen für physikalische Vorgänge möglich, also mentale Verursachung?

In den letzten Jahren hat sich Jaegwon Kim wie kaum ein anderer mit diesen Fra-

¹⁰Paradoxerweise bekommt ausgerechnet diese schwache und mit dem Reduktionismus noch am ehesten verträgliche Form von DC damit platonische (oder, wie Emmeche et al. meinen, sogar deistische) Züge, weil sie eine vorgegebene Attraktorenmenge festlegt, die alle Möglichkeiten bereits im Vorhinein fixiert. Im Unterschied dazu werden bei mittlerer DC diese Attraktoren erst durch den Prozess der Emergenz und die dadurch wirkenden Constraints erstellt.

gen auseinandergesetzt [Kim93, Kim98]. Eine ausführliche Diskussion von Kims Werk und der Kritik daran würde den Rahmen dieser Arbeit sprengen; hier soll jedoch zur Orientierung eines seiner zentralen Argumente summarisch wiedergegeben werden.¹¹ Dieses baut im wesentlichen auf drei Thesen auf:

These 1 (Supervenienz) *Für jede mentale Eigenschaft M gilt: wenn etwas M hat, muss es eine physikalische Basiseigenschaft P haben, womit auch alles, was P hat, auch M hat. Sprich: M superveniert auf P.*

In anderen Worten: jeder Realisierung einer mentalen Eigenschaft entspricht die Realisierung einer physikalischen Basiseigenschaft. Weiters ist zur Realisierung der mentalen Eigenschaft die Realisierung der jeweiligen physikalischen Basiseigenschaft *hinreichend*. Die Supervenienzthese stellt somit die Kovarianz mentaler und physikalischer Eigenschaften dar; mentale Eigenschaften sind stets in physikalischen Eigenschaften verankert und schweben nicht entkoppelt von diesen herum. Zu beachten ist, dass mit „superveniert auf“ keine Aussage über Downward Causation, ja nicht einmal über Emergenz gemacht wird: auch resultierende mentale Eigenschaften (wenn es so etwas gibt) supervenieren auf einer physikalischen Basis.¹²

These 2 (Kausale Geschlossenheit) *Jedes verursachte physikalische Ereignis hat eine physikalische Ursache.*

Es gibt somit keine physikalischen Ereignisse, die nur nichtphysikalische Ursachen haben. Die physikalische Welt ist daher kausal geschlossen.¹³

¹¹Die Darstellung folgt im wesentlichen den kritischen Ausführungen von Crisp und Warfield [CW01]; vgl. auch die Erläuterungen von Stephan ([Ste99], Kap. IV) und El-Hani und Pereira [EHP00].

¹²„... ‘arise out of’ and ‘supervene on’ are neutral with respect to the emergent/resultant distinction: both emergent and resultant properties of a whole supervene on, or arise out of, its microstructural, or micro-based, properties.“ ([Kim00], S. 308) Wie bereits erwähnt hat dieser Terminus jedoch nicht bei allen Autoren eine derartig neutrale Stellung, sondern gilt häufig als Synonym für Downward Causation.

¹³Man beachte, dass sich diese Interpretation der kausalen Geschlossenheit von einer weitaus

These 3 (Eigenschaftsdualismus) *Mentale Eigenschaften unterscheiden sich von physikalischen Eigenschaften.*

Dies ist eine notwendige Voraussetzung der Nichtreduzierbarkeit mentaler Eigenschaften: diese sind zumindest nicht identisch mit physikalischen Eigenschaften.

Kim argumentiert nun folgendermaßen. Erstens: falls Supervenienz nicht gilt, kausale Geschlossenheit und Eigenschaftsdualismus jedoch schon, dann sei mentale Verursachung nicht möglich (oder zumindest nicht verständlich). Hier mangelt es an der Kopplung der beiden Eigenschaften durch die Supervenienzrelation; die mentalen Eigenschaften schweben reibungslos über der physikalischen Welt.

Zweitens, der schwierigere Fall: falls Supervenienz gilt, kausale Geschlossenheit und Eigenschaftsdualismus ebenfalls, dann ist mentale Verursachung auch nicht möglich. Dies liegt Kim zufolge zum einen daran, dass die Verursachung einer mentalen Eigenschaft M^* durch die mentale Eigenschaft M nur möglich ist, indem der Umweg über die zugrundeliegenden physikalischen Basiseigenschaften P^* bzw. P genommen wird: M verursacht M^* , indem es P^* verursacht (siehe Abb. 2.1). Um eine mentale Eigenschaft zu verursachen, muss die physikalische Basiseigenschaft verursacht werden, auf der die mentale superveniert¹⁴ – dies gilt auch dann, wenn eine mentale Eigenschaft eine andere mentale Eigenschaft verursachen soll. Zum anderen lässt sich nun behaupten, dass P und M , unabhängig voneinander hinreichende Ursachen von P^* seien – solche kausale Überdeterminierung ist laut Kim logisch nicht haltbar. Folgt man diesen beiden Behauptungen, erfordert jedoch *jede* mentale Verursachung kausale Überdeterminierung und ist daher nicht haltbar.

Die Konsequenz aus diesen Betrachtungen ist für Kim, dass zwischen M und M^* kein „echter“ kausaler Zusammenhang herrschen kann. M superveniert auf P , P

strengerer unterscheidet, die als Exklusion bezeichnet werden kann: „Jedes verursachte physikalische Ereignis hat nur physikalische Ursachen.“ Hier ist die Geschlossenheit der physikalischen Verursachung vollständig: jedes physikalische Ereignis lässt sich auf eine physikalische Ursache zurückführen. Diese Definition schließt automatisch jede mentale Verursachung physikalischer Ereignisse aus.

¹⁴Crisp und Warfield [CW01] geben ein Beispiel: um Schmerz in meinem Arm zu verursachen, muss ein Bienenstich jenen neuronalen Zustand verursachen, auf dem der Schmerz superveniert.



Abbildung 2.1: *Supervenienz und mentale Verursachung*

verursacht P^* , M^* superveniert auf P^* . Zwischen M und M^* herrscht lediglich „superveniente Verursachung“.

Kim (selbst ein deklarerer reduktiver Physikalist) erweitert diese Argumentationsweise auch auf Downward Causation und akzeptiert daher nur eine epistemologische Form von Downward Causation [Kim00].

2.2.4 (In-)Determinismus

Ein weiterer, wichtiger Aspekt für Emergenztheorien ist jener der Bedeutung von (diachronem) Determinismus oder Indeterminismus: ist Emergenz etwas, das den Determinismus in Frage stellt? Sind emergente Eigenschaften deterministisch oder nicht?

Zur Untersuchung dieser Fragen ist es zuerst nötig, die strikte Verschränkung von ontologischem Aspekt (Determinismus) und epistemologischem Aspekt (Vorhersagbarkeit) zu lockern. Wenn es etwas wie objektiven Zufall tatsächlich gibt, ist dies ein Fall von Indeterminismus – unter dieser Voraussetzung folgt Unvorhersagbarkeit als logische Konsequenz (auch die genaueste Erkenntnis kann nichts an der Indeterminiertheit eines Ereignisses ändern). Der Umkehrschluss ist jedoch nicht zwingend: wenn es keinen objektiven Zufall gibt (also völlige Determiniertheit herrscht), folgt daraus nicht unbedingt, dass das Systemverhalten selbst durch einen im Sinne des Laplaceschen Dämons allwissenden Beobachter vorhersagbar

ist. Emmeche et al. betonen die Notwendigkeit dieser Trennung, verweisen aber darüber hinaus darauf, dass selbst in der Physik der Determinismus ein nicht ganz erreichbares Ideal ist – wenn man diese Aufweichung des Begriffs akzeptiert, wären auch emergente Phänomene als deterministisch zu bezeichnen [EKS97].

Die emergentistischen Ideen Karl Poppers sind von einer explizit antideterministischen Haltung geprägt, die auf dem Konzept der Propensitäten aufbaut. Propensitäten sind dabei objektive (also nicht aus einem Wissensmangel des Beobachters resultierende) Verwirklichungs-Wahrscheinlichkeiten, die einer Situation innewohnen:

Zurückliegende Situationen ... determinieren keineswegs die zukünftige Situation. Vielmehr bedingen sie veränderliche *Propensitäten, die zukünftige Gegebenheiten beeinflussen, ohne sie jedoch auf eine ganz bestimmte Weise festzulegen*. ... Die Zukunft ist *offen: objektiv offen*. ([Pop95], Hervorhebung im Original)

Diese objektive Wahrscheinlichkeit hat ihre Wurzeln nach Popper darin, dass alle physischen Systeme in Wirklichkeit „Wolken“ sind, also ihrem Wesen nach indeterministische Gebilde (im Gegensatz zu der mechanistischen Vorstellung von „Uhren“, völlig deterministischen Gebilden). Dies ist Voraussetzung für die Emergenz von hierarchischen Stufen und der kausalen Interaktion (auch Downward Causation) zwischen ihnen (siehe [PE82], Kap. P1).

Eine weitere indeterministische Emergenz- und Selbstorganisationstheorie wird von Hofkirchner vertreten.¹⁵ Die bijektive, mechanistische Relation zwischen Ursache und Wirkung („*actio est reactio*“) ist demnach nur dann gültig, wenn sich ein System im oder nahe dem thermodynamischen Gleichgewicht befindet. Wenn Systeme Selbstorganisation zeigen – was nur fernab vom Gleichgewicht geschehen kann – ist das Systemverhalten nicht mehr eindeutig determiniert:

¹⁵Über die Beziehung zwischen Emergenz und Selbstorganisation gibt es wohl ebenso viele Auffassungen wie über diese Begriffe selbst; allein dieses Geflecht aufzulösen, könnte die eine oder andere Diplomarbeit füllen (vgl. die Aufsatzsammlungen von Krohn und Küppers [KK90, KK92]). Festzustellen ist, dass Theorien der Selbstorganisation die Entstehung und Erhaltung von Ordnung (was immer damit im jeweiligen Fall gemeint ist) beschreiben – und damit etwas, wovon auch der diachrone Aspekt der Emergenztheorien handelt.

Different inputs may lead to the same output, and the same input may lead to different outputs. So causes and effects are not coupled unambiguously. Causes may give rise to novel effects. Little causes may have big effects. Similar causes may have dissimilar effects. Thus *causa non aequat effectum, actio non est reactio*. Due to mathematical short cuts not being applicable, emergent phenomena cannot be predicted in detail. There is no mechanistic transformation which turns the cause into the effect.¹⁶ There is an activity of the system itself which selects one of the several possible ways of reacting. [FH97]

Diese objektive Wahlmöglichkeit des Systems ist kein völliger Indeterminismus in dem Sinn, dass *alles* möglich ist. Es gibt lediglich eine (endliche) Zahl verschiedener Entwicklungspfade, die vom System eingeschlagen werden können. Dies wird als „incomplete“ oder „less than strict“ determinism bezeichnet [Hof01].

In der „klassischen“ Emergenztheorie von Lloyd Morgan, Alexander und Broad wird im allgemeinen *kein* Indeterminismus unterstellt (vgl. [Ste99], S. 26f).

2.2.5 Kohäsion statt Irreduzibilität und Unvorhersagbarkeit?

John Collier verfolgt einen deutlich anderen Ansatz zum Thema der Emergenz. Statt den üblichen Kriterien Irreduzibilität und Unvorhersagbarkeit gilt für ihn „Kohäsion“ als das wesentliche Kriterium, das emergente Phänomene von Epiphänomenen unterscheidet [CM98].

Kohäsion ist für Collier die *Basis dynamischer Identität*, weil sie den inneren Kausalzusammenhang von Dingen bezeichnet. Dinge – jedenfalls physische Objekte – zeichnen sich dadurch aus, dass sich ihre Teile (mehr oder minder) kohärent durch

¹⁶Hier wird evident, dass in dieser Beschreibung keine saubere Trennung zwischen indeterministischen Situationen und jenen von deterministischem Chaos erfolgt. Auch im deterministischen Chaos können ähnliche Ursachen unähnliche Effekte haben, genauso wie mathematische Abkürzungen nicht anwendbar sind (vgl. 3.1.2). Trotzdem gibt es eine mechanistische Transformation, die Ursache in Wirkung verwandelt. Erst die Behauptung, dass *gleiche* Inputs zu verschiedenen Outputs führen können (und dass dies auf eine Wahl des Systems zurückzuführen ist), geht über deterministisches Chaos hinaus.

Zeit und Raum bewegen.¹⁷

Als Ursachen spatio-temporaler Kohärenz kommen drei Möglichkeiten in Frage:

- Zufall. Als typische Beispiele gelten Sternbilder und zufällige Muster in Wolken („rabbits in clouds“) – die Verdinglichung dieser Erscheinungen ist eine rein subjektive; die angeblichen „Objekte“ werden vom Betrachter durch Analogiebildung konstruiert.
- Korrelierte unabhängige Ursachen. Regenbögen oder Schrotschüsse ergeben sich aus einer gemeinsamen Ursache, zwischen ihren jeweiligen Teilen herrscht jedoch kein kausaler Zusammenhalt. Dies ist für Collier die Definition von Epiphänomenen: „they are not merely coincidental, but they are not proper things, either ... they get their coherence from the regularities of their underlying causes, but have no internal binding“.
- Kausale Interaktion zwischen den Teilen: Kohäsion. Objekte, die von kausalen Zusammenhängen ihrer Einzelteile zusammengehalten werden, reichen von einem einfachen Stein bis zu einem Vogelschwarm. Kohäsion bezeichnet genau diesen inneren Zusammenhalt, der (wie in diesen Beispielen) stärker oder schwächer sein kann und das Objekt vor der Auflösung durch innere oder äußere Fluktuation bewahrt.

Kohäsion ist, wie Collier selbst feststellt, nicht direkt zu beobachten: „any direct cause can be imitated by epiphenomenal causes“. Ob, und wie stark kohäsiv ein System ist, kann empirisch festgestellt werden, indem der innere Zusammenhalt im Vergleich mit der Umgebung des Systems in verschiedenen Situationen getestet wird – Kohäsion ist immer relativ zum jeweiligen Kontext zu sehen. So zeichnet sich ein fallender Regentropfen durch eine gewisse Kohäsion gegenüber seiner gasförmigen Umgebung aus (durch die Oberflächenspannung), die jedoch beim Auftreffen auf eine Wasseroberfläche verloren geht: mit der Kohäsion geht dann auch das Objekt „Regentropfen“ verloren.

¹⁷Derartige Kohärenz ist auch für Goldstein [Gol99] eines der Charakteristika von Emergenz, das Kriterium der Neuartigkeit ist jedoch mindestens genauso wichtig. Emergenz ist „the arising of novel and coherent structures, patterns and properties during the process of self-organization in complex systems“ (zitiert nach [Cor02])

Kohäsion ist somit identitätsstiftend für Objekte. Der Zusammenhang mit dem Emergenzbegriff geht über die Kohäsion nicht nur der Teile eines Objekts, sondern auch der *Eigenschaften*. Als kohäsiv gilt eine Eigenschaft nach Collier dann, wenn sie weitgehend unempfindlich („largely insensitive“) gegenüber externen oder internen Fluktuationen ist. Als ein Beispiel gilt etwa der Druck von Gasen: dieser ist der Durchschnitt unzähliger einzelner Molekülbewegungen, der jedoch im Gegensatz zu einem mathematischen Konstrukt (wie etwa einem Durchschnittseinkommen) kausal wirksam ist. Fluktuationen der einzelnen Molekülbewegungen sind dabei irrelevant; ein und dieselbe Wirkung kann durch viele verschiedene Konfigurationen erreicht werden.

Reduktion scheitert hier, weil der Druck a) keine abstrakte Eigenschaft (ein rein mathematisches Konstrukt) ist, sondern selbst kausal wirksam ist¹⁸, und b) er sich aus der Makrostruktur des Systems ergibt (Behälter). Bei der Reduktion auf die Ebene der einzelnen Gasmoleküle verschwindet die Makrostruktur, die eine notwendige Voraussetzung der zu erklärenden Eigenschaft ist: ohne Behälter kein Druck. Ähnliches gilt für klassische Beispiele von nichtlinearen Systemen, wie etwa Rückkopplungsschleifen.

Während der Kohäsionsbegriff mit dem Kriterium der Irreduzibilität durchaus vereinbar ist und es auch impliziert, gilt dies nach Collier nicht für das Kriterium der Unvorhersagbarkeit, das seiner Ansicht nach für Emergenz wenig relevant ist:

- Die oft zitierten Beispiele des „deterministischen Chaos“ sind nicht a priori unvorhersagbar; die hierbei wichtige praktische Grenze der Mess- bzw. Rechengenauigkeit ist vom wissenschaftlichen Kenntnisstand abhängig und ändert sich somit: was heute mangels genauer Kenntnis von relevanten Parametern als unvorhersagbar (und damit möglicherweise emergent) gilt, muss in Zukunft möglicherweise aus der Kandidatenliste für Emergenz gestrichen werden.¹⁹

¹⁸ „... thermodynamic properties are themselves causally efficacious. The average family cannot get into a car. Pressure, however, can do work.“ [CM98]

¹⁹Stephan deutet interessanterweise genau auf diese praktische Grenze der Messgenauigkeit (in Verbindung mit deterministischem Chaos) als Begründung für die tatsächliche Existenz von Strukturemergenz (vgl. [Ste99], S. 245 f). Es ist allerdings Definitionsfrage, ob man irgendwelche Beschränkungen der Messgenauigkeit als „prinzipielle“ Unvorhersagbarkeit akzeptiert, oder nicht.

- Unvorhersagbarkeit ist keine hinreichende Bedingung für Emergenz – auch „single-level“ Systeme können unvorhersagbare Eigenschaften haben, sind aber per Definition nicht emergent.

Die zweite Anknüpfung an den Emergenzbegriff – neben der Irreduzibilität – ist darin zu sehen, dass Kohäsion neuartige Möglichkeiten schafft. Epiphänomene haben selbst keine kausale Kraft, diese ist nur auf ihrer konstitutiven Mikroebene zu suchen. Erst der kohäsive Zusammenhalt erlaubt es, neue Eigenschaften zu bilden: „When cohesion arises at higher levels, ... the potential is created for the formation of new cohesive structures or properties formed through causal interactions amongst the old ones. Emergent phenomena are novel because they create new capabilities.“ [CM98]

2.2.6 Metaphysik der Prozesse als Lösung des Emergenzproblems?

Eine unorthodoxe Position in der Emergenzdebatte, die hier der Vollständigkeit halber erwähnt sein sollte, vertritt Mark Bickhard: seine Kritik richtet sich gegen die heute verbreitete Metaphysik der (Mikro-)Partikel und ihrer Eigenschaften, die dem Reduktionismus so entgegenkommt und den Emergentismus so problembe-laden erscheinen lässt: „The inexorable reality of quantum particles keeps grab-bing all of the causal powers, leaving nothing for purported emergents. Perhaps we must accept ... that we ourselves are mere epiphenomena“ [BC00].

Ein Lösungsansatz, so Bickhard, könnte in einer Prozessontologie liegen, wie sie die Quantenfeld-Theorie verspricht: alles ist „organization of process“, auch das, was wir heute üblicherweise als Substanz bezeichnen. Hierbei bestehe nicht die Gefahr, dass dies einfach eine andere Form von Reduktionismus ist, die die Welt statt auf Partikelinteraktion auf Quantenfelder reduziert. Quantenfelder sind Prozesse und können ausschliesslich in diversen Mustern existieren – die ganze Welt besteht aus Mustern von Prozessen, quer über alle Ebenen der Realität. Muster kleinen und großen Maßstabs sind kausal völlig gleichberechtigt, eine Reduktion auf „Kleinstmuster“ ist sinnlos. Die prinzipielle Existenz nichtreduzierbarer Eigenschaften wäre in einer solchen Welt trivial.

Ein Anknüpfungspunkt für die vorliegende Arbeit, der über das allgemeine Kon-

zept von Emergenz hinausgeht, ist Bickhards Kritik an heute gängigen Modellen der künstlichen Intelligenz. Analog zur Kritik an einer Partikel-Metaphysik sieht er die Bestrebungen, die Repräsentation von Wissen und Bedeutung zu modellieren, in einer Sackgasse: die künstliche Intelligenz operiere mit fixen „Repräsentationspartikeln“, ohne das Entstehen (die Emergenz) von Bedeutung erfassen zu können. Auch hier sei als Ausweg eine völlige Abkehr von grundlegenden Annahmen der Symbolmanipulation anzustreben, um stattdessen eine Prozessontologie zugrunde zu legen [BT95].

2.2.7 Emergenzbegriffe nach Stephan

Eine umfassende und aktuelle Analyse des Emergenzbegriffes aus philosophischer Sicht legte Achim Stephan vor [Ste99]. Von besonderem Nutzen für die vorliegende Arbeit ist dabei das von Stephan geprägte Begriffssystem, das die Einordnung verschiedener Emergenzbegriffe – von der Frühzeit des Emergentismus bis zu aktuellen Entwicklungen – erleichtert. Obwohl die Klassifikation der Emergenzbegriffe nach Stephan nicht das gesamte Spektrum des emergentistischen Denkens abdeckt (siehe z.B.: „Kohäsion“), ist sie dennoch eine nützliche Grundlage zur Einordnung von (schein-)emergenten Phänomenen. Im Folgenden soll dieses Begriffssystem (vgl. [Ste99]) zusammenfassend wiedergegeben werden.

Ausgangspunkt: schwacher Emergentismus

Der schwache Emergentismus ist von 3 Grundprinzipen geprägt:

- **Physischer Monismus:** alle Dinge der Welt sind aus physikalischen Bausteinen zusammengesetzt (dies ist im wesentlichen die naturalistische Abgrenzung von *élan vital* und Interventionen der *res cogitans* des Dualismus). Übersinnliche Kräfte oder Entitäten stehen somit außer Diskussion.
- **Systemische Eigenschaften sind möglich:** dies sind Eigenschaften eines Systems, die keiner seiner Bestandteile hat (so besteht etwa ein lebendes Wesen aus nicht-lebenden Bestandteilen, ein sprechender Mensch besteht aus nicht-sprechenden Zellen, usw.). Wenn keines der verstärkenden Kriterien (s.u.)

hinzukommt, sind solche systemischen Eigenschaften nach Kim *resultierend*, nicht emergent [Kim00].

- Synchroner Determiniertheit: die Eigenschaften eines Systems sind an dessen Mikrostruktur gekoppelt - keine Änderung von Systemeigenschaften ist möglich, die nicht auch eine Änderung der Systemstruktur oder von Eigenschaften der Bestandteile ist.²⁰

Die obigen Kriterien sind für sich vollständig mit dem reduktiven Physikalismus vereinbar, weshalb dieser Emergenzbegriff tatsächlich als (sehr) schwach anzusehen ist. Er ist allerdings als Ausgangspunkt für die engeren Emergenzbegriffe nützlich und - wie sich im Laufe dieser Arbeit herausstellen wird - in der Literatur zum Emergent Computing durchaus verbreitet.

Verstärkung: Irreduzibilität

Das Kriterium der Irreduzibilität systemischer Eigenschaften führt zur synchronen Emergenz.²¹ Während reduzierbare systemische Eigenschaften aus den Eigenschaften der Systembestandteile abgeleitet werden können, trifft dies auf irreduzible nicht mehr zu. Die irreduzible Systemeigenschaft ist prinzipiell nicht auf die Eigenschaften der Bestandteile reduzierbar, die sie isoliert oder in anderen Verbindungen zeigen.

²⁰Um auf das unter „Reduktionismus“ skizzierte Beispiel des analysierten Menschen zurückzukommen: auch im Emergentismus würde beim Zusammensetzen der Einzelteile wieder der selbe Mensch mit den selben Eigenschaften erscheinen. Zu beachten ist weiters, dass hier keine Aussage über die Richtung der Kausalität gemacht wird - die Annahme der Synchronen Determiniertheit sagt nichts über „Downward“ oder „Upward“ Causation aus, sondern über die zwingende Kopplung von Struktur und Eigenschaften.

²¹Nicht zuletzt wegen der Ähnlichkeit mit dem Kriterium der synchronen Determiniertheit scheint der Begriff „analytische Emergenz“ passender. Hier soll aber trotzdem dem etablierten Begriffssystem Stephans gefolgt werden.

Verstärkung: Unvorhersagbarkeit

Das Kriterium der (prinzipiellen) Unvorhersagbarkeit führt die zeitliche Komponente ein und leitet damit zur diachronen, evolutionären Sichtweise über. Es besagt, dass im Laufe der Entwicklung der Welt völlig neue Strukturen, Entitäten und Eigenschaften auftauchen, von denen zumindest einige prinzipiell (also auch nicht durch einen allwissenden Laplaceschen Dämon) unvorhersagbar sind. Stephan unterscheidet zwei Sichtweisen:

- Die neuen Eigenschaften sind unvorhersagbar, weil die Strukturen, die diese Eigenschaften besitzen, neuartig und unvorhersagbar sind (neues Kriterium: Struktur-Unvorhersagbarkeit). Dies könnte in einer deterministischen Welt zumindest bei aperiodischen, chaotischen Prozessen der Fall sein, weil die Prognose hier auf praktische und prinzipielle Beschränkungen trifft.²² Geht man von einer indeterministischen oder zumindest nicht vollständig deterministischen Welt aus, ist derartige Unvorhersagbarkeit eine logische Folge.
- Die neuen Eigenschaften sind unvorhersagbar, weil sie irreduzibel sind (siehe oben). Diese Ausprägung des Emergentismus ist eine logische Konsequenz aus dem bereits erwähnten Kriterium der Irreduzibilität, woraus allerdings auch eine Einschränkung folgt: die prinzipielle Unvorhersagbarkeit gilt nur in dem Sinn, dass der Beobachter vollständiges Wissen über die einzelnen Teile des Systems besitzt, jedoch noch nie zuvor ein gleich aufgebautes System untersucht hat. Es ist somit nicht vorhersagbar, welche Eigenschaften ein völlig neuartiges System besitzt, bevor dieses zum ersten Mal erzeugt und analysiert wurde. Sind die Eigenschaften dieses neuartigen Systems erst einmal bekannt, sind damit auch die Eigenschaften aller identischen Systeme vorhersagbar – sie sind identisch.²³ Diese Art von Unvorhersagbarkeit

²²vgl. dazu das Konzept der Simulation in 3.1.2

²³Bekommt ein Reduktionist in einer emergentistischen Welt ein analytisch zerlegtes System in Einzelteilen mit einer vollständigen Beschreibung dieser Teile, ihrer Eigenschaften und einem Bauplan, jedoch ohne einer Beschreibung des ganzen Systems und seiner Eigenschaften, kann er beim Zusammensetzen überrascht werden: gelegentlich tauchen Eigenschaften auf, die sich prinzipiell nicht aus dem Wissen um die Teile vorhersagen ließen. Einen starken Emergentisten überrascht nicht, dass so etwas überhaupt passiert, er kann aber genausowenig vorhersagen, wann oder wo

kann als Aufwertung der Empirie verstanden werden: erst wenn das System tatsächlich gebaut wird, können seine Eigenschaften bestimmt werden.

Es ist an dieser Stelle wichtig festzustellen, dass auch diese starken Formen von Emergenz keiner völligen Ablehnung des Reduktionismus gleichkommen: es wird nicht behauptet, dass *alle* Systemeigenschaften irreduzibel oder unvorhersehbar sind. Der Emergentismus ist in dieser Hinsicht eine Ergänzung des Reduktionismus, keine unvereinbare Gegenposition wie der Holismus.²⁴

2.3 Zusammenfassung

Alle Emergenztheorien gehen von einer gemeinsamen ontologischen Basis aus: der Existenz von Systemeigenschaften und der Existenz einer hierarchischen Struktur der Welt. Dies sind notwendige, aber nicht hinreichende Bedingungen für „starke“ Emergenz.

Darüber hinaus vertreten Emergenztheorien eine oder mehrere der folgenden Thesen, die teils voneinander abhängen:

- Irreduzibilität
- Unvorhersagbarkeit
- Downward Causation
- Indeterminismus

es passiert, oder welche neuen Eigenschaften auftauchen werden. Dies jedoch unter der Voraussetzung, dass keiner der beiden vorher ein ähnliches System analysiert hat und daher aus Erfahrung weiss, dass aus bestimmten Konfigurationen bestimmter Teile bestimmte Eigenschaften emergieren.

²⁴Diese Tatsache – dass der Emergentismus eine deutlich abgegrenzte Position zwischen Reduktionismus und Holismus einnimmt – wird leider nicht selten ignoriert; Holismus und Emergentismus werden so einfach gleichgesetzt oder als Spezialfall der jeweils anderen Theorie angesehen (siehe z.B. [Hei95], wo Lloyd Morgan als prominenter Vertreter des Holismus bezeichnet wird).

Es soll an dieser Stelle bewusst kein neues Begriffssystem erfunden werden, das sämtliche Theorien mit emergentistischen Zügen aufnehmen kann und diese etwa noch mit Bewertungen wie „schwach“, „stark“, „echt“ usw. versieht. Wie sich noch zeigen wird, genügen die obigen Kriterien, um Emergenz im Kontext der Informatik zu untersuchen. Dies ist das Ziel des folgenden Kapitels.

Kapitel 3

Emergent Computing

3.1 Zum Begriff und Selbstverständnis des Emergent Computing

In einer der ersten umfangreichen Publikationen zum Thema Emergent Computing gibt Stephanie Forrest [For91] einen Überblick über dieses Feld, der auch zu einer ersten groben Definition¹ führt:

Researchers ... have begun to explore computational models in which the behaviour is in some sense more than the sum of its parts. ... In these systems interesting global behavior *emerges* from many local interactions. When the emergent behavior is also a computation, we refer to the system as an *emergent computation*.² [For91]

Als wesentliche Vorteile des Emergent Computing gelten:

¹Emergent *Computation* und Emergent *Computing* werden manchmal synonym gebraucht, meistens sind die Begriffe jedoch so zu verstehen: Emergent Computing ist jenes Feld der Informatik, das sich mit „emergent computations“ beschäftigt.

²sic! Aus dem Kontext wird allerdings ersichtlich, dass nicht „das System“ per se als „emergente Berechnung“ angesehen wird, sondern dass das System solches Verhalten zeigt oder, genauer gesagt, sein Verhalten so interpretierbar ist.

- **Höhere Effizienz.** Der Verzicht auf rechenintensive Kontroll- und Steuerungsmechanismen senkt die gesamte Rechenzeit. Es ist allerdings evident, dass dieser mögliche Gewinn durch einen Verlust an Sicherheit über die zu erwartenden Resultate erkaufte wird (begrenzte praktische Vorhersagbarkeit).
- **Höhere Flexibilität.** In der Interaktion mit dynamischen, komplexen Umgebungen ist es kaum möglich, sämtliche zukünftigen Erscheinungen abzu- sehen und die Reaktionen darauf explizit zu programmieren. EC-Systeme können flexibler auf solche Situationen reagieren – mit dem selben Nachteil wie schon im obigen Punkt erwähnt.
- **Vorteile der Repräsentation.** In einigen Bereichen (z.B. der Meteorologie) mangelt es an umfassenden formalen Modellen der Makroebene; hier bietet sich also der bottom-up-Ansatz des Emergent Computing zur Simulation an.
- **grounding** Harnad [Har91] sieht in den üblicherweise konnektionistisch geprägten Ansätzen des EC einen wesentlichen Vorteil gegenüber symbolorientierten (und damit „top-down“-orientierten) Ansätzen der AI: die Verankerung in den empirisch festgestellten physischen Grundlagen der *natürlichen* Intelligenz. Während sich bislang im menschlichen Gehirn keine Mechanismen zur Symbolverarbeitung isolieren ließen, sind künstliche neuronale Netze ein – vereinfachtes – Modell des Zentralnervensystems, wie wir es heute kennen.

Emergent Computation wird von Forrest deutlich von Emergenz im allgemeinen Sinne abgegrenzt; entscheidend ist die Nutzbarkeit, dass damit also „interesting and useful computational systems“ gebaut werden, und dass dieser Ansatz in manchen Situationen sogar der einzig Erfolg versprechende ist. Emergent Computation ist in dieser Diktion somit nur eine Untermenge jener Effekte, die aus nichtlinearer Interaktion in Computersystemen entstehen können – die restlichen dieser Phänomene sind also für Emergent Computation genauso unerwünscht (oder zumindest uninteressant) wie in den Standardverfahren der Informatik, die sich auf lineares Verhalten konzentrieren.

Drei bestimmende Eigenschaften von EC-System sind nach Forrest:

1. Eine Menge von Agenten, die jeweils expliziten Anweisungen folgen,

2. Interaktionen zwischen den Agenten (entsprechend den Anweisungen), die implizit Muster auf globaler Ebene bilden (Epiphänomene),
3. Eine natürliche Interpretation dieser Epiphänomene als *computations*.

Dass bei Forrest die Definition eines als „emergent“ deklarierten Phänomens dieses ausdrücklich als Epiphänomen bezeichnet, macht den hier verwendeten Emergenzbegriff äußerst schwach, wenn nicht überhaupt selbstwidersprüchlich. Umso erstaunlicher die kurz darauf folgende Andeutung einer Art von Downward Causation, die jedoch nicht weiter ausgeführt wird: „Global patterns may influence the behavior of the lower-level local instruction, that is, there may be feedback between the levels.“ ([For91], S. 2)

Chris Langton, einer der bekanntesten Vertreter des Emergent Computing, vertritt einen weitgehend ähnlichen Emergenzbegriff wie Forrest. Für ihn ist jedoch die unmittelbare Nutzbarkeit oder Interpretierbarkeit kein Kriterium mehr: unter der Voraussetzung, dass es keinerlei Steuerung des globalen Verhaltens gibt, ist *jedes* Verhalten auf einer höheren Ebene als jener der einzelnen Agenten emergent ([LTFR92], S. 4). Der Unterschied zwischen Forrest und Langton liegt in der Sichtweise der „emergenten“ Phänomene: für Forrest zählt nur der Aspekt der Berechnung, Langton sieht dies nur als eine von mehreren Möglichkeiten.

Während sich die meisten EC-Proponenten damit befassen, künstliche Systeme herzustellen, deren emergente Eigenschaften beobachtet und in irgendeiner Form nutzbar gemacht werden könnten, geht Langton zusätzlich davon aus, dass EC eine Eigenschaft natürlicher, physikalischer Systeme ist. Die Arbeitshypothese lautet daher:

Computation may emerge spontaneously and come to dominate the dynamics of physical systems ... [Lan91]

Viele Verfahren des Emergent Computing lehnen sich an natürliche Systeme an, ohne den Anspruch, diese natürlichen Systeme zu erklären oder exakt zu imitieren: dieser technische, anwendungsorientierte Ansatz ist typisch für die angewandte AI, etwa in der Mustererkennung.

Langton hingegen versucht mit seiner Arbeit Grundprinzipien der Natur zu modellieren, was konsequenterweise in Artificial Life mündet (siehe 3.5). Er ist jedoch nicht der einzige, sondern nur einer der bekannteren Vertreter dieser Ansicht. Das Verständnis von Emergent Computing als einem konstitutiven Grundprinzip aller möglichen natürlichen Systeme – physisch, biologisch, sogar psychisch – ist allgemein kennzeichnend für die Forschung des Santa Fe Institute (SFI), das sich in den letzten 10–15 Jahren zu einem Zentrum dieses Feldes entwickelt hat. Stellvertretend sei hier aus einer am SFI verfassten Dissertation zitiert:

The decentralized spatially extended systems that we see in nature today, in particular biological ones, have evolved. Shaped by the process of natural selection, the individuals in these systems have adapted over time to take advantage of their collective emergent pattern forming behavior to perform global information processing that benefits the system as a whole. The reason evolution has produced so many decentralized spatially extended systems capable of emergent computation, ... is likely because of the mentioned advantages of efficiency, flexibility, and robustness for emergent computation as opposed to centralized computation. Alternatively, it could be that it is simply too difficult for evolution to produce fully centralized systems from scratch. [Hor99]

Eine der Schwierigkeiten mit dieser „emergenten Informationsverarbeitung“ besteht darin, dass die zentralen Begriffe „computation“ und „information“ oft nur vage definiert werden (wie häufig in der Literatur des Emergent Computing), andererseits offenbar vom üblichen Sprachgebrauch abweichen. So stellt Emmeche [Emm94] die Frage, welchen Sinn es hat, von Berechnung zu sprechen, wenn es weder eine konzeptionelle Struktur von Symbolen, Manipulationsregeln und Axiomen gibt, noch jemanden oder etwas, für den diese Dinge und die Berechnung selbst irgendeinen Sinn ergeben.

3.1.1 Clarks Emergent Explanation

Clark [Cla96] spricht im Kontext von A-Life (siehe 3.5) von einer „Triade der Erklärungen“, die sich aus „homunkularer“, „interaktiver“ sowie „emergenter“ Er-

klärung zusammensetzt. Erklärungsbedürftig sind nach Clark zwei wesentliche Erscheinungen:

... (1) the unfolding of patterns over time and (2) the emergence of interesting or adaptive features from clever couplings between agents and environments (including other agents). [Cla96]

Clark grenzt seinen Emergenzbegriff von jenen ab, die Emergenz mit Überraschung des Beobachters bzw. Abweichung von explizit programmiertem Verhalten gleichsetzen – diese seien nämlich zu beobachterzentrisch. Er folgt dagegen der Linie von Steels: Emergenz ist dann im Spiel, wenn ein Phänomen seine Wurzeln in unkontrollierten Variablen hat. Steels' Unterscheidung kontrollierter und unkontrollierter Variablen:

A controlled variable can be directly influenced by a system, for example, a robot can directly control its forward speed. ... An uncontrolled variable changes due to actions of the system, but the system cannot directly impact it, only through a side-effect of its actions. For example, a robot cannot directly impact its distance to a wall; it can only change its direction of movement which will then indirectly change the distance. ([Ste94] zitiert nach [Cla96])

Dies ist aus mehreren Gründen problematisch.³ Es ist zum einen schwer zu sehen, wie zwischen einer resultierenden Systemeigenschaft und einer als emergent bezeichneten Eigenschaft unterschieden werden könnte, bzw. falls es überhaupt einen solchen Unterschied gibt. Zum anderen können hier nach Belieben aus praktisch jedem Ereignis emergente Phänomene erzeugt werden, indem der Bezugsrahmen verändert wird: ist der Bezugsrahmen der Roboter plus der Untergrund, auf dem er fährt, ist die Richtungsänderung nicht emergent. Beziehen wir irgendein Objekt (wie etwa die Wand) ein, kann die Entfernung zur Wand als unkontrollierte Variable gelten, und schon ist die Bewegung „emergent“. Das funktioniert jedoch

³Neben der hier geäußerten Kritik ist festzustellen, dass Clarks Emergenzbegriff teilweise Ähnlichkeit mit dem von Stephan deutlich kritisierten Emergenzbegriff Hermann Hakens aufweist (siehe [Ste99], 18. Kapitel).

auch in die andere Richtung: strenggenommen beeinflusst die Elektronik des Roboters nicht die Bewegungsrichtung des Roboters, sondern vorerst nur einen Stromfluss, der einen oder mehrere Motoren zur Richtungsänderung antreibt. In diesem Sinne ist schon die Richtungsänderung (oder gar die Rotationsgeschwindigkeit der Motoren) selbst eine unkontrollierte Variable und somit Basis eines emergenten Phänomens, ganz ohne äußere Bezugspunkte. Aus diesem Argument folgt auch, dass Clarks Emergenz genauso beobachterabhängig ist wie die von ihm kritisierten Emergenzbegriffe: der Rahmen ist eine Entscheidung des Beobachters, genauso wie die Wahl der zu messenden Variablen.

Tatsächlich gibt Clark selbst zu, dass derartig definierte Emergenz „neither breathtaking nor rare“ sei – es stellt sich die Frage, welchen Nutzen dieser Emergenzbegriff überhaupt haben soll. Wenn er etwa einen einfachen Rückkopplungsmechanismus als emergent bezeichnet, mit dem ein Roboter eine Wand entlang fahren kann⁴, muss man wohl jedem trivialen Regelmechanismus (z.B. einem Heizungs-thermostaten oder einer Toilettenspülung) emergentes Verhalten zugestehen. Für diese Arbeit besteht jedoch darin ein Nutzen, dass Clark erstmals verdeutlicht, welche Konsequenzen sich ergeben, wenn der im Emergent Computing oft vage und undifferenziert eingesetzte Begriff „Emergenz“ ernst genommen, genauer expliziert und zu Ende gedacht wird.

3.1.2 Simulationsrelative Emergenz

In [Bed97] formuliert Mark Bedau, ebenfalls vor dem Hintergrund von A-Life und den bereits von Forrest skizzierten Grundprinzipien des Emergent Computing, seinen Emergenzbegriff. Gegeben ist ein System S , das aus Teilen besteht, die die Mikroebene des Systems darstellen. Jeder Teil hat seinen eigenen Zustand (im Systemkontext: Mikrostatus) und eine Mikrodynamik D , die diesen Zustand beeinflusst, typischerweise abhängig vom Zustand der Teile in unmittelbarer Nähe. Ein Makrostatus P des Systems besteht aus strukturellen Eigenschaften, die vollständig durch

⁴Der Roboter ist so gebaut, dass er bei seiner Fahrt normalerweise leicht nach rechts zieht. Registriert er mit einem Sensor eine Kollision auf der rechten Seite, lenkt er kurzfristig nach links – das Resultat ist, dass der Roboter die Wand entlangfährt, indem er immer wieder mit ihr kollidiert, ein Stück wegfährt, kollidiert, usw.

den Mikrostatus der Teile und etwaige exogene Parameter determiniert sind; Irreduzibilität ist demzufolge ausgeschlossen.

„Schwache Emergenz“ ist nach Bedau dann und nur dann gegeben, wenn jedes P aus D und den exogenen Parametern prinzipiell *erklärt* werden kann (das folgt logisch aus den erwähnten Prämissen), aber ein zukünftiges P prinzipiell nur durch *Simulation* vorhergesagt werden kann.

Unter Simulation wird dabei die iterative Berechnung verstanden: will man im Zeitpunkt t_0 den Makrostatus im Zeitpunkt t_n wissen, müssen n Simulationsschritte gemäß der Mikrodynamik D durchgeführt werden. Eine rechnerische Abkürzung durch ein vereinfachtes Modell würde zu drastisch anderen Resultaten führen. Hierin liegt eine deutliche Analogie zu aperiodischen, deterministisch chaotischen Systemen.

Beachtenswert an dieser für das Emergent Computing selten klaren Begriffsdefinition ist unter anderem die Tatsache, dass Bedau etwaige indeterministische Einflüsse auf D explizit berücksichtigt: diese werden als exogene Parameter modelliert. Für die Verhältnisse innerhalb eines – per definitionem deterministischen – Computersystems ist dies völlig korrekt, denn dessen grundlegendste Transformationsregeln besitzen keine Freiheitsgrade. Eine z.B. durch einen Pseudo-Zufallszahlengenerator oder eine Benutzereingabe eingebrachte „Zufallszahl“ ergibt sich nicht aus Zuständen des Systems, sie ist innerhalb des betrachteten Systems in den Worten Rosens „unentailed“, somit ein Parameter. Ihr kausaler Einfluss auf die endogenen Variablen des Systems ist jedoch wieder durch die grundlegenden Transformationsregeln determiniert. Würde man z.B. in einem zellulären Automaten (siehe 3.4) die Übergangsfunktion mit einer Zufallskomponente versehen, müsste dies mittels einer solchen exogenen Variable geschehen. Echter Indeterminismus liegt dabei nicht vor, es sei denn, man ignoriert willkürlich den kausalen Einfluss dieser Variable. Tatsächlich handelt es sich also lediglich um einen „Exogenismus“, weshalb solche Versuche keine drastisch anderen Eigenschaften hervorbringen können, als sie von zellulären Automaten bekannt sind, deren Übergangsfunktionen nicht „indeterministisch“ sind.

„Weak emergence“ schliesst somit Irreduzibilität aus und neutralisiert indeterministisches Verhalten, indem es als exogen gegeben modelliert wird. Downward

Causation ist gar kein Thema; die effiziente Kausalität steckt allein in der Mikro-dynamik *D*. Der Fokus liegt damit auf Unvorhersagbarkeit, die allerdings nicht aus mangelnder Information eines Beobachters rührt – in einem in diskreten Zeitschritten arbeitenden, digitalen Rechner ist es bekanntlich kein Problem, sämtliche Systemzustände und Übergangsfunktionen völlig exakt zu kennen – sondern durch eine unausweichliche mathematische Beschränkung entsteht: auch der Laplacesche Dämon hätte in dieser Situation keinen Informationsvorsprung, er könnte ebenfalls keine Abkürzungen in der Vorhersage machen.⁵

Im wesentlichen gleichwertig ist die von Darley vorgeschlagene Emergenzdefinition: „a true emergent phenomenon is one for which the optimal means of prediction is simulation“ [Dar]. Während es bei einem nicht-emergenten System aus dem Anfangszustand und dem Wissen um die Übergangsregeln leicht möglich ist, einen zukünftigen Zustand hinreichend genau vorherzusagen (basierend auf einem vereinfachten Modell des Systems, das solche Vorhersagen gestattet), scheitert dies bei einem emergenten System. Als Gradmesser für die „Emergenz“ eines Systems gilt dabei der Rechenaufwand für die jeweilige Art der Vorhersage: „emergence is purely the result of a phase change in the amount of computation necessary for optimal prediction of certain phenomena“. Der Übergang zwischen nicht-emergent und emergent ist nicht sprunghaft zu sehen: je „emergenter“ das System, desto höher der Aufwand für die modellbasierte Vorhersage im Vergleich zur schrittweisen Simulation.

Um die Begriffsverwirrung in Grenzen zu halten, seien diese verwandten Konzepte von Emergenz in weiterer Folge als „simulationsrelative Emergenz“ bezeichnet.⁶

⁵„The modal terms in this definition are metaphysical, not epistemological. ... Our need to use a simulation is due neither to the current contingent state of our knowledge nor to some specifically human limitation or frailty. Although a Laplacian supercalculator would have a decisive advantage over us in simulation speed, she would still need to simulate.“ [Bed97] Es handelt sich hier übrigens auch nicht um das Problem mangelnder Darstellungsgenauigkeit von nicht-ganzen Zahlen in binärer Darstellung. In solchen Fällen kumulieren sich kleinste Darstellungs- bzw. Rundungsfehler über viele Rechenschritte zu großen Abweichungen im Endresultat. Dies ist aber nur dann von Belang, wenn ein System simuliert werden soll, das genauer „rechnet“ als das simulierende System (siehe dazu [EF94], Kapitel 2.3. sowie [PH91])

⁶Es ist interessant zu beobachten, dass Bedau und Darley ein gleichartiges Konzept von Emergenz sehr unterschiedlich wertend als schwach bzw. „truly“ emergent bezeichnen. Die in den Artikeln zu findenden Literaturreferenzen lassen einen möglichen Grund erkennen: während Bedau

3.1.3 Carianis modellrelative Emergenz

Peter Cariani [Car92] unterscheidet drei Emergenzbegriffe: computational, thermodynamic, relative to a model, wovon allerdings nur zwei genauer beschrieben werden:

Computational emergence basiert auf der ursprünglichen Definition Forrests: komplexes, nicht zentral gesteuertes Verhalten auf Makroebene entspringt lokaler Interaktion auf der Mikroebene und ist somit „emergent“ – sehr viel deutlicher geht Cariani jedoch nicht darauf ein. Er stellt jedoch klar fest, dass diese Form von Emergenz auf einer Grundannahme des Reduktionismus basiert: „micro-orders determine macro-orders but not vice versa“ – Downward Causation und Irreduzibilität sind somit ausgeschlossen; über die Vorhersagbarkeit wird keine Aussage gemacht. Seine in den Raum gestellte Frage „If we randomly come across a computer simulation and we have no clue as to its purpose, can we tell if its computations are emergent?“ (Hervorhebung im Original) deutet auf jenes Problem hin, das Clark mit seinem oben kritisierten Emergenzbegriff verdeutlicht hat: wenn Emergenz als Synonym für nicht-explicit programmiertes Verhalten gilt, hängt sie unweigerlich von der (vermuteten) Intention des Programmierers und dem eigenen Beobachtungsrahmen ab.

Emergence relative to a model ist der von Cariani für A-Life favorisierte Emergenzbegriff. Es handelt sich dabei zuvorderst um ein rein erkenntnistheoretisches Konzept: Emergenz ist die Abweichung des Verhaltens eines Systems von jenem Modell, das der Beobachter davon hat. Ihre Anwendung findet diese Variante von „Emergenz“ primär in der Beurteilung der *Adaptivität* eines Systems, d.h. in der Fähigkeit, sich selbst in Reaktion auf äußere Einflüsse zu verändern.

Zur Erläuterung wird auf eine erweiterte Hertzsche Modellierungsrelation zurückgegriffen⁷, die für „organisms and devices“ gleichermaßen gelten soll (Abbildung 3.1). Ein mit der Umwelt interagierendes formales System (beispielsweise ein Ro-

sich des Hintergrunds der philosophischen Emergenztheorien offenbar bewusst ist (vor dem sein Emergenzbegriff tatsächlich vergleichsweise schwach ist), bezieht sich Darley ausschließlich auf Publikationen aus Informatik und Physik.

⁷Rosen [Ros91] geht sehr ausführlich auf die von Cariani nur kurz angedeuteten Details dieser Modellierungsrelation ein, er benutzt dabei allerdings etwas anderes Vokabular.

Semantik fixiert. Wird das fixe Programm durch einen Lernalgorithmus (z.B. ein neuronales Netz, siehe 3.2) erweitert, liegt ein syntaktisch adaptives System vor. Ein semantisch adaptives System könnte seine Eingabe- und Ausgabemechanismen ändern, ein allgemein evolutionäres System würde dies zusätzlich mit syntaktischer Adaptivität verbinden.⁸ Wo ist hier der Emergenzbegriff verborgen? Er hängt davon ab, welche Freiheitsgrade das Modell des Beobachters vom beobachteten System hat. Geht der Beobachter von einem syntaktisch und semantisch fixierten System aus, wäre jede beobachtete Syntaxänderung emergent.

Cariani macht in der Beschreibung seiner „modellrelativen Emergenz“ einige interessante, kritische Bemerkungen zur Methodologie des A-Life und zu den dort gängigen Emergenzbegriffen. Wie bereits erwähnt, ist sein Ansatz jedoch in erster Linie der einer Adaptivitäts- oder Autonomiebeurteilung und hat mit den etablierten Emergenzbegriffen der Philosophie so gut wie nichts zu tun: hierarchische Struktur ist bestenfalls ein Seiteneffekt, Irreduzibilität von Systemeigenschaften ist gar kein Thema, Neuartigkeit nur in sehr abgeschwächter Form.⁹ Im Emergent Computing sieht er starke Emergenz (wie im vorigen Kapitel beschrieben) daher bestenfalls in der Kognition der menschlichen Betrachter: „Rather than emergent devices in their own right, these computer simulations are catalysts for emergent processes in our minds; they help us create new ways of seeing the world.“ ([Car92], S. 790)

3.1.4 Zusammenfassung

In der Literatur des Emergent Computing zeigt sich allgemein eine Neigung, den Begriff „emergent“ zwar ausgiebig zu verwenden, aber selten klar zu definieren. Die oben erwähnten Definitionsversuche sind rare Ausnahmen, die jedoch das weite Spektrum des Emergent Computing grob vorstrukturieren. Als ein Zwischenresultat kann gelten, dass die in Kapitel 2 diskutierten Aspekte Irreduzibilität, Down-

⁸Ein derartiges System wäre nicht mehr nur adaptiv, sondern auch in gewissem Sinn autonom, weil es die Kategorien seiner Beobachtung und deren Bewertung selbst wählt – es besitzt interne Pragmatik.

⁹In dieser Hinsicht trägt Cariani leider dazu bei, das ohnehin schon überfrachtete Wort „Emergenz“ noch mehrdeutiger zu machen.

ward Causation sowie Indeterminismus für diese Emergenzdefinitionen keine ernsthafte Rolle spielen. In den nächsten Abschnitten sollen vor diesem Hintergrund einige typische Beispiele aus der Praxis des Emergent Computing diskutiert werden. Diese kurze Auflistung von Beispielen ist bei weitem nicht vollständig, sie markiert aber die Eckpunkte jenes Bereichs, in dem sich Emergent Computing abspielt: praktisch alle modernen EC-Anwendungen leiten sich aus Variation und Kombination der folgenden Beispiele her.

3.2 Beispiel: Konnektionismus

Künstliche neuronale Netze sind eine der frühesten Anwendungen der konnektionistischen¹⁰ Spielart der künstlichen Intelligenz. Ihr Aufbau lehnt sich an die Strukturen des menschlichen bzw. tierischen Nervensystems an. Grundelement ist das *Neuron*, ein deutlich vereinfachtes Modell biologischer Neuronen [Bra95, Roj96].

3.2.1 Grundlagen

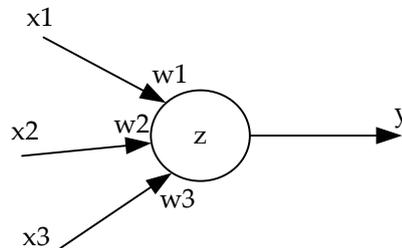


Abbildung 3.2: *Modellneuron*

Das Modellneuron verfügt über Eingänge x_i (biologisch: Dendriten), die mit Gewichten w_i versehen sind (biologisch: Synapsen) und gemäß einer Aktivitätsfunktion zur Aktivierung z führen. Im allgemeinen Fall ist diese Aktivitätsfunktion eine gewichtete Summe:

$$z(w, x) = \sum_{j=1}^n w_j x_j$$

Zu beachten ist, dass Gewichte auch negative Werte annehmen und damit die Aktivität des Neurons unterdrücken können („inhibitorische Synapsen“). Die Aktivität am Ausgang y (das „Feuern“ des Neurons) wird durch eine Ausgabefunktion S beschrieben:

¹⁰Ein sehr nützlicher, kompakter Überblick über die diversen Varianten des Konnektionismus ist in [Far91] zu finden.

$$y = S(z)$$

Typischerweise werden in der Ausgabefunktion zusätzliche Parameter wie etwa ein Schwellwert s_1 (minimale Aktivierung, um eine Ausgabe zu erzeugen) sowie ein Sättigungswert s_2 (Begrenzung der Ausgabe z) berücksichtigt. Eine beliebte Form von Aktivierungsfunktion ist die Sigmoidalfunktion, die den gewünschten Sättigungseffekt mitbringt:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Schwellwerte können ihrerseits wiederum zeitabhängig sein. Diese *time-varying thresholds* begrenzen die maximale Impulsrate am Ausgang des Neurons. Ermüdung (*fatigue*) modelliert Beschränkungen durch den Stoffwechsel biologischer Neuronen.¹¹

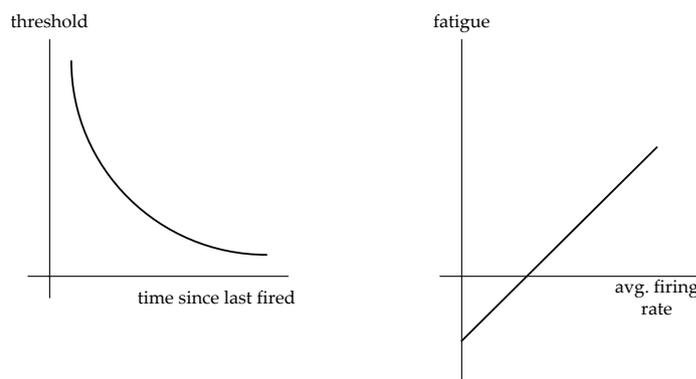


Abbildung 3.3: Zeitabhängige Schwellwerte und Ermüdung (nach [Hol97])

Werden mehrere Neuronen in Form eines gerichteten Graphen miteinander verbunden (d.h. Ausgänge von Neuronen an Eingänge anderer Neuronen angeschlossen), spricht man von einem neuronalen Netz. Ist dieser Graph zyklensfrei, gilt das

¹¹Diese Parameter machen das so einfach scheinende Modellneuron im Sinne von Foersterns[von84] zur nicht-trivialen Maschine, weil sie einen inneren Zustand einführen, der zukünftige Ausgabewerte mitbestimmt.

Netz als vorwärts gerichtet (*feed-forward*): in diesem Fall kann das Ausgangssignal eines Neurons nicht mehr in dessen Eingangssignale einfließen. Ein derartiges Netz hat eine vergleichsweise einfache Dynamik und kann als Implementation einer Serienschaltung nichtlinearer Funktionen betrachtet werden [Far91]. Andernfalls, wenn Ausgangssignale wieder rückgeführt werden, gilt das Netz als rückgekoppelt (*recurrent*). Rückgekoppelte Netze können unter anderem – im Gegensatz zu rein vorwärts gerichteten Netzen – als Informationsspeicher dienen, ähnlich elektronischen „flip-flop“-Schaltungen. In beiden Fällen ist das neuronale Netz als eine mehrfache Überlagerung (potentiell vieler) einfacher Funktionen zu verstehen, was auch als Superpositionsprinzip bezeichnet wird.

Analog zur stufenweisen Informationsverarbeitung in biologischen neuronalen Netzen werden auch künstliche typischerweise in Schichten (*layers*) unterteilt, in denen mehrere parallel arbeitende Neuronen zusammengefasst werden. Die Wahl der Netztopologie erfolgt üblicherweise nach Erfahrungswerten (da sich im Laufe der Entwicklung gewisse Netzstrukturen als besonders geeignet für bestimmte Anwendungen erwiesen haben), kann aber auch adaptiv erfolgen (etwa mit Unterstützung durch genetische Algorithmen).

Lernen

Neuronale Netze können beliebige stetige Funktionen realisieren. In der Praxis ist es jedoch bei strukturell komplexeren Netzwerken praktisch unmöglich, die zur Realisierung einer gegebenen, nicht-trivialen Funktion nötigen Parameter im vorhinein explizit zu bestimmen. Es ist daher nötig, diese Parameter schrittweise zu approximieren. Im Konnektionismus wird dieser Vorgang – etwas suggestiv¹² – als Lernen bezeichnet. Lernen ist in diesem Kontext somit als Optimierungsvorgang zu verstehen.

Im Fall des überwachten Lernens wird diese Parameteränderung exogen während einer separaten Trainingsphase erreicht: das Netz wird durch einen menschlichen Bediener so angepasst, dass die gewünschte Relation zwischen Eingangs- und Aus-

¹²Oder auch viel zu suggestiv: Lenz und Meretz argumentieren in [LM95], dass diese unkritische Begriffsverwendung mit einer unkritischen Anwendung der neuronalen Netze als universales Erklärungsmodell für psychische Vorgänge einhergeht.

gangssignalen entsteht. Ein sehr bekanntes Beispiel für einen Algorithmus, der diesen Vorgang unterstützt, ist das „Backpropagation“-Verfahren (siehe [LM95], S. 38f. und [Bra95]).¹³

Unüberwachtes Lernen soll ohne diese ständige Bedienerintervention auskommen. Beim *Hebbschen Lernen*¹⁴, einer Form von unüberwachtem Lernen, werden automatisch „erfolgreiche“ Eingänge durch Gewichtsänderung verstärkt: folgt auf eine Aktivierung des Eingangs x_i im Zeitpunkt t eine Aktivierung des Ausgangs y im Zeitpunkt $t+1$, wird das Gewicht w_i erhöht. In Erweiterungen dieses Ansatzes werden die anderen Gewichte verringert („synaptische Konkurrenz“), das Gewicht w_i mit Abklingfunktionen versehen, usw. (siehe [Bra95], Kap. 1.5).

Bei *stochastischen* neuronalen Netzen wird in Anlehnung an das „simulated annealing“-Optimierungsverfahren thermales Rauschen simuliert, indem in den Eingängen ein Zufallssignal eingestreut wird. Ziel ist auch hier, den Lernprozess schneller und sicherer zum erwünschten Optimum zu bringen (siehe [Roj96], Kap. 14).

Selbstorganisierte Merkmalsgewinnung / Self-Organising Maps

Self-organising maps, entwickelt von Kohonen [Koh84] sind eine Form neuronaler Netze, die nicht zuletzt wegen ihrer suggestiven Bezeichnung gern als Paradebeispiele des EC gesehen werden.

Ein wesentliches Element von self-organising maps (SOMs) ist die Annahme einer Nachbarschaft zwischen Neuronen der selben Schicht.¹⁵ Neuronen können sich damit innerhalb der Schicht gegenseitig aktivieren bzw. hemmen. Ziel von SOMs ist es, eine interne Repräsentation eines Eingangssignals zu erreichen, und zwar unter der Voraussetzung des Topologie-Erhalts (oder zumindest der Topologie-

¹³Die bereits erwähnte Unterscheidung zwischen vorwärtsgerichteten und rückgekoppelten Netzen bezieht sich übrigens nur auf die sogenannte Funktionsphase des Netzes, also den „Normalbetrieb“ – jene Rückkoppelung, die durch das Lernen (überwacht oder nicht) geschieht, ist davon nicht berührt.

¹⁴Dieses basiert auf den in den 1940er Jahren entwickelten neurologischen Theorien von D. Hebb [Heb49].

¹⁵Diese Nachbarschaft ist durchaus mit jener zu vergleichen, die in zellulären Automaten existiert (siehe 3.4), wobei es bei letzteren aber kein exogenes Eingangssignal gibt.

Approximation): benachbarte Punkte des Eingabemusters werden auch auf benachbarte Neuronen abgebildet, obwohl es gleichzeitig zu einer Dimensionsreduktion kommen kann. Die sogenannte selbstorganisierende Eigenschaft von SOMs besteht nun darin, dass die Gewichtsvektoren des Netzes durch geeignete Parametrisierung automatisch – also unüberwacht, ohne Bedienerintervention – die gewünschte Abbildung des Eingangsmusters erreichen.¹⁶ Jedes Neuron soll sich dabei möglichst auf eine bestimmte Region der Eingabedaten spezialisieren, was durch den von Kohonen entwickelten Lernalgorithmus sichergestellt wird. Ein Beispiel ist in Abbildung 3.4 zu sehen: die anfangs in einem kleinen Bereich um den Mittelpunkt verteilten Gewichtsvektoren passen sich sukzessive den gleichmäßig über den Musterraum verteilten Eingangsmustern an. Dieses „Entfalten“ wird durch die über die Nachbarschaft der Neuronen festgelegten Zusammenhänge vorangetrieben.

SOMs haben biologische Vorbilder: so ist die topologieerhaltende Dimensionsreduktion von Eingangssignalen eine der grundlegenden Aufgaben, die in der Signalverarbeitung des Gehirns gelöst werden müssen (vgl. [Roj96], 15.1). Technische Anwendungen von SOMs sind sehr weit gefächert, z.B. in der automatischen Inhaltsklassifikation von Texten, in der inversen Kinematik (zur Steuerung von Roboterarmen) u.a.

Boolean Networks

Boolean Networks sind eine in den Grundelementen radikal vereinfachte Version konnektionistischer Netze, die jedoch sehr starke Ähnlichkeiten mit zellulären Au-

¹⁶Fort und Pagès [FP96] leiten aus dieser Eigenschaft eine Definition von Selbstorganisation ab. „Starke“ Selbstorganisation sei dann gegeben, wenn die Menge \mathcal{O} organisierter Zustände absorbierend ist (d.h., sobald der Prozess \mathcal{O} erreicht, wird diese Menge nicht mehr verlassen) und die Zeit zwischen unorganisiertem Ausgangszustand und Eintreffen in \mathcal{O} „nahezu sicher endlich“ ist. Es ist höchst fraglich, ob selbst dieser „starke“ Selbstorganisationsbegriff mit dem in [Hof01] beschriebenen in Einklang zu bringen ist, allein zumal Kausalität und Determinismus gar nicht zur Diskussion gestellt werden. Rojas [Roj96] unterscheidet selbstorganisierende von nicht-selbstorganisierenden Netzen allgemein so, dass bei ersteren der korrekte Output nicht a priori definiert werden kann (und daher keine Quantifizierung einer Abweichung von einem Lernziel möglich ist).

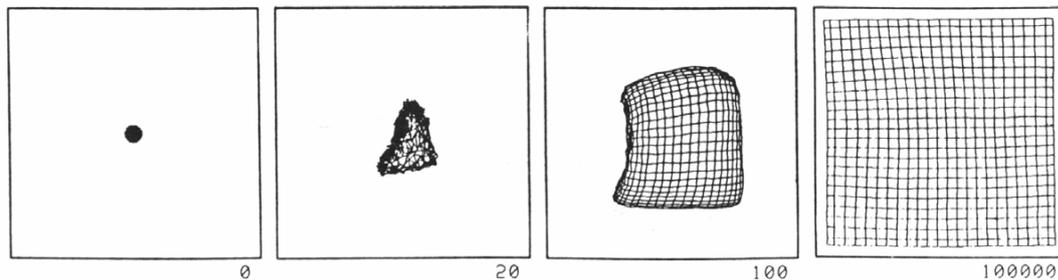


Abbildung 3.4: Selbstorganisation einer SOM. Die rechteckige Fläche stellt den Musterraum dar, Gewichtsvektoren benachbarter Neuronen sind durch Linien verbunden (aus [Koh84])

tomaten aufweisen.¹⁷ Die historischen Wurzeln der Boolean Networks lassen sich bis zu Alan Turings Frühwerk zurückverfolgen (siehe dazu Teuschers ausführliche Darstellung in [Teu02]), an dieser Stelle soll aber exemplarisch die aktuellere Forschung von Stuart Kauffman herausgegriffen werden [Kau93]. Wie bei den meisten Verfahren des Emergent Computing gibt es bei den Netzwerken von Kauffman einen biologischen Hintergrund, und zwar die Frage des Zusammenhangs zwischen Genotyp (DNA) und Phänotyp: wie ergibt sich aus gegebenem Genotyp ein bestimmter Phänotyp? Wie differenzieren sich in der Entwicklung eines Organismus verschiedene Zelltypen; wie wird dieser Vorgang gesteuert?

Die DNA kann – der Informatik sehr entgegenkommend – als endlich langer String über einem endlichen Alphabet kodiert werden. Es ist weiters bekannt, dass zwischen den einzelnen Symbolen dieses Strings Zusammenhänge herrschen; die Aktivität eines Gens ist somit von der anderer Gene abhängig.¹⁸ Kauffman vereinfacht nun das Gen-Alphabet auf nur zwei Zustände und modelliert die Zusammenhänge zwischen den Genen als logische Operatoren wie AND und OR:

Boolean networks are made up of binary, on-off variables. A network

¹⁷Die Einordnung wäre daher auch bei den zellulären Automaten möglich, hier soll jedoch der in der Literatur üblicheren Einteilung gefolgt werden.

¹⁸... und von einer Reihe anderer Faktoren [Fut90].

has N such variables. Each variable is regulated by some of the variables in the network, which serve as its inputs. The *dynamical behavior* of each variable, whether it will be active (1) or inactive (0) at the next moment, is governed by a *logical switching rule or Boolean function*. . . . Let K stand for the number of input variables regulating a given binary element. ([Kau93], S. 188, Hervorhebung im Original)

Ein sehr einfaches Boolean Network ist in Abbildung 3.5 zu sehen. Aus der Kombination der Wahrheitstabellen für die einzelnen Knoten des Netzwerkes lässt sich eine Tabelle ermitteln, die für jeden Zustand der Knoten den nächsten Zustand determiniert. Für das gegebene Netz ist etwa der Nachfolgezustand von (0, 0, 0) wiederum (0, 0, 0): aus diesem Zustand gibt es kein Entkommen mehr, er ist daher auch als einfacher Attraktor zu verstehen.

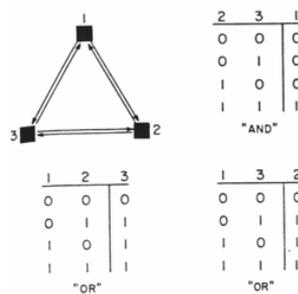


Abbildung 3.5: Ein einfaches Boolean Network mit $N = 3$ und $K = 2$. Die Wahrheitstabellen geben den jeweils nächsten Zustand des Knotens an, der vom Zustand der anderen Knoten abhängt. Aus [Kau93].

In komplizierten Netzen können sich sogenannte zyklische Attraktoren ergeben: jeder Zustand im Einzugsbereich des Attraktors führt in einen Zyklus von Zuständen, der danach nicht mehr verlassen wird (siehe Abbildung 3.6).

Für Kauffman sind nun diese Attraktoren die Basis der verschiedenen Zelltypen, die sich aus einem genetischen Code ergeben: „a cell type is an attractor of the genetic regulatory system“. Tatsächlich ist die Anzahl der Attraktoren, die in einem

bundenen Neuronen, das optische Eingangssignale über eine „Retina“ empfängt. Zusätzlich wird ein kontrastsuchender Reflex angenommen, der den Fokus der Retina auf starke Kontraste (wie etwa Ecken und Kanten) lenkt, solange die Aktivität im neuronalen Netz unter einer gewissen Schwelle liegt. Im Beispiel wird davon ausgegangen, dass ein kontrastreiches Dreieck auf die Retina projiziert wird.

Synchronisation Die Anwendung des Hebbschen Lernens in einem rückgekoppelten Netz führt dazu, dass Rückkopplungen verstärkt werden (positive Rückkopplung). Liegt der Fokus des Systems auf einer Ecke des Dreiecks, beginnen die mit dieser Region der Retina verbundenen Neuronen sich aufzuschwingen: sowohl die Gewichte der mit der Retina verbundenen Synapsen, als auch die Gewichte der Verbindungen zwischen den beteiligten Neuronen werden verstärkt. Alle anderen Verbindungen werden gleichzeitig geschwächt, wodurch sich eine abgegrenzte, zusammenhängende Region von Neuronenaktivität herausbildet. In der Terminologie von Hebb ist dies ein *cell assembly*, das bereits auf geringe Reize hin synchron zu schwingen beginnt. Diese Schwingung wird jedoch durch die Ermüdung (*fatigue*) bedämpft, wodurch die Aktivität nach einiger Zeit wieder abebbt.

Ist die Aktivität des *cell assembly* abgeklungen, führt der kontrastsuchende Reflex den Fokus der Retina auf die nächste (benachbarte) Ecke des Dreiecks. Wiederum kommt es zur Bildung eines *cell assembly*, das sich jedoch nicht mit dem vorherigen *cell assembly* überschneiden kann, weil die Neuronen in dessen Bereich noch stark ermüdet sind und aufgrund dieser „Erfolglosigkeit“ durch die Hebbsche Regel weiter geschwächt werden. Ein Neuron kann somit nicht beiden *cell assemblies* zugeordnet sein, sondern nur einem. Das neuronale Netz hat auf diese Weise klar getrennte Repräsentationen zweier Ecken des Dreiecks erlernt. Die Erweiterung auf die dritte Ecke funktioniert analog, worauf dann wieder auf eine der anderen Ecken fokussiert wird, usf.: es entsteht ein stetiger Abtastzyklus, der für jede Ecke des Dreiecks zum Auslösen ganz spezifischer Neuronenaktivität führt.

Antizipation Da das gerade aktive *cell assembly* die Aktivität in den benachbarten *assemblies* aktiv unterdrückt, „erholen“ sich diese Zellen von der Ermüdung und werden dadurch im Vergleich zu anderen Regionen des Netzes (die an

keinem der assemblies beteiligt sind) besonders empfindlich. Nach Holland werden diese assemblies damit auf das Feuern vorbereitet; das Netz *antizipiert* somit die nächste Ecke. Diese antrainierte regionale Empfindlichkeit kann dazu führen, dass auf eine im Vergleich zu den anderen Ecken wesentlich kontrastschwächere Ecke trotzdem reagiert wird, wenn sie im Bereich eines bereits etablierten cell assembly liegt (Interpolation).

Hierarchiebildung Innerhalb der zufällig verbundenen Neuronen des Netzes können einige existieren, deren Verbindungen geeignet sind, um gemeinsam auf die oben skizzierte Dreiersequenz abwechselnd aktivierter cell assemblies zu reagieren. In diesem Fall bildet sich ein cell assembly, das die Aktivitäten der drei als „Eckendetektoren“ wirkenden assemblies zusammenfasst und somit selbst zum „Dreiecks-Detektor“ wird.

Die in diesem Gedankenexperiment¹⁹ skizzierten „emergenten Phänomene“ finden sich in vielen kleineren, aber formaler beschriebenen Beispielen von unüberwachtem Lernen bzw. selbstorganisierter Merkmalsgewinnung in neuronalen Netzen, z.B. in den schon erwähnten *self-organising maps*.

Was, wenn überhaupt, emergiert in neuronalen Netzen? Es ist unbestreitbar, dass das oben skizzierte Beispiel systemische Eigenschaften besitzt: ein einzelnes Neuron erkennt keine Dreiecke, das Gesamtsystem schon. Diese Eigenschaften sind aber vollständig auf die Eigenschaften der einzelnen Elemente des Systems reduzierbar: das Irreduzibilitätskriterium spielt keine Rolle.

Weiters ist festzustellen, dass sich ohne menschliche Intervention eine Strukturbildung vollzieht: am Beginn steht ein zufällig verbundenes Netz; nach der Trainingsphase finden wir ein Netz vor, in dem sich aus dem Gewirr von Einzelneuronen deutlich abgegrenzte, zusammenhängende Regionen entwickelt haben, die in erkennbarer Beziehung zum Eingangssignal und sogar in hierarchischer Beziehung untereinander stehen. Dieser Fall könnte als ein Beispiel für die von Levins [Lev73] gezeigte Neigung komplexer Systeme zur Hierarchiebildung und Verringerung der internen Konnektivität gelten. Eine der Voraussetzungen für diese Strukturierung

¹⁹Obwohl es sich um ein „Gedankenexperiment“ handelt, gibt es keine grundsätzlichen Hindernisse, dieses System zu implementieren. Einzig die Tatsache, dass eine große Zahl von Neuronen vorausgesetzt wird, macht ein solches Experiment sehr rechenaufwendig.

ist die Existenz eines Rückkopplungsmechanismus – in diesem Fall zum einen das Hebb'sche Lernen (das Rückkopplung auf unterster Ebene zwischen einzelnen Neuronen darstellt), zum anderen der als gegeben angenommene kontrast-suchende Reflex (der von der Gesamtaktivität des Netzes abhängig ist). Ein rein vorwärtsgerichtetes Netz, egal ob mehrstufig oder nicht, ist im wesentlichen eine fixe Transformationsmaschine, die keine derartige Strukturierung vollziehen kann. Trotzdem ist die Strukturbildung deterministisch, selbst wenn es sich um ein stochastisches Netz handelt (das „Rauschen“ ist in diesem Fall einfach ein weiterer Input, der deterministisch verarbeitet wird). Weiters ist sie prinzipiell vorhersagbar: für jeden Eingangswert ist der nächste Schritt determiniert. Fraglich ist lediglich, ob die Vorhersage nur durch Simulation möglich ist. Auf die meisten in der angewandten Informatik eingesetzten Varianten neuronaler Netze dürfte dies nicht zutreffen (diese Modelle sind oft auf bestimmte Anwendungen maßgeschneidert, ihr Verhalten daher gut abschätzbar), obwohl sich chaotisches Verhalten durch geeignete Struktur und Parameterwahl sehr wohl erreichen lässt (siehe [Bra95]).

Boolean Networks sind in der beschriebenen Form durch ihre starre Struktur und enge Kopplung keine Kandidaten für einen Emergenzbegriff, der über den im letzten Absatz skizzierten hinausgeht. Kauffman spricht zwar häufig von „spontaner Ordnung“, was jedoch in einem eher metaphorischen Sinn zu verstehen ist, zumal die „Attraktoren“ eines solchen Netzes durch die Wahrheitstabelle im vorhinein festgelegt sind. „Emergent“ ist das Verhalten dieser Netze nur in dem Sinn, dass es verblüffend komplex und dabei geordnet erscheinen kann.²⁰

²⁰ „ $K = 2$ nets crystallize spontaneous order. Random Boolean networks with $K = 2$ inputs exhibit unexpected and powerful collective spontaneous order“ ([Kau91], Hervorhebung hinzugefügt)

3.3 Beispiel: Evolutionäre Verfahren

Evolutionäre Verfahren in der Informatik orientieren sich an den Grundprinzipien der Evolutionstheorie: Selektion und Mutation. Aus diesem relativ weiten Feld sollen hier die zwei bekanntesten Verfahren herausgegriffen werden: Genetische Algorithmen und Classifier Systems.

3.3.1 Genetische Algorithmen

Genetische Algorithmen (GA, in den 1970er Jahren von Holland entwickelt [Hol92]) dienen vor allem der Lösung nichtlinearer (Optimierungs-)Probleme. Die Grundidee ist eine adaptive Suche, ein sich ständig wiederholender Prozess von Informationsfindung und -auswertung. Analog zu den Grundprinzipien der biologischen Evolution werden Selektion, Genkombination und Mutation eingesetzt, um optimales bzw. neuartiges Verhalten zu generieren. Ein entscheidendes Merkmal von GA (im Gegensatz zu früheren auf Mutation und Selektion basierenden Optimierungsansätzen) ist die Existenz ganzer Populationen von Lösungen. Statt nur eine einzige Lösung weiterzuentwickeln, werden viele verschiedene Lösungen über den Suchraum gestreut, um durch Anwendung evolutionärer Mechanismen die bestmögliche Lösung „heranzuzüchten“. Der wesentliche Vorteil dieses Vorgehens ist, dass der GA weniger leicht in lokalen Optima stecken bleibt als andere Verfahren, die immer nur eine einzige Lösung weiterentwickeln.

Genetische Algorithmen werden im Emergent Computing häufig als Zusatz zu anderen Verfahren wie neuronalen Netzen oder zellulären Automaten verwendet, um gewisse Parameter (wie z.B. die Netztopologie) zu optimieren.

Grundlagen

Das Vorgehensmodell der Genetischen Algorithmen lässt sich am besten als Pseudocode beschreiben:²¹

²¹Die folgende Darstellung orientiert sich an jenen von Schöneburg et al. [SHF94] und Teuscher [Teu02]

1. Wähle eine geeignete genetische Kodierung für das „Genom“
2. Initialisiere zufällig eine Population von Genomen (Generation 0)
3. Evaluation: bewerte alle Elemente der aktuellen Generation gemäß einer Fitnessfunktion
4. Crossover: selektiere Paare aus der Population und erzeuge durch Crossover Nachkommen der aktuellen Generation
5. Mutation: mutiere die Nachkommen
6. Generationenwechsel: ersetze Elemente der aktuellen Generation durch Nachkommen, evt. gemäß einem Ersetzungsschema
7. Falls die Abbruchbedingung noch nicht erreicht ist, gehe zu Schritt 3.

Jeder einzelne Schritt wirft einige Fragen für den Entwurf eines GA auf, so zum Beispiel: was wird im Genom kodiert, und in welcher Form? Wie groß soll die Population sein? Nach welchen Kriterien soll die Fitness von Genomen beurteilt werden? Wie sollen Crossover und Mutation erfolgen? Was soll die Abbruchbedingung sein?

Die erste Frage beschäftigt sich mit Inhalt und Repräsentation des Genoms. Der Inhalt des Genoms ist typischerweise die zu optimierende Größe – dies könnte der Wert irgendeiner reellen Funktion, aber auch eine Menge von möglichen Entscheidungen eines Individuums sein. Die Repräsentation dieser Größe erfolgt typischerweise in Form eines binären Strings fixer Länge, wobei allerdings die Frage der Kodierung eine Reihe von Problemen aufwirft, die die Konvergenz des GA beeinflussen (siehe [SHF94], S. 189f.).

Bei der Evaluation stellt sich die Frage nach der Fitnessfunktion, gemäß der die Genome beurteilt werden sollen. In Optimierungsproblemen entspricht dies der jeweiligen Zielfunktion (in der Ökonomie z.B. eine Nutzenfunktion).

Beim Crossover geht es darum, wie die gemäß der Fitnessfunktion selektierten Paare von Genomen miteinander gekreuzt werden sollen. Das Grundprinzip: Bits an bestimmten Positionen werden zwischen den beiden Genomen vertauscht. Im einfachsten Fall, dem sogenannten Einpunkt-Crossover, werden einfach alle Bits ab

einer bestimmten Position im String ausgetauscht. Im Zweipunkt-Crossover wird dagegen ein bestimmter Abschnitt (von Position p_a bis Position p_b) ausgetauscht. Das universellste Verfahren ist das mittels einer zufällig generierten Crossover-Maske, die die auszutauschenden Bits festlegt. Die Wahl eines geeigneten Crossover-Mechanismus bestimmt entscheidend, wie gut der Suchraum abgedeckt wird.

Mutation erfolgt für jedes Genom einzeln: je nach Methode werden entweder einzelne Bits oder ganze Sequenzen mit einer gewissen Wahrscheinlichkeit invertiert. Mutationen sind besonders wichtig, um die allzu rasche Konvergenz zu lokalen Optima zu vermeiden.

Im Generationenwechsel wird entschieden, wie die nächste Generation von Genomen zusammengesetzt ist. Im einfachsten Fall werden einfach alle alten Genome ersetzt, in verfeinerten Varianten wird z.B. beurteilt, ob die neuen Genome überhaupt höhere Fitness als die alten aufweisen – wenn nicht, ist die Ersetzung möglicherweise nicht sinnvoll.

3.3.2 Classifier Systems

Classifier Systems (CS, oft auch als „Learning Classifier Systems“ bezeichnet) wurden wie die Genetischen Algorithmen von Holland [Hol92] entwickelt. Sie können als Systeme gesehen werden, in dem GAs eine tragende Rolle spielen, weisen aber auch gewisse Ähnlichkeiten mit Boolean Networks auf [Far91]. CS werden in allen Bereichen verwendet, in denen möglichst selbstständig „lernende“²² Systeme angestrebt werden.²³

Grundlagen

Holland erklärt seine Motivation zur Entwicklung der Classifier Systems mit einer Frage: „How does a system improve its performance in a perpetually novel envi-

²²Lernen ist hier, wie bei den neuronalen Netzen, in erster Linie als Optimierungsprozess zu verstehen.

²³Siehe [LSW00] für einen Überblick über die Forschung der 1990er Jahre. Eine der außergewöhnlichsten Anwendungen ist wohl jene von Smith et al.: ein relativ einfaches CS wurde erfolgreich eingesetzt, um neuartige Flugmanöver in Luftkampfsituationen zu entwickeln [SDR⁺00].

ronment where overt ratings of performance are only rarely available?" ([Hol92], S. 172). Ein CS ist also als ein mit seiner Umwelt interagierendes System zu verstehen, das unüberwacht lernen soll, ohne genaue Informationen zu haben, *wie* es lernen soll. Es bekommt nur vage Bewertungen wie „gut“ oder „schlecht“, diese möglicherweise stark zeitverzögert (erst nach mehreren Aktionen stellt sich heraus, ob der eingeschlagene Weg sinnvoll war) und muss daraus selbst geeignete Adaptionentscheidungen treffen.

Die Verbindung zur Umwelt geschieht im CS über *detectors* und *effectors* (siehe Abb. 3.7), sowie über *payoffs*, über die die Beurteilung des CS-Verhaltens einlangt. Die Detektoren erzeugen standardisierte Informationspakete (*messages*), wie sie im CS an mehreren Stellen eingesetzt werden. Messages sind binäre Strings fixer Länge (z.B. 011101 für die Länge $k = 6$) und landen stets auf der *message list*.

Mögliche Aktionen des Systems werden durch die *classifier* festgelegt. Jeder Classifier ist ein Paar aus Bedingung (*condition part*) und Aktion (*action part*). Die Bedingung legt fest, welche Messages zum Auslösen der Aktion führen; die Aktion besteht darin, wiederum eine bestimmte Message an die Liste oder an die Effectors zu senden. Die Messages sind somit ein Kommunikationsmittel, das das System nicht nur für die Interaktion mit der Umwelt, sondern auch für interne Kommunikation zwischen Classifiern verwendet.

Die beiden Teile eines Classifiers sind Strings fixer Länge, wobei der Bedingungs- teil ein ternäres Alphabet erlaubt: zu den Symbolen 0 und 1 kommt das Symbol # hinzu, das eine Position des Strings als „beliebig“ kennzeichnet. So trifft z.B. eine Bedingung 11##### auf alle Messages zu, die mit 11 beginnen – die Positionen danach können beliebige Werte haben. Der fixe Teil der Bedingung (11) wird als *tag* bezeichnet, d.h. Nachrichten mit dieser Markierung werden durch ganz bestimmte Classifier abgedeckt. Je mehr #-Zeichen eine Bedingung enthält, desto unspezifischer ist sie. Im Standardmodell von Holland hat jeder Classifier zwei Bedingungen, die durch ein logisches UND verknüpft werden. Die Bedingungen können auch logisch negiert werden (d.h. die Bedingung ist nur erfüllt, wenn keine Message auf der Message List dem String entspricht).

Die Ausführung eines Classifier System erfolgt durch Wiederholung der folgenden Schritte:

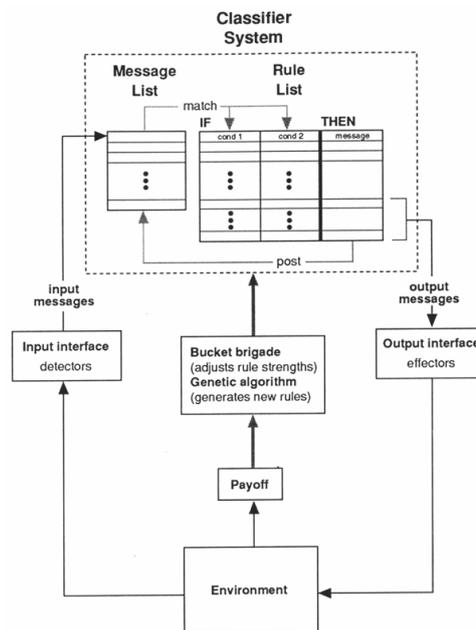


Abbildung 3.7: Aufbau eines Classifier System (aus [Hol92])

1. Messages vom Detektor werden auf die Message List gelegt.
2. Jede Bedingung jedes Classifiers wird mit allen Messages auf der Liste verglichen, um passende Classifier (d.h. solche, die für wenigstens eine Message erfüllt sind) zu finden.
3. Diejenigen Classifier, bei denen beide Bedingungen erfüllt wurden, dürfen an einem Wettbewerb (*competition*) teilnehmen.
4. Diejenigen Classifier, die den Wettbewerb gewonnen haben, dürfen ihre Messages absenden (an die Message List oder an die Effectors).
5. Die Messages des letzten Zyklus werden von der Liste gelöscht.

Der – nach einem idealisierten Auktionsprinzip durchgeführte – Wettbewerb der Classifier basiert unter anderem auf einer Bewertung (*strength*), die den bisherigen

Nutzen jedes Classifiers darstellt: so werden bewährt erfolgreiche Verhaltensregeln bevorzugt ausgeführt. Die Nutzenbewertung und „Belohnung“ erfolgreicher Classifier erfolgt nach einem Bewertungsalgorithmus (*bucket brigade*), der berücksichtigen muss, dass der Erfolg der Ausführung einer Regel oft erst viel später durch ein Payoff-Signal bestätigt wird. Erfolglose Classifier werden mit der Zeit durch neu generierte Classifier ersetzt: hier kommt der Genetische Algorithmus zum Zug.

3.3.3 Emergenz?

Es ist schwierig, in Genetischen Algorithmen sinnvolle Anwendungen des Begriffs „Emergenz“ zu finden. Genetische Algorithmen sind ein Verfahren, um einen gegebenen Raum von Lösungsmöglichkeiten zu durchsuchen und darin existierende Lösungen zu finden, die einer Optimierungsaufgabe möglichst gut entsprechen. Evident ist, dass sie für diesen Zweck ausgesprochen nützlich sind (vgl. die Beispiele in [SHF94]), und dass die gefundenen Lösungen keineswegs immer offensichtlich oder praktisch vorhersehbar sind. Eine interessante Erscheinung ist, dass sich im Laufe der simulierten Evolution sogenannte „building blocks“ herausbilden: das sind kompakte Teillösungen, die Crossover und Selektion gut überstehen und zusammengesetzt eine (hinreichend) optimale Lösung ergeben können (eine Konsequenz des Schemata-Theorems von Holland, siehe [Hol92]). Dies ändert jedoch nichts an der Tatsache, dass sich hier keine Ansätze von Irreduzibilität, Indeterminismus oder prinzipieller Unvorhersagbarkeit orten lassen. Die erfolgreichen Teillösungen sind nur in dem sehr schwachen Sinn emergent, dass sie nicht explizit vom menschlichen Experimentleiter vorgegeben wurden.

In Classifier Systems orten Forrest und Miller viele „emergente Verhaltensweisen“; auf die folgenden drei „Aspekte emergenten Verhaltens“ gehen sie genauer ein [FM91]:

Symbolisches Schließen „Of central importance to the theme of emergent computation is understanding how high-level structures can emerge in learning classifier systems and come to have a selective advantage over direct input/output maps“ ([FM91], S. 214). Mit solchen Strukturen höherer Ebene ist im Prinzip das gemeint, was bei den neuronalen Netzen als Hierarchiebildung bezeichnet

net wurde: es bilden sich durch Koadaption Einheiten (Regeln bzw. Mengen von Regeln) aus, die nicht mehr direkt zwischen Ein- und Ausgabe vermitteln, sondern sich auf andere Einheiten beziehen. Hierbei handelt es sich um eine neue Systemeigenschaft (die Regeln werden zu Foersterschen nicht-trivialen Maschinen [von84]), die aber weder indeterministisch noch irreduzibel ist. Forrest und Miller weisen lediglich auf die Schwierigkeiten hin, solches Verhalten gezielt zu erreichen – und damit auf eine *praktische*, aber keine prinzipielle Unvorhersagbarkeit.

Lernen Hier geht es vor allem um die Interaktion mit der – möglicherweise höchst komplexen – Außenwelt, die beim Lernen notwendig wird. Die Autoren sehen dabei jedoch Emergenz nicht im Kontext des Gesamtsystems aus Classifier System und Umwelt, sondern im Classifier System selbst: auch in diesem Fall kann es sich daher nur um praktische Unvorhersagbarkeit handeln.

Dynamische Eigenschaften Während die einzelnen Bestandteile von Classifier Systems gut verstanden werden, gibt es noch viel Unklarheiten über das Verhalten des Gesamtsystems: „there is little solid understanding of why some classifier system designs are successful at solving a given problem and others are not“ ([FM91], S. 217). Diese Probleme könnten, so die Autoren, durch die Übernahme von Konzepten aus der Theorie dynamischer nichtlinearer Systeme (Attraktoren, Statuszyklen) besser behandelt werden.

Ob dies zutrifft, soll hier nicht beurteilt werden. Für die Emergenzdiskussion ist entscheidend, dass es hier um Systemeigenschaften geht, die aus dem deterministischen Zusammenspiel von bekannten Komponenten resultieren – Irreduzibilität und Indeterminismus stehen nicht zur Debatte. Einzig das Kriterium der Unvorhersagbarkeit könnte hier zum Zug kommen, jedoch nur in der Variante der simulationsrelativen Emergenz (vgl. 3.1.2).

3.4 Beispiel: Zelluläre Automaten

Zelluläre Automaten (cellular automata, kurz CA) sind diejenige Technik des Emergent Computing, die dessen Grundprinzipien in der reinsten Form implementieren. Ihr besonderer Reiz liegt neben der einfachen Realisierung wohl nicht zuletzt daran, dass sie besonders eindrucksvoll zu visualisieren sind. CA sind daher trotz – oder gerade wegen – ihrer relativen Einfachheit auch nach Jahrzehnten der Forschung immer noch eines der beliebtesten Beispiele für Emergent Computing.

3.4.1 Grundlagen

Der zelluläre Automat ist ein vereinfachtes, aber vom Grundprinzip her abbildungsgetreues Modell des physikalischen Reduktionismus. Bei der Realisierung am Computer stellt jener eine abgeschlossene Simulationsumgebung dar, wie sie auch ein Laplacescher Dämon haben müsste: der Ausgangszustand und die „Naturgesetze“ (in diesem Fall die Übergangsfunktionen) sind bekannt, somit kann jeder zukünftige Zustand durch Iteration hergestellt werden.

Zelluläre Automaten bestehen aus einem n -dimensionalen Feld diskreter „Zellen“ (Partikel), die als endliche Automaten betrachtet werden können. Jede Zelle kann einen von endlich vielen (typischerweise zwei) Zuständen annehmen. Ein eindimensionaler CA mit zwei Zuständen (0, 1) kann als unendliches Band oder auch als Ring betrachtet werden:

. . . 0 1 0 1 0 1 1 1 1 0 0 1 0 1 0 1 0 . . .

Für jede Zelle c gilt eine Übergangsfunktion, die die Zustände der „benachbarten“ Zellen auf den Zustand der Zelle c abbildet. Unter Nachbarschaft werden all jene Zellen verstanden, die in einer bestimmten, relativ zu c definierten und endlich großen Umgebung liegen.²⁴ Diese Übergangsfunktion wird für jede Zelle wiederholt berechnet, womit die Dimension *Zeit* eingeführt wird – diese läuft in diskreten Schritten ab. Die Übergangsfunktion legt somit den Zustand von c in $t + 1$ fest,

²⁴Typischerweise wird auch die Zelle c selbst als Teil der Nachbarschaft angesehen.

wobei die Zustände der benachbarten Zellen im Zeitpunkt t die Parameter dieser Funktion sind.

In einem Beispiel sei die Nachbarschaft N definiert als die Zelle c_l (links von c), die Zelle c selbst sowie die Zelle c_r (rechts von c). Eine mögliche Übergangsfunktion lässt sich damit als Wahrheitstabelle angeben:

Tabelle 3.1: Wahrheitstabelle für die Übergangsfunktion eines 1-dimensionalen CA

$c_l(t)$	$c(t)$	$c_r(t)$	$c(t+1)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Der typische Ablauf eines CA-Experiments folgt den Schritten:

1. Festlegen der Parameter (Dimensionen des Feldes, Anzahl möglicher Zustände jeder Zelle, Übergangsfunktionen)
2. Setzen des Ausgangszustandes: typischerweise werden hierbei die Zellen (zumindest teilweise) zufällig initialisiert, in der englischsprachigen Literatur anschaulich *seeding* genannt.
3. Ausführung: Die Ausführung besteht schlicht darin, in endloser Wiederholung in jeder einzelnen Periode für jede einzelne Zelle die Übergangsfunktion zu berechnen und dementsprechend den neuen Zustand zu setzen.
4. Abbruch: der Zeitpunkt des Abbruchs wird vom Experimentator grundsätzlich willkürlich bestimmt, kann aber in einigen Fällen automatisch ermittelt werden. So gibt es in CAs (mindestens) einen Gesamtzustand, der ein End- oder Haltezustand ist; d.h. aus dem heraus kein anderer Gesamtzustand mehr erreicht werden kann – weiteres Ausführen des CA ist dann sinnlos.

Reversible und irreversible CA

CA lassen sich in *reversible* und *irreversible* zelluläre Automaten unterteilen. Eine grundlegende Anforderung an alle CA ist, wie bereits erwähnt, dass zu jedem Zeitpunkt t der Zustand im Zeitpunkt $t + 1$ (und somit, rekursiv angewandt, *jeder* zukünftige Zustand) deterministisch bestimmt ist. Daraus folgt jedoch noch nicht, dass jeder *vorherige* Zustand bestimmbar ist: in vielen CA bestimmt eine arithmetische Summe – und somit eine irreversible mathematische Operation – den nächsten Zustand. Typisch ist etwa eine Abhängigkeit von der Anzahl (Summe) jener Nachbarzellen, die einen bestimmten Zustand haben. In diesem Fall gibt es mehrere mögliche Zustände in t , die zum gleichen Zustand in $t + 1$ führen können: in anderen Worten, es gibt Attraktoren.

Daraus ergibt sich ein wesentlicher qualitativer Unterschied zwischen reversiblen und irreversiblen CA: eine „Selbststrukturierung“ oder „Selbstordnung“ ist in reversiblen Varianten nicht möglich; verschiedene Zustände bzw. Zustandsketten (Trajektorien) können nie konvergieren.²⁵ Im Zusammenhang mit Emergent Computing werden daher im allgemeinen nur irreversible CA erwähnt.

Metriken: Wolframs Klassen und der Lambda-Parameter

CA zeigen nicht immer interessantes Verhalten: so ebbt in vielen Fällen die Aktivität auf dem Feld sehr rasch ab oder geht in lokale Oszillationen über, in anderen Fällen wiederum wirkt das Verhalten unstrukturiert und chaotisch „verrauscht“. Wolfram [Wol84] schlug eine qualitative Klassifikation zur Bewertung dieser unterschiedlichen Entwicklungen vor: von Klasse I (Entwicklung zu einem homogenen, statischen Endzustand - „quiescence“) bis zur interessantesten Klasse IV (komplexe, teils langlebige Muster lokalisierter Strukturen).

Eine quantitative Beurteilung der Dynamik von CAs erlaubt der von Langton entwickelte λ -Parameter. Langton zeigte, dass die von Wolfram vorgeschlagenen Klassen in verschiedenen Wertebereichen von λ zu finden sind [Lan91]. Klasse-IV-CAs finden sich dabei an einem kritischen Phasenübergang zwischen zwei Bereichen: unterhalb dieses Punkts im λ -Raum sind jene CAs einzuordnen, die gegen einen

²⁵vgl. [EF94], Kap 2.2., 3: „Irreversibilität ist [eine] Voraussetzung für Selbstorganisation“

Ruhezustand oder unendliche lokale Oszillationen streben (Klasse I und II). Darüber befinden sich jene CAs, die sich chaotisch entwickeln (Klasse III).²⁶

3.4.2 Emergenz?

Die Beliebtheit der CA in der wissenschaftlichen und besonders der populärwissenschaftlichen Literatur dürfte nicht zuletzt an den Visualisierungsmöglichkeiten liegen.

So wird von „gliders“, „glider cannons“, „pulsars“ und ähnlichen Mustern in Life (einem von John Conway entwickelten zweidimensionalen CA) gesprochen, als wären sie autonom agierende Objekte [Gar83]. Der „glider“ ist eine Formation, die sich auf dem zweidimensionalen Feld stetig in eine Richtung weiterbewegt und dabei (sofern ihr Weg nicht Hindernisse wie 1-Zellen kreuzt) ihre Form zyklisch ändert (Abbildung 3.8).

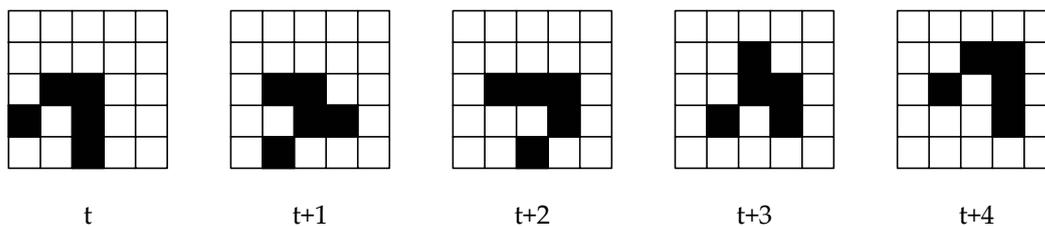


Abbildung 3.8: *Bewegungszyklus einer glider-Formation in Life*

Russell Standish sieht dieses Phänomen als emergent an. Nach seiner Definition ist dann von Emergenz zu sprechen, wenn ein Phänomen durch eine Beschreibung auf Makroebene erfasst werden kann, nicht jedoch durch eine Beschreibung auf Mikroebene (im Fall des CA wäre das die Ebene der einzelnen Zellen):

A glider can appear at any location within the CA, and may have one of four possible orientations. The description cannot represent the fact that

²⁶Die genaue Zuordnung zu den Klassen I, II oder III gilt jedoch als formal unentscheidbar [Dar].

two gliders separated diagonally by 1 cell in along each axis with the same orientation are temporally related. A glider, as an object-in-itself, is a pure macrodescription object. [Sta01]

Standish zielt damit auf den zyklischen, ständig verändernden Charakter des Musters „glider“ ab: das so beschriebene Muster ist in jedem t verschieden von $t+i$, für jedes i , das kein Vielfaches von 4 ist. Dass die Beschreibung auf der Ebene einzelner Zellen keinen Zusammenhang zwischen *Zellformationen* wiedergeben kann, ist nun zum einen trivial. Zum anderen lässt sich sehr wohl auf Mikroebene (wenn wir darunter mehr als nur eine einzige Zelle, sondern ganze Felder von Zellen verstehen) induktiv zeigen, dass die Teilmatrix zum Zeitpunkt $t+4$ identisch zu jener ist, die im Zeitpunkt t um jeweils eine Zelle verschoben zu finden war – den „Beweis“ dafür liefert die Ausführung des CA.

Was ist hier also, will man der Argumentation Standishs folgen, „emergent“? Dass einer interessant aussehenden Kette von Transformationen (oder statischer ausgedrückt, einer Klasse von 4 Zuständen einer Teilmatrix), ein eigener, zusammenfassender Name und somit „Ding-Status“ zugewiesen wird, kann durchaus als Beschreibung auf Makroebene gesehen werden. Es steht jedoch in keiner Weise im Widerspruch zum Reduktionismus, denn eine Irreduzibilität der Eigenschaften oder prinzipielle Unvorhersagbarkeit der Struktur wird gar nicht behauptet. Was in diesem Fall emergiert, ist vor allem ein neuer Name. Ein Emergenzbegriff, der sich auf jedes beliebige Phänomen anwenden lässt, dem ein eigener Name zugestanden wird, ist jedoch in der Praxis wenig hilfreich.

Für Holland steht die praktische Unvorhersagbarkeit der glider-Formation im Vordergrund:

The possibility of such a spatially coherent moving pattern is not something easily determined by direct inspection of the laws of Conway's universe. . . . [it] only exists because of the strongly nonlinear interaction of the particles (states) in adjacent cells. . . . no extant analytical technique will predict the existence of a glider pattern. [Hol97]

Es ist evident, dass hier gar nicht der Versuch unternommen wird, starke Kriterien der Emergenz zu behaupten. Es kann jedoch sehr wohl behauptet werden,

dass die Vorhersage des Verhaltens von System S mit der Ausführung des Systems gleichzusetzen ist: Vorhersage (Prädiktion) bedeutet, das Verhalten des Systems abschätzen zu können, *bevor* dieses Verhalten tatsächlich eintritt. Es sei nun vorausgesetzt, dass das System (wie in Fällen von deterministischem Chaos) durch ein vereinfachtes Modell S' *nicht* brauchbar abgeschätzt werden kann, weil bereits kleine Abweichungen (z.B. der Zustand einer einzigen Zelle) zu drastischen Unterschieden im Systemverhalten führen können. Diese Voraussetzung ist vermutlich in CA der Klassen III und IV gegeben und wird von Wolfram als *computational irreducibility* bezeichnet:

Cellular automaton evolution can be considered as a computation. A procedure can short cut this evolution only if it involves a more sophisticated computation. But there are cellular automata capable of universal computation that can perform arbitrarily sophisticated computations. So at least in these cases no short cut procedure can in general be found. The cellular automaton evolution corresponds to an irreducible computation, whose outcome can be found effectively only by carrying it out explicitly. [Wol85]

Korrekte Vorhersage des Systemverhaltens erfordert somit, dass der Ausgangszustand des Systems S in einem Modellsystem S_v völlig exakt nachgebildet wird (was im Falle eines CA kein Problem ist, da der Ausgangszustand vollständig und genau bekannt ist) und die Transformationsregeln exakt angewendet werden, bis der gewünschte Zeitpunkt erreicht ist. Diese Ausführung des geklonten Systems S_v ist aber tatsächlich in allen Zuständen und auch im Rechenaufwand völlig identisch mit der Ausführung des untersuchten Systems S – man hätte genausogut S ausführen können. Der Begriff „Vorhersage“ macht in diesem Kontext wenig Sinn; genauso wie der Begriff „Modell“ für S_v deplaziert wirkt. Wie S ist S_v ein formales System, und dazu noch eine völlig identische Kopie von S .²⁷ Diese Situation ist

²⁷Immerhin lässt sich aus dieser Modellierungs-, oder besser Simulationsrelation eine gute Illustration des Leibnizschen Dualismus (siehe 2.1.3) gewinnen: wenn S und S_v gleichzeitig und mit gleicher Ausgangskonfiguration gestartet und in identischen Zeitschritten iteriert werden, verläuft das Geschehen in ihnen völlig parallel, ohne dass es einen Kausalzusammenhang der Systeme gibt. Dieses Beispiel ist passender als das von Schleichert [Sch92] kritisierte der beiden parallel laufen-

exakt jene, die der simulationsrelative Emergenzbegriff Bedaus und Darleys (siehe 3.1.2) beschreibt.

Kann dem glider im Sinne von Collier Kohäsion zugesprochen werden? Festzustellen ist, dass dieses Muster nicht durch Zufall erhalten bleibt – wie ja der Zufall bei CA generell ausgespart bleibt, es sei denn in der anfänglichen Phase des „seeding“. Der Zusammenhalt dieses Haufens von Zellen ist nicht nur einer gemeinsamen Ursache zu verdanken, sondern auch der inneren Struktur: jede einzelne Zelle des glider ist mit ihren Nachbarzellen gekoppelt. Diese Kopplung besteht aber eben nicht nur mit der „Innenwelt“ des glider, sondern auch mit der „Außenwelt“, und zwar genauso stark: eine Zelle am Rand des glider ist gegenüber Zuständen außerhalb der Formation genauso empfindlich wie gegenüber Zuständen innerhalb der Formation. Schon eine einzige Zelle in der Nachbarschaft einer Randzelle des glider kann die gesamte Formation zerstören, wie auch eine (z.B. durch äußeren Einfluss) veränderte Zelle im Inneren der Formation (siehe dazu Abbildung 3.9). Nun ist aber genau die relative Unempfindlichkeit gegenüber internen und externen Fluktuationen eines der bestimmenden Kriterien von Kohäsion. Der glider-Formation kann somit kein kohäsiver Zusammenhalt zugesprochen werden: sie ist genauso fragil wie jedes andere Muster dieser Größenordnung.

Nach Hordijk, der seinen Begriff von Emergent Computing auf dem Forrests aufbaut, ist die Emergenz nicht in Formationen wie dem glider selbst zu finden:

In the „Game of Life“ emergent structures called gliders and glider guns are used to create explicitly a particular initial configuration such that the CA dynamics mimics computation with logical gates. This seems to be somewhere in between emergent and explicitly programmed. The intrinsic computation embedded in a CA's dynamics, however, appears to be truly emergent. ([Hor99], S. 12)

Diese „intrinsic computation“ ist eines der immer wiederkehrenden Themen in der Arbeit Langtons. Langton zieht einige (spekulative) Schlüsse aus seinen Beobachtungen von CA über komplexe Systeme im allgemeinen. So wäre der Übergang

den Uhren: zelluläre Automaten sind kausal geschlossene Welten (wenn man davon absieht, dass Ausgangskonfiguration und Übergangsfunktion exogen gegeben sind), Uhren sind es nicht.

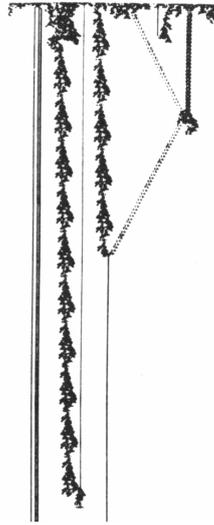


Abbildung 3.9: Ein eindimensionaler CA der Klasse IV; die Zeitachse verläuft vertikal von oben nach unten. Aus der Ausgangskonfiguration entwickeln sich mehrere komplexe, langlebige Muster, die jedoch früher oder später alle durch Kollision mit relativ simplen Mustern zerstört werden. Aus [Lan91].

zwischen den Klassen I+II sowie III als Übergang zwischen zwei fundamentalen Klassen dynamischen Verhaltens zu deuten, nämlich „fest“ und „flüssig“. Nur an dieser Grenze zwischen Stillstand und Chaos könne sich interessantes Verhalten entwickeln, woraus sich das Schlagwort „computation at the edge of chaos“ ableitet. Für die Frage nach der Emergenz ist hier allerdings wenig an neuen Einsichten zu gewinnen: mehr als die oben beschriebene simulationsrelative Emergenz ist auch an der „edge of chaos“ nicht zu finden.²⁸

²⁸Mitchell et al. konnten einige Resultate der CA-Experimente Langtons nicht nachvollziehen [MHC93] und stellen damit die große Bedeutung der vielzitierten „edge of chaos“ (trotz sehr vorsichtiger Formulierung) in Frage.

3.5 Artificial Life: die neue Heimat des Emergent Computing

In der Artificial Intelligence ist Emergent Computing nur eine Denkschule von mehreren: die „klassische“ künstliche Intelligenz verfolgt einen top-down-Ansatz und konzentriert sich auf höherliegende Konzepte wie Repräsentation, Induktion, Modellbildung. Anders im relativ jungen Forschungsgebiet Künstliches Leben (Artificial Life, A-Life): hier ist Emergent Computing das deklarierte Programm. Nach Boden ist *Selbstorganisation* ein zentrales Konzept des A-Life: in der Erklärung

Self-organization involves the emergence (and maintenance) of order, or complexity, out of an origin that is ordered to a lesser degree ... This development is 'spontaneous', or 'autonomous', following from the intrinsic character of the system itself ... instead of being imposed on the system by some external designer ([Bod96b], S. 3)

findet sich eine Zusammenfassung der Grundprinzipien von Emergent Computing. Es kommt daher nicht überraschend, dass zelluläre Automaten (siehe 3.4) als eines der ersten und einfachsten Beispiele für A-Life gelten. Eine eindeutige Zuordnung der diversen EC-Techniken zu AI oder A-Life ist jedoch nicht immer möglich: so lassen sich z.B. genetische Algorithmen (siehe 3.3) beiden Bereichen zuordnen.

Ähnlich wie in der AI können im A-Life zwei wesentliche Ziele unterschieden werden: das Verständnis des (biologischen) Lebens, wie wir es kennen („life as we know it“), und die Simulation oder gar Realisation neuer Lebensformen („life as it could be“). Der Anspruch, Leben nicht nur zu simulieren, sondern zu *realisieren*, also genuin lebendige Wesen zu erschaffen, wird als *starkes A-Life* bezeichnet.²⁹

Der fundamentale Unterschied zur Biologie ist nach einem der Pioniere dieses Feldes, Langton [Lan96], dass es in A-Life um „life made by man rather than by nature“ gehe – dieses *making* bezieht sich nicht nur auf das Bauen theoretischer, sondern konkreter Modelle, vom zellulären Automaten bis zum Roboter. Während die

²⁹Diese Spielart ist zwar die medienwirksamste und umstrittenste, macht aber gemessen an wissenschaftlichen Publikationen und prominenten Vertretern nur einen kleinen Bereich des Feldes aus.

Biologie generell einen analytischen top-down-Ansatz verfolge, konzentriere man sich im A-Life auf einen synthetischen bottom-up-Ansatz, der die selbstorganisierte Natur des Lebens besser zu erklären vermag:

Life depends critically on principles of dynamical self-organization that have remained largely untouched by traditional analytic methods. . . . these self-organizing dynamics are fundamentally non-linear phenomena, and non-linear phenomena in general depend critically on the interactions *between* parts . . . [such] phenomena are most appropriately treated by a *synthetic* approach ([Lan96], S. 40, Hervorhebung im Original)

Langton reklamiert somit für A-Life, im Gegensatz zur Biologie eine *Systemsicht* einzubringen, indem die Relationen zwischen den einzelnen Komponenten berücksichtigt werden – es erscheint jedoch etwas problematisch, dies mit dem Begriff „Synthese“ gleichzusetzen, zumal sich Analyse nicht unbedingt auf isolierte Einzelkomponenten beschränken muss. Auch die Behauptung, dass sich die Biologie auf das analytische Zerlegen beschränke, dürfte Biologen missfallen: Ökologie, Populationsbiologie, Evolutionsbiologie arbeiten durchaus nicht nur rein analytisch, sondern verfolgen systemische, integrative Ansätze.

A-Life ist jedoch damit ein klassischer Fall eines jener Forschungsprogramme, die Emergent Computing exemplifizieren: das methodologische Grundprinzip ist der Bau von computerbasierenden Systemen, die komplexes Verhalten zeigen, aber aus einfachen, parallel arbeitenden Komponenten zusammengesetzt sind (siehe Abbildung 3.10). Das Ziel: Leben, oder zumindest lebensähnliches Verhalten, sollen in ebendiesen Systemen emergieren. Dieser Anspruch führte auch dazu, dass A-Life die Diskussion zur Frage „was ist Leben?“ in den letzten Jahren neu entfacht hat.³⁰ Die vom jeweiligen Forscher akzeptierte Lebensdefinition bestimmt entscheidend, welche Ansätze aus dem breiten Spektrum des EC verwendet werden, um dem Ziel näher zu kommen.

Für die Emergenzdiskussion bietet A-Life dann nichts neues, wenn es sich um konventionelle Verfahren der Informatik oder um Variationen und Kombinationen der

³⁰Einige im Bezug auf starkes A-Life eher optimistische Beiträge sind in [Bod96b] gesammelt, ausführliche Kritik an den Grundannahmen von starkem A-Life kommt unter anderem von Robert Rosen [Ros91, Ros00].

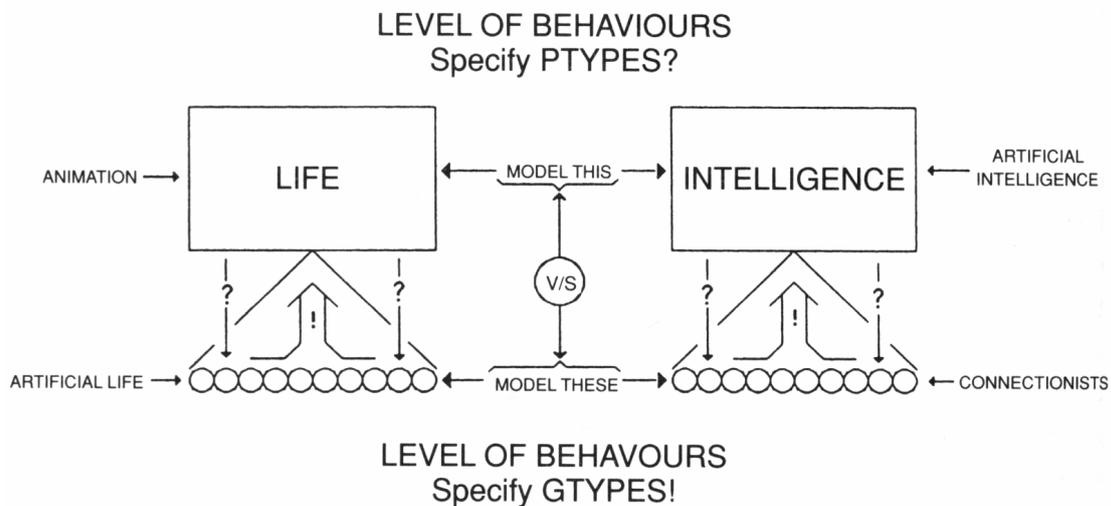


Abbildung 3.10: Langtons Illustration der verschiedenen Ansätze in A-Life und analog AI: die „Fragezeichen“ der top-down-Methodologien stehen den „Rufzeichen“ der bottom-up-Methodologien entgegen; die beiden Ebenen werden metaphorisch als GTYPE (Genotyp) und PTYPE (Phänotyp) bezeichnet (aus [Lan96]).

bereits erwähnten EC-Verfahren handelt. So sind etwa Langtons *self-replicating loops* [Lan96] eine Anwendung zellulärer Automaten, Rays bekanntes „Tierra“ [Ray92] kann als erweiterte Mischform von Genetischen Algorithmen und zellulären Automaten gelten, Lindenmayers L-Systeme [LP89] sind im wesentlichen rekursive Funktionen.

Ein gern zitiertes Beispiel soll hier etwas genauer betrachtet werden: die computergestützte Simulation von Schwarmverhalten im „Boids“-Modell von Reynolds [Rey87]. Reynolds' primäres Ziel war keineswegs die Schaffung von A-Life im heutigen Sinn, sondern ein praktisches Problem der Computergraphik: wie kann realistisch aussehendes Schwarmverhalten von Vögeln und anderen Tieren simuliert werden, ohne jedes einzelne Mitglied des Schwarms einzeln explizit steuern zu müssen? Wie kann dies trotz Hindernissen in der Flugbahn sichergestellt werden? Die Lösung: ein System, in dem jeder simulierte Vogel („Boid“) ein parallel laufender Prozess ist. In diesen Prozessen wird ein relativ einfaches Programm ab-

gearbeitet, das den Flug des Boid durch den virtuellen Raum nach folgenden Kriterien (in absteigender Wichtigkeit) steuert: 1) es ist ein Mindestabstand zu Hindernissen einzuhalten – auch zu anderen Boids, 2) die eigene Fluggeschwindigkeit sollte jener der benachbarten Boids nahekommen, und 3) es sollte immer eine Position in der Nähe des virtuellen Schwerpunkts aller benachbarten Boids angestrebt werden. Durch geeignete Parametrisierung entsteht aus diesen vergleichsweise simplen Verhaltenskomponenten tatsächlich höchst natürlich aussehendes Schwarmverhalten; die Schwärme schaffen es sogar, Hindernisse wie etwa Säulen zu „umfließen“, d.h. sich bei Bedarf aufzuspalten und nach dem Hindernis wieder zu vereinigen.

Dieses Modell ist deutlich aufwändiger als etwa das eines zellulären Automaten, entspricht aber genauso dem Grundprinzip des Emergent Computing: einfaches lokales Verhalten führt zu komplexem globalem Verhalten, zentrale Kontrolle des globalen Verhaltens entfällt. Die wesentlichsten Unterschiede zu den bisher besprochenen EC-Beispielen sind der Wegfall einer starren Struktur (die Systemteile können sich selbst frei bewegen) und die losere Kopplung zwischen den Systemteilen (ohne die das Verhalten des Schwarms unnatürlich, weil zu starr wirken würde). Reynolds' Modell ist eines der ersten in einer langen Reihe von strukturell ähnlichen A-Life-Modellen, in denen das Verhalten der einzelnen Teile immer komplexer bzw. adaptiver, und die Kopplung zwischen ihnen immer loser wird – wohl auch dank der immer größeren Rechenleistung verfügbarer Computer.³¹ Der höhere Aufwand ändert jedoch trotzdem nichts an einer Tatsache, die Langton bereits in den späten 1980er Jahren klar ausgesprochen hat: „The field of Artificial Life is unabashedly mechanistic and reductionist“ [Lan89].

³¹Nicht zufällig setzte mit der Weiterentwicklung des A-Life und der steigenden Komplexität der simulierten „Lebewesen“ eine Diskussion zum Thema Autonomie ein [Bod96a].

Kapitel 4

Zusammenfassung und Diskussion

4.1 Emergent Computing: angewandter Reduktionismus?

Einheitsprogramme üben stets einen verführerischen Glanz aus ... Der Teufel steckt freilich bekanntlich im Detail ... : Wieweit ist es möglich, sinnvoll und legitim, Schlüsselbegriffe aus einzelnen Disziplinen in andere Disziplinen so zu übertragen, daß bei der Übertragung der ursprüngliche Bedeutungsgehalt gewahrt bleibt und gleichzeitig mit diesen Begriffen in der neuen Disziplin praktisch operiert werden kann - m.a.W., wieweit sind Übertragungen nicht nur metaphorische Sprachspiele? [Klü90]

Was heute als „Emergent Computing“ bezeichnet wird, hat mit tatsächlichen Emergenztheorien wenig zu tun. Es verwundert daher kaum, dass der Begriff „Emergenz“ in der Informatik zwar von Systemeigenschaften ausgeht und somit im Sinne Stephans „schwache Emergenz“ abdeckt, aber mit den eingangs skizzierten starken philosophischen Emergenzbegriffen wenig gemeinsam hat. Stattdessen lassen sich grob zwei Konzepte von Emergenz identifizieren:

Phänomenologische Emergenz Was nicht explizit programmiert wurde, gilt als emergent: es erscheint zur Laufzeit des Systems¹. In diesem Sinn ist „emergent computing“ nichts weiter als „implicit computing“. Dieser Emergenzbegriff hängt mit dem Grundprinzip zusammen, dass komplexes Verhalten aus einfachem Verhalten entstehen kann – wie es etwa bei den zellulären Automaten besonders anschaulich zu sehen ist.

Simulationsrelative Emergenz Im Sinne Bedaus „weak emergence“ oder im Sinne Wolframs „computational irreducibility“. Diese trifft auf Systeme zu, deren Verhalten mit perfekter Information zwar simulierbar, aber nicht auf andere Weise vorhersehbar ist.

Die Begriffsunterschiede zur emergentistischen Philosophie liegen nicht etwa an anderen Sprachgewohnheiten der Disziplin, sondern an ihrer zugrundeliegenden „Metaphysik“: die Basis des Emergent Computing ist reduktionistisch. Die tatsächlich entscheidenden Dinge passieren auf der Mikroebene in lokalen Interaktionen; das – wenn auch möglicherweise für den Beobachter überraschende – Verhalten auf Makroebene ist direktes, epiphänomenales Resultat davon. Ein sehr deutliches Beispiel dieser Idee gibt John Hollands Buch „Emergence“ ab, in dem Holland alle hier beschriebenen Beispiele von Emergent Computing durch das Modell der „constrained generating procedure“ (cgp) vereinigt [Hol97]. Jede cgp besteht entweder selbst aus mehreren miteinander verbundenen cgp's oder – im Falle einer „elementaren cgp“ aus einem einfachen Mechanismus, vergleichbar einer Übergangsfunktion eines zellulären Automaten. Jede cgp kann somit auf einen Graphen von miteinander verbundenen deterministischen Mechanismen *reduziert* werden.² In anderen Worten: das Verhalten jeder cgp, sei es noch so kompliziert, ist letztlich das *Resultat* von elementaren Mechanismen und ihrer Kopplung. Was diese Position vom mechanistischen Bild der Welt als Uhrwerk³ unterscheidet ist lediglich die Betonung der nichtlinearen Interaktion zwischen den Mechanismen: „the whole is more than the sum of its parts“ – jedoch nur in dem Sinn, dass es Systemeigenschaften gibt, die sich nicht trivial aus einer Aufsummierung der Ei-

¹vgl. Hordijk, Clark in Kap. 3, sowie die Bemerkungen Cornings in Kap. 2.

²vgl. die Boolean Networks 3.2

³vgl. Descartes/Leibniz in 2.1.1

enschaften von Bestandteilen ergeben.⁴ Das Ganze und seine Eigenschaften sind trotzdem lediglich eine kompakte Beschreibung der Teile und ihres Zusammenwirkens, sie haben *keine eigenständige kausale Kraft*, sie sind *Epiphänomene*, sie sind *nicht emergent*.

Was Emergent Computing von hergebrachten Methoden unterscheiden, ist daher *nicht* die Position in der Debatte um Holismus, Emergentismus oder Reduktionismus. Es geht vielmehr um die Frage, wie in einer reduktionistischen Welt Probleme gelöst werden können.

In klassischen Ansätzen der Informatik wird versucht, ein gegebenes Problem durch schrittweise Analyse in immer kleinere Teile zu partitionieren, wobei jeder Analyseschnitt (sic!) die Constraints der Teile fixiert: *divide et impera*. Auf die Problemteile wird dann ein explizit definierter Algorithmus angewendet.⁵ Diese Ansätze geraten meist dann in Schwierigkeiten, wenn das System hochgradig adaptiv sein soll oder der Lösungsweg – und damit die Schnittführung im Analyseprozess – sich nicht bereits aus der Problemdefinition ergibt. Dies ist etwa dann der Fall, wenn das Problem nicht klar definiert ist⁶, es für das Problem keine hinreichende Makrotheorie gibt oder es sich um ein schwieriges mathematisches Optimierungsproblem handelt.

Im Emergent Computing hingegen wird die analytische Schnittführung möglichst weitgehend durch Synthese ersetzt: die Methodik ist explorativ, die Lösung soll aus dem gemeinsamen Verhalten der einzelnen Teile *resultieren* (in der technischen Anwendung wird dieser Vorgang durch entsprechende Rahmenbedingungen in die gewünschte Richtung gelenkt – etwa in den Lernverfahren der neuronalen Netze). Wenn hier von Emergenz gesprochen wird, dann hat dies immer mit einem Mangel an Vorhersage zu tun, sei dieser nun eine prinzipielle (Vorhersage im engeren Sinn ist unmöglich, außer durch Simulation), praktische (Vorhersage ist möglich, aber

⁴Nachdem Langton A-Life als „unverfroren mechanistisch und reduktionistisch“ deklariert hat, fügt er hinzu: „However, this *new mechanism*—based as it is on multiplicities of machines and on recent results in the fields of nonlinear dynamics, chaos theory, and the formal theory of computation—is vastly different from the mechanism of the last century“ ([Lan89], S. 6).

⁵NB: auch „Heuristiken“ werden algorithmisch implementiert!

⁶vgl. die Schwierigkeiten, „Intelligenz“ und „Leben“ zu definieren

aus irgendwelchen Gründen sehr aufwendig) oder faktische⁷ (Vorhersage wäre möglich, wurde aber nicht durchgeführt) Beschränkung.

Damit ist die zweite Frage beantwortet, die in der Einleitung gestellt wurde: was Emergent Computing heute mit echten Emergenztheorien zu tun hat. Die erste Frage ist die ungleich schwierigere, grundsätzlichere: *kann* Emergenz im Sinne der Emergenztheorien im Computing überhaupt erreicht werden? Im folgenden sollen einige Aspekte skizziert werden, die mit der Lösung dieses Problems zusammenhängen.

4.1.1 Kann es „starkes“ Emergent Computing geben?

Zunächst soll überlegt werden, aus welcher Weltsicht heraus eine derartige Frage gestellt werden kann, und welche Probleme sich daraus ergeben (siehe Abbildung 4.1).

(Digitale) Computer seien hier als Maschinen im Sinne Krämers [Krä88] definiert. Eine Maschine ist demnach ein realisierter Algorithmus, der den Bedingungen Elementarität, Determiniertheit, Allgemeinheit und Endlichkeit⁸ entspricht. Die Elementarität macht dabei den reduktionistischen Aspekt aus: „[die] Zerlegung eines komplexeren Prozesses in Grundoperationen, die selbst nicht weiter zerlegbar sind...“ ([Krä88], S. 159).

Aus der Sicht eines reduktiven Physikalisten unterscheidet sich der Computer nicht fundamental von seiner Umgebung: es wird unterstellt, dass die Welt prinzipiell verlustfrei auf ein formales System (oder zumindest eine endliche Zahl formaler

⁷Faktisch heisst hier, dass die Vorhersage – aus welchem Grund auch immer – gar nicht erst versucht wurde. Beispielsweise könnte der wandfolgende Roboter (siehe 26) durch einen Konstruktionsfehler entstanden sein. Daraus resultiert der schwächste Emergenzbegriff, der sich von der umgangssprachlichen Bedeutung des „Auftauchens, Erscheinens“ noch unterscheidet: das Verhalten kommt einfach für einen bestimmten Beobachter überraschend.

⁸Hierunter können drei Teilbedingungen verstanden werden: 1) der Algorithmus selbst kann als endliche Folge von Zeichen niedergeschrieben werden, 2) die Ausführungszeit jedes Rechenschrittes ist endlich, 3) der Algorithmus hält nach einer endlichen Anzahl von Schritten. Nur die ersten beiden Teilbedingungen sind in der Algorithmusdefinition notwendig (siehe [Krä88], S. 160), womit auch die Existenz einer Endlosschleife nichts am Status als Algorithmus ändert.

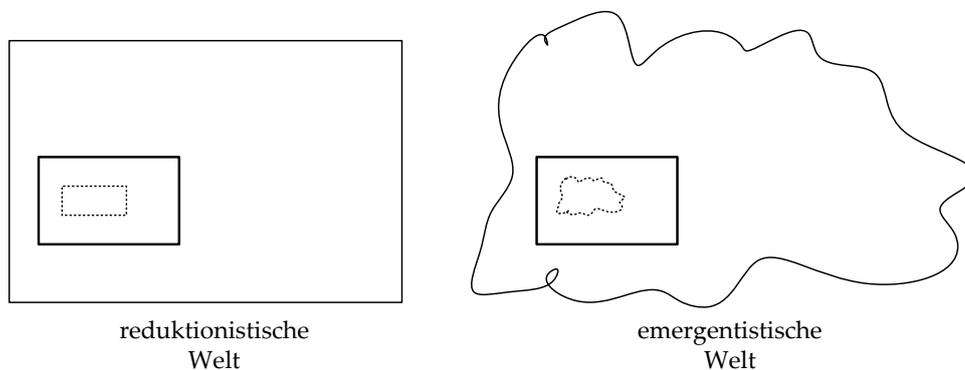


Abbildung 4.1: In einer reduktionistischen Welt besteht kein prinzipielles Hindernis, Eigenschaften der Welt in einem formalen System exakt abzubilden. In einer emergentistischen Welt trifft dieser Versuch auf grundsätzliche Schwierigkeiten.

Systeme) abzubilden wäre, das im Computer repräsentierbar ist.⁹ Ob dies als Simulation oder als Realisierung verstanden wird, ist eher zweitrangig. Was den Computer *erkenntnistheoretisch* vom Rest der Welt unterscheidet und nützlich macht, ist die Tatsache, dass seine Zustände und Zustandsübergänge absolut vollständig und absolut genau erfass-, beschreib- und messbar sind: der Status des menschlichen Beobachters entspricht dabei jenem des Laplaceschen Dämons. Der Versuch, in dieser „Maschine in der Maschine“ etwas eben *nicht* Maschinenhaftes, also z.B. Emergentes, zu erzeugen, dürfte aus dieser Weltanschauung kaum sinnvoll (weil ohne Realitätsbezug) und prinzipiell aussichtslos erscheinen.

Für den Emergentisten weisen Computer sehr wohl drastische Unterschiede zur Umgebung auf. Es handelt sich dabei um Maschinen, die mit einigem Aufwand reduktionistische Inseln in einer nicht – oder zumindest nicht vollständig – re-

⁹So zumindest die unterstellte Arbeitshypothese. Ob es reduktiven Physikalismus gibt, der nicht formalisierbar wäre, sei an dieser Stelle ausgeklammert. Wir treffen hier auch auf das Problem, dass eine vollständige Abbildung der Welt im Computer eine unendliche Rekursion bedeutete: auch der Computer ist Teil der Welt und müsste im Computer abgebildet werden – das formale System müsste deshalb *in sich* Aussagen *über sich* enthalten, was jedoch logisch unmöglich ist ([Krä88], S. 156). Ein etwas bescheidenerer Anspruch des Physikalisten wird sein, dass im Computer nicht „die Welt“ vollständig abgebildet wird, sondern nur die Mechanismen, nach denen sie funktioniert.

duktionistischen, vielleicht sogar indeterministischen Welt schaffen. In diesem Sinn könnte es auch dem Emergentisten aussichtslos erscheinen, innerhalb dieser mühsam geschaffenen nicht-emergentistischen, nicht-indeterministischen Kunstwelt wieder eine tatsächlich emergentistische zu schaffen. Tatsächlich emergentistisch heisst dabei, dass es nicht nur um den bloßen Anschein von Emergenz handelt, der durch algorithmische (und damit reduktionistische) Tricks erreicht wird. So ist es etwa grundsätzlich vorstellbar, dass ein irreduzibles System simuliert wird, indem für das Gesamtsystem neue Axiome eingeführt werden, die für die Bestandteile nicht gelten (d.h. aus den für die Bestandteile gültigen Axiomen nicht deduziert werden können). Diese könnten jedoch nur exogen hinzugefügt werden: es wäre ein Widerspruch, wenn sich nicht-ableitbare Eigenschaften aus den bekannten Eigenschaften der Bestandteile herleiten ließen. Ähnliches gilt für den Indeterminismus: da wir Berechnung im Computer als Vorgang definieren, in dem aus Eingabedaten, Axiomen und Transformationsregeln strikt deterministisch alles andere folgt, ist es ein recht offensichtlicher Zirkelschluss, dass aus Berechnung kein indeterministisches Verhalten kommen kann. Versuche, in ein solches System Zufall zu injizieren (vgl. die Bemerkungen in 3.1.2), können dessen strikte interne Kausalitätsverhältnisse nicht ändern. Selbst wenn man dies als zumindest noch als Simulation von Indeterminismus akzeptieren würde, hat es noch nichts mit der für manche Emergenztheorien wesentlichen Idee von unvollständigem Determinismus (siehe 2.2.4) zu tun. In diesen Modellen geht es nicht einfach um Zufall (z.B. eine Zufallskomponente in der Übergangsfunktion eines CA), sondern um Freiheitsgrade eines Systems, die sich in bestimmten Phasenübergängen ergeben. Wann und wo genau sollte also der Indeterminismus ins Spiel kommen? Dies dürfte nicht exogen, sondern endogen („the effect is not predictable because it is *the system* that intervenes in the chain of cause and effect“ [Hof01]) entschieden werden. Es zeigt sich somit, dass schon die *Simulation* starker Emergenz auf einige Probleme trifft.

Gibt es trotzdem Auswege aus dieser Situation? Wenigstens zwei Stoßrichtungen erscheinen denkbar: die erste bestünde darin, die Mächtigkeit von Computern bzw. Turingmaschinen zu erweitern. Teuscher ([Teu02], Kap. 5) gibt einen kurzen Überblick über solche Versuche, die derzeit jedoch nicht über den Status von Gedankenexperimenten hinauskommen. Die andere bestünde darin, Emergenz nicht im geschlossenen Computersystem selbst, sondern in Gesamtsystemen zu suchen, in de-

nen die natürliche Umwelt ein wesentlicher Bestandteil ist. Wenn etwa ein „Insektenschwarm“ aus Robotern in einer natürlichen Umgebung arbeitet (siehe [KBK00]), kann aus Sicht eines Emergentisten ausgeschlossen werden, dass dieses *Gesamtsystem* emergentes Verhalten zeigen könnte?¹⁰

4.2 Schwache Emergenz: stark genug für wichtige Fragen

Rückt man von der strikten Forderung nach *prinzipiellen* Unwägbarkeiten ab, bleibt immer noch in vielen Bereichen des Emergent Computing praktische Unvorhersagbarkeit – in einem Maß, das bei weitem ausreicht, höchst nützliche Resultate in der Künstlichen Intelligenz zu erreichen, und den Bau komplexer Softwaresysteme trotz aller Fortschritte im Software Engineering schwierig und fehleranfällig¹¹ zu machen.

4.2.1 Emergent Computing im Software Engineering

Forrest weist darauf hin, dass heutige Programmiersprachen von Grund auf zur Vermeidung von EC ausgelegt sind: Eigenschaften wie Kontextfreiheit minimieren die nichtlineare Interaktion von Programmelementen so weit als möglich ([For91], S. 5f).

Die Grundprinzipien des Emergent Computing weisen deutliche Ähnlichkeiten mit den heute üblichen Entwürfen von Informationssystemen, auf. Es ist daher zu erwarten, dass sich Eigenschaften von gezielt für EC entwickelten Systemen zumindest in abgeschwächter Form auch in komplexen Softwaresystemen finden, bei denen unerwartete Effekte absolut unerwünscht sind. Tatsächlich entspricht es der Erfahrung jedes Software Engineer, dass sich trotz sorgfältiger Analyse, Entwurf und Implementation im fertiggestellten System nicht vorhersehbare Effekte

¹⁰Vgl. dazu Wegner [Weg97] und Cariani (3.1.3).

¹¹Siehe dazu Neumanns beeindruckende Aufstellung und Analyse verschiedenster Fälle von versagenden Computersystemen [Neu95].

einstellen, die offenbar in Zusammenhang mit der Struktur des Gesamtsystems stehen.¹²

Die Ursachen können dabei vielerlei sein: von einfachen Programmierfehlern über inadäquaten Systementwurf bis hin zu unvollständigem oder falschem Wissen über die Umgebung, mit der das System interagiert (z.B. Hardware oder andere Softwarekomponenten). Die Suche nach der konkreten Ursache (oder zumindest dem *Auslöser* des Problems, dessen Auftreten in komplexeren Systemen durch eine ganze Reihe von Rahmenbedingungen überhaupt erst ermöglicht wird), gestaltet sich dabei oft mühselig und zeitaufwendig.

Es besteht die Möglichkeit, dass das Auftreten „emergenter Bugs“ ausgerechnet durch die moderne objekt- und komponentenbasierte Softwareentwicklung begünstigt wird, die strukturelle Ähnlichkeiten mit den konnektionistischen Techniken des Emergent Computing aufweist. Das Paradigma der Objektorientierung strukturiert Software und Hardware in „Objekte“¹³: gekapselte Einheiten aus Daten und Programmen, die über wohldefinierte Schnittstellen mit anderen Objekten interagieren können. Während beim Entwurf prozeduraler Software immer ein klarer roter Faden des Programmflusses im Vordergrund steht, ist die OO-Denkweise stärker von Objekten, ihren Eigenschaften und der gegenseitigen Interaktion geprägt [Weg97]. Illustriert werden kann dies durch einen Vergleich üblicher Entwurfsmethoden (Abbildungen 4.2 und 4.3).

Zwar wird im objektorientierten Entwurf sehr wohl auch der Programmfluss berücksichtigt, der klare „rote Faden“ vom Anfangs- bis zum Endzustand ist aber sowohl im Entwurf als auch im fertig implementierten Produkt wesentlich schwieriger erkennbar. Tatsächlich ist eine typische und nicht immer leicht zu beantwortende Frage beim Reverse Engineering bzw. Refactoring bestehender objektorientierter Software: „das sind also die Objekte mit ihren Eigenschaften und Methoden; wo ist nun eigentlich *das Programm?*“¹⁴.

¹²Ein Phänomen, das z.B. von Hofstadter [Hof79] anschaulich gemacht wird: der Autor beschreibt den Effekt, dass auf dem Timesharing-Rechner seiner Universität ab einer bestimmten Anzahl von Benutzern die Systemperformance schlagartig einbricht – diese Skalierungsgrenze ist jedoch nirgends im System explizit kodiert, sie ergibt sich als Systemeigenschaft.

¹³Der Begriff impliziert in diesem Zusammenhang keine Subjekt-Objekt-Beziehung.

¹⁴Dieses Problem des Re(verse)-Engineering korrespondiert mit einer generellen Schwäche des objektorientierten Entwurfs, größere, langlebige Arbeitsabläufe („work flows“) zu modellieren.

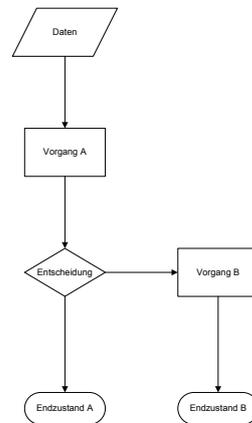


Abbildung 4.2: Flussdiagramm – das klassische Entwurfsmodell rein prozedural orientierter Software.

Besonders deutlich wird die Ähnlichkeit, wenn Teile von Software nebenläufig (*concurrent*) arbeiten können, also im Fall von Multiprocessing oder Multithreading, sowie bei verteilten Systemen. Die Situation ähnelt dann konnektionistischen Techniken wie den besprochenen neuronalen Netzen: auch dort verschwimmt die strikte Trennung von Daten und Code, auch dort agieren die Systemteile nebenläufig und tauschen nur so viel Information wie unbedingt nötig über möglichst einfach gehaltene Schnittstellen aus (in diesem Sinn lose Kopplung zwischen Systemteilen gilt als eines der Ideale im Software-Design, vgl. [Som96], [BMR⁺98]). Die Frage „wo ist das Programm?“, also die Frage nach dem Verhalten des Systems, ist bei einem neuronalen Netz ebenfalls schwer zu beantworten, denn „das Programm“ steckt nicht in den vergleichsweise simplen Einzelneuronen, es ergibt sich implizit. Je ähnlicher das Softwaresystem einem solchen konnektionistischen Netz wird, desto *impliziter* wird sein Verhalten – ein Umstand, der dem Streben des Software Engineering nach Dokumentierbarkeit und Determinierbarkeit diametral entgegensteht.

Die Frage lautet nun, wie mit dieser Situation umzugehen ist. Zum einen ist es nötig, ein theoretisches Instrumentarium zur Beschreibung und Analyse von (im

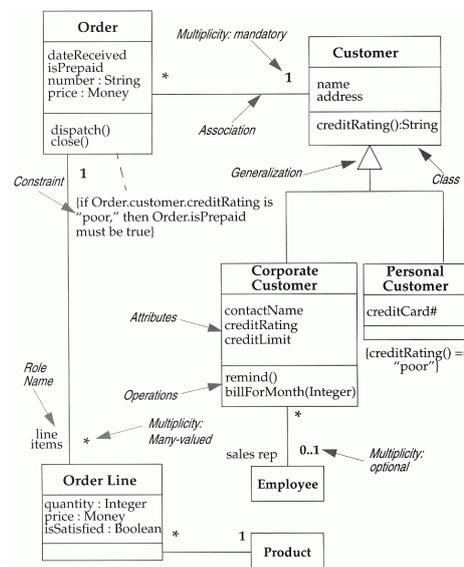


Abbildung 4.3: Klassendiagramm – „truly central within object-oriented methods“ (aus [FS97])

schwachen Sinn) emergenten Eigenschaften in Software zu entwickeln. Zum anderen müssen Methoden geschaffen werden, diese Eigenschaften zu beherrschen, will man nicht den entgegengesetzten Weg einschlagen und – wie im Emergent Computing – die Emergenz zum Weg oder gar zum Ziel erklären. Die Forschung zu diesen Themen steckt derzeit noch in den Anfängen; einige Ansätze sollen hier erwähnt werden.

Chroust [Chr02] beschäftigt sich konkret mit dem Problem emergenter Eigenschaften in komponentenbasierter Software. Zum einen schlägt er für komponentenbasierte Software eine Klassifikation von Systemeigenschaften vor, die „emergente“ Eigenschaften explizit berücksichtigt. Ob die Emergenz (hier ist damit im besonderen die praktische Unvorhersagbarkeit gemeint) dieser Eigenschaften eine Folge unzureichender Modellierungsansätze oder ein grundsätzlicheres Problem (etwa eine Folge prinzipieller Unvorhersagbarkeit) ist, wird von Chroust vorerst bewusst außer acht gelassen. Eine emergente Systemeigenschaft ist definiert als: „a property which is not determined solely from the properties of the system’s components,

but which is additionally determined by the system's structure (i.e. by the way the parts are connected to form the system)".

Schwierig zu analysierende Phänomene wie thrashing, das sich im wesentlichen als Performanceeinbruch eines Systems äußert, oder deadlocks, deren Auswirkungen von kurzfristigen, auf einzelne Teile beschränkten Aussetzern bis hin zum völligen Systemstillstand reichen können, könnten möglicherweise durch bessere Modellierungstechniken beherrscht werden, die nicht nur die Struktur, sondern auch die Dynamik komponentenbasierter Systeme erfassen. Daher greift auch die obige Definition emergenter Systemeigenschaften zu kurz: nicht nur die Struktur (der Komposition eines Systems), sondern auch der zeitliche Ablauf bestimmen das Systemverhalten. Zwei Systeme könnten exakt dieselbe Struktur, aber völlig verschiedene Laufzeitcharakteristika haben.

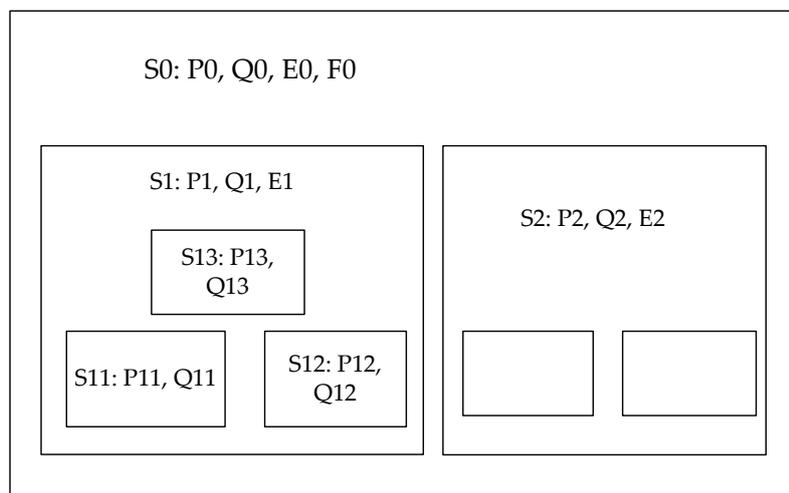


Abbildung 4.4: Eigenschaften komponentenbasierter Softwaresysteme (nach [Chr02]). Das System S_0 besteht aus den Subsystemen S_1, S_2 usw. Die Eigenschaften P_1, P_2 sowie Q_1, Q_2 gelten als nicht-emergent und werden im Gesamtsystem nach einer bekannten Funktion zu P_0 bzw. Q_0 aggregiert. Die emergenten Eigenschaften E_1 bzw. F_0 tauchen dagegen auf der jeweiligen Systemebene neu auf und lassen sich – zumindest praktisch – nicht aus den Eigenschaften der Teilsysteme erklären oder vorhersagen.

Als Lösungsansatz zur Eliminierung oder zumindest Einschränkung solcher Eigenschaften wird vorgeschlagen, nur eine kleine, definierte Untermenge möglicher Kontrollstrukturen (Sequenz, Alternative, Asynchrone Parallelausführung, Schleife, endlicher Automat) zuzulassen. Angesichts der erstaunlichen Komplexität des Verhaltens so primitiver Systeme wie etwa der eindimensionalen zellulären Automaten stellt sich dabei allerdings die Frage, ob dieser Ansatz tatsächlich eine wesentliche Reduzierung des „Überraschungseffekts“ für den Softwareingenieur bringen kann. Eine weitere Beschränkung ist darin zu sehen, dass er sich auf zentralisierte Steuerung beschränkt, und nur schwer auf ereignisgetriggerte Steuerung umzulegen ist. Letztere ist in besonders in hardwarenahen sowie verteilten, lose gekoppelten Systemen üblich und entspricht auch eher dem OO-Paradigma.¹⁵

Long und Denning [LD95] schlagen mit „ultra-structure“ einen neuen Weg ein, um Analyse, Entwurf und Wartung komplexer Systeme vorhersagbarer zu machen, ohne durch exzessive Rigidität spätere Änderungen zu erschweren. Ultra-structure teilt zwei Grundprinzipien mit Emergent Computing: 1) Komplexität entsteht aus der Interaktion einfacher Elemente, 2) die herkömmliche Trennung in Daten und Code verschwimmt. Im deutlichen Gegensatz zu EC stehen jedoch nicht Elementarteilchen (Zellen, Neuronen, Objekte) mit ihren dynamischen Eigenschaften (Übergangsfunktionen, Aktivitätsfunktionen, Methoden) im Vordergrund, sondern *Regeln*. Regeln sind gleichzeitig Programm und Daten; sie können sich in hierarchischer Weise auf andere Regeln beziehen und somit als Constraints wirken bzw. selbst von Constraints beeinflusst werden. Dies ist jedoch nur auf den ersten Blick eine Rückkehr zur prozeduralen Arbeitsweise der prä-OO-Ära, weil Regeln als Datenstrukturen vorliegen. Weiter oben wurde ausgeführt, dass die heute so populäre objektorientierte Arbeitsweise jener von Emergent Computing ähnelt und damit auf, wie Long und Denning es ausdrücken, „substantialist assumptions“ basiert.¹⁶ Was Ultra-structure im Kontext dieser Arbeit interessant macht, ist die Tatsache, dass die Probleme komplexer Software angegangen werden, indem genau diese Annahmen fallengelassen oder zumindest deutlich verändert werden.

Power [Pow90] beschäftigt sich mit komplexen verteilten Systemen und schlägt zur

¹⁵Die Begriffe „Systemstruktur“, „zentrale Steuerung“ und „ereignisgetriggerte Steuerung“ werden hier in der Definition Sommervilles ([Som96], Kap. 13) verwendet.

¹⁶vgl. die Bemerkungen zu Bickhard in 2.2.6

Beherrschung von unerwarteten, praktisch unvorhersagbaren Ereignissen vor, sich an den Prinzipien von lernenden Classifier Systems zu orientieren. Ziel ist dabei jedoch nicht mehr, das Systemverhalten für den Entwickler vorhersagbarer zu machen, sondern dem System selbst die Fähigkeit zur Adaption an seine Umwelt zu geben – ein gutes Beispiel für jene Ansätze, die Emergent Computing im Software Engineering nicht verhindern, sondern explizit nutzen wollen.

Wegners Plädoyer für einen Paradigmenwechsel von der algorithmischen zur interaktionistischen Denkweise im Entwurf von Software geht in eine ähnliche Richtung, aber noch weiter [Weg97]. Es sei Zeit, den Schritt von der unflexiblen, ausdrucksarmen, aber formal vollständig erfassbaren Welt der Algorithmen zur flexibleren, ausdrucksreicheren, aber formal nicht mehr vollständig erfassbaren Welt der Interaktionsmaschinen zu wagen. Für Wegner ist dies keine Kapitulationserklärung vor der hohen Komplexität von Software, sondern ein notwendiger Schritt in der Weiterentwicklung des Software Engineering.

4.3 Schlussbemerkung

Keine Entwurfstheorie für Software – sei sie nun emergentistisch, holistisch oder prozessontologisch orientiert – kann etwas an der Tatsache ändern, dass heutige Computer im Grunde immer noch deterministische Symbolverarbeiter sind und nach den bewährten von-Neumann-Prinzipien arbeiten. Die Programmierung von Computern ist und bleibt im Kern mechanistisch. Der Entwurf komplexer Software kann und muss jedoch *konzeptionell* weit genug von der untersten Ebene entfernt sein, um mit völlig anderen, abstrakten Begriffen operieren zu können. Man vergleiche hier die bekannten Überlegungen Herbert Simons zur hierarchischen Strukturierung im Bau komplexer Systeme: von der mechanischen Taschenuhr bis zum Softwaresystem ist Komplexität nur durch Abstraktion bewältigbar [Sim73].

Wenn solche Begriffe unkritisch verwendet werden, führt dies vordergründig zu kaum mehr als begrifflicher Verwirrung und überzogenen Erwartungen in die Informatik: „machine learning“ und „artificial intelligence“ haben auch nach Jahrzehnten der Forschung wenig mit menschlichem Lernen und Intelligenz zu tun, „artificial life“ ist zwar künstlich, aber noch immer kein Leben, und „emergent

computing“ liefert den Emergenzphilosophen keine neuen Erkenntnisse oder gar Bestätigung ihrer Theorien.

Sind diese Schlagwörter somit nur leere, wirkungslose *buzzwords*? Nein. Sie prägen Denk- und Arbeitsweisen in der Informatik und darüber hinaus: Technik existiert nicht unabhängig von der Gesellschaft, die sie hervorbringt; Gesellschaft existiert nicht unabhängig von der durch sie hervorgebrachten Technik. Aus der Technik werden Theorien über die Welt abgeleitet (die Welt als Uhrwerk, als Dampfmaschine, als zellulärer Automat), oder sie werden durch die Technik „bewiesen“ (wenn auch oft nur im metaphorischen Sinn). Diese Theorien haben stets ethische Aspekte, weil sie Antworten auf Fragen nach dem richtigen Handeln geben. Gerade weil der Informatik heute dank „Informationszeitalter“ eine so große Rolle zugestanden wird, sollten ihre Schlagwörter kritisch untersucht werden.

Literaturverzeichnis

- [AA96] AHL, Valerie ; ALLEN, Timothy F.: *Hierarchy Theory : A Vision, Vocabulary, and Epistemology*. New York : Columbia University Press, 1996
- [ACF00] ANDERSEN, Peter B. (Hrsg.) ; CHRISTIANSEN, Peder V. (Hrsg.) ; FINNEMANN, Niels O. (Hrsg.): *Downward Causation : Minds, Bodies and Matter*. Århus : Aarhus University Press, 2000
- [BC00] BICKHARD, Mark H. ; CAMPBELL, Donald T.: Emergence. In: [ACF00], S. 322 f.
- [Bed97] BEDAU, Mark A.: Weak emergence. In: TOMBERLIN, J. (Hrsg.): *Philosophical Perspectives : Mind, Causation, and World* Bd. 11. Malden : Blackwell, 1997, S. 375–399
- [Bli92] BLITZ, David: *Emergent Evolution : Qualitative Novelty and the Levels of Reality*. Dordrecht : Kluwer, 1992
- [BMR⁺98] BUSCHMANN, Frank ; MEUNIER, Regine ; ROHNERT, Hans ; SOMMERLAD, Peter ; STAL, Michael: *Pattern-orientierte Software-Architektur : Ein Pattern-System*. Bonn ; Paris [u.a.] : Addison-Wesley, 1998
- [Bod96a] BODEN, Margaret A.: Autonomy and artificiality. [Bod96b], S. 95–108
- [Bod96b] BODEN, Margaret A. (Hrsg.): *The Philosophy of Artificial Life*. Oxford [u.a.] : Oxford Univ. Pr., 1996 (Oxford Readings in Philosophy)
- [Bra95] BRAUSE, Rüdiger W.: *Neuronale Netze : Eine Einführung in die Neuroinformatik*. Stuttgart : Teubner, 1995

- [BT95] BICKHARD, Mark H. ; TERVEEN, Loren: *Foundational Issues in Artificial Intelligence and Cognitive Science : Impasse and Solution*. Amsterdam [u.a.] : Elsevier, 1995 (Advances in Psychology 109)
- [Cam74] CAMPBELL, Donald T.: 'Downward Causation' in hierarchically organised biological systems. In: AYALA, F. (Hrsg.) ; DOBZHANSKY, T. (Hrsg.): *Studies in the Philosophy of Biology*. Berkeley : University of California Press, 1974, S. 179–186
- [Car92] CARIANI, Peter: Emergence and Artificial Life. In: [LTFR92], S. 775–797
- [Chr02] CHROUST, Gerhard: System properties under composition. In: [Tra02], S. 203–208
- [Cla96] CLARK, Andy: Happy couplings : emergence and explanatory interlock. In: [Bod96b], S. 262–281
- [CM98] COLLIER, John D. ; MULLER, Scott J.: The dynamical basis of emergence in natural hierarchies. In: *Emergence, Complexity, Hierarchy, and Organization. Selected and Edited Papers from the ECHO III Conference* Finnish Academy of Technology, 1998 (Acta Polytechnica Scandinavica MA91)
- [Cor02] CORNING, Peter A.: The re-emergence of emergence : a venerable concept in search of a theory. In: *Complexity* (2002)
- [CW01] CRISP, Thomas M. ; WARFIELD, Ted A.: Kim's master argument. In: *NOÛS* 35 (2001), Nr. 2, S. 304–316
- [Dar] DARLEY, Vince: Emergent phenomena and complexity, S. 411–416
- [EF94] EBELING, Werner ; FERSTEL, Rainer: *Chaos und Kosmos : Prinzipien der Evolution*. Heidelberg ; Berlin ; Oxford : Spektrum Akademischer Verlag, 1994
- [EHP00] EL-HANI, Charbel N. ; PEREIRA, Antonio M.: Higher-level descriptions: why should we preserve them? In: [ACF00], S. 118–142
- [EKS97] EMMECHE, Claus ; KØPPE, Simo ; STJERNFELT, Frederik: Explaining emergence : towards an ontology of levels. In: *Journal for General Philosophy of Science* (1997), Nr. 28, S. 83–119

- [EKS00] EMMECHE, Claus ; KØPPE, Simo ; STJERNFELT, Frederik: Levels, emergence, and three versions of downward causation. In: [ACF00], S. 13 f.
- [Emm94] EMMECHE, Claus: *Das lebende Spiel : Wie die Natur Formen erzeugt*. Rowohlt, 1994
- [Far91] FARMER, J. Doyne: A Rosetta Stone for connectionism. In: [For91], S. 153 f.
- [FH97] FLEISSNER, Peter ; HOFKIRCHNER, Wolfgang: Actio non est reactio : an extension of the concept of causality towards phenomena of information. In: *World Futures* 49 und 50 (1997), Nr. 3–4 und 1–4
- [FM91] FORREST, Stephanie ; MILLER, John H.: Emergent behavior in classifier systems. In: [For91], S. 213–227
- [For91] FORREST, Stephanie (Hrsg.): *Emergent Computation : Self-Organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks*. Cambridge, Mass. : MIT Press, 1991 (Special issues of Physica D)
- [FP96] FORT, Jean-Claude ; PAGÈS, Gilles: About the Kohonen algorithm : strong or weak self-organization? In: *Neural Networks* 9 (1996), Nr. 5, S. 773–785
- [FS97] FOWLER, Martin ; SCOTT, Kendall: *UML Distilled : Applying the Standard Object Modeling Language*. Reading, Mass. [u.a.] : Addison-Wesley, 1997
- [Fut90] FUTUYMA, Douglas J.: *Evolutionsbiologie*. Basel ; Boston ; Berlin : Birkhäuser, 1990
- [Gar83] GARDNER, Martin: *Wheels, Life, and Other Mathematical Amusements*. New York : Freeman, 1983
- [Gol99] GOLDSTEIN, Jeffrey: Emergence as a construct : history and issues. In: *Emergence* (1999), Nr. 11, S. 49–72
- [Ham95] HAMLYN, D.W.: Gestalt theory. In: *The Oxford Companion to Philosophy*. Oxford University Press, 1995

- [Har91] HARNAD, Stevan: The symbol grounding problem. In: [For91], S. 335 f.
- [Heb49] HEBB, D. O.: *The Organization of Behavior : a Neuropsychological Theory*. New York : Wiley, 1949
- [Hei95] HEIL, John: Holism. In: *The Oxford Companion to Philosophy*. Oxford University Press, 1995
- [Hof79] HOFSTADTER, Douglas R.: *Gödel, Escher, Bach : An Eternal Golden Braid*. Basic Books, 1979
- [Hof01] HOFKIRCHNER, Wolfgang: The hidden ontology : real-world evolutionary systems concept as key to information science. In: *Emergence* (2001), Nr. 3.3
- [Hol92] HOLLAND, John H.: *Adaptation in Natural and Artificial Systems : An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. 1st MIT Press ed. Cambridge, Mass. ; London : MIT Press, 1992
- [Hol97] HOLLAND, John H.: *Emergence. From Chaos to Order*. Reading, Mass. : Addison-Wesley, 1997
- [Hor99] HORDIJK, Wim: *Dynamics, Emergent Computation, and Evolution in Cellular Automata*, University of New Mexico, Diss., 1999
- [Joh01] JOHNSON, Steven: *Emergence : The Connected Lives of Ants, Brains, Cities and Software*. Penguin Press, 2001
- [Kau91] KAUFFMAN, Stuart A.: Requirements for evolvability in complex systems : orderly dynamics and frozen components. In: [For91], S. 135–152
- [Kau93] KAUFFMAN, Stuart A.: *The Origins of Order : Self-Organization and Selection in Evolution*. New York ; Oxford : Oxford Univ. Press, 1993
- [KBK00] KRIEGER, Michael J. ; BILLETTER, Jean-Bernard ; KELLER, Laurent T.: Ant-like task allocation and recruitment in cooperative robots. In: *Nature* (2000), August, Nr. 406

- [Kim93] KIM, Jaegwon: *Supervenience and Mind*. Cambridge, Mass. : Cambridge Univ. Press, 1993
- [Kim98] KIM, Jaegwon: *Mind in a Physical World*. Cambridge, Mass. : MIT Press, 1998
- [Kim00] KIM, Jaegwon: Making sense of downward causation. In: [ACF00], S. 305 f.
- [KK90] KROHN, Wolfgang (Hrsg.) ; KÜPPERS, Günter (Hrsg.): *Selbstorganisation : Aspekte einer wissenschaftlichen Revolution*. Braunschweig ; Wiesbaden : Vieweg, 1990
- [KK92] KROHN, Wolfgang (Hrsg.) ; KÜPPERS, Günter (Hrsg.): *Emergenz : Die Entstehung von Ordnung, Organisation und Bedeutung*. Frankfurt : Suhrkamp, 1992
- [Kla02] KLAUNINGER, Bert: Causality and emergence. In: [Tra02], S. 193–202
- [Klü90] KLÜVER, Jürgen: Auf der Suche nach den Kaninchen von Fibonacci, oder : Wie geschlossen ist das Wissenschaftssystem? In: [KK90], S. 201 f.
- [Koh84] KOHONEN, Teuvo: *Self-Organisation and Associative Memory*. Berlin : Springer, 1984
- [Krä88] KRÄMER, Sybille: *Symbolische Maschinen : die Idee der Formalisierung in geschichtlichem Abriß*. Darmstadt : Wissenschaftliche Buchgesellschaft, 1988
- [Lan89] LANGTON, Christopher G. (Hrsg.): *Artificial Life*. Redwood City : Addison-Wesley, 1989 (Santa Fe Institute Studies in the Sciences of Complexity, Proceedings VI)
- [Lan91] LANGTON, Christopher G.: Computation at the edge of chaos : phase transitions and emergent computation. In: [For91], S. 12 f.
- [Lan96] LANGTON, Christopher G.: Artificial Life. In: [Bod96b], S. 39 f.

- [LD95] LONG, Jeffrey G. ; DENNING, Dorothy E.: Ultra-structure : a design theory for complex systems and processes. In: *Communications of the ACM* 38 (1995), Nr. 1, S. 103–120
- [Lev73] LEVINS, Richard: The limits of complexity. In: [Pat73], S. 111 f.
- [LM95] LENZ, Anita ; MERETZ, Stefan: *Neuronale Netze und Subjektivität : Lernen, Bedeutung und die Grenzen der Neuro-Informatik*. Braunschweig ; Wiesbaden : Vieweg, 1995 (Theorie der Informatik)
- [LP89] LINDENMAYER, Aristid ; PRUSINKIEWICZ, Przemyslaw: Developmental models of multicellular organisms. In: [Lan89], S. 221–249
- [LSW00] LANZI, Pier L. (Hrsg.) ; STOLZMANN, Wolfgang (Hrsg.) ; WILSON, Stewart W. (Hrsg.): *Learning Classifier Systems : from Foundations to Applications*. Berlin ; Heidelberg [u.a.] : Springer, 2000 (Lecture Notes in Artificial Intelligence 1813)
- [LTFR92] LANGTON, Christopher G. (Hrsg.) ; TAYLOR, Charles (Hrsg.) ; FARMER, J. Doyne (Hrsg.) ; RASMUSSEN, Steen (Hrsg.): *Artificial Life II*. Redwood City : Addison-Wesley, 1992 (Santa Fe Institute Studies in the Sciences of Complexity, Proceedings X)
- [MHC93] MITCHELL, Melanie ; HRABER, Peter T. ; CRUTCHFIELD, James P.: Revisiting the edge of chaos : evolving cellular automata to perform computations. In: *Complex Systems* (1993), Nr. 7, S. 89–130. – SFI Working Paper 93-03-014
- [Neu95] NEUMANN, Peter G.: *Computer-Related Risks*. Reading, Mass. [u.a.] : ACM Press, 1995
- [Pat73] PATTEE, Howard H. (Hrsg.): *Hierarchy Theory : The Challenge of Complex Systems*. New York : Braziller, 1973
- [PE82] POPPER, Karl R. ; ECCLES, John C.: *Das Ich und sein Gehirn*. München : R. Piper, 1982
- [PH91] PALMORE, Julian ; HERRING, Charles: Computer arithmetic, chaos and fractals. In: [For91], S. 99 f

- [Pop95] POPPER, Karl R.: *Eine Welt der Propensitäten*. Tübingen : J.C.B. Mohr, 1995
- [Pow90] POWER, June: Distributed systems and self-organization. In: *Proceedings of the 1990 ACM Annual Conference on Cooperation*, 1990, S. 379–384
- [Ray92] RAY, Thomas S.: An approach to the synthesis of life. In: [LFR92], S. 371–408
- [Rey87] REYNOLDS, Craig W.: Flocks, herds, and schools : a distributed behavioral model. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press, 1987, S. 25–34
- [Roj96] ROJAS, Raúl: *Neural Networks : a Systematic Introduction*. Berlin ; Heidelberg [u.a.] : Springer, 1996
- [Ros91] ROSEN, Robert: *Life Itself : A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life*. New York : Columbia University Press, 1991
- [Ros00] ROSEN, Robert: *Essays on Life Itself*. New York : Columbia University Press, 2000
- [Rus95] RUSE, Michael: Reductionism. In: *The Oxford Companion to Philosophy*. Oxford University Press, 1995
- [Sal85] SALTHER, Stanley N.: *Evolving Hierarchical Systems : Their Structure and Representation*. New York : Columbia University Press, 1985
- [Sch92] SCHLEICHERT, Hubert: *Der Begriff des Bewußtseins : eine Bedeutungsanalyse*. Frankfurt am Main : Klostermann, 1992
- [SDR⁺00] SMITH, Robert E. ; DIKE, Bruce A. ; RAVICHANDRAN, B. ; EL-FALLAH, A. ; MEHRA, R.K.: The fighter aircraft LCS : a case of different LCS goals and techniques. In: [LSW00], S. 283–300
- [SHF94] SCHÖNEBURG, Eberhard ; HEINZMANN, Frank ; FEDDERSEN, Sven: *Genetische Algorithmen und Evolutionsstrategien*. Bonn ; Paris [u.a.] : Addison-Wesley, 1994

- [Sim73] SIMON, Herbert A.: The organization of complex systems. In: [Pat73], S. 3 f.
- [Smu27] SMUTS, J. C.: *Holism and Evolution*. 2nd edition. London : Macmillan, 1927
- [Som96] SOMMERVILLE, Ian: *Software Engineering*. 5th edition. Harlow ; Reading, Mass. [u.a.] : Addison-Wesley, 1996
- [Sta01] STANDISH, Russell K.: On complexity and emergence. In: *Complexity International* 09 (2001)
- [Ste94] STEELS, Luc: The Artificial Life roots of Artificial Intelligence. In: *Artificial Life* (1994), Nr. 1, S. 75–110
- [Ste99] STEPHAN, Achim: *Emergenz : Von der Unvorhersagbarkeit zur Selbstorganisation*. Dresden ; München : Dresden Univ. Press, 1999 (Theorie und Analyse 2)
- [Teu02] TEUSCHER, Christof: *Turing's Connectionism : an Investigation of Neural Networks Architectures*. London ; Berlin [u.a.] : Springer, 2002
- [Tra02] TRAPPL, Robert (Hrsg.): *Cybernetics and Systems 2002. Proceedings of the 16th European Meeting on Cybernetics and Systems Research*. Bd. 1. Austrian Society for Cybernetic Studies, 2002
- [von84] VON FOERSTER, Heinz: Principles of self-organization – in a socio-managerial context. In: ULRICH, H. (Hrsg.) ; PROBST, G. J. B. (Hrsg.): *Self-Organization and Management of Social Systems*. Berlin ; Heidelberg [u.a.] : Springer, 1984, S. 2 f.
- [Weg97] WEGNER, Peter: Why interaction is more powerful than algorithms. In: *Communications of the ACM* 40 (1997), Nr. 5, S. 80–91
- [Wol84] WOLFRAM, Stephen: Universality and complexity in cellular automata. In: *Physica D* (1984), Nr. 10
- [Wol85] WOLFRAM, Stephen: Twenty problems in the theory of cellular automata. In: *Physica Scripta* (1985), Nr. T9, S. 170–183