

Die approbierte Originalversion dieser Dissertation ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (<http://www.ub.tuwien.ac.at>).

The approved original version of this thesis is available at the main library of the Vienna University of Technology (<http://www.ub.tuwien.ac.at/englweb/>).



TECHNISCHE
UNIVERSITÄT
WIEN

VIENNA
UNIVERSITY OF
TECHNOLOGY

DISSERTATION

Ein computationales Modell für die Allgemeine Systemtheorie

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften unter der Leitung von

ao.Univ.-Prof. Dr.phil. Wolfgang Hofkirchner

E187

Institut für Gestaltungs- und Wirkungsforschung

eingereicht an der Technischen Universität Wien
bei der Fakultät für Informatik

von

Dipl.-Ing. Martin Schliefnig

Matrikelnummer: 0160919

Engerthstraße 214E/3/20, 1020 Wien

Wien, am 11. Oktober 2009

Kurzfassung

Die Welt und die Vorgänge, die sich in ihr abspielen, können im Rahmen von unterschiedlichen Systemtheorien beschrieben werden. Eine Allgemeine Systemtheorie geht davon aus, dass es bestimmte Eigenschaften und Regeln in allen Arten von Systemen – von der atomaren bis hinauf zu einer globalen, sozialen Ebene und weiter – gibt, die allgemein gültig sind. Gelingt es der Wissenschaft, eine derartige Allgemeine Systemtheorie hinreichend zu beschreiben, wäre dies nicht nur ein Vorteil für interdisziplinäre Arbeiten und damit für viele unterschiedliche Wissenschaftsgebiete, sondern würde auch ganz allgemein neue Perspektiven auf die Welt als solches und ihre Probleme erlauben.

Die Dissertation verfolgt das Ziel, ausgehend von existierenden Ansätzen, die notwendigen Begriffe und Konzepte einer Allgemeinen Systemtheorie in einem formalen, computationalen Modell zusammen zu fassen und die damit verbundenen Charakteristika zu untersuchen. Von besonderer Bedeutung sind dabei die Arbeiten des Biologen und Systemtheoretikers Ludwig von Bertalanffy und der von ihm mitgegründeten “Society for General Systems Research”, die im auch in dieser Arbeit als solches bezeichneten Bertalanffy-Programm zusammengeführt werden. Das formale Modell, das dabei im Rahmen der Dissertation konstruiert wird – Computational General System Theory (CGST) –, sowie die Ergebnisse in Hinblick auf praktische Anwendbarkeit und Komplexität desselben, sollen als Grundlage für die Fortführung des Bertalanffy-Programms im 21. Jahrhundert dienen. Die formale Abbildung und Konstruktion erfolgt daher basierend auf bestehenden Modellen der modernen Informatik wie Semantic-Web-Sprachen und Zellulären Automaten.

Ausgehend vom CGST-Modell erfolgt in der Arbeit schließlich eine wissenschaftstheoretische Auseinandersetzung mit den für eine Allgemeine Systemtheorie besonders wesentlichen Konzepten der Emergenz und Selbstorganisation. Dazu wird eine Abbildung der Arbeiten von Kurt Gödel zu unentscheidbaren Sätzen in der *Principia Mathematica* auf die Eigenschaften emergenter Systeme vorgenommen und mit der Notwendigkeit selbstreferenzieller Zustände in ebensolchen Systemen in Verbindung gebracht.

“Wer nicht die Idee der Physik (das heißt nicht die physikalische Wissenschaft selbst, sondern das vitale Weltbild, das sie geschaffen), nicht die historische und biologische Idee, jenen philosophischen Weltenplan sich zu eigen gemacht hat, der ist kein gebildeter Mensch. Und wenn keine außergewöhnlichen, spontanen Gaben diesen Mangel ausgleichen, so ist es mehr denn unwahrscheinlich, dass ein Mensch dieser Art ein wirklich guter Arzt, Richter oder Techniker zu sein vermag.”

Ortega y Gasset, *“Die Aufgabe der Universität”*,
gelesen bei Ervin Laszlo, *“Systemtheorie als Weltanschauung”*

Den gebildeten Menschen *Ludwig von Bertalanffy, Kurt Gödel, Ilya Prigogine, Erich Jantsch* und *Wolfgang Hofkirchner* für ihre vorangegangenen Arbeiten in großer Anerkennung und Dankbarkeit verbunden.

Für meinen Großvater,
einen ebenso gebildeten Menschen.

Inhaltsverzeichnis

1	Einleitung	5
1.1	Kurze Geschichte der Allgemeinen Systemtheorie	6
1.1.1	Das Bertalanffy-Programm	10
1.1.2	Warum es heute keine Wissenschaft der Allgemeinen Systemtheorie gibt	15
1.1.3	Im Dunstkreis einer Allgemeinen Systemtheorie: Luhmann, Kybernetik, Kelsen	17
1.2	Kritik bestehender Arbeiten	20
	Lars Skyttner: General Systems Theory.	21
	Hartmut Bossel: Systeme, Dynamik, Simulation.	22
2	Motivation	25
2.1	Wozu eine Allgemeine Systemtheorie?	26
2.1.1	“Projekt eine Welt”	26
	Climate Change.	27
	Schutz von Leben.	29
	Intelligent Design.	31
2.2	Wissenschaftsphilosophie einer Allgemeinen Systemtheorie	33
2.3	Praktischer Nutzen einer Allgemeinen Systemtheorie	35
	Metasprache für Informationssysteme	36
	Ganzheitliche Problemsicht	36

	Modernes humanistisches Weltbild	36
3	Systeme	37
3.1	Reductive Levels	38
3.2	Was ist ein System?	45
3.3	Systeminstanzen	51
3.4	Regellogik im statischen System	54
3.5	Dynamische Systeme	54
3.6	System und Umwelt	59
3.7	“Avoiding such unnecessary Duplication of Labor”	61
3.7.1	Ähnliche Systeme	62
3.7.2	Ähnliche Systeminstanzen	62
3.7.3	Ähnliches Systemverhalten	64
4	Computational General System Theory (CGST)	68
4.1	Ontologische Systembeschreibung	69
4.1.1	Eine System-Ontologie basierend auf der Web Ontology Language	71
4.1.2	Komplexität und Ausdrucksstärke	76
4.2	Semantic Web Rule Language als Regelkonzept für CGST	77
4.3	Von der Systemontologie zum Zellulären Automaten	80
4.3.1	NetLogo – Programmierung Zellulärer Automaten	82
4.3.2	Ableitung von NetLogo-Gerüsten aus OWL mittels XSL	85
4.4	Technische Umsetzung	87
4.4.1	Verwaltung von Γ , Δ und Θ in einem Triplestore	88
4.4.2	Scriptbasierte Erzeugung der OWL-basierten Automaten	93
5	Das Serendipity System	95

5.1	Ein einfaches System	96
	Simulation 1 – Leben und Sterben.	97
	Simulation 2 – Mutation und Selektion.	99
	Simulation 3 – Kooperation.	103
5.2	Ein komplexes System	105
5.3	CGST-Ähnlichkeits-Algorithmik am Beispiel von Lotka-Volterra-Systemen	110
	5.3.1 Erkennen von Lotka-Volterra-Abhängigkeitsmustern in CGST-OWL-Subgraphen	112
	5.3.2 Erkennen von Lotka-Volterra-Verhaltensmustern durch Trajektorienanalyse	113
6	Selbstorganisation und Emergenz	116
6.1	Emergenz: Wie kommt das Neue in die Welt?	117
	6.1.1 Formalisierbarkeit von Emergenz	118
6.2	Die Selbstorganisation des Universums	132
	Fischschwärme und Soziale Insekten.	134
	Dictyostelium discoideum.	136
	6.2.1 Selbstorganisation im CGST-Modell	139
	6.2.2 Selbstorganisation und Emergenz in Multi-Level-CGST-Instanzen?	141
6.3	Die Münchhausen-Vermutung der Selbstreferenz	146
7	Integration bestehender Wissensbasen	150
7.1	Proteine als Systeme	151
7.2	Kollaborative Ontologieentwicklung	155
	7.2.1 Collaborative Protege – Protege Plugin	156
	7.2.2 myOntology	156
	7.2.3 Ontoverse	158
8	Zusammenfassung	161

8.1	Motivation	162
8.2	Das CGST-Modell	163
8.3	Selbstorganisation und Emergenz	164
8.4	CGST-Weltbild und Ausblick	164
A	C-GST Ontologie im OWL-Format	166
B	CGST Transformationsscript Aufruf	170
C	NetLogo-Serendipity-Implementierung (Kooperation)	172
D	NetLogo-Implementierung für Prigogine's Brüsselator	177

Kapitel 1

Einleitung

*“Die Wahrheit wird gelebt, nicht doziert.”
Josef Knecht in Hermann Hesses “Glasperlenspiel”*

1943 veröffentlichte Hermann Hesse sein “Glasperlenspiel”. Das Werk erzählt die Geschichte des ‘Magister Ludi’ Josef Knecht, eines Meisters des in naher Zukunft konzipierten, abstrakten Glasperlenspiels. Ein einzelnes Glasperlenspiel besteht dabei aus der Herstellung von Analogien aus unterschiedlichen Wissenschaftsdisziplinen, wobei insbesondere die Mathematik und die Musik eine besondere Stellung einnehmen. Das Glasperlenspiel selbst beschreibt eine Menge an Regeln, die die Spieler anwenden, um Glasperlenspiele zu erstellen. Obwohl Hesses Intentionen, die er mit dem Werk verfolgte, vor allem in der Skizzierung eines Zustandes, in dem es nichts Neues mehr zu entdecken gibt und die Werbung für eine humanistische Weltauffassung waren, erlangte vor allem die Idee des Glasperlenspiels selbst – obgleich nie im Detail erklärt, sondern nur teilweise durch Beispiele angedeutet – weltweite Bekanntheit.

Als Ludwig von Bertalanffy 1969 sein “General System Theory”-Kompendium publizierte, kam er um einen Verweis auf Hesses “Glasperlenspiel” (und Nikolaus von Kues “De ludo globi”, in dem Analogien zwischen einem im 15. Jhdt. üblichen Spiel und den Abläufen in der Welt hergestellt werden) nicht herum¹:

“ The literary gourmet may remember Nicholas of Cusa’s ‘De ludo globi’ (1463) and Hermann Hesse’s Glasperlenspiel, both of them seeing the working of the world reflected in a cleverly designed, abstract game.”

Was in Hesses Szenario nur noch ein Spiel ist, weil die Gegebenheiten der

¹Siehe [3], Seite 11.

Welt ohnehin bereits vollständig erkundet sind, ist für Bertalanffy das Hauptanliegen seiner Allgemeinen Systemtheorie²:

“In fact, similar concepts, models and laws have often appeared in widely different fields, independently and based upon totally different facts. There are many instances where identical principles were discovered several times because the workers in one field were unaware that the theoretical structure required was already well developed in some other field. General system theory will go a long way towards avoiding such unnecessary duplication of labor.”

Während Hesses Magister Ludi darum bemüht ist, möglichst schöne Analogien aus unterschiedlichen Wissenschaftsbereichen, insbesondere in Verbindung mit der Musik und der Mathematik, zu gestalten, ist es das Anliegen einer Allgemeinen Systemtheorie, eine Theorie, Modelle und Methoden zur Verfügung zu stellen, um derartige Analogien zu formalisieren und automatisiert ableiten zu können. Also tatsächlich neues Wissen zu erzeugen, um das erneute Herausarbeiten in einem Wissenschaftszweig zu vermeiden, wenn die dadurch entstehende Kenntnis bereits als Wissen in einem anderen Wissenschaftszweig formuliert ist. Auf diese und weitere Motivationen soll im nächsten Kapitel noch genauer eingegangen werden.

1.1 Kurze Geschichte der Allgemeinen Systemtheorie

Da es nicht das Ziel dieser Arbeit ist, die Entstehungsgeschichte des Konzepts einer Allgemeinen Systemtheorie aufzuarbeiten, können wir an dieser Stelle nur einen kurzen und sicherlich unvollständigen Überblick über die ‘Meilensteine’ geben. Der interessierte Leser sei aber auf die sehr umfassende und sorgfältige Arbeit von Klaus Müller³ und die Arbeiten eines von Rupert Riedl initiierten Programms⁴ zur Relevanz des Bertalanffy-Programms hingewiesen.

Sehr oft wird im Zusammenhang mit dem Konzept einer Allgemeinen Systemtheorie und hier insbesondere mit der Eigenschaft der Emergenz, eine Verbindung zu Aristoteles’ Satz “Das Ganze ist mehr als die Summe der Teile” aus den Arbeiten zur Metaphysik hergestellt. Obgleich wir später den Begriff der Emergenz als Systemeigenschaft noch genauer untersuchen werden, ist diese Assoziation mit der griechischen Philosophie doch etwas leichtfertig hergestellt.

²Siehe [3], Seite 33.

³Siehe [34], insbesondere Kapitel 3, 11 und 12.

⁴Siehe dazu eine Einleitung in [17].

Noch früher und auch deutlich “systemischer” hat sich der griechische Philosoph Empedokles mit dem Lauf der Dinge auf diese Art beschäftigt. Empedokles erklärte die Entstehung und den Zerfall von Lebewesen und Dingen aus den vier Elementen Feuer, Luft, Wasser und Erde mittels der Kräfte Liebe und Hass. Während die Liebe für die Vermischung von Elementen und die dadurch entstehende Neubildung von Dingen verantwortlich ist, ist es der Hass für die Dekonstruktion von Dingen. Aus diesem Ansatz leitete Empedokles auch als erster Philosoph eine Grundelement-Lehre ab, die so etwas wie kleinste Teile, aus denen alle Dinge der Welt und des Universums zusammengesetzt sind – wenngleich er diese Idee auch nicht sehr tiefgreifend verfolgte oder gar den Begriff des Atoms oder ein vergleichbares Konzept beschrieb. Von Empedokles lässt sich übrigens eine Brücke zu Bertalanffy schlagen: und zwar über Hölderlin, der das Schicksal des Empedokles in einem seiner Gedichte beschrieb⁵.

Aristoteles beruft sich in seiner Metaphysik auch auf die Elementen-Lehre des Empedokles, lehnt diese aber in ihrer konkreten Ausprägung ab. Doch auch Aristoteles beschäftigt sich mit dem Ursprung allen Seins und begibt sich auf die Suche nach einer “Ersten Philosophie” (er selbst spricht ja nie von einer Metaphysik), deren Aufgabe sich auch stark mit jener einer Allgemeinen Systemtheorie überschneidet⁶.

“Zuerst also werde die Frage, die wir als erste aufwarfen, behandelt, nämlich ob es Aufgabe einer oder mehrerer Wissenschaften ist, alle Gattungen der Ursachen zu betrachten. Wie aber könnte es wohl Aufgabe einer Wissenschaft sein, Prinzipien, die nicht entgegengesetzt sind, zu kennen?”

Aristoteles Betrachtungen führen über eine naturwissenschaftlich begründete und umfassende Philosophie hin zu theologischen Überlegungen und verlieren dabei die Konstruktion einer Allgemeinen Theorie über die Regeln und Prinzipien der Welt immer stärker aus den Augen. Er ist sich dabei aber sehrwohl über die Bedeutung, aber eben auch über die Schwierigkeit einer “Wissenschaft vom Allgemeinen” bewusst und stimmt in der Argumentation mit seinen späteren Nachfolgern aus dem Bertalanffy-Programm überein, wenn er sagt⁷:

“Notwendigerweise trifft darunter das Merkmal, alles zu wissen, auf den zu, der am meisten über die Wissenschaft vom Allgemeinen

⁵Hölderlins Empedokles beschreibt den Tod des Philosophen, der sich laut historischen Überlieferungen in den Ätna stürzte. Diese Version vom Selbstmord des Empedokles wird in der Geschichtsschreibung jedoch angezweifelt, was Hölderlin aber nicht an der literarischen Umsetzung hinderte. Bertalanffy befasste sich in jungen Jahren mit diesem Werk Hölderlins – siehe dazu Ludwig von Bertalanffy, “Hölderlins Empedokles”, Zeitschrift für Ästhetik und allgemeine Kunstwissenschaft 20, Seite 241-248.

⁶Aristoteles, “Schriften zur ersten Philosophie”, III. Buch, Absatz 2.

⁷Aristoteles, “Schriften zur ersten Philosophie”, I. Buch, Absatz 2.

verfügt; denn dieser kennt gewissermaßen alles, was dem Allgemeinen untergeordnet ist. Doch gerade dies, das Allgemeinste, ist für die Menschen am schwierigsten zu erkennen; ist doch der Abstand zu den Sinneswahrnehmungen am weitesten.”

Wenn wir uns nun auf die Suche nach den Vorläufern einer holistischen oder universalen Betrachtung und Beschreibung der Welt abseits der Theologie begeben, so finden wir bei dem bereits in der Einleitung kurz erwähnten Nikolaus von Kues aber vor allem bei Immanuel Kant und Leibnitz entsprechende Gedanken. Mit Nikolaus von Kues und seinem Werk “De ludo globi” hat sich Bertalanffy schon sehr früh in [2] beschäftigt.

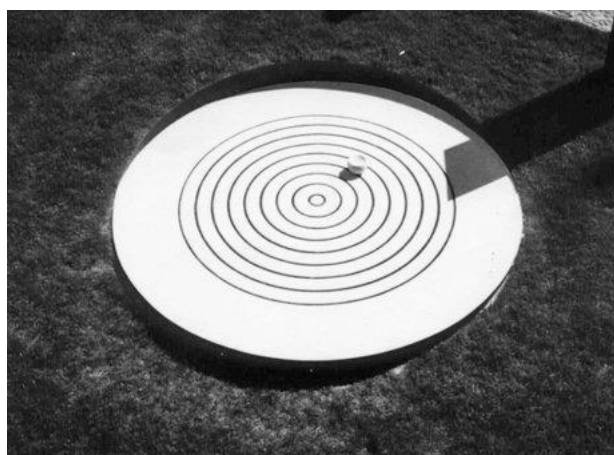


Abbildung 1.1: Im Hof des Diözesanmuseums in Brixen findet sich eine leicht adaptierte Nachbildung des Globus- oder Kugelspiels. Bildquelle: <http://www.hofburg.it>

In “De ludo globi” geht es um eine philosophische Unterhaltung über den höheren Sinn des so genannten Globusspiels, ein Spiel, bei dem ein halb geöffneter Globus auf einer schneckenartigen Kreisbahn so gerollt werden muss, dass er schließlich die Mitte erreicht (siehe 1.1). Die unterschiedlichen spielbestimmenden Faktoren werden in dem Dialog dabei mit den Herausforderungen des Lebens verglichen⁸. Der Dialog wird dabei von Johann, Herzog von Bayern, und Nikolaus, Kardinal des Titels von Sankt Peter in Ketten, geführt. Eine Rezension findet sich in [2]. Zur Bedeutung des Spiels heißt es im Dialog etwa⁹:

“Johann: Jeder Globus ist von eigener Art, anders als der andere gekrümmt. Also kann einer dem anderen nicht folgen.

Kardinal: Das ist wahr. Keiner kann genau der Spur des anderen folgen. Sondern es ist notwendig, dass jeder die Neigungen seines Globus und die Leidenschaften beherrscht, dadurch dass er sich selbst

⁸Siehe [27]

⁹Siehe [27], Seite 59 f.

übt; [...] Dies ist die mystische Kraft des Spieles: Durch tüchtige Übung kann sogar ein gekrümmter Globus reguliert werden, so dass nach vielen unstillen Biegungen die Bewegung im Reich des Lebens zur Ruhe kommt. ”

Interessant ist aber – man bedenke den frühen Entstehungszeitpunkt des Werks im 15. Jahrhundert – vor allem auch die Analogiebildung und die systemische Sichtweise auf die Welt, die sich in dem Gespräch erkennen lässt¹⁰:

“*Johann*: Nach dieser Meinung kann man wohl zugeben, dass die Welt dreifaltig ist; die kleine, der Mensch, die größte, die Gott ist, die Universum genannt wird; die kleine ist Gleichnis der großen, die Große Gleichnis der größten. Doch ich bin im Zweifel, ob der Mensch, da er kleine Welt ist, auch Teil der großen Welt sei.

Kardinal: Der Mensch ist durchaus in der Weise kleine Welt, dass er auch Teil der großen ist. Denn in allen Teilen strahlt das Ganze wider, weil der Teil Teil des Ganzen ist. So ist der Widerschein des ganzen Menschen in der Hand, die in Proportion zum Ganzen steht, aber dennoch strahlt im Kopf die ganze Vollkommenheit des Menschen in besserer Weise wider; ebenso ist der Widerschein des Universums in jedem seiner Teile. [...]

Johann: Wenn ich das richtig verstehe, dann ist, so wie das Universum ein großes Reich ist, auch der Mensch ein Reich, aber ein kleines in dem großen Reich, so wie das Reich Böhmen im römischen Reich oder universalen Kaiserreich. ”

Ob des frühen Entstehungszeitpunkts der Arbeit von Bertalanffy über Nikolaus von Kues im Jahr 1928 wäre es interessant zu erfahren, ob Bertalanffy sich mit “De ludo globi” beschäftigt hat, weil er damals bereits überzeugt von einem holistischen Ansatz war, oder aber ob ihn ebendieses Werk einen Denkanstoß in diese Richtung geliefert hat. Keine der hier erwähnten Biografien, [14] und [22], geht auf diesen besonderen Aspekt ein, weshalb es schwierig ist, dieses Detail zu beurteilen – vermutlich handelt es sich hier um eines jener wenigen Geheimnisse, die Bertalanffy nicht in einer seiner zahlreichen Publikationen erwähnt, sondern mit ins Grab genommen hat.

Um von Kues zum nächsten Meilenstein der Entwicklung systemischen Denkens zu gelangen, ist ein großer Sprung notwendig. Gottfried Wilhelm Leibniz wirkte zu Beginn des 18. Jahrhunderts, sein Kollege Immanuel Kant in dessen zweiter Hälfte. Beide trugen dazu bei, eine durch die Mathematik und neue Form der Metaphysik revolutionäre Weltansicht zu schaffen, die auch heute noch wichtige Fundamente westlicher Denkweisen, Wissenschaftsprinzipien und Gesellschaftsstrukturen bildet. Und beide verfolgten dabei gewissermaßen einen

¹⁰Siehe ebenda, Seite 43 f.

systemischen Ansatz: Eine entsprechende Aufarbeitung der begrifflichen und konzeptuellen Überschneidungen soll an dieser Stelle nicht erneut aufgearbeitet werden, weil sie in [34] sehr gut beschrieben ist¹¹.

Um nach den wirklichen Anfängen einer Allgemeinen Systemtheorie auch im Sinne ihrer eigentlichen Intention und Zielsetzungen zu suchen, ist ein weiterer Sprung an den Beginn des 20. Jahrhunderts notwendig. Vorarbeiten auf diesem Gebiet leisteten dabei etwa Martin Heidegger mit seinen Untersuchungen über den ontologischen Seinsbegriff und die Stellung des Seins in der Zeit sowie Max Weber mit seiner neuen Form der Soziologie und der Reduktion sozialer Systeme auf Elemente sozialen Handelns. Diese und weitere Denker wie Edmund Husserl, aber auch die Kritik am Wiener Kreis und die Diskussionen mit seinem Doktorvater Moritz Schlick bereiteten den Unterboden für Bertalanffys Programmatik einer Allgemeinen Systemtheorie.

1.1.1 Das Bertalanffy-Programm

Als geistiger Vater eines allgemeinen systemischen Ansatzes in einer hinreichenden Deutlichkeit kann zweifelsohne der Wiener Biologe Ludwig von Bertalanffy genannt werden. Davidson nennt¹² das Jahr 1937 und den Ort der Universität von Chicago für die erste öffentliche Präsentation Bertalanffys seiner Allgemeinen Systemtheorie. Aufgrund von Kritik hat dieser aber für fast 20 Jahre seine Ansätze auf diesem Gebiet anschließend im Geheimen fortgeführt und erst, als er 1954 bei einem Mittagessen mit Boulding und Rapoport, auf die gleich noch zu sprechen kommen sein wird, eine Bestätigung für seinen Ansatz erhält, die öffentliche Arbeit daran wieder ernsthaft aufgenommen. In den darauffolgenden Jahren entwickelte sich eine Gruppe von Wissenschaftern, die gemeinsam mit ihm das Thema bearbeiteten und schließlich auch eine 1957 in "Society for General Systems Research" umbenannte Organisation zu diesem Zwecke gründeten – die Jahresschriften dieser Organisation werden in dieser Arbeit noch an einigen Stellen von entscheidender Bedeutung sein. Ob seiner dokumentierten theoretischen Pionierarbeit auf diesem Gebiet und aufgrund seines Einsatzes und des Umstands, dass Bertalanffy seine Schwerpunktsetzung auf dieses Thema legte, ist es sicher nicht falsch, die um ihn herum entstandenen Arbeiten zu dieser Zeit unter dem Begriff "Bertalanffy-Programm" zusammen zu fassen¹³.

Alles ist im Fluss, nichts bleibt so, wie es ist. Dieser Gedanke, der auch schon in der Antike eine zentrale These philosophischer Überlegungen bildete, ist einer der Eckpfeiler der Bertalanffy'schen Allgemeinen Systemtheorie, die sich aus einer "Allgemeinen Theorie offener Systeme und des Fließgleichgewichts" in sei-

¹¹Siehe [34], Seite 27 ff.

¹²Siehe [14], Seite 136 f.

¹³Dieser Begriff wird unter anderem auch von Müller in [34] verwendet.

nen frühen Arbeiten herausbildete. Vom Molekül, über Lebewesen und politische Systeme bis hin zu Galaxien und noch weiter. Vom einfachen Fortbewegungslauf einer Kugel auf einer Schräge bis hin zum menschlichen Bewusstsein. Alle Vorgänge in der Natur sind einer kontinuierlichen Veränderung unterworfen und bedingt durch den Austausch von Energie und Materie. Schränkt man die Beobachtungen im Rahmen wissenschaftlicher Untersuchungen auf Teilmengen solcher Prozesse oder Objekte ein, die man als System betrachtet, so erhält man ein Modell eben jener offenen Systeme, die sich im Fließgleichgewicht befinden – und die, wie Bertalanffy bereits in [4] beschreibt, wichtige gemeinsame Eigenschaften besitzen. Auf die Beschreibung solcher Eigenschaften, die eine dahinter liegende Allgemeine Systemtheorie erforderlich oder zumindest wünschenswert machen, legt Bertalanffy auch bei seinen darauf folgenden Arbeiten ganz besonderen Wert: Wachstum, Stoffwechsel, Homöostase und Rückkopplung sind nur einige der wesentlichen Begriffe in diesem Zusammenhang, die von Bertalanffy in seinen Publikationen in bedeutenden Wechselwirkungen beschrieben wurden und so oft begriffliche Weiterentwicklungen erfuhren.

Wissenschaftler aus unterschiedlichen Disziplinen haben sich am Bertalanffy-Programm beteiligt und die Ansätze von Bertalanffy teilweise weiterentwickelt, teilweise aber auch vollkommen neue Aspekte konstruiert. Einer der bedeutendsten ist Anatol Rapoport, auf dessen Ansatz einer Allgemeinen Systemtheorie wir hier ebenfalls kurz eingehen wollen. Ähnlich wie Bertalanffy sah auch Rapoport in der Möglichkeit, durch Analogiebildung die Wissenschaft zu bereichern, ein enormes Potential. So schreibt er in seinem wichtigsten Werk zu diesem Thema einleitend¹⁴:

“Die Allgemeine Systemtheorie lässt sich als ein Programm mit dem Ziel auffassen, die Analogiebildung in theoretisch fruchtbare Kanäle zu lenken.”

Rapoport, wie Bertalanffy ebenfalls stark an der Biologie und ihrer theoretischen Beschreibung interessiert, nähert sich dem Thema der Allgemeinen Systemtheorie in erster Linie mathematisch und hier sehr stark mit einem Fokus auf gesellschaftliche Vorgänge. Für die Analogiebildung erwähnt er – im Gegensatz zu Bertalanffy, der fast ausschließlich auf konzeptueller Ebene argumentierte – in seinem GST-Werk ganz konkrete Beispiele, die er auch mathematisch beschreibt.

Ein klassisches Beispiel, in dem er die Wirksamkeit von Analogien über verschiedene Wissenschaftsdisziplinen hinweg reflektiert sieht, sind dabei Katastrophenszenarien¹⁵ in Systemen im Allgemeinen und die sogenannte Cusp- oder Spitzenkatastrophe im Speziellen. Eine Cusp-Katastrophe ist ein Szenario

¹⁴Siehe [38], Seite 1.

¹⁵Siehe [38], Seite 67 ff.

in einem dynamischen System, das den kontinuierlichen oder plötzlichen Übergang eines Systems in einen Extremzustand beschreibt, wobei im Gegensatz zu einem einfachen Katastrophenszenario zwei unterschiedliche Katastrophenzustände erreicht werden können.

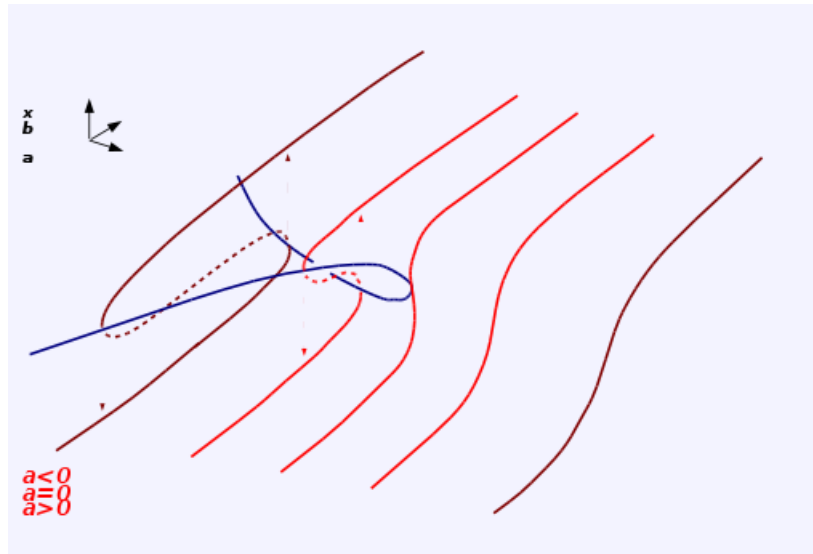


Abbildung 1.2: Graph einer Cusp-Katastrophe. Bei $a < 0$ ist ein Zustandsübergang in beide Katastrophenzustände möglich. Bildquelle: Wikimedia Commons.

Rapoport nennt nun als Beispiele für ein solches Szenario Probleme aus unterschiedlichen Wissenschaften: So genannte “Ambivalente Zeichnungen” aus dem Bereich der Psychologie; einen durch Stress verursachten Erregungszustand bei Hunden; und schließlich Bayes’sche Entscheidungsmodelle für Entscheidungen unter Unsicherheit. Eine Cusp-Katastrophe, wie sie auch in Abbildung 1.2 grafisch veranschaulicht ist, wird mathematisch durch eine Funktion der Form

$$V = ax^4 + ax^2 + bx$$

beschrieben, wobei a und b als Kontrollparameter der Funktion bezeichnet werden und entscheidende Eigenschaften des Systems in Hinblick auf seine Stabilität darstellen. Interessant für “echte” Spitzenkatastrophen sind dabei – wie man aufgrund der Beschaffenheit der Potentialfunktion sieht – jene Systemkonfigurationen, in denen $a < 0$ gilt. Beim Beispiel mit dem Erregungszustand eines Hundes etwa beschreibt a den aktuellen Stress- bzw. Erholungsfaktor, wobei $a > 0$ für einen erholten, $a < 0$ für einen gestressten Zustand steht. Je nachdem, wie stark der Hund nun mit wahrgenommenen Provokationen (über b abgebildet) konfrontiert wird, kann das Verhalten in zwei verschiedenen Extremzuständen, Angst und Wut, und den dazugehörigen Aktionen, Flucht und Angriff, eskalieren.

Als Analogie sieht Rapoport – und bezieht sich dabei auf die Arbeiten zu



Abbildung 1.3: Beispiel für eine Kippfigur. Quelle: Wikimedia Commons.

diesen Themen von Poston und Stewart bzw. Zeeman – eine Thematik aus der Gestaltpsychologie, so genannte ambivalente Zeichnungen oder Kippfiguren. Hierbei handelt es sich um Bilder, die durch unscharfe Aspekte zwei unterschiedliche Interpretationen über deren Inhalt zulassen. Ein Beispiel ist in Abbildung 1.3 dargestellt: Die Zeichnung kann als der zur Seite geneigte Kopf einer jungen Frau oder als der Kopf einer alten Frau wahrgenommen werden. Der entscheidende Kontrollparameter a beschreibt hier den Detailgrad der Zeichnung: Würden entweder mehr Details eines Kopfes einer alten Frau oder eines Oberkörpers einer jungen Frau skizziert werden, wäre die Wahrscheinlichkeit, ein zweites Objekt darin zu sehen, weit geringer. Rapoport veranschaulicht das mit einem selbst gezeichneten Beispiel in Form einer Betrachtungsmatrix sehr gut¹⁶.

Der dritte Anwendungsfall befindet sich in der Domäne der Entscheidungstheorie und beschreibt Entscheidungsverhalten unter Unsicherheit, d.h. in solchen Situationen, wo eine oder mehrere Auswirkungen bzw. die Wahrscheinlichkeit ebendieser, bei einer Entscheidung zwischen zwei Alternativen im Vorhinein nicht bekannt sind. Je höher der Grad der Unsicherheit, desto größer ist grundsätzlich die Wahrscheinlichkeit, dass beide Alternativen in Frage kommen können. Sind jedoch im Vorhinein die meisten Auswirkungen bekannt, so sinkt diese Wahrscheinlichkeit. Für eine genauere Erläuterung sei wieder auf Rapoport¹⁷ verwiesen.

Rapoport sieht diese Szenarien als Beispiel für die Möglichkeit und Sinnhaftigkeit einer Analogiebildung bei (komplexen) Systemen. Ist das Verhalten ei-

¹⁶Siehe [38], Seite 74 f.

¹⁷Siehe [38], Seite 76 f.

ner Cusp-Katastrophe in einem Bereich bereits relativ gut beschrieben, können Wissenschaftler anderer Domänen, wenn sie auf dieses Problem stoßen, unter Umständen bereits sehr viel aus den verwandten Arbeiten lernen (und unnötige doppelte Arbeit vermeiden, würde Bertalanffy jetzt anmerken). Vorausgesetzt natürlich, sie wissen überhaupt, dass es derartige verwandte Arbeiten bereits gibt. Oder, worauf hier ja hingearbeitet werden soll, sie haben ein Informationssystem zur Verfügung, das durch eine einheitliche Art der Beschreibung von Systemverhalten Abfragen über ähnliche Probleme erlaubt.

Während Bertalanffy und Rapoport in ihren Arbeiten stets auf biologische Vorgänge fokussiert waren und daraus dann gesellschaftliche oder philosophische Ableitungen versuchten, beschäftigte sich Kenneth Boulding mit dem Ansatz einer Allgemeinen Systemtheorie in Hinblick auf die Anwendbarkeit in den Wirtschaftswissenschaften. Boulding war Ökonom und versuchte die Vorgänge aus der Natur für die Funktionsweise von Gesellschafts- und Wirtschaftssystemen zu übersetzen und stellte dabei insbesondere die Evolution und ihre Vorgänge in den Vordergrund. Er war überzeugt von einer Art emergentem Sprung, der sich mit der Ausbildung der menschlichen Art einstellte und wies dem menschlichen Handeln in seinen Arbeiten daher stets eine besondere Rolle zu. Boulding spricht hier zwar nicht von Emergenz, formuliert das aber eindeutig in diese Stoßrichtung¹⁸:

“When the know-how of the genosphere had risen to the point where it could produce a human being, that is, Adam and Eve, or whatever their names were, evolution on this planet went into a new gear. Two billion years or so earlier, the introduction of life had set the dynamics of earth into a new gear.”

Boulding sieht die Entstehung des Menschen und den Beginn gesellschaftlicher Evolution als qualitativ auf der selben Ebene befindlichen Sprung wie die Entstehung des Lebens selbst. Die gesellschaftliche Evolution sieht er als eine “Evolution von Artefakten”, wobei die Sprache das wichtigste und bedeutendste dieser Artefakte darstellt.

Boulding hatte stets ein großes Naheverhältnis zu Bertalanffy und Rapoport, was sich auch in zahlreichen Veröffentlichungen in den System-Science-Publikationen widerspiegelt. Neben Rapoport und Boulding, die zu den engsten Vertrauten zählten, hatte Bertalanffy noch – zu Beginn wenig, dann mehr, am Ende wieder weniger – weitere Weggefährten, deren Anbindung und Ausrichtung wohl am besten in den System-Science-Publikationen dokumentiert ist. Fakt ist jedoch, dass es heute trotz einiger systemtheoretisch fokussierter Organisationen (wie auch der ISSS¹⁹ als Nachfolgeorganisation der Society for

¹⁸Siehe [11], Seite 121 ff.

¹⁹International Society for the Systems Sciences, siehe <http://www.iss.org>.

General Systems Research) keine umfassenden Programme zur Erarbeitung einer Allgemeinen Systemtheorie und schon gar keine große Öffentlichkeit, wie sie etwa für die Quantenphysik oder die Evolutionstheorie besteht, für dieses Thema gibt.

1.1.2 Warum es heute keine Wissenschaft der Allgemeinen Systemtheorie gibt

In den vergangenen 150 Jahren bediente man sich beim Lösen von Problemen in unterschiedlichen Wissenschaftsbereichen vermehrt der Anwendung von Differentialgleichungen, da diese bis dahin nicht darstellbare Verhaltensweisen der Natur sehr gut beschreiben konnten – zurecht kann die genaue mathematische Beschreibung dieser Gleichungsform daher als eine maßgebliche und revolutionäre Entdeckung in der Geschichte der Wissenschaft bezeichnet werden. Differentialgleichungen können eingesetzt werden, um Wachstum in Bakterienkulturen oder Ökosystemen zu beschreiben, Schwingungen und ihr Verhalten darzustellen oder um etwa in der Elektrotechnik Strom- und Spannungsverläufe mit Speicherelementen zu berechnen.

Gleichzeitig lassen sich auch Muster in unterschiedlichen Wissenschaftsgebieten erkennen, die mit der gleichen oder sehr ähnlichen Differentialgleichungen beschrieben werden können. Dieser Umstand veranlasste Ludwig von Bertalanffy Zeit seines Lebens dazu, auf diese Form der formalen Repräsentation von Wissen seine Allgemeine Systemtheorie hin auszurichten. Ein Beispiel, an dem Bertalanffy das beschreibt, sind Wachstumsfunktionen²⁰.

“In mathematics, the exponential law is called the ‘law of natural growth’ and with ($a_1 > 0$) is valid for the growth of capital by compound interest. Biologically, it applies to the individual growth of certain bacteria and animals. Sociologically, it is valid for the unrestricted growth of plant or animal population, in the simplest case for the increase of bacteria when each individual divides into two, these into four, etc. In social science, it is called the law of Malthus and signifies the unlimited growth of a population, whose birth rate is higher than its death rate.”

Differentialgleichungen und insbesondere Differentialgleichungssysteme sind aber mit einer Reihe von theoretischen und praktischen Problemen verbunden, auf deren Lösung Bertalanffy zur Zeit der Ausformulierung seines Ansatzes vertraut hat, die aber bis zum heutigen Zeitpunkt und trotz der großartigen Leistungen der Angewandten Informatik nach wie vor bestehen. Theoretische Probleme ergeben sich dabei aus der Natur der Differentialgleichungen an sich:

²⁰Siehe [3], Seite 62.

Aufgrund der Heterogenität der Differentialgleichungen und ebensolcher Systeme, ist es nicht möglich, a priori eine Lösungsmethode zu bestimmen, auch können Probleme in Bezug auf Annahmewerte (Anfangsrandwertprobleme) auftreten. Je verschachtelter ein Differentialgleichungssystem ist, umso schwieriger ist die Auflösung. Insbesondere im Bereich von partiellen Differentialgleichungen sind für manche Typen (noch) gar keine mathematischen Lösungsverfahren bekannt, die die Möglichkeit oder gar die Ausprägung einer Lösung auf formal-theoretischer Ebene herleiten können (wohl aber praktisch simuliert mittels numerischer Lösungsverfahren). Ein Beispiel, das sich auch in GST-Publikationen findet, sind Strömungsmechanik-Probleme. Eine spezielle Klasse bilden hier jene Grundgleichungen, die mittels Navier-Stoke-Gleichungen beschrieben werden: Für diese speziellen Differentialgleichungen wurde bis dato keine Herleitung von globalen Lösungen gefunden.

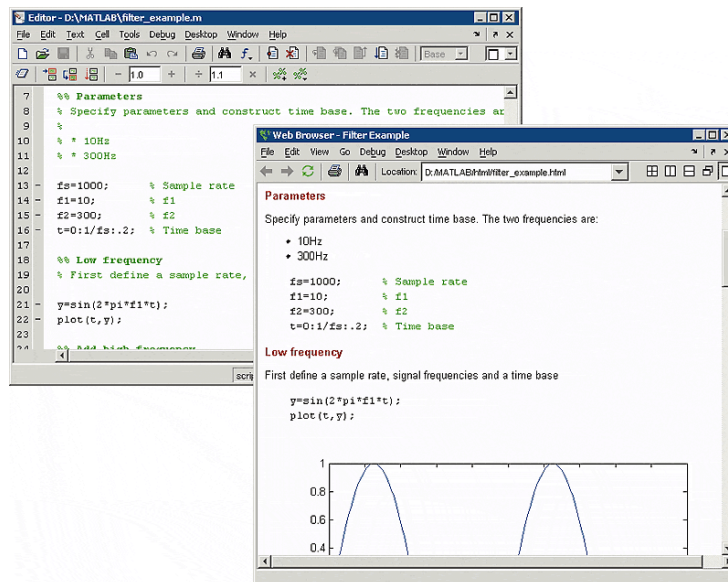


Abbildung 1.4: Entwicklung in MATLAB.

Viel wesentlicher als die theoretischen Probleme, die sich ja – mit Ausnahme von Berechnungskomplexitätsproblemen – meist nur auf spezielle Fälle beschränken, scheinen die praktischen Probleme, die bei der Simulation von komplexem Verhalten mittels Differentialgleichungen auftreten. Die Informatik ist heute in der Lage, auf Anwendungsebene Tools und Hilfsmittel zur Verfügung zu stellen, die Verschachtelungen von Differentialgleichungen sehr einfach modellieren lassen, zusätzlich gibt es mächtige Programme wie MATLAB (siehe Abbildung 1.5) und Mathematica (siehe Abbildung 1.4), die Berechnungsklassen und Simulationen für die gängigsten Probleme bieten.

Ein praktisches Arbeiten im Rahmen einer vergleichenden Allgemeinen Systemtheorie würde sich aber trotzdem enorm schwierig gestalten, da es (1) keine gemeinsame formale Sprache zur Wissenrepräsentation auf einer syntaktischen oder semantischen Ebene gibt, (2) der Ansatz, Systeme durch Differential-

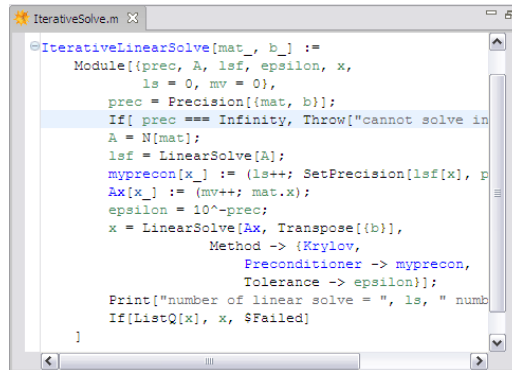


Abbildung 1.5: Entwicklung in Mathematica.

gleichungen und ihr mathematisches Verhalten zu repräsentieren grundsätzlich schon einmal ein analytischer und kein konstruktiver ist und damit mit jedem zusätzlichen Schritt immer schwieriger zu handhaben ist und (3) das Denken in mathematischen Funktionen, das zur Modellierung auf diese Art notwendig ist, zwar Vertreter jener Wissenschaftsbereiche, die tagtäglich damit konfrontiert sind, leicht fällt, mit Sicherheit aber kein mehrheitsfähiger Modellierungsansatz ist.

Freilich gibt es auch auf dem Gebiet der mathematisch fokussierten computationalen Modellierung große Erfolge. Der jüngste ist die Suchmaschine **wolframalpha.com** des Mathematica-Erfinders Stephen Wolfram, die, basierend auf effizienten Suchalgorithmen und der Kalkulation durch Zelluläre Automaten in der Lage ist, zahlenbezogene Suchabfragen aller Art mathematisch zu übersetzen und zu beantworten.

1.1.3 Im Dunstkreis einer Allgemeinen Systemtheorie: Luhmann, Kybernetik, Kelsen

Aber auch mehr oder weniger unabhängig von Bertalanffy und seinem Programm hat es Entwicklungen von universellen Systemtheorien gegeben. Einer der prominentesten Vertreter dabei ist der deutsche Soziologe Niklas Luhmann²¹, der durch seine Arbeit tiefreichende Spuren hinterlassen hat und dessen Theorie sozialer Systeme in einer solchen Aufzählung nicht fehlen sollte.

Es wird in Kapitel 3 noch eine detaillierte, kritische Auseinandersetzung mit Luhmanns Grundkonzepten geben. An dieser Stelle soll aber vorerst neutral die Intention Luhmanns und die Kernidee seiner Arbeit kurz vorgestellt werden.

²¹Völlig unabhängig war freilich auch Luhmanns Ideenfindung nicht von den Ansätzen des Bertalanffy-Programms, so hat er etwa ebenfalls in den System-Science-Publikationsreihen veröffentlicht.

Luhmann beginnt das Vorwort zu seinem Hauptwerk²², an dessen Fragmenten er 30 Jahre lang gearbeitet hat, mit einer Rechtfertigung für sein umfassendes Theorie-Programm:

“ Die Soziologie steckt in einer Theoriediskussion. Eine im ganzen recht erfolgreiche empirische Forschung hat unser Wissen vermehrt, hat aber nicht zur Bildung einer facheinheitlichen Theorie geführt.”

Diese Theoriediskussion zu überwinden war eines der Ziele, die Luhmann mit seiner (in Bezug auf die Themenfelder soziologischer Wissenschaften) universellen Systemtheorie erreichen wollte. Zwar ist ihm das zu Lebzeiten nicht gelungen, doch erfahren Luhmanns Arbeiten heute eine immer stärkere Rezeption und haben Eingang in die Studien von Soziologen, Politikwissenschaftlern und Wirtschaftsforschern gefunden. Insofern wurde Luhmann's Systemtheorie bisher sicherlich nicht zur einheitlichen Theorie der Soziologie, wohl aber zu einer ihrer stärksten und bedeutendsten Grundthesen. Und: Der Trend, die Soziologie systemisch zu erfassen, hält nach wie vor an und ist sicherlich auf die Arbeiten von Luhmann und Talcott Parsons zurück zu führen.

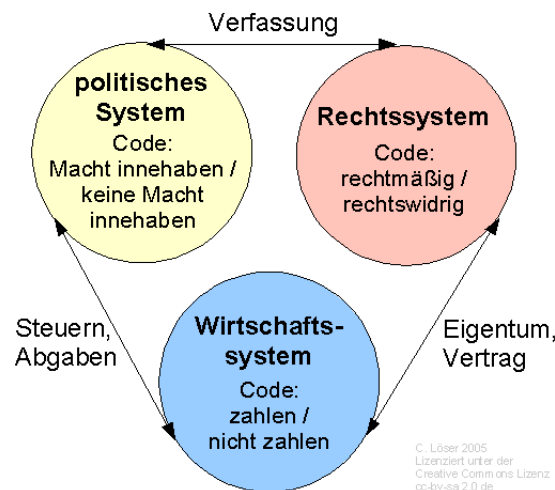


Abbildung 1.6: Schematische Darstellung der strukturellen Kopplung zwischen einzelnen sozialen Systemen bei Luhmann. Bildquelle: Wikimedia Commons.

Wie sieht dieser Grundentwurf von Niklas Luhmann aber jetzt aus? Luhmann erkennt, dass die Kommunikation das wesentliche Element in Sozialen Systemen darstellt und gründet auf dieser Feststellung seine Theorie. Unterschiedliche soziale Systeme bedienen sich seiner Ansicht nach so genannter symbolisch generalisierter Kommunikationsmedien, die Kommunikation wahrscheinlich machen und dadurch zu einer Ausdifferenzierung eines bestimmten sozialen Systems führen. So funktionieren Politische Systeme über das Kommunikationsmedium Macht, Wirtschaftssysteme über das Medium Geld usw. Seiner Ansicht nach müssen diese Teilsysteme dann auch unabhängig voneinander

²²Siehe [31].

betrachtet werden, da die Ausdifferenzierung dermaßen radikal und stark pas-
siert, dass die Schnittstellen zu anderen Teilsystemen fast vernachlässigbar sind.
Daraus ergibt sich auch der zweite wesentliche Kernpunkt in Luhmanns Ansatz:
Die Differenzierung zwischen System und Umwelt, d.h. etwa die Feststellung,
dass das Wirtschaftssystem alles beinhaltet, was über das Kommunikationsme-
dium Geld transportiert werden kann, politische Prozesse aber grundsätzlich
außerhalb dieses Systems stehen und daher nur zur Umwelt gehören und nicht
in entscheidende Analysen mit einbezogen werden können. Diese grundsätzlich
problematischen Aussagen für eine Allgemeine Systemtheorie werden aber zu
geeigneter Stelle in der Arbeit noch einmal aufgegriffen und diskutiert werden.

Vor Luhmann kam bereits Talcott Parsons mit einer ähnlichen Idee. “Tal-
cott Parsons war der – vorläufig – letzte Altmeister der Soziologie, der ver-
suchte in einem Schema das Gesamthänomen Gesellschaft in den Griff zu be-
kommen”, heißt es im Vorwort zu einer Neuauflage seines Hauptwerks “Das
System moderner Gesellschaften” von Dieter Claessens im Jahr 1985, also zu
einer Zeit, als Luhmann bereits an seiner Theorie Sozialer Systeme arbeite-
te, von einem internationalen Durchbruch aber noch ein paar Jahre entfernt
war. Parsons systemische Herangehensweise an die Soziologie und die Gesell-
schaft mit ihren Subsystemen ist prinzipiell strukturierter und weniger komplex
und soziologisch-abstrakt als jene von Luhmann, der aber auch nach eigenen
Angaben eben darauf aufbaut. Parsons machte in seiner Arbeit aber einen ent-
scheidenden Fehler, die sich darauf beziehende Kritik war und ist so heftig
und vernichtend, dass sich kaum jemand wagte, seine Modelle ernsthaft als
Fundament für weitere Arbeiten herzunehmen: Der für die Zeit der Entste-
hung sehr guten systemtheoretisch-analytischen Betrachtung sozialer Systeme
folgen mehrere Kapitel westlich-androzentrischer, historischer Gesellschaftsana-
lyse, die in ihrer kulturellen Perspektive so einseitig und ignorant erscheinen,
dass der Blickwinkel und die Objektivität Parsons zurecht angezweifelt wurden
und werden.

Ebenfalls bereits vor Luhmann beschäftigte sich eine andere Gruppe von
Wissenschaftlern mit systemischen Ansätzen in den Wissenschaften, und auch
hier stand die Kommunikation im Zentrum der Beobachtungen: Die Kybernetik,
begründet von Norbert Wiener und von weiteren bedeutenden Wissenschaftlern
wie Heinz von Förster betreut, verfolgte ein ähnliches Ziel wie die Ansätze einer
Allgemeinen Systemtheorie. Vorgänge in der Natur, aber auch in Gesellschaften
sollten anhand der Kommunikation und Regelungen und ihrer formalen, ma-
thematischen Beschreibungen in eine vereinende Wissenschaft, die Kybernetik,
gebracht werden. Eine Unterscheidung zwischen der Kybernetik und GST ist
nicht ganz einfach, bei genauerer Betrachtung aber doch sehr klar möglich: Die
Kybernetik ist die mechanistische Schwester der GST und stellt nicht die grund-
legenden Relationen und Elemente von Systemen, sondern die daraus resultie-
renden Prozesse, Regelungen und Kommunikationswege in den Mittelpunkt ih-
rer Beobachtungen. Die Kybernetik hatte, wie auch das Bertalanffy-Programm,

entscheidenden Einfluss auf heutige systemische Ansätze in den Wissenschaften.

Ein prominenter Österreicher, der im Zusammenhang mit einer Allgemeinen Systemtheorie eigentlich nie genannt wird, ist der "Vater" der österreichischen Verfassung, Hans Kelsen. Kelsen war Rechtspositivist und Zeitgenosse (1881 - 1973) von Bertalanffy und lehrte sogar ebenfalls an der Universität Wien. Trotzdem gibt es keine Querverweise oder sonstige Hinweise, dass die beiden sich Zeit ihres Lebens getroffen oder gar gemeinsam ihre Ideen ausgetauscht haben. Das scheint erstaunlich, wenn man sich den Titel des sicherlich spannendsten, wenngleich auch nicht populärsten Werks von Kelsen ansieht: "General Theory of Law and State" (1949). Kelsen hatte also – lange vor Luhmann, noch vor Parsons und sogar vor dem Erscheinen von Bertalanffys Hauptwerk – von diesem Begriff in einer durchaus sinnverwandten Bedeutung Gebrauch gemacht. Im Prinzip beschreibt Kelsen hier die Konzeption seiner Reinen Rechtslehre und der gesellschaftlichen und moralischen Verknüpfungen. Kelsens Reine Rechtslehre ist in Zusammenhang mit dieser Arbeit eine durchaus spannende Angelegenheit: beschreibt sie doch ein prinzipiell in sich geschlossenes Rechtssystem, das sich selbst auf Basis der ursprünglichen oder aktuellen Rechtsregeln weiterentwickeln kann und darf. Jede daraus abgeleitete rechtliche und damit staatliche – denn Kelsen setzt das Recht und den Staat gleich, sie sind also eins – Konsequenz ist damit aus systemischer Sicht prinzipiell ordnungsgemäß, unabhängig von ihrer gesellschaftlichen Implikation. Kelsen beschreibt dazu in [26] auch eine "Hierarchy of the Norms" und "Determinations of the Judicial Act".

Dieser Ansatz ist insofern spannend, als dass Gesetze auf der selben logischen Ebene wie Prädikatenlogik "funktionieren" und grundsätzlich auch in ebendiese abbildbar sind. Betrachtet man diesen Umstand in Kombination mit Kurt Gödels Resultaten, die später noch eine entscheidende Rolle in dieser Arbeit spielen werden, etwas genauer, so entstehen interessante Fragestellungen in Hinblick auf die formale Widerspruchsfreiheit, Vollständigkeit und die Beweisbarkeit des Rechtssystems an sich und damit, nach Kelsen, auch des Staats. Selbstverständlich war Kelsen kein Vertreter einer Allgemeinen Systemtheorie wie wir sie oben beschrieben haben – trotzdem sollen seine Arbeiten zumindest an dieser Stelle kurz erwähnt werden, um zu zeigen, dass eine Allgemeine Systemtheorie auch durchaus streng formal auf gesellschaftlicher Ebene, nämlich in Form des Rechtssystems, eine stabile Ausprägung finden kann.

1.2 Kritik bestehender Arbeiten

Systemisches Denken hat heute den Einzug in viele Wissenschaftsdisziplinen gefunden, sowohl was die theoretische Auseinandersetzung, als auch was den

praktischen, experimentellen Teil betrifft.

Die Informatik hat dazu ob ihrer systemfordernden Eigenschaft einen wesentlichen Teil beigetragen, der Einfluss der wissenschaftlichen Pioniere auf diesem Gebiet ist aber sicherlich auch nicht zu leugnen. Während sich also neue Zweige wie “Systemische Biologie”, “Soziale Systemtheorie”, “Systemische Psychotherapie”, uvm. entwickelt und etabliert haben, ist der Begriff einer “Allgemeinen Systemtheorie” kaum im wissenschaftlichen Diskurs präsent. Auch die Kybernetik als verwandte Disziplin führt außerhalb der sie tragenden wissenschaftlichen Gemeinschaften ein Schattendasein:

“Tatsächlich gehört ‘die Kybernetik’ selbst eher schon wieder der Vergessenheit an.”

heißt es da etwa im Vorwort einer 2002 erschienenen Publikation²³ zum Andenken an Norbert Wiener. Ein wichtiger Grund dafür ist sicherlich das Scheitern des Bertalanffy-Programms und der Kybernetik in Hinblick auf das Hauptziel, die Entwicklung und Etablierung einer praxistauglichen Meta-Wissenschaft, die allgemeine Regeln und Sachverhalte für alle damals und heute bestehenden Wissenschaften zur Verfügung stellt und als Kommunikationsbasis zwischen ebendiesen dient.

Ein weiterer Grund dafür ist aber auch die Art und Weise, wie die wenigen Ansätze, eine Allgemeine Systemtheorie weiterzuerfolgen, heute konzipiert werden. Anstatt auf die Entwicklungen der Wissenschaft der letzten 40 Jahre und insbesondere die neuen Möglichkeiten der Informatik zurückzugreifen, erleben wir in eigentlich allen praxisbezogenen Versuchen die selben Fehler, die auch schon Bertalanffys Ansatz ins Leere laufen ließen. Nur kann man sich, wenn man seine Schriften und Denkweisen sowie seinen Hang zu interdisziplinärem Arbeiten kennt, zu sagen trauen, dass Bertalanffy und seine Weggefährten, würden sie heute leben, sicherlich anders vorgehen würden, andere Ansätze wählen würden.

Schauen wir uns anhand einiger Beispiele kurz an, wovon hier die Rede ist. Die folgenden Publikationen sind alle vergleichsweise aktuell und beschäftigen sich mit Modellen einer Allgemeinen Systemtheorie bzw. mit allgemein gültigen Systemmodellen:

Lars Skyttner: General Systems Theory. Skyttner bietet in [43] eine gute inhaltliche Zusammenfassung der philosophischen Historie systemischen Denkens. Er bezieht dabei auch viele wesentliche Denkmodelle systemischer Wissenschaften mit ein und weist an mehreren Stellen auf die Bedeutung der

²³Siehe [16], Seite 3.

Informatik und das Potential einer Allgemeinen Systemtheorie hin. Der Untertitel “Problems – Perspective – Practice” ist aber irreführend und hält nicht, was er verspricht: Vor allem die “Practice” beschränkt sich auf die Anwendung systemischen Denkens in theoretischen Modellen einzelner Wissenschaften: Informatik, Management-Theorie, etc. Praktische Beispiele für die Allgemeine Systemtheorie an sich und eine Evaluation auf Basis der in der Publikation sehrwohl aufgegriffenen Ziele einer ebensolchen werden nicht behandelt. Das macht dieses Werk, das derzeit wohl das jüngste mit dem Titel “General Systems Theory” sein dürfte, zwar zu einem guten Sammelpunkt für im Zusammenhang mit Systemtheorie wesentlichen Konzepten, trägt aber kaum etwas zur Weiterentwicklung der angesprochenen Ziele bei.

Hartmut Bossel: Systeme, Dynamik, Simulation. Hartmut Bossel arbeitet seit vielen Jahren als Systemwissenschaftler an der mathematischen Beschreibung und Simulation von Systemen in unterschiedlichen Wissenschaftsgebieten mit einem Fokus auf die Modellierung und Simulation ökologischer Prozesse und der angrenzenden Bereiche. In [9] gibt er einen Überblick, inwieweit bestehende Anwendungen der Informatik die Modellierung und Simulation komplexer, dynamischer Systeme unterstützen können und führt auch eine Reihe praktischer Beispiele und Lösungswege an. Bossel arbeitet dabei mit grafikbasierten Prozess-Modellierungssystemen wie STELLA, die im Hintergrund auf der Verkettung und Kombination von Differentialgleichungen basieren. In einer Publikationsreihe, dem “Systemzoo”, analysiert Bossel außerdem explizit die Anwendbarkeit derartiger Konzepte und Programme auf unterschiedliche Wissenschaftsbereiche: Physikalische Systeme (1), Ökosysteme (2), Gesellschaftliche Systeme (3).

Die fachkundige Modellierung mittels derartiger Anwendungen kann brauchbare und praxisnahe Resultate hervorbringen – das zeigt Bossel mit seinen Ergebnissen und Publikationen Schritt für Schritt und daher gut nachvollziehbar. Zwei Probleme, die dafür verantwortlich sein dürften, dass sich Programme wie STELLA (siehe Abbildung 1.7) abseits akademischer Untersuchungen im Gegensatz zu in dieser Dissertation vorgestellten Konzepten kaum durchgesetzt haben, werden aber auch in den angesprochenen Publikationen deutlich: Zum einen ist trotz der graphischen Modellierungsunterstützung ein tiefgreifendes mathematisches Verständnis der Eigenschaften von Differentialgleichungen notwendig, um sinnvolle Resultate zu erzeugen, zum anderen ist die Integration verteilten Wissens und die kollaborative Arbeit an Systeminstanzen durch den Fokus auf die prozedurale und nicht die systemische, d.h. auf die Elemente und Relationen des Systems bezogene, Modellierung kaum oder nur innerhalb von kleinen Expertengruppen möglich. Zum Erreichen der Ziele einer Allgemeinen Systemtheorie, nämlich auch bereits vorhandenes Wissen zu nutzen, um unnötige Arbeiten zu vermeiden, bedarf es aber genau dieser Charakteristika der verwendeten formalen Modelle und praktischen Anwendungen.

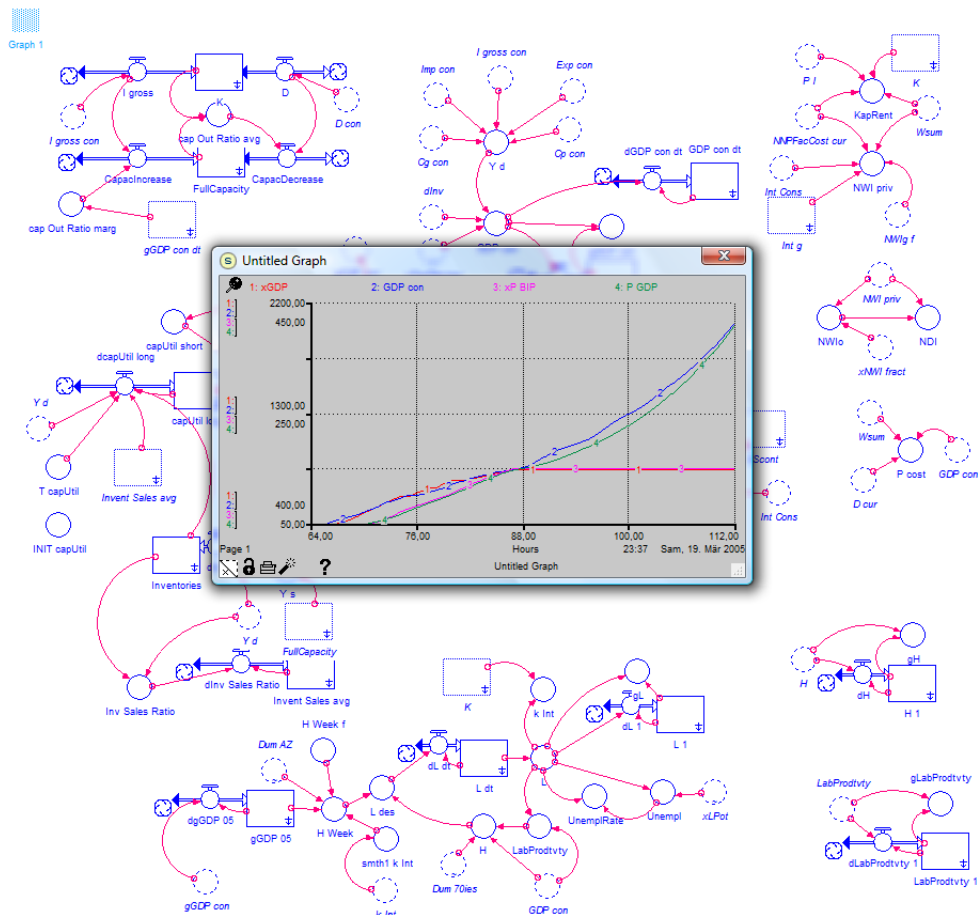


Abbildung 1.7: Auszug eines STELLA-System-Modells zur Modellierung der Österreichischen Wirtschaft von Gerhart Bruckmann und Peter Fleissner (1989). Die einzelnen Graphen stellen Teilprozesse des Wirtschaftssystems dar. Quelle: <http://peter.fleissner.org/MathMod/AUSTRIA.STM>

Interessant erscheint bei beiden Aufbereitungen die Gemeinsamkeit, dass die Autoren zwar Bertalanffys Hauptwerk, jedoch weder Publikationen der Society for General Systems Research noch Anatol Rapoport in der Literatur-Liste führen und auch nicht auf vergangene Modelle einer Allgemeinen Systemtheorie eingehen. Diese fehlende Auseinandersetzung mit der Historie einer Allgemeinen Systemtheorie ist eine mögliche Erklärung für das Aufgreifen ähnlicher mathematischer Ansätze, die, wie wir noch im Detail zeigen werden, für ein umfassendes, praktisch relevantes Modell zu unhandlich sind.

Zusammenfassend vorweggenommen können wir folgende Kritikpunkte an bisherigen Versuchen, ein praxistaugliches Modell für eine Allgemeine Systemtheorie zu etablieren, anführen:

- Die Anwendung von Differentialgleichungssystemen zur direkten Abbildung komplexer Systeme ist in Einzelwissenschaften etabliert und zielführend, aufgrund ihrer Komplexität und der schwierigen Nachvollziehbarkeit durch den menschlichen Beobachter aber nicht für eine vergleichende, interdisziplinäre Forschung geeignet.
- Die Beschreibung einer formalen Systemdefinition als Basis für darauf aufbauende Modelle und Applikationen ist aufgrund der nachhaltigen Auswirkungen, insbesondere der Klarheit in Bezug auf die Grundbausteine der einzelnen Systeme, im Rahmen einer Modellbildung für eine Allgemeine Systemtheorie unverzichtbar.
- Ein proprietäres Modell, das sich nicht oder nur mit sehr hohem Aufwand auf bestehende Standards, insbesondere solche im Bereich der Wissensrepräsentation abbilden lässt, ist kaum zu etablieren, da ausreichende Datenmengen fehlen, um eine kritische Masse an Systeminstanzen und Systemwissenschaftlern zu erreichen.

Auf all diese Punkte wollen wir in unserem Modell besondere Rücksicht nehmen und sie in der Phase der formalen Beschreibungen wie auch des Entwurfs des konkreten Modells und der Veranschaulichung durch Beispiele referenzieren.

Kapitel 2

Motivation

*“Ganz klein am Horizont
kann man Dinge sehen,
Dinge die wir nicht verstehen,
das Geschehen lässt uns auseinander gehen,
hinein in einen Wald aus Zeichen.
Die Weichen sind gestellt,
in einer Welt, deren Umriss uns gefällt.”
Tocotronic, “Hi Freaks”*

Der Mensch ist aus der natürlichen Entwicklung auf diesem Planeten als jenes Lebewesen hervorgegangen, das mit Abstand die am weitesten entwickelten intellektuellen Fähigkeiten besitzt. Dabei zeichnet ihn nicht so sehr die Fähigkeit zur Bildung sozialer Gemeinschaften – und damit wie wir später sehen werden die Entwicklung von Systemen auf einer höheren Stufe – aus, denn diese Fähigkeiten besitzen andere Lebewesen grundsätzlich auch: Den großen qualitativen Sprung hat der Mensch in seiner Entwicklung mit der Ausbildung seiner Kommunikationsfähigkeit und schließlich dem Lernen und Anwenden einer hochkomplexen Sprache getätigt. Diese Fertigkeit hat dem Menschen im Laufe der weiteren Geschichte einen Vorteil verschafft, der es ihm heute erlaubt, die Herrschaft über den Planeten und das auf ihm befindliche Leben zu besitzen.

Durch die Sprache war es möglich, immer differenziertere soziale Gemeinschaften zu bilden, die durch Arbeitsteilung eine enorme Effizienz im Vergleich zu primitiven Stammesgesellschaften der Frühzeit erreichen konnten. Dadurch war es möglich, Zeit für Weiterentwicklungen, insbesondere durch Technik und Wissenschaft, zu schaffen und so immer größere Fortschritte durch Erfindungen, die die Lebensqualität erhöhten und die Herrschaft des Menschen immer stärker festigten, zu erzielen. In diesem Kontext muss man dann auch suchen, wenn man sich mit der Frage der Wissenschaftsgeschichte beschäftigt. Warum

ist die Differenzierung der Wissenschaftsgebiete so entstanden, wie wir sie heute erleben? Wie lassen sich die Grenzen zwischen den einzelnen Wissenschaften begründen, warum gibt es heute Biologen, Physiker, Soziologen, etc. und nicht eine andere Aufteilung auf die Welt der zu lösenden Probleme, die den Menschen beschäftigen? Der Verlauf der Geschichte und ihr Einfluss auf die Ausbildung strukturierter Wissenschaft, auf die Gründung der Universitäten und Schulsysteme, auf die Bewertung des Stellenwertes einzelner Errungenschaften etc. ist nun einmal so, wie er ist.

Die Aufteilung der heutigen Wissenschaftswelt ist vermutlich auch ganz gut und sinnvoll so, es gibt wenig, das dafür spricht, dass sich hier eine schlechte, ineffiziente Differenzierung durchsetzen könnte. Aber wie in allen anderen Bereichen ist durch die Differenzierung sehr stark der Blick auf das Gesamte, auf “die eine Welt” und ihre Beschaffenheit, verloren gegangen. Das führt in einer komplexen Gesellschaft wie wir sie heute vorfinden zu einer vorwiegenden Spezialistensicht auf die großen Probleme, die oft nicht in der Lage ist, das Wesentliche zu erkennen, weil sie für eine ganzheitliche Analyse des Problems blind ist.

2.1 Wozu eine Allgemeine Systemtheorie?

Dieser “blinde Fleck der Wissenschaft” ist eines von zwei historischen Hauptargumenten für eine Allgemeine Systemtheorie. Das zweite Argument, das heute noch aktueller und wichtiger ist als zu Bertalanffys Zeit, ist die “Vermeidung doppelter Arbeit” durch den Austausch der Wissenschaften untereinander über eine allgemeine Komplementärtheorie, die als Kommunikationsmittel zwischen den Spezialwissenschaften vermitteln kann.

2.1.1 “Projekt eine Welt”

Dieser Titel wurde von Wolfgang Hofkirchner für seine Habilitationsschrift¹ gewählt, in der er die Zusammenhänge der Vorgänge in der Welt und ihre Beobachtungen und Untersuchungen auf unterschiedlichen Ebenen, in unterschiedlichen Wissenschaften untersucht. Hofkirchner untersucht dabei auch die Entwicklung des oben beschriebenen Verhältnisses bezogen auf die Frage, ob der Mensch zu einer ganzheitlichen Erfassung der Welt und ihrer Prozesse tendiert oder nicht. Ein einheitliches wissenschaftliches Weltbild soll dabei das Ziel der Entwicklungen der Einzelwissenschaften sein.

Dieses neue Weltbild soll den blinden Fleck für ganzheitliche Probleme auf unserer Welt auffüllen. Von welchen “ganzheitlichen Problemen” sprechen wir

¹Siehe [23].

hier aber überhaupt? Betrachten wir hierzu vorerst exemplarisch drei ausgewählte aktuelle Diskurse der letzten Jahre und die unterschiedlichen Positionen und Wissenschaftsfelder, die daran beteiligt sind.

Climate Change. Der Klimawandel ist sicherlich jenes Thema, das im Verlauf der letzten Jahre und Jahrzehnte am stärksten zugenommen hat, was die Aufmerksamkeit der Öffentlichkeit und die Bearbeitung innerhalb der (Populär-)Wissenschaft betrifft. Der Diskurs stellt sich dabei als sehr schwierig und oft auch unsachlich dar. *Worum geht es:* Der Mensch und seine Umwelt sind stark von den Klima- und Wetterbedingungen abhängig, hinzu kommt bei dieser Thematik, dass der Mensch das Verhalten von Klima und Wetter so gut wie gar nicht bewusst in eine bestimmte Richtung steuern kann – das ist in einer ansonsten hochtechnisierten Umgebung eine mittlerweile außergewöhnliche Situation, die verständlicherweise für Aufregung und Panik sorgt. Die Vernetzung der Mediengesellschaft liefert zudem verstärkt Katastrophenszenarien und besetzt das Thema in einer sensationsgerichteten Art und Weise.

Welche Akteure sind vorrangig in den Diskurs involviert: Klimaforscher und Naturwissenschaftler, die schwere Industrie und die Automobilindustrie, Umweltbewegungen und natürlich wie bei allen ausgewählten Themen Politik und Medien.

Tatsächlich sind die anthropogenen Effekte der Klimaveränderung unbestritten, da der Mensch seit Beginn des Industriezeitalters maßgeblich in biologische, chemische und geologische Prozesse der weltweiten bio-geo-chemischen Flüsse eingreift. Auch scheint dies im speziellen Fall des planetaren Kohlenstoffkreislaufs und seiner Störung durch die Eingriffe des Menschen signifikant nachweisbar, sowohl was die Quellen als auch was die unmittelbaren Auswirkungen dieser Veränderungen betrifft. Auch sind die meisten involvierten Speicher und Flüsse des globalen Kreislaufs relativ gut bekannt, was in Abbildung 2.1 dargestellt ist. Trotzdem gibt es eine ganze Reihe ungeklärter Fragen und offener Probleme in diesem Zusammenhang, die zeigen, wie wenig wir eigentlich trotz all der technischen Möglichkeiten von globalen Kreisläufen und ihren Auswirkungen und Wechselwirkungen verstehen.

Noch offenkundiger wird die Problematik, wie so oft, wenn man die Reaktion politischer Systeme auf die Herausforderungen des Klimawandels betrachtet. Ein vielversprechender Ansatz, der bereits von James Lovelock² skizziert wurde, heute aber auf Basis globaler Daten viel zielgerichteter verfolgt werden kann, ist die Betrachtung des Planeten als komplexes lebendes Ökosystem. In [13] werden etwa globale Stoffkreisläufe beschrieben und analysiert und ein Paradigmenwechsel von der abgeschlossenen, analytischen Ökologie hin zu einer

²Siehe dazu auch [30].

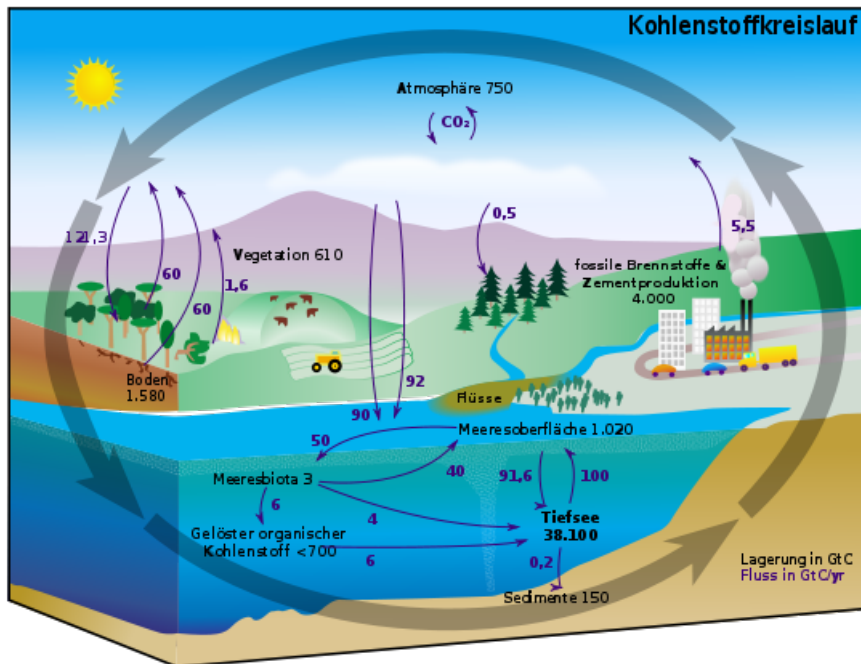


Abbildung 2.1: Diagramm des Kohlenstoffkreislaufes. Die schwarzen Zahlen zeigen wie viele Milliarden Tonnen Kohlenstoff (Gt C) in den verschiedenen Reservoiren vorhanden sind. Die blauen Zahlen zeigen an, wie viel Kohlenstoff zwischen den einzelnen Speichern pro Jahr ausgetauscht wird. Bildquelle: Wikimedia.

“Earth System Science” gefordert, die einen umfassenden systemischen Ansatz zur Beschreibung und Untersuchung globaler Zusammenhänge verfolgen soll. Unter einem Ökosystem verstehen die Autoren dabei folgendes³:

“An ecosystem consists of all the organisms and the abiotic pools with which they interact. Ecosystem processes are the transfers of energy and materials from one pool to another.”

Tatsächlich wissen Naturwissenschaftler heute noch vergleichsweise wenig über all diese Komponenten: Wie groß sind die einzelnen Pools? Wie genau laufen die Prozesse ab, welche Faktoren beeinflussen sie? Zwei Beispiele, die in [13] betrachtet werden und die zeigen, wie schwer es der Wissenschaft selbst heute unter Zuhilfenahme leistungsfähiger Rechner noch fällt, globale Ökoprozesse zu verstehen, sind der El-Nino-Southern-Oscillation-Effekt (ENSO) und die Frage nach der Aufteilung globaler Kohlenstoffquellen. Beide Themen wurden bereits in unzähligen Publikationen und auf zahlreichen Konferenzen diskutiert und sind für die Menschheit von großer Bedeutung. Insbesondere die El Nino / La Nina Effekte und ihre Ursachen führen regelmäßig zu großen wirtschaftlichen und humanitären Katastrophen, ohne dass man bisher in der Lage wäre, das

³Siehe [13], Seite 4.

Auftreten dieser periodischen Naturprozesse voraus zu sagen. Viele Ökologen sehen die Lösung des Rätsels in einer stärker fokussierten systemischen Betrachtungsweise, in ganzheitlichen Modellen, eben in einer “Earth System Science”, wie sie auch in [13] angeregt wird.

Schutz von Leben. Die Thematik des Umgangs von Gesellschaften mit dem Schutz von – menschlichem aber auch tierischem – Leben ist deshalb eine so interessante für unsere Arbeit, weil so viele Akteure aus unterschiedlichen wissenschaftlichen und gesellschaftlichen Bereichen involviert sind. *Worum geht es:* In den Grenzbereichen der Souveränität bzw. der Anerkennung von wertvollem Leben gibt es eine Reihe von Streitpunkten, die global immer wieder zu heftigen Auseinandersetzungen auf gesellschaftlicher und politischer Ebene führen. Drei wesentliche sind dabei (1) der Schutz von ungeborenem Leben bzw. das Recht (von Frauen) auf Abtreibung, (2) das Leisten von aktiver Sterbehilfe für unheilbar kranke Menschen und (3) der Tierschutz, insbesondere in jenen Bereichen, wo von Tierrechtsbewegungen von einer Verletzung der Würde und der Rechte von Tieren durch den Menschen gesprochen wird. *Welche Akteure sind vorrangig in den Diskurs involviert:* Die Katholische Kirche und andere Glaubensgemeinschaften, Mediziner, Emanzipations- und Tierrechtsbewegungen und Lebensmittelindustrie.

Die entscheidenden Fragen in diesem Diskurs sind “Was ist Leben?” und “Was ist wertvolles und schützenswertes Leben und warum?”. In der sogenannten Abtreibungsdebatte geht es darum, dass es heutzutage medizinische Möglichkeiten gibt, mit geringem physischen Risiko für die Frau eine ungewollte Schwangerschaft abzuberechnen. Das Recht dazu wurde in den meisten europäischen Ländern als Folge der zweiten Welle der Frauenrechtsbewegung in den 1970er-Jahren gesetzlich verankert, obgleich die Regelungen im Detail in den Ländern mitunter sehr unterschiedlich aussehen, wie auch in Abbildung 2.2 veranschaulicht wird.

Das Thema hält sich aber konstant im öffentlichen Diskurs und wird immer wieder aufgegriffen, weil zumeist entweder die Kirche oder aber die Frauenbewegungen eine Abänderung des bestehenden Rechts in die eine oder andere Richtung fordern. Die Katholische Kirche leitet ihre Forderungen dabei aus dem Gebot “Du sollst nicht töten” ab und schreibt im Katechismus⁴:

“Das fünfte Gebot verbietet als schwerwiegende Verstöße gegen das Sittengesetz die direkte Abtreibung, als Ziel oder als Mittel gewollt, und die Mitwirkung daran; dieses Vergehen wird mit der Exkommunikation bestraft, weil das menschliche Wesen von der Empfängnis an in seiner Unversehrtheit absolut zu achten und zu schützen ist.”

⁴Siehe “Katechismus der Katholischen Kirche”, Dritter Teil, Zweiter Abschnitt, Absatz 470.

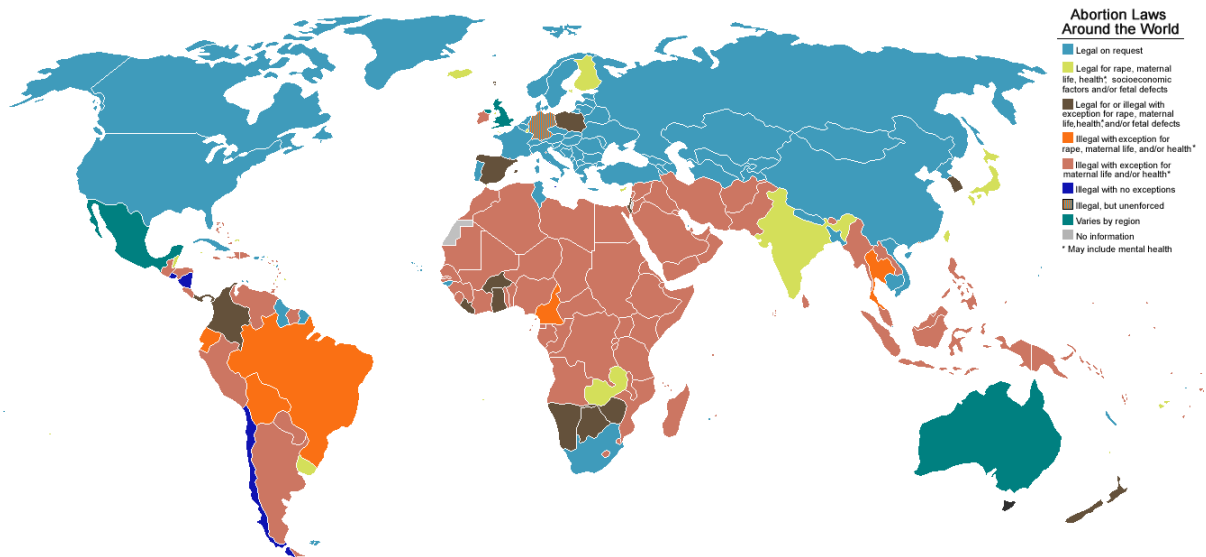


Abbildung 2.2: Grafische Darstellung der rechtlichen Situation zur Möglichkeit eines Schwangerschaftsabbruchs. Bildquelle: Wikimedia.

Die Befürworter der Abtreibung hingegen stellen zwar den Schutz menschlichen Lebens und das Verbot zu töten nicht in Frage, definieren menschliches Leben und ab wann dieses als wertvoll bzw. schützenswert zu behandeln ist aber anders und sehen in der frühen Phase einer Schwangerschaft das Recht der Frau auf Selbstbestimmung als das stärker zu gewichtende. Nun hat eine Allgemeine Systemtheorie, wie wir im Laufe dieser Arbeit zeigen, ohne Zweifel sehr viel Lösungspotential – sie wird aber sicherlich keine Konflikte zwischen Glaube und Wissenschaft beilegen können, weil sie nur in den Bereich der Wissenschaft, nicht aber in jenen des Glaubens hineinreicht. Wieso diskutieren wir dieses Thema dann an dieser Stelle? Hier geht es darum zu zeigen, wie sehr dieser spezielle Konflikt auf einer bestimmten Ebene, nämlich im Wechselspiel zwischen Glaube und Frauenbewegung, festgefahren ist. Jeder Politiker tut derzeit gut daran, das Thema nicht anzusprechen, weil die Gefahr groß ist, dass er zwischen diesen Fronten inhaltlich verloren geht. Tatsächlich ist die Bevölkerungsentwicklung und ihre Steuerung über derartige Möglichkeiten wie auch über die Empfängnisverhütung etc. eine Frage, die viel zu wichtig ist, als dass sich ihre Behandlung nur in diesem Teilbereich abspielen dürfte.

Ohne jetzt politische Forderungen irgendeiner Art ableiten zu wollen, muss es verstärkt Möglichkeiten geben die Entwicklungen von Gesellschaften und die Simulation von Regulativen in diesem Bereich zu berechnen, um die Ergebnisse in sachliche politische Diskurse einbinden zu können. Das wird spätestens dann ein ernstes und heftig diskutiertes Thema werden, wenn die Migration als Prozess zur Aufrechterhaltung westeuropäischer Staaten und ihrer Volkswirtschaften nicht mehr ausreicht, weil die Ursprungsländer der Immigranten aufgrund der steigenden Lebensqualität und Zivilisationsprozesse mit den selben demo-

graphischen Auswirkungen konfrontiert sein werden wie heute Westeuropa. Erst mit der Entwicklung von Modellen und Simulationen zu beginnen, wenn das Problem bereits akut ist, wird aber zu ähnlichen Diskurseigenschaften führen wie im Falle des Klimawandels. Und vermutlich werden die Diskussionen um die Möglichkeiten zur Aufrechterhaltung der Sozialsysteme auch die Diskussion um die Möglichkeiten und Auswirkungen aktiver Sterbehilfe beeinflussen, da dieses Thema sich ja nur “am anderen Ende” der selben Wirkungskette befindet. Dass Wachstum – hier Bevölkerungswachstum – ein klassisches GST-Thema ist, zeigen außerdem die zahlreichen Publikationen zu dem Thema in den General Systems Yearbooks bzw. bei Bertalanffy selbst, bei Boulding und anderen GST-Vertretern.

Die Thematik des Tierschutzes mag im Vergleich dazu “unspektakulär” wirken, ist aber ob ihrer potentiellen Dimension eine nicht minder interessante. Die Geschichte zeigt, dass mit steigender Lebensqualität in Staaten die Bereitschaft in der Zivilgesellschaft wächst, Minderheiten zu unterstützen und Diskriminierung zu bekämpfen. Die Abschaffung der Sklaverei und des Leibeigentums, die Emanzipation der Schwarzen in Nordamerika und die weltweite Frauenbewegung sind nur einige der bekannten Prozesse, die auch auf solche Weiterentwicklungen der zugrunde liegenden Gesellschaften zurückzuführen sind. Der Tierschutz und die Erklärung von Tieren zu einer schützenswerten “gesellschaftlichen” Gruppe rückt in den letzten Jahrzehnten immer mehr in das Aufmerksamkeitsfeld der Öffentlichkeit. Dabei gibt es im Zusammenhang mit den Forderungen von Tierrechtsbewegungen auch durchaus interessante wissenschaftliche Auseinandersetzungen mit den Grundfragen dieses Diskurses. So hat der österreichische Tierrechtsaktivist Martin Balluch sich vor einigen Jahren in [1] mit der Frage des Bewusstseins bei Tieren und der daraus entstehenden Konsequenz erweiterter Tierrechte auseinandergesetzt.

Intelligent Design. Dieses letzte Thema in diesem Zusammenhang wurde auch deshalb ausgewählt, weil hier ein interessanter Bezug zum Leben und Wirken Bertalanffys besteht: So hat Bertalanffy 1955 eine Vortragsreihe und eine Publikation⁵ initiiert mit dem Titel “Schöpfungsglaube und Evolutionstheorie”, in dem Theologen und Wissenschaftler zu dieser Thematik zu Wort kommen sollten – auch Bertalanffy selbst hat einen Beitrag mit dem Titel “Die Evolution der Organismen” dazu verfasst. 2007 erschien als Antwort auf den damals medial stark präsenten Diskurs um die Idee des “Intelligent Design” und den aufstrebenden Kreationismus in den USA sowie die Partizipation des österreichischen Kardinals Christoph Schönborn eine Publikation herausgegeben von Marianne Popp, quasi einer “Nachbarin” Bertalanffys an der Biologie

⁵Die Publikation ist der Abdruck einer 1955 vom Süddeutschen Rundfunk veranstalteten Vortragsreihe mit ebendiesem Titel, bei der neben Bertalanffy auch weitere Naturwissenschaftler sowie Theologen Fragen im Grenzbereich zwischen Wissenschaft und Religion betrachteten. Siehe “Schöpfungsglaube und Evolutionstheorie – Eine Vortragsreihe bekannter Theologen und Naturwissenschaftler”, Alfred Körner Verlag, Stuttgart.

der Universität Wien, mit dem Titel “Schöpfung und Evolution – zwischen Sein und Design”.

Worum geht es: Das Christentum hat wie auch alle anderen großen Weltreligionen eine im Alten Testament überlieferte Erklärung über den Ursprung und die Entstehung der Welt. Der biblische Schöpfungsglaube aus dem Buch Genesis ist dabei nach heutigem Wissen weit von dem entfernt, was wir aufgrund der Erkenntnisse der Naturwissenschaften wissen. Trotzdem gibt es – vor allem in den USA – religiöse Gruppen, die einen Unterricht über den Schöpfungsglauben als Alternative zur Evolutionstheorie fordern. Mit diesem radikalen Ansatz, also der wortwörtlichen Auslegung der biblischen Schöpfung, wollen wir uns hier aber nicht weiter beschäftigen, da sie wohl als unwissenschaftlich zu bezeichnen ist. Interessanter ist eine wissenschafts-tolerantere Version, die in den letzten Jahren unter dem Begriff des Intelligent Design zusammengefasst wurde: Die Kernthese des Intelligent Design besagt dabei, dass aufgrund der hohen Komplexität und der Art der Ausprägungen der Natur und des Menschen eine höhere Intelligenz, also eine Art allmächtiger Uhrmacher (der Gottesbegriff wird meist aufgrund der gesetzlichen Situation in den USA vermieden) hinter der Konzeption der natürlichen Entwicklungsprozesse steht. Kritisiert wird dabei insbesondere die Evolutionstheorie und die Erklärung der natürlichen Entwicklung durch Mutation und Selektion.

Der Erfolg der Idee legt gleichzeitig einen vermeintlich wunden Punkt der Naturwissenschaften offen: Oft fehlen Teilstücke in den Evolutionstheorien und noch immer können komplexe biologische Evolutionsprozesse anhand der bestehenden Theorien nicht ausreichend plausibel erklärt werden. Vermeintlich aber deshalb, weil die Vertreter des Intelligent Design sich hier oft auf Fehlinterpretationen etwa der Evolutionstheorie Darwins beziehen und zum Teil bewusst auf die tatsächlich nicht ausreichenden Mechanismen reiner Mutations- und Selektionsprozesse und des in dieser Form oft falsch angewandten “Survival of the fittest” verweisen.

Tatsächlich ist die Biologie heute aber schon viel weiter und aktuelle Ansätze im Bereich der EvoDevo-Theorien (Evolutionary Development) und Arbeiten zum Prinzip der Selbstorganisation können schon heute viele vermeintlich als zu komplex angesehene Vorgänge durchaus plausibel erklären. Diese Entwicklungen werden wir uns aber später noch etwas genauer ansehen. Tatsache ist, dass die hier vorgestellten Prinzipien auch in anderen Wissenschaftsbereichen – etwa in den Sozial- und Wirtschaftswissenschaften, aber auch auf der Ebene von Zellverbänden und zur Erklärung für die Entstehung von Sonnensystemen – immer stärkeren Einzug finden und uns im Laufe der nächsten Jahrzehnte sicherlich viele neue Erkenntnisse bringen werden.

Gerade aufgrund der hohen Komplexität der zuletzt genannten Beispiele

und weil es sich hier tatsächlich um allgemeine Prinzipien der Ordnung der Welt auf unterschiedlichen Ebenen handelt, ist es umso wichtiger, eine Allgemeine Theorie mit einem passenden Vokabular und der Möglichkeit zum Austausch zwischen den Wissenschaftswelten zu schaffen, um effizient bereits erzielte Resultate wiederverwenden zu können.

2.2 Wissenschaftsphilosophie einer Allgemeinen Systemtheorie

Es wurden bewusst drei Beispiele mit starkem Bezug zum Menschen und seiner Beschaffenheit und persönlichen Handlungsweise ausgewählt. Wie in der Einleitung beschrieben wurde, hatte Bertalanffy mit seiner Allgemeinen Systemtheorie ja stets ein humanistisches Programm vor Augen.

Auffallend ist, dass in allen drei betrachteten Diskursen – und das gleiche gilt auch für alle anderen aktuellen “großen” Probleme und Fragen der Menschheit – Politik und Medien als wichtige Akteure involviert sind. Daraus lässt sich ein wichtiges drittes Argument für eine Allgemeine Systemtheorie ableiten: Sie wäre jedenfalls ein wichtiges Instrument zur Unterstützung von Entscheidungsprozessen in Politischen Systemen.

Interessant erscheint in diesem Zusammenhang dann aber die Frage, ob der Ansatz einer Allgemeinen Systemtheorie, wie er in dieser Arbeit basierend auf den Theorien von Bertalanffy beschrieben wird, eine bestimmte Weltanschauung bzw. eine bestimmte philosophische Weltsicht impliziert. Ausführlicher mit dieser Frage hat sich der Philosoph und Systemwissenschaftler Ervin Laszlo beschäftigt. In [29] beschäftigt er sich mit den Ansätzen einer auf systemische Betrachtungsweisen basierenden Philosophie. Dabei erkennt auch er eine ähnliche Motivation für eine Allgemeine Systemtheorie⁶:

“Wollen wir die Wirklichkeit richtig begreifen, so bleibt uns nichts anderes übrig, als unsere Forschungsobjekte als Systeme mit eigenen Eigenschaften und Strukturen aufzufassen. Auf diese Weise können wir Systeme verschiedener Art vergleichen, ihre gegenseitigen Beziehungen im Rahmen noch größerer Systeme untersuchen und den Grundzusammenhang zwischen ihnen allen deutlich machen. Wollen wir wirklich verstehen, wer wir sind und wie unsere Welt aussieht, ist die Entwicklung einer Allgemeinen Systemtheorie eine zwingende Notwendigkeit.”

Laszlo sagt damit also – sehr radikal –, dass wir die Welt heute noch nicht

⁶Siehe [29], Seite 21.

“wirklich verstehen”, sondern durch die Spezialistensicht eben nur unzureichend und nur teilweise wirklich wahrnehmen. Noch haben wir aber keine ganzheitliche Sicht, kein “richtiges Begreifen der Wirklichkeit”. Dabei stimmen Laszlo viele andere Geistes- und Naturwissenschaftler zu. Dass die Allgemeine Systemtheorie des Rätsels Lösung, der Schlüssel zum richtigen Begreifen der Wirklichkeit ist, wird da schon umstrittener sein. Interessant ist eine derartig hoffnungsvolle und optimistische Ableitung aber allemal. Laszlo bemüht sich auch – im Gegensatz zu den Vertretern des Bertalanffy-Programms – nicht wirklich um eine Harmonisierung mit den bestehenden Wissenschaften⁷:

“Die ganzheitliche Sicht der Systemwissenschaft steht im Gegensatz zur atomistischen, mechanistischen Weltanschauung der klassischen Naturwissenschaften.”

Er sieht diese beiden Denkrichtungen also als entgegengesetzt, zumindest was ihre philosophischen Implikationen betrifft, und nicht als komplementär, wie es Bertalanffy selbst gesehen hat. Das ist ein wesentlicher Unterschied, denn tatsächlich ist die Frage nach der Vereinbarkeit der Resultate und Auffassungen einer allgemeinen systemischen Weltansicht eine schwierige und ernste. Gesellschaftliche Themen wie sie zu Beginn dieses Kapitels angesprochen wurden, aber auch naturwissenschaftliche Paradigmen wie jenes des Freien Willens oder der Künstlichen Intelligenz könnten aus der Sicht einer Allgemeinen Systemtheorie neue Bewertungen, Gesellschaften aufgrund der Tragweiten neue Werte erhalten. Es ist nicht absehbar, wie stark sich die damit zwangsweise verbundenen gesellschaftlichen Interessenskonflikte ausprägen würden bzw. werden, wenn man von der Überzeugung ausgeht, dass ein solches Weltbild, wie es Laszlo skizziert, nur eine Frage der Zeit ist. Wie “dringend” ein solcher Umbruch notwendig ist und wie rasch oder schleichend er erfolgen soll, ist daher eine wichtige Frage⁸:

“Die Verlagerung des Interesses von der klassischen zur systemwissenschaftlichen Weltansicht ist dringend geboten. Weltanschauungen sind Gesamtheiten von Begriffen, Vorstellungen, Werten und Gewohnheiten, die in einer Gemeinschaft gelten und Handlungen ihrer Mitglieder steuern. [...] Sie kann dazu beitragen, dass der Mensch das Wesen der Welt, in der er lebt, versteht und durchschaut, ebenso seine Rolle und Identität in dieser Welt.”

Das ist freilich nur die positive Seite des Umbruchs, die die möglichen Konflikte und Narben bei einer dringenden Umsetzung außen vorlässt. Laszlo bringt aber den Nutzen einer systemischen Weltanschauung noch einmal auf den Punkt, er sieht ihn nämlich darin,⁹

⁷Siehe ebenda, Seite 22.

⁸Siehe ebenda, Seite 24.

⁹Siehe ebenda, Seite 24 f.

“dass die Avantgarde der modernen Wissenschaftler Möglichkeiten zur Schaffung einer nicht atomistischen, sondern ganzheitlichen Welt-sicht zur Verfügung stellt, die das Bedürfnis nach praktischer Lebenshilfe in unserer Zeit durchaus befriedigen mag. Diese neue Welt-sicht kann uns die Schlüssel, die Symbole, die Orientierung, ja sogar detailliertere Modelle in die Hand geben, durch die es uns möglich wird, die kritischen Probleme dieser immer mehr globalisierten und deshalb ganzheitlich wirkenden Welt zu lösen.”

Das ist eine wichtige Feststellung, das Hinarbeiten in diese Richtung von jenen Wissenschaftlern und Intellektuellen, die dieses Konzept und ihren Wert heute bereits verstanden haben, deshalb eine hehre Aufgabe. Der Bertalanffy-Biograf Davidson formuliert die Auswirkungen des Weltbilds auf die Lage und Zukunft der Menschheit¹⁰ noch etwas kräftiger:

“Heute, da die Menschheit technologisch in der Lage ist, ihre eigene Zukunft zu formen oder sich selbst jegliche Zukunft zu versagen, kann die philosophische Sicht, die wir vom Leben und unserer eigenen Spezies haben, sehr wohl die wichtigsten Entscheidungen in der gesamten Geschichte beeinflussen.”

Legt man das auf Themen wie die Klimapolitik, Friedens- und Kriegspolitik und die internationalen wirtschaftlichen Prozesse um, scheint ein besseres Verständnis der Welt als ganzheitliches System doch wieder dringender notwendig. Wenn uns die Theorie komplexer Systeme aber etwas lehrt, dann wohl, dass es ohnehin durch die Eigendynamik der involvierten sozialen bzw. der Wissenschaftssysteme zu einer von Individuen oder kleinen Gruppen nur schwer beeinflussbaren Entwicklung dieses Umbruchs kommen wird, was der Diskussion über den Weg und das Zustandekommen eines ebensolchen zu einer Randdiskussion verurteilt.

2.3 Praktischer Nutzen einer Allgemeinen Systemtheorie

Eine Theorie ist erst dann wirklich wertvoll, wenn sie in der Praxis angewandt werden kann und hier auch nützliche Resultate (in der Regel in Form von neuen Erkenntnissen) produziert und eine gesellschaftliche Wertschöpfung zur Folge hat. Das Ziel aller Wissenschaft, manchmal kurzfristig, manchmal sehr langfristig ausgerichtet, ist es, die Lebensbedingungen des Menschen zu verbessern und zum Wohle der Gesellschaft einen Beitrag zu leisten. Aus den vorangegangenen Untersuchungen lassen sich zumindest drei maßgebliche Punkte für einen entscheidenden praktischen Nutzen einer Allgemeinen Systemtheorie aus heutiger Sicht ableiten:

¹⁰Siehe [14], Seite 56.

Metasprache für Informationssysteme Die Entwicklungen der Informatik haben Informationssysteme hervorgebracht, die die Vision, das Wissen über Systeme in einer Form zu verwalten, dass sie vergleichbar sind und neues Wissen daraus abgeleitet werden kann, wirklich werden lassen können. Die Realisierung dieser Vision und derartiger Systeme kann langfristig und nachhaltig dazu beitragen, die Welt aus Sicht der einzelnen Wissenschaftsdisziplinen besser zu verstehen, Lösungen für Probleme schneller zu finden und “doppelte Arbeit” an ähnlichen Problemstellungen zu verhindern. All diese Punkte würden direkt durch eine Kostenreduktion im Forschungsweg oder indirekt durch die effizientere Ausrichtung von Forschungsfragen eine spürbare Wertschöpfung für die Wissenschaft hervorbringen. Eine Allgemeine Systemtheorie ist in diesen Informationssystemen dann die Metasprache, auf die die einzelnen Wissenschaftsdomänen abgebildet werden und die Abfragen zu Problemstellungen und das damit verbundene wissenschaftliche Arbeiten mittels auf diesem Modell basierenden Anwendungen unterstützt.

Ganzheitliche Problemsicht Durch die digitale Vernetzung und die Globalisierung sowie den anhaltenden Trend zur Schaffung supranationaler Gesellschaftsformen war die umfassende Analyse der großen Diskurse und Probleme der Menschheit in Hinblick auf die Politikgestaltung und die Auflösung politischer Konflikte noch nie von so großer Bedeutung wie dies heute der Fall ist. Ganzheitliche, systemische Analysen von elementaren Fragestellungen, mit dem Ziel, gesellschaftlich fördernde, umsetzbare Konsenslösungen hervorzubringen, erhalten dadurch eine stark wachsende Bedeutung. (Politische) Diskurse scheitern oft in der Unfähigkeit, ein Problem auf seine Kernelemente zu reduzieren, um so eine sachliche Diskursebene zu erreichen, weil Teilaspekte des Problems in unterschiedlichen Politikrahmen betrachtet und nur innerhalb dieser Grenzen analysiert werden. Eine Allgemeine Systemtheorie kann eine ganzheitliche Problemperspektive maßgeblich unterstützen und fördern und damit zu einer entsprechenden Beurteilung und einer Ausweitung der jeweiligen Politikrahmen führen, wenn basierend auf einem Modell für eine Allgemeine Systemtheorie eine entsprechende Diskursmethodik geschaffen wird.

Modernes humanistisches Weltbild Eine Allgemeine Systemtheorie, die einen umfassenderen Blick auf die Beschaffenheit der Natur erlaubt und der Einheit der Wissenschaft als Metasprache dient, hat auch das Potential, eine ganzheitliche Weltsicht hervorzubringen, die das Bewusstsein und die Identifikation des einzelnen Menschen und der von ihm gebildeten Gesellschaften in eine stärker konstruktiv und kooperativ ausgerichtete Prägung beeinflusst. Gesellschaftliche Konflikte und Kriege sind oft das Resultat unüberbrückbarer Differenzen, denen unterschiedliche Sichtweisen und Paradigmen zu Grunde liegen. Eine systemtheoretisch begründete Weltsicht kann zu einer Homogenisierung der Auffassung der Welt und gesellschaftlicher Ziele beitragen und damit helfen, Differenzen zu beseitigen oder abzuschwächen.

Kapitel 3

Systeme

*“Herr Professor Luhmann, welche Kritiker Ihrer Systemtheorie fürchten Sie am meisten?”
“Die Dummen.”
Niklas Luhmann in einem Interview 1973*

Eine Allgemeine Systemtheorie steht und fällt mit der Möglichkeit, einen allgemeinen Systembegriff zu definieren, der nicht nur konzeptueller Platzhalter, sondern sinnvolle abstrakte Instanz ist. Die Kritikpunkte an einer Allgemeinen Systemtheorie und am Bertalanffy-Programm fokussieren sich oft auf dieses Unterfangen. Klaus Müller bringt es auf den Punkt¹:

“ Was aber ist mit einem Systembegriff gewonnen, der von jeder inhaltlichen Aussage absieht? Welche Aussagen sind einer Systemtheorie, die sich Systemen schlechthin widmet, überhaupt möglich? [...] Ein allgemeiner Systembegriff hat freilich von vornherein mit einem Dilemma zu kämpfen: um auf Systeme jeglicher Art anwendbar zu sein, muss er mit dem Anspruch auf universale Zuständigkeit antreten; um nicht zu einer Leerformel herabzusinken, muss er gleichwohl spezifische Aussagen über Systeme im Allgemeinen zulassen. ”

Im Diskurs der Society for General Systems Research wird das Problem ebenfalls deutlich. Im ersten Jahrbuch der Gesellschaft, das 1956 erschienen ist, findet sich ein Beitrag zur Definition von Systemen im Allgemeinen², auf den wir gleich noch näher eingehen werden. Trotz zahlreicher weitergehender Versuche blieb dieses oft als minimalistisch bezeichnete Modell stets die einzige formale Abstraktion, die hinreichend breit anerkannt wurde, um sie als Grund-

¹Siehe [34], Seite 199 ff.

²Siehe [21].

lage einer Allgemeinen Systemtheorie verwenden zu können. Müller kritisiert diesen Minimalismus so:

“Zwar scheinen Hall und Fagen ihren formalen Systembegriff von den ideologischen Altlasten einer Ganzheitslehre abgekoppelt zu haben. Die Relationen und Funktionen eines derart definierten Systems sagen nichts über das Substrat der Systemprozesse aus; sie beziehen sich weder auf stoffliche, noch auf energetische oder informationsförmige Kopplungen zwischen seinen Elementen. [...] Doch entsteht daraus sogleich das oben angesprochene, bis in die jüngste Systemtheorie kontrovers gebliebene Problem. Wie nämlich lassen sich aus einem universalistischen Systembegriff, der sich auf keinen spezifischen Forschungsbereich festlegen will, hinreichend scharfe Aussagen über spezielle Systeme herleiten? ”

Diese nicht nur von Müller geäußerte Kritik an einem allgemeinen Systembegriff ist allerdings insofern irreleitend, als dass die Ableitung, nach der hier verlangt wird, überhaupt kein Ziel einer Allgemeinen Systemtheorie ist. Weder bei Bertalanffy noch bei Rapoport finden wir den Anspruch, dass aus dem Systembegriff an sich Aussagen über spezielle Systeme, die über ebendiese Definition hinausgehen, abgeleitet werden können. Das würde ja gleichwohl Münchhausens Trick mit dem Aufziehen am eigenen Schopf entsprechen und muss deshalb als Ziel und Anforderung ausgeschlossen werden.

Die Frage nach dem Wert und der Sinnhaftigkeit eines allgemeinen Systembegriffs hat aber trotzdem durchaus ihre Berechtigung. Und wir werden in dieser Arbeit nicht zu formalen Modellen und technischen Realisierungsmöglichkeiten voranschreiten, ehe wir nicht dieses Begriffsdilemma zumindest in Bezug auf das hier vorzustellende Modell und seine Ansprüche aufgelöst haben. Wir wollen also an dieser Stelle versuchen – mit Rücksichtnahme auf die vorgebrachten Kritikpunkte –, eine formale Definition eines Systembegriffs zu erarbeiten, die für alle Systeme Gültigkeit besitzt und dabei über ein nutzloses tautologisches Konzept hinausgeht.

3.1 Reductive Levels

Was heißt dabei “für alle Systeme”? Wir wollen uns bei der Beantwortung dieser Frage zunächst an der von Oppenheim und Putnam³ vorgeschlagenen Klassifikation von Systemen orientieren, die zwischen folgenden Ebenen von Systemen

³In ihrem 1958 erschienenen Beitrag “Unity of Science as a Working Hypothesis”, siehe [36], beschäftigen sie sich mit dem Problem der bis dato unklassifizierten Auseinandersetzung mit einer einheitlichen Herangehensweise an Wissenschaft im Allgemeinen. Das Resultat ihrer Überlegungen ist eben jenes Schema, das in dem Beitrag detaillierter vorgestellt wird.

unterscheidet: (1) Elementarteilchen, (2) Atome, (3) Moleküle, (4) Zellen, (5) Organismen und (6) Soziale Gruppen.

An die Klassifikation (im folgenden als OP-Modell bezeichnet) wurden dabei unter anderem die Anforderungen gestellt, dass Systeme auf einer bestimmten Ebene insofern abgeschlossen sind, als dass sie keine Elemente auf übergeordneten Ebenen besitzen. Weiters war es Oppenheim und Putnam ein Anliegen, auf damals bestehende Wissenschaftsfelder Rücksicht zu nehmen, wobei sie hier eine Einschränkung auf empirisch behandelbare Wissenschaften vornahmen. Diese beiden Anforderungen sollen auch für unsere adaptierte Variante, die wir aufgrund bestehender Schwächen des OP-Modells definieren, Gültigkeit besitzen. Eine Bedingung, die wir so in ihrer Universalität jedoch nicht übernehmen können, ist die folgende⁴:

“Any thing of any level except the lowest must possess a decomposition into things belonging to the next lower level. In this sense each level, will be as it were a 'common denominator' for the level immediately above it.”

Diese Bedingung würde unser gesamtes Modell von vornherein auf einen rein deterministischen Ansatz beschränken, was, wie wir später sehen werden, nicht nur metaphysische sondern auch rein formale Probleme mit sich bringen würde und deshalb verworfen wird. Die wesentlichen Adaptionen des OP-Modells finden sich aber auf der Ebene der Reductive Levels selbst, wie in Abbildung 3.1 veranschaulicht wird. Man sieht sofort, dass es nur zwei wesentliche Änderungen an der Spitze des Modells gibt: Soziale Systeme werden durch Ökosysteme ersetzt und eine neue, siebente Ebene Astronomische Systeme kommt hinzu.

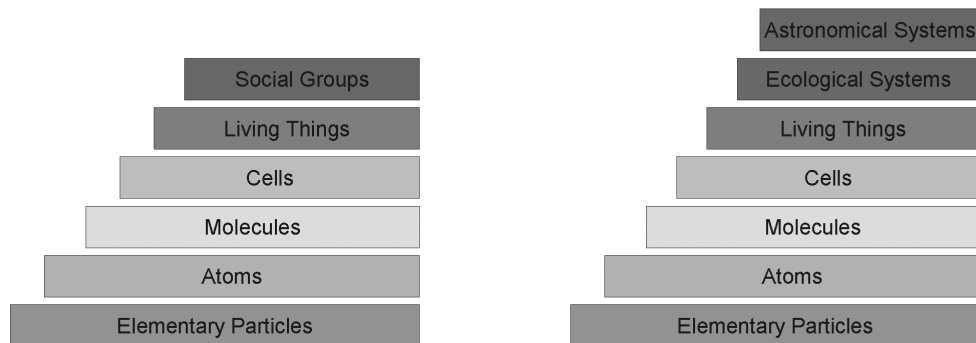


Abbildung 3.1: Gegenüberstellung des OP-Modells (links) und des CGST-Modells für Reductive Levels.

Da wir ansonsten mit den Abbildungen von Oppenheim und Putnam der Wissenschaften auf die Reductive Levels übereinstimmen, gibt es an dieser Stelle keine detaillierte Aufstellung ebendieser. Vielmehr wollen wir nun die beiden

⁴Ebenda, Seite 9.

Änderungen am Modell begründen.

Schauen wir uns dazu den Sprung von der Ebene der Organismen zu jener der Sozialen Systeme (oder Sozialen Gruppen) genauer an. Die Frage, die wir klären müssen, ist: Sind die Zusammenhänge in einem rein Sozialen System, wie es von Oppenheim und Putnam oder noch restriktiver etwa von Niklas Luhmann definiert wird, stärker als jene Zusammenhänge, die sich in einem Ökosystem, also in einer Gemeinschaft von Menschen, Tieren und Pflanzen – kurz, in einer Gemeinschaft von Organismen – finden lassen? Und unsere Antwort wird lauten: Nein. Warum ist das die Frage, die wir stellen müssen? Weil nach dem OP-Modell auf die Ebene der Organismen eine nächste Ebene folgen muss, die die entsprechenden Bedingungen erfüllt. Diese Ebene muss sich irgendwie aus Elementen bzw. Systemen unterer Ebenen zusammen setzen und durch Wissenschaftsfeld und Klassifikation der Elemente von den unteren Ebenen so abheben, dass Systeme unterer Ebenen keine Elemente auf dieser neuen Ebene beinhalten. Die Schlussfolgerung⁵ im OP-Modell sieht dazu so aus:

“By comparison with what we shall find on lower levels, the micro-reduction of level 6 to lower ones has not yet advanced very far, especially in regard to human societies. [...] Many writers believe that there are some laws common to all forms of animal association, including that of humans. [...] With respect to the problem of human social organization, as will be seen presently, two things are striking: (1) the most developed body of theory is undoubtedly in the field of economics, and this is at present entirely micro-reductionistic in character; (2) the main approaches to social theory are all likewise of this character.”

Oppenheim und Putnam ordnen alle sozialen Tiere auf dieser Ebene ein, ein Ameisenhaufen würde sich hier also auch untersuchen lassen. Folgt man der Idee der heutigen Sozialwissenschaften, etwa der Definition Luhmanns, eines sozialen Systems, scheidet der Ameisenhaufen aus. Dieser Unterschied sei hier noch der Vollständigkeit halber erwähnt. Beides ist uns aber für den nächsten Schritt zu eingeschränkt: Pflanzen, Tiere und Menschen bilden gemeinsam sensible Systeme genau eine Ebene über jener, in der ihre individuelle Existenz angesiedelt ist und beeinflussen sich gegenseitig, kommunizieren auch untereinander. Ein Politisches System oder eine Familie ist da eine mehr oder weniger abgrenzbare Untereinheit, sicherlich aber keine Zwischenstufe auf einem Stufenmodell der Komplexität von Systemen. Es wäre wahrscheinlich aufgrund der humanistischen Fokussierung im Sinne Bertalanffys, von Menschen gebildeten Gruppen einen besonderen Stellenwert zu verleihen – hier müssen wir dann aber mit der humanistischen Intention der Allgemeinen Systemtheorie Bertalanffys ausnahmsweise brechen und um der Konsistenz willen Gruppen von Tieren und Pflanzen auf die selbe Ebene wie solche von Menschen stellen. Denn was den

⁵Siehe [36], Seite 16 ff.

Menschen vom Tier unterscheidet, ist, mag es in seinen Konsequenzen noch so großartig sein, doch nur eine andere Art der Konfiguration auf zellulärer Ebene bzw. die Auswirkungen, seien sie nun emergent oder deterministisch, ebendieser.

Die Einführung der neuen obersten Ebene Astronomischer Systeme ist weit weniger problematisch. Es findet sich keine klare Aussage darüber in Oppenheim und Putnams Beitrag, warum auf der Ebene der Sozialen Systeme bereits Schluss ist. Unbestritten hat die theoretische und praktische Untersuchung der Eigenschaften astronomischer Ereignisse mit Fragestellungen wie dem Ursprung des Universums, seiner Ausdehnung und der Bewegung von Himmelskörpern sowie den engen Überschneidungen mit der Quantenphysik, aber auch durch wissenschaftlich historische Momente wie der bemannten Mondlandung in den letzten 50 Jahren seit der Definition des Modells von Oppenheim und Putnam ein überdurchschnittliches Wachstum erfahren – empirisch belegen lässt sich diese Behauptung etwa durch die Steigerung der Ausgaben weltweit für die Raumfahrt, stellvertretend sei hier die Gründung der NASA im Jahr 1958 und die darauffolgende Entwicklung ihres Budgets erwähnt⁶.

Auch trägt das Buch von Erich Jantsch, auf das wir später noch zu sprechen kommen werden, bewusst den Titel “Die Selbstorganisation des Universums” und nicht “Die Selbstorganisation der Welt” – zu deutlich sind nämlich auch die Organisationprozesse etwa bei der Bildung von Sonnensystemen, wenn sich aus Staub und Gas Protosterne und protoplanetare Scheiben und in weiterer Folge Sterne und Planeten bilden. Überhaupt kommt einer der interessantesten Fragen zu Systemen, der sich auch Bertalanffy in [4] gewidmet hat, nämlich jener nach dem Fließgleichgewicht, in ihrer Ausprägung auf der astronomischen Ebene eine ganz besondere Bedeutung zu: Während für Systeme auf allen unteren Ebenen stets eine Differenzierung zwischen Systemen im Fließgleichgewicht und solchen, die dem 2. Hauptsatz der Thermodynamik folgen, möglich ist, ist heute die Frage nach der Beschaffenheit des Universums und dessen vermeintlichen Zusteuerns auf den Wärmetod eine der meist diskutierten in der Physik. Diese und viele weitere Entwicklungen und Besonderheiten der Astrophysik und verwandter Disziplinen fordern die Einführung unserer neuen obersten Schicht im CGST Reductive-Levels-Modell.

Diese Form der Klassifizierung der Natur bzw. von Systemen ist nicht ohne gewisse Schwächen. Insbesondere sagt die Hierarchie sehr wenig über die Qualität der Organisation und Komplexität der am höchsten entwickelten (bekanntesten) Systeminstanzen auf der jeweiligen Ebene aus. Nach heutigem Wissensstand würden wir dem Menschen und seiner einzigartigen Fähigkeit zu Selbsterkenntnis, vorausschauendem Denken und Reflexion einen besonderen Qualitätsstatus zusprechen, der sich sonst auf keiner anderen Ebene – als der

⁶Siehe http://en.wikipedia.org/wiki/NASA_budget#Annual_budget.2C_1958-2008

der Lebewesen bzw. der Ökosysteme, wenn man Gesellschaften als Teilsysteme von ebensolchen betrachtet – finden lässt. Diese Tatsache legt nahe, dem Menschen als Lebewesen und Gesellschaften als soziale Gruppen eigenständige Ebenen mit ganz besonderen Eigenschaften zuzuweisen. Es wird an späterer Stelle in dieser Arbeit noch die These verteidigt, dass nicht nur der Turing-Test einmal erfolgreich zugunsten der Computer ausgehen wird, sondern Maschinen auch prinzipiell in der Lage dazu sind, Bewusstsein und Kreativität zu entwickeln. Wäre dies der Fall, stellt sich aber auch hier dann das Problem der hohen Qualität eines Systems auf einer noch niedrigeren Ebene (jener der physikalischen Systeme nämlich). Auch gibt es Ansätze in der Wissenschaft, die Bewusstsein und damit qualitativ hochwertige Systemstrukturen auf höheren Ebenen, insbesondere auf jener des globalen Ökosystems eines Planeten wie der Erde vermuten. Eine wissenschaftliche Basis für komplexes planetares Verhalten und das Aufzeigen von etwa homöostatischen Prozessen publizierte James Lovelock im Rahmen seiner Gaia-Hypothese. Heute werden insbesondere auf Basis der weltweiten Kommunikation über das Internet soziologische Ansätze wie jener der Noosphäre diesem Bereich zugeordnet. Freilich sind Theorien, die von einer Bewusstseinsstufe auf der Ebene von Ökosystemen ausgehen, der Wissenschaft kaum zugänglich und werden daher zurecht auch eher dem Bereich der Esoterik zugeschrieben – völlig außer Acht lassen oder in die Absurdität abschieben darf man aber auch solche Ansätze nicht, denn aus rein systemischer oder systemtheoretischer Sicht gibt es keinen Hinweis auf eine Regel oder einen Umstand, der derartigen Systemen eine Bewusstseinsentwicklung bzw. die dafür notwendigen Strukturen und Prozesse unmöglich machen würde.

Freilich gibt es zum hier vorgestellten, vom OP-Modell abgeleiteten Schema der Reductive Levels auch alternative Ansätze. Kenneth Boulding etwa publizierte im ersten Jahrbuch der Society for General Systems Research – im nächsten Abschnitt wird eine weitere Publikation ebendort sehr wichtig für die Ausarbeitung des Systembegriffs selbst sein – einen Ansatz⁷ “to the organization of general system theory”, den er als “levels of theoretical discourse” bezeichnete. Boulding sah grundsätzlich zwei Möglichkeiten, sich einer Klassifizierung im Rahmen einer GST anzunähern: Entweder über “general phenomena which are found in many different disciplines”⁸ wie etwa Wachstumsgesetze, System-Umwelt-Beziehungen, Informationstheorie u.ä. Oder aber eben über eine Art der Klassifizierung von Systemen aus unterschiedlichen Wissenschaften selbst. Hier beschreibt er ein neunstufiges Modell⁹: (1) frameworks, (2) clockworks, (3) thermostat, (4) open system, (5) plant, (6) animal, (7) human, (8) social organization und (9) transcendental systems. Es finden sich viele Überschneidungen mit dem OP-Modell, so ist (1) die Ebene der Atome, mit (4) ist die Ebene der Zellen gemeint u.ä. Bouldings Schema ist aber nicht so analytisch durchdacht wie jenes von Oppenheim und Putnam. Nach heutigem Stand der Wissenschaft sind der Übergang zwischen Pflanzen- und Tierwelt und auch die

⁷Siehe [10].

⁸Siehe ebenda, Seite 13.

⁹Siehe ebenda, Seite 14 ff.

zugeordneten Eigenschaften weitaus fließender als noch vor 50 Jahren vermutet. Ein "harter Schnitt" ist hier wahrscheinlich gar nicht mehr möglich. Boulding versucht zwar, eine Art qualitatives Schema einzuführen, stößt dabei aber bei der Zuordnung der Ebenen zu einzelnen Problemen der Wissenschaftsfelder auf den Umstand, dass eben die vermeintliche Hierarchieordnung wenig über die tatsächliche Qualität der Ebene aussagt bzw. hier keine hierarchische Ordnung ohne Bruch möglich ist. Auch stellt seine oberste Ebene der transzendenten Systeme, die er so begründet¹⁰

"To complete the structure of systems we should add a final turrent of transcendental systems, even if we may be accused at this point of having build Babel to the clouds. There are however the ultimates and absolutes and the inescapable unknowables, and they also exhibit systematic structure and relationship. It will be a sad day for man when nobody is allowed to ask questions that do not have any answers."

einen wissenschaftlich nicht weiter erklärten Bruch mit einer sonst klar technisch bzw. wissenschaftlich begründeten Klassifizierung dar. Diese neunte Ebene scheint Boulding eine Art Rettungsboot zu sein, in das man später alle Probleme, an deren Lösung man scheitert, hineinwerfen kann um sie der nicht zugänglichen Ebene der Transzendenz zuzuschreiben. Dinge, die dem menschlichen Geist und damit der Wissenschaft nicht zugänglich sind, haben aber auch in einem wissenschaftlichen Modell bzw. einer Theorie nichts verloren. Jedenfalls sind Bouldings Ebenen zu undeutlich beschrieben, um aus ihnen eine formale, mathematische Definition abzuleiten.

Was aber im OP-Modell bleibt ist das Fehlen einer Qualitätskennzeichnung der einzelnen Ebenen bzw. ihrer möglichen Teilsysteme. Es wäre hilfreich, klassifizieren zu können, dass wir im System des Menschen bzw. im System der sozialen Gruppen Charakteristika vorfinden, die eine einzelne Zelle nicht aufweisen kann; und zwar Charakteristika, die nicht nur einfache oder komplexere Eigenschaften des Systems beschreiben, sondern die von derart zentraler Bedeutung für das System sind, dass sie tatsächlich als qualitativ anderswertig einzustufen sind. Eine Möglichkeit dazu, die von den bisher vorgestellten Arbeiten abgeleitet werden kann, und bei Hofkirchner bereits auf den Punkt gebracht ist, ist eine Einteilung in verschiedene Kognitionsstufen (K1) Musterbildung, (K2) Symbolbildung und (K3) Ideenbildung, die für das CGST Modell als Kennzeichnungen für Systeminstanzen übernommen und integriert werden können.

Diese Qualitätskennzeichnung könnte auch umfangreicher berücksichtigt werden, wie das etwa Stanley Salthe in seinen Publikationen vorschlägt¹¹.

¹⁰Siehe ebenda, Seite 16.

¹¹Siehe [41], Seite 9.

“The real question for us is what is the basic structure (or order) of the world? [...] It must be a structure that is spontaneously stable. It must be a structure that generates things, boundaries, at least in our interaction with it. Very few will do all these; one that will is hierarchical structure – that is, nature viewed as a hierarchy of entities existing at different discrete levels of organization.”

Salthe beschreibt dann grundsätzlich zwei Möglichkeiten der sinnvollen Herangehensweise¹².

“Note that the major characteristic of these patterns is difference in scale or importance; some things are bigger than others, some exercise greater (broader) influence than others. [...] It will be the view of this work that the world is ordered in both of these ways simultaneously and that, since nesting has characteristics of both, the material aspect of the world are ordered in a nested manner.”

Das Resultat, zu dem er nach einer Betrachtung bis dato bestehender anderer Schemata, etwa von Bunge oder MacMahon, kommt, ist ein zweidimensionales Modell, bestehend aus einer “Ecological Hierarchy of Nature” und einer “Genealogical Hierarchy of Nature”. So ein zweidimensionales Modell mag Sinn machen, wenn man, wie Salthe, auf dieser Hierarchie die weiterführende Weltansicht begründet. Im CGST-Modell sind die Reductive Levels aber nur eine Hilfestellung zur Clusterung der Systemwissenschaften und ihrer Einheiten, um eine Grundstruktur zur Betrachtung der Probleme zur Verfügung zu stellen. Tatsächlich finden Übergänge zwischen den Stufen so vielfältig und in so unterschiedlicher Qualität statt, dass trotz vermeintlicher emergenter Sprünge von “einer Welt”, und nicht von einem in Naturstufen zerpflückten Weltbild die Rede sein muss. Deshalb scheint von allen bisherigen Ansätzen die gewählte, adaptierte Variante des OP-Modells hier am geeignetsten.

Eine abschließende Ergänzung zum Verhältnis der einzelnen Reductive-Levels zueinander können wir einer generischen Stufenmodellbeschreibung aus [23] entnehmen, die – die Begriffe der Selbstorganisation und Emergenz seien hier bereits vorweggenommen – sehr treffend das Zusammenspiel der einzelnen Ebenen in einer umfassenden Abbildung der Realität beschreibt:

“So wird die Theorie evolutionärer Systeme die Realität als Gesamtheit der Bewegung auseinander hervorgegangener, sich gegenseitig beeinflussender und auch weiterhin in Entwicklung befindlicher Systeme begreifbar. Das heutige Universum erscheint mit all seinen Teilsystemen als aus einer Kette ineinandergeschachtelter Selbstorganisationszyklen hervorgegangen, in denen jeweils ein System der

¹²Siehe ebenda, Seite 9.

Stufe n zur Basis für die Emergenz eines Systems auf der Stufe n+1 wird.”

3.2 Was ist ein System?

Systeme gibt es in allen Wissenschaftsdisziplinen unserer Reductive Levels – das ist die Grundannahme, die auch Oppenheim und Putnam in ihrem Beitrag verfolgen. Anders formuliert kann man sagen: In all diesen Wissenschaftsbereichen gibt es auch systemische Ansätze, um sich wissenschaftlich an die Analyse von Verhalten und Eigenschaften der involvierten Objekte und ihrer Zusammenhänge anzunähern. Das Zusammenspiel von Molekülen, die wiederum als Zusammenschluss von einzelnen atomaren Elementen betrachtet werden können, zu sich selbst reproduzierenden Lebewesen erfährt schon seit vielen Jahren populäre systemische Annäherungen, wie in Kapitel 6 später noch gezeigt wird; eine Gesellschaft kann als soziales System beschrieben und untersucht werden¹³; das Zusammenspiel von Menschen, Tieren, Pflanzen und nicht-lebendiger Materie kann in Ökosystemen und ihren Prozessen zusammengefasst werden¹⁴. Nun ist es aber, wie wir in Kapitel 2 festgestellt haben, nicht so einfach, eine allgemeine formale Grundlage zu schaffen.

Bertalanffy beginnt einen Aufsatz¹⁵ in [3] zwar mit dem Versuch, das Systemkonzept pragmatisch von “unten” herauf zu konstruieren, belässt es aber bei einer sehr philosophischen Umschreibung desselben. Interessant ist dennoch seine Unterteilung der unterschiedlichen Möglichkeiten, Differenzierungen in Systemen in Bezug auf deren Elemente formal abzubilden. Er teilt diese Möglichkeiten in drei Gruppen ein und beschreibt, Systeme können sich anhand der Anzahl, des Typs und der Anordnung (also der Relationen zwischen den Elementen) unterscheiden. Das war es dann auch schon – womit er prinzipiell Recht hat und was später für die Wahl der formalen Abbildung der statischen Komponente in unserem Modell eine wichtige Rolle spielen wird.

Sehr praktisch endet dieser Entwurf dann aber, wie wir bereits festgestellt haben, schlussendlich nicht, da die zu Beginn vorgestellten Graphenmuster schon bald in nicht mehr durchschaubaren Differentialgleichungssystemen münden. Eine wichtige Art der Differenzierung hat Bertalanffy außerdem vergessen oder zumindest nicht hinreichend deutlich beschrieben: nämlich jene der unterschiedlichen Relationen zwischen den Elementen.

Hall und Fagen sind das Ganze dann schon etwas geordneter angegangen¹⁶.

¹³Siehe [31]

¹⁴Siehe [13].

¹⁵Siehe [3], Seite 54 f.

¹⁶Der Aufsatz von Bertalanffy stammt aus dem Jahre 1950, der Entwurf von Hall und Fagen aus 1956.

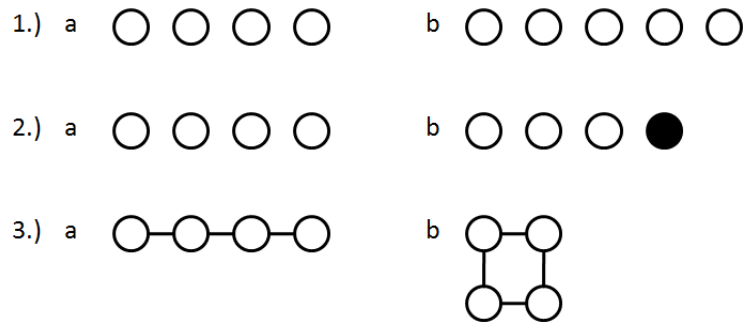


Abbildung 3.2: Klassifikation von Unterschieden von Elementbeziehungen in [3].

Mit ihrer Systemdefinition wollen wir uns nicht nur jetzt begrifflich ausführlicher auseinander setzen, sondern diese Analyse dann auch im nächsten Kapitel in die Grundlage des computationalen Modells, die Systemontologie, überführen. Hall und Fagen beschränken sich bei der Einleitung zu ihren Definitionen auf die folgende Aussage¹⁷:

“The plan of the present paper is to discuss properties of systems more or less abstract; that is to define systems and to describe the properties that are common to many systems and which serve to characterize them.”

Den Systembegriff selbst beschreiben sie dann einfach aber treffend folgendermaßen¹⁸:

“A system is a set of objects together with relationships between the objects and between their attributes.”

Mehr ist und darf aus dem Systemkonzept an sich auch nicht abgeleitet werden: Ein System besteht aus einer Menge von Elementen (*objects*). Diese Elemente können atomare Eigenschaften (*attributes*) besitzen. Außerdem können Beziehungen unterschiedlicher Art (*relationships*) zwischen den einzelnen Elementen bestehen. Hall und Fagen präsentieren keine formalen logischen Definitionen für die Begriffe und Zusammenhänge, die sie einführen, sondern beschränken sich auf klar abgetrennte natürlichsprachliche Aussagen – was angesichts der untergeordneten Bedeutung der mathematischen Logik zu Beginn der zweiten Hälfte des 20. Jahrhunderts nicht weiter verwunderlich ist. Diese fehlenden, jedoch zwingend notwendigen logischen Formulierungen werden wir nun nachholen. Wir werden uns dabei in diesem Kapitel auf abstrakte, mittels Description Logics (DL) nicht näher definierter Ausdrucksstärke, logische

¹⁷Siehe [21], Seite 18.

¹⁸Siehe ebenda, Seite 18.

Beschreibungen beschränken und diese im nächsten Kapitel in die konkrete Sprache bzw. den konkreten Dialekt unseres Modells überführen.

Ein System wird definiert als die Vereinigungsmenge von unterschiedlichen Element-Konzepten. Hier tritt aus logischer Sicht auch schon gleich die erste interessante Frage auf: Sind Elemente in der Folge in ihrer Konzepteigenschaft als atomar zu betrachten oder können sie aus logischen Verknüpfungen anderer Elemente hervorgehen bzw. derart definiert werden? Da uns “komplexe” Elemente später in Entscheidungsprobleme bringen würden und wir dem Elementbegriff in dieser Arbeit eben diese atomare konzeptuelle Rolle zuweisen, werden wir keine abgeleiteten Elemente auf der Ebene der statischen logischen Definitionen erlauben. Für jede Art des Zusammenschlusses von Elementen in einer Art und Weise, die noch nicht das System als solches ausfüllt, führen wir den Begriff des Subsystems ein, der sich auch bereits bei Hall und Fagen findet¹⁹:

“It is clear from the definition of system and environment that any given system can be further subdivided into subsystems.”

Bevor wir hier zu einer endgültigen logischen Definition kommen, müssen wir noch die Beziehungen und Attribute der Elemente definieren. Beide sind sich formal sehr ähnlich, da bei den Beziehungen Relationen zwischen Elementen und bei den Attributen Relationen zwischen jeweils einem Element und einem atomaren Datentyp, der formal als sprachliche Konstante definiert ist, definiert werden. Zur einfacheren Handhabung führen wir aber noch Mengen für die Elemente (Γ), die Relationen (Θ) und die Attribute (Δ) des Systems ein. Für das System selbst wählen wir S als Konzeptbezeichnung. Eine erste intuitive Überführung in eine Description-Logic-Formel wäre dann

$$S \equiv \Gamma \sqcap \Theta \sqcap \Delta$$

d.h. ein System ist exakt definiert durch seine Elemente, Attribute und Relationen. Das ist aus mehreren Gründen problematisch: Zuerst bedeutet der \equiv -Operator eine Unterklasse-Oberklasse-Definition, d.h. eine echte Gleichheit der beiden Ausdrücke links und rechts. Das ist bei unseren Anwendungen vielleicht nicht immer wünschenswert, weil wir ja auch unterschiedliche Systeme, z.B. S_1 und S_2 haben möchten, die nur aufgrund ihrer Benennung eine Identität besitzen und beide etwa eine ganz einfache Bestandsmenge $\Gamma_1 = \{A, B\}$ besitzen. Aus der Definition oben würde aber folgen

$$S_1 \equiv \Gamma_1 \text{ und } S_2 \equiv \Gamma_1 \text{ impliziert } S_1 \equiv S_2$$

¹⁹Siehe ebenda, Seite 20.

Weil wir aber vielleicht Unterschiede im Sinn haben, die sich erst durch die System-Dynamik ergeben, wollen wir die begriffliche Identität auf jeden Fall beibehalten und müssten deshalb eigentlich sagen

$$S \sqsubseteq \Gamma \sqcap \Theta \sqcap \Delta$$

Auch diese Definition ist noch immer praktisch nicht ganz glücklich, obwohl sie formal korrekt hinzubekommen wäre. Wir haben jetzt nämlich noch – ohne bisher auf die Details der Relationen eingegangen zu sein – folgendes logisches Problem: Sind Elemente oder Relationen in einem intendierten System nur optional, müssten wir noch entsprechende Restriktionen einführen, um die Ausdrücke logisch konsistent zu halten. Wenn wir etwa ein System

$$\text{Gruppe} \equiv \Gamma_{\text{Gruppe}} \text{ mit } \Gamma_{\text{Gruppe}} = \{\text{Mann}, \text{Frau}\}$$

definieren, bekommen wir ob der tatsächlichen logischen Auflösung der Menge – unabhängig davon, welchen Weg wir wählen – ein Problem, wenn wir auch Gruppen, die nur aus Frauen bestehen, zulassen wollen. Denn die naheliegende allgemeine Variante

$$\Theta_{\text{Gruppe}} \sqsubseteq \exists \text{hatGruppenmitglied.Mann} \sqcap \exists \text{hatGruppenmitglied.Frau}$$

der Auflösung von $\Theta_{\text{Gruppe}} = \{\text{hatGruppenmitglied}\}$ scheitert in diesem speziellen Fall genauso wie

$$\Theta_{\text{Gruppe}} \sqsubseteq \forall \text{hatGruppenmitglied.Mann} \sqcap \forall \text{hatGruppenmitglied.Frau}$$

wobei diese zweite Variante an sich schon logisch nicht mehr konsistent ist. Eine Quantifizierung der entsprechenden Relationen wäre hier die einzige formal korrekte Lösung, d.h. eine Annotation der jeweiligen Relationen mit ≥ 0 , die dann auch eine leere Lösungsmenge einzelner relationaler Ausdrücke erlauben würde. Das macht die Ausdrücke aber nicht nur unübersichtlich und komplexer, sondern führt auch später bei der Übertragung in die Ontologiesprache zu unerwünschten Effekten, weshalb wir hier eine andere, formal lockerere Variante zur Darstellung wählen, die sich später auch einfacher ontologisch abbilden lässt. Wir definieren das System an sich als Tripel

$$S = \langle \Gamma, \Theta, \Delta \rangle$$

wobei die drei Teilmengen jeweils genauer definiert sind durch

$$\Gamma = \{Element_1, \dots, Element_n | \forall x \in \Gamma : x \sqsubseteq Element \wedge hasElement(S, x)\}$$

Element ist dabei ein generisches Konzept für Elemente, das das allgemeine Objekt eines Elements in der Allgemeinen Systembeschreibung kennzeichnet. *hasElement* ist eine generische Beziehung zwischen dem System und allen Elementen des Systems, die diese Zuordnung formal repräsentiert. Bei Hall und Fagen lesen wir zum Element-Konzept²⁰:

“Objects are simply the parts or components of a system, and these parts are unlimited in variety. Most systems in which we will be interested consist of physical parts: atoms, stars, switches, masses, springs, wires, bones, neurons, genes, muscles, gases, etc. We also admit as objects abstract objects such as mathematical variables, equations, rules and laws, processes, etc.”

Wichtig an dieser Stelle ist die Differenzierung zu atomaren Eigenschaften, da Elemente wie bereits oben beschrieben in unterschiedlichen formalen Beziehungen zu sich selbst oder anderen Elementen stehen können müssen²¹:

“The relationships to which we refer are those that tie the system together. It is, in fact, these relationships that make the notion of system useful.”

Die Menge Θ aller Elementrelationen muss dabei jeweils Domäne und Abbildungsraum der Relationen definieren, wobei beide jeweils wieder aus einer Teilmenge von Γ bestehen können und sich nicht auf einzelne Element-Konzepte beschränken müssen.

$$\Theta = \{ElementRelation_1(Element_{x_1}, Element_{x_2}), \\ ElementRelation_2(Element_{x_3}, Element_{x_4}), \\ \dots, ElementRelation_n(Element_{x_{n-1}}, Element_{x_n}) | \\ \forall x \in \Theta : x(e_m, e_n) \rightarrow elementRelation(e_m, e_n)\}$$

elementRelation ist dabei ein generisches Objektattribut für eine Element-Element-Relation, das die allgemeine Beziehung zwischen Elementen in der Allgemeinen Systembeschreibung kennzeichnet. Als Ergänzung dazu werden nun noch die atomaren Element-Eigenschaften definiert, die sich – zumindest in der jeweiligen Systeminstanz – nicht weiter zerlegen lassen und bei Hall und Fagen als “attributes” bezeichnet werden²²:

²⁰Siehe ebenda, Seite 18.

²¹Siehe ebenda, Seite 18.

²²Siehe ebenda, Seite 18.

“Attributes are properties of objects. For example, in the preceding cases, the objects listed have, among others, the following attributes: atoms – the number of planetary electrons, the energy states of the atoms, the number of atomic particles in the nucleus, the atomic weight; stars – temperature, distances from other stars, relative velocity.”

In der Informatik werden atomare Eigenschaften meist auf vordefinierte Datentypen – Strings, Zahlen, Booleans, u.ä. – abgebildet, um Speicherauslastung, Eigenschaften der Werteausprägungen etc. im Programm effizienter verwalten zu können. Description Logics behelfen sich hier mit einem “Trick”, indem sie Datentypen wie Konzepte behandeln, die aber in der späteren Übersetzung in XML-Syntax wieder auf einen entsprechenden Datentyp-Standard verweisen.

$$\Delta = \{ \text{Attribut}_1(\text{Element}_{x_1}, \text{Datatype}_1), \\ \text{Attribut}_2(\text{Element}_{x_2}, \text{Datatype}_2), \\ \dots, \text{Attribut}_n(\text{Element}_{x_n}, \text{Datatype}_n) \mid \\ \forall x \in \Delta : x(e_m, d_n) \rightarrow \text{attribute}(e_m, d_n) \}$$

attribute ist dabei ein generisches Datenattribut für eine Element-Datentyp-Relation, das die allgemeine Zuordnung von atomaren Eigenschaften zu Elementen in der Allgemeinen Systembeschreibung kennzeichnet.

Weiters definieren wir beispielhaft einige Axiome, um die Intention des Modells durch den Ausschluss sinnfreier Systeminstanzen zu verstärken. Insbesondere wollen wir dabei die folgenden Kriterien gewährleisten.

A1. Minimale Systeminstanzen. Ein System mit $\Gamma = \{\}$, $\Theta = \{\}$ und $\Delta = \{\}$ das ausschließlich durch den Begriff definiert wird, soll aufgrund der nun gewählten formalen Repräsentationsform nicht erlaubt werden. Wir fügen deshalb unserem Modell die folgenden Axiome hinzu

$|\Gamma| \geq 1$, d.h. es muss mindestens ein Element-Konzept im System geben

A2. Transitivität der Subsysteme. Subsysteme bezeichnen untergeordnete Organisationseinheiten in Systemen, bei denen es sich formal zwar ebenfalls um Systeme handelt, die aus Perspektive der wissenschaftlichen Betrachtung aber keine abgeschlossene autonome Systemeinheit bilden. Für solche Subsysteme muss gelten, dass sie eine klare hierarchische Ordnung besitzen. Ist also S_2 Subsystem von S_1 und S_3 Subsystem von S_2 , dann ist auch S_3 Subsystem von S_1 . Formal muss also gelten

$$\text{hasSubSystem}(S_1, S_2) \wedge \text{hasSubSystem}(S_2, S_3) \rightarrow \text{hasSubSystem}(S_1, S_3)$$

A3. Eindeutige Zuordnung zu Reductive Levels. Systeme und Elemente müssen den Regelungen des OP-Modells folgend klar einem Reductive Level, und nur einem Reductive Level zugeordnet werden. D.h. ist ein System bereits einem Reductive Level zugeordnet, kann es keinem zweiten, unterschiedlichen Reductive Level zugeordnet werden.

$$hasReductiveLevel(S_1, R_1) \wedge hasReductiveLevel(S_1, R_2) \rightarrow R_1 = R_2$$

Dasselbe muss auch für Elemente indirekt über deren Zuordnung zu Systemen (auch zu unterschiedlichen Systemen, denn das ist ja grundsätzlich erlaubt) gelten.

$$\begin{aligned} & | hasReductiveLevel(S_1, R_1) \wedge hasElement(S_1, E_1) | \rightarrow | \\ & || \neg \exists S_x | hasReductiveLevel(S_x, R_y) \\ & || \wedge hasElement(S_x, E_1) \\ & || \wedge R_y \neq R_1 \end{aligned}$$

Die Verletzung eines dieser beiden Reductive-Levels-Axiome würde die weitere Verarbeitung durch die Inkonsistenz im Stufenmodell erheblich erschweren. Sehrwohl ist eine multiple indirekte Zugehörigkeit zu verschiedenen Reductive Levels aber natürlich bei Attributen möglich, da diese ja nur grundsätzliche Eigenschaften von Elementen auch unterschiedlicher Natur repräsentieren. So kann also zum Beispiel das *Gewicht* oder das *Alter*, wenn es als Attribut repräsentiert wird, durchaus sowohl einem Element auf der Ebene R_1 als auch einem Element auf der Ebene R_2 zugeordnet werden, ohne dass sich dabei Konsistenzprobleme ergeben.

Eine Reihe weiterer Axiome werden bei Hall und Fagen, aber auch bei Bertalanffy informell angedeutet. Viele dieser Vorschläge bedürfen aber sicherlich einer eingehenderen Betrachtung und der Prüfung auf Praxistauglichkeit, sodass die Ausarbeitung eines Axiomenkatalogs für CGST in nachfolgenden Arbeiten eine umfassende, aber notwendige Aufgabe sein wird, um das Modell zu vervollständigen.

3.3 Systeminstanzen

Bisher haben wir nur einen ersten Schritt unternommen, die statische Systembeschreibung allgemein mit Hilfe von Description-Logic-Ausdrücken zu formalisieren. Dabei sind wir noch nicht darauf eingegangen, wie wir denn später konkrete Systeminstanzen anhand dieser Beschreibungen logisch einordnen werden. Es ergibt sich nämlich – wie in vielen Bereichen, die mittels DLs formal abbilden – eine grundsätzliche Problematik der Differenzierung zwischen Modell und In-

stanz.

Um das genauer zu erklären muss zuerst kurz auf die Art und Weise der Erstellung von Instanzen mittels DLs eingegangen werden. Konzepte und Definitionen, also der Beschreibungsbaum eines DL-Systems werden in der sogenannten Terminologie oder TBox ausgedrückt. Die Instanzen, die diese Konzepte und Definitionen für ein konkretes Modell erfüllen, bezeichnet man dann als “Weltbeschreibung” oder ABox. TBox und ABox gemeinsam repräsentieren die vollständige Wissensbasis des formalen Systems. Das bedeutet aber gleichzeitig auch, dass es keine Ebene dazwischen gibt und man sich bei der Modellierung nur entweder für eine Zuordnung zur TBox oder ABox entscheiden kann. Eine wirklich elegante Lösung zur Einführung von Zwischenebenen gibt es nicht, praktisch erfolgreiche Ansätze, die mit ähnlichen Problemen konfrontiert waren, schon.

Sehen wir uns die Problematik nun etwas genauer an: Wenn schlussendlich die Ontologie für unsere Allgemeine Systemtheorie in Form einer Web-Ontology-Language-Beschreibung (OWL-Beschreibung) fertig gestellt ist, enthält diese wie oben bereits vorbereitet abstrakte generalisierte Konzepte für Elemente, Relationen und Attribute sowie ein Konzept für das System selbst, für Subsysteme etc. Wenn wir jetzt, wie wir das auch in Kapitel 5 ausführlicher skizzieren, ein spezielles System auf Basis des CGST-Modells definieren, wollen wir ja auch einzelne Instanzen dieser ebenfalls noch abstrakten Beschreibung erzeugen. *Beispiel.* Ein System zur Definition von Molekülen wird basierend auf der CGST-Ontologie definiert und enthält ein Element Atom sowie Verbindungen zwischen den Atomen als Element-Relation. Eine einzelne Systeminstanz würde dann eine Menge von Molekülen mit konkreten Ausprägungen der Atome – etwa eine Instanz für das H_2O -Molekül – enthalten. Damit haben wir aber drei Stufen in unserem Modell:

- Die CGST-Ontologie zur Beschreibung von Systemeigenschaften im Allgemeinen;
- die einzelnen Systembeschreibungen mit unterschiedlichen Element-Konzepten, Relationen etc.;
- sowie die Instanzen der Systembeschreibungen mit konkreten Ausprägungen der Konzepte, Relationen etc.

Eine ähnliche Situation findet sich im Bereich des Semantic-Web-basierten E-Commerce, wo Produkte und Angebote basierend auf einer einheitlichen Ontologie zur Beschreibung von E-Commerce-Sachverhalten im Semantic Web dargestellt werden sollen. Hier gibt es nämlich ebenfalls eine abstrakte Beschreibung von Produkten und Dienstleistungen an sich (z.B. ein Konzept Mobiltelefon), die Zwischenebene der Produktmodelle (z.B. das Gerät Nokia 1100)

sowie die Instanzebene der konkreten einzelnen Produkte (z.B. ein ganz konkretes Nokia 1100 mit der Seriennummer 000001). Eine in der Praxis erfolgreiche Auflösung der Problematik findet sich in der Umsetzung der GoodRelations-Ontologie²³, wo für die Zwischenebene so genannte Proxy-Instanzen eingeführt wurden, die formal gesehen aber ebenfalls als ontologische Konzepte repräsentiert werden.

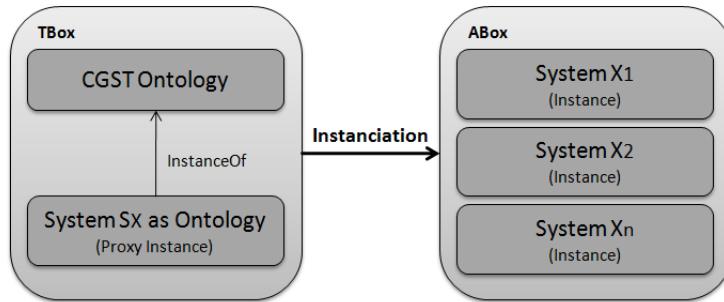


Abbildung 3.3: Abbildung der konkreten Systemmodelle als Proxy-Instanzen auf der TBox-Ebene.

Umgelegt auf unser Systemmodell bedeutet das nun, dass spezielle Systeme als Subklassen des allgemeinen Systemkonzepts, Elemente als Subklassen des allgemeinen Element-Konzepts, Element-Relationen als Subeigenschaften der allgemeinen Elementrelation, usw. repräsentiert werden und die Instanzebene erst für die tatsächlichen Systeminstanzen mit konkreten Ausprägungen herangezogen wird (siehe Abbildung 3.3). Um also etwa ein Systemmodell für Moleküle zu definieren, muss die TBox-Ebene nicht verlassen werden – erst bei der Instanzierung konkreter Moleküle wie etwa des H_2O -Moleküls erfolgt der Schritt auf die ABox-Ebene.

Dass dieser “Workaround” notwendig ist, hat zum einen mit der Entwicklung von DLs aus PL1, zum anderen mit der ursprünglichen Intention der Semantic-Web-Sprachen, zwischen Schemata und Instanzen – also wie in anderen Datenbanken auch, z.B. relationale Tabellen-Beschreibungen und deren Inhalte – zu differenzieren, zu tun. Als das Problem schließlich in der Praxis tragend und in den W3C-Arbeitsgruppen diskutiert wurde, war es für einen Paradigmenwechsel bereits zu spät bzw. sahen jene Vertreter, die das Semantic Web als große Datenbank (und nicht mehr) sahen, auch keine Notwendigkeit dazu. In der Beschreibungsrealität einer Allgemeinen Systemtheorie kommt man, wie in vielen anderen Bereichen auch, um diese “Dreifaltigkeit” von allgemeiner Systemontologie, spezieller Systemontologie und Systeminstanz aber nicht herum, sodass das gewählte Modell eine gute Methode zur Auflösung dieses formalen Problems darstellt.

²³Siehe <http://purl.org/goodrelations/>.

3.4 Regellogik im statischen System

Moderne Wissenssysteme bestehen konzeptuell in der Regel aus zwei Teilen: Aus der Wissensbasis an sich, in der alle Informationen des Systems in einer strukturierten Form hinterlegt sind, und einem Satz von allgemein gültigen Regeln, die – im Normalfall bei jeder Änderung in der Wissensbasis – auf ebene angewandt werden, um neues Wissen logisch abzuleiten. Dabei sind die damit verbundenen Paradigmen entscheidend: Wird die Wissensbasis ausgekoppelt und isoliert und zu einem beliebigen Zeitpunkt wieder mit dem Regelsatz verbunden, so muss die selbe, um inferiertes Wissen erweiterte Wissensbasis entstehen, d.h. die Ausführungsreihenfolge darf keinen Unterschied erzeugen. Außerdem darf die Wissensbasis zu keinem Zeitpunkt Inkonsistenzen enthalten, da die für die Ableitung verwendeten Reasoner an ebensolchen scheitern würden (“Aus einem Widerspruch kann alles hergeleitet werden”).

Eine Regellogik ist auch für CGST von großer Bedeutung, da ja gerade in einer Allgemeinen Systemwissensbasis die Analogiebildung und ähnliche Methoden ausgenutzt werden sollen, um aus bekanntem Wissen neues abzuleiten. Ein Teil der Regellogik wird sich dabei bei genauerer Betrachtung der Überführung der Konzepte, die bisher vorgestellt wurden, in eine konkrete Sprache von selbst ergeben: OWL integriert bereits wesentliche allgemeine mathematische Regeln, die untrennbar mit der jeweiligen Subsprache verbunden sind und damit systemimmanent bereits in CGST vorhanden sind.

Für individualisierte Regeln gebe es prinzipiell die Möglichkeit, sich auf die Ebene von PL1 zu begeben und hier auch auf eine ähnliche Syntax aufzubauen. Was aber jedenfalls auch für die Regelkomponente gelten soll, weil es eine der wesentlichen Anforderungen für CGST ist: Die einfache Bearbeitbarkeit, Transparenz und Kompatibilität mit bereits bestehenden formulierten Regeln soll gegeben sein, um die praktische Arbeit - auch mit Regeln und Ableitungen – möglichst gut zu unterstützen. Die hier zur Verfügung stehenden Ansätze sind allesamt nicht optimal und sind insbesondere in Hinblick auf die Optimierung der gekoppelten Reasoner bzw. Inferencer noch ausbaufähig. Trotzdem soll im nächsten Kapitel ein konkreter Ansatz präsentiert werden, der sich wohl auch innerhalb der Semantic-Web-Landschaft als Standard für dieses Problem durchsetzen dürfte.

3.5 Dynamische Systeme

Der Begriff der Ontologie, wie wir ihn von der Semantik der Semantic-Web-Sprachen übernommen haben, ist für eine Theorie von Systemen im Allgemeinen, die auch deren Entwicklung beschreiben soll, eigentlich etwas irreführend. Obwohl der Teil der Systeme, der ontologisch beschrieben wird, sich in der

Simulation – zumindest in der Dimension, in der wir diese Untersuchungen betrachten – nicht ändert und daher korrekt als ontologisch bezeichnet werden kann, handelt es sich bei der Simulation von systemischen Prozessen, insbesondere dort, wo wir Prozesse wie jenen der Selbstorganisation beobachten wollen, vielmehr um eine Ontogenese.

Systeme entwickeln sich also – mit der Zeit, würde man jetzt wohl sagen wollen. Tatsächlich werden wir aber im Gegensatz zu Bertalanffy und seinen Kollegen auf das Konzept einer kontinuierlichen, nahtlosen Zeit ohne Sprünge verzichten. Wenn Bertalanffy eine Systembeschreibung wie hier²⁴

$$\begin{aligned}\frac{dQ_1}{dt} &= a_{11}Q_1 + \dots a_{1s}Q_s + \dots \\ \frac{dQ_s}{dt} &= a_{s1}Q_1 + \dots \\ \frac{dQ_n}{dt} &= a_{ns}Q_s + \dots a_{n1}Q_1 + \dots\end{aligned}$$

präsentiert, in der die Entwicklung eines Systems mit seinen Teilkomponenten über ein Differentialgleichungssystem beschrieben wird, so übernimmt er diese Variante aus den gängigen mathematisch gestützten Wissenschaften wie der Physik, die auf ebendiesem kontinuierlichen Zeitkonzept basiert. Tatsächlich ist die Annahme, Systeme entwickeln sich nicht sprunghaft, sondern in einem nicht auf kleinste Schritte zerlegbaren Fluß, nicht zufriedenstellend auflösbar. Auf eine Zusammenfassung philosophischer oder metaphysischer Ansichten über das Wesen der Zeit werden wir an dieser Stelle jetzt verzichten. Aber spätestens seit den Erkenntnissen der Quantenphysik, insbesondere im Zusammenhang mit Untersuchungen zu Gravitationsbeziehungen, ist die Eigenschaft der Zeit als Kontinuum auch physikalisch umstritten: Im Bereich der Planck-Zeit von $5.39121 \cdot 10^{-44}$ s verlieren die sonst gültigen Naturgesetze ihre Gültigkeit, was ein Hinweis auf das Fehlen der Eigenschaft der Zeit eines Kontinuums in diesem Bereich ist. Eine ausführlichere physikalische Diskussion darüber findet sich etwa in [7]. Wesentlich ist dabei ganz einfach die Hypothese, dass es wie im Raum auch in der Zeit eine kleinste Einheit t_ϵ gibt, die nicht weiter zerlegbar ist – allerkleinste Zeitsprünge also anstatt einer kontinuierlichen, reellen Zeitachse also.

Auch die jüngsten viel Aufsehen erregenden Arbeiten des deutschen Astrophysikers Martin Bojowald²⁵ bedienen sich dieses Konzepts und bauen darauf neue Erklärungsmodelle für die Entstehung des Universums und den Zustand des Urknalls bzw. der Singularität und der “Zeit” davor. Bojowald teilt dabei das Universum in so genannte “Raumzeitatome” ein, d.h. er geht also ebenfalls von einer kleinsten Zeiteinheit, die nicht weiter zerlegbar ist, aus – und ist damit nicht alleine, wie die Repliken auf seine Theorien zeigen. Die Physik befindet

²⁴Siehe [3], Seite 71.

²⁵Für eine Zusammenfassung siehe Martin Bojowald, “Zurück vor den Urknall. Die ganze Geschichte des Universums.”, S. Fischer, Frankfurt am Main 2009 und Martin Bojowald et al., “Forget the Big Bang. Now it’s the Big Bounce.”, Scientific American, Oktober 2008.

sich also aktuell in einem Zeitdiskurs, und auch, wenn hier in den nächsten Jahren und Jahrzehnten keine Auflösung erwartet werden kann, so kann doch zumindest festgestellt werden, dass die unendlich kleine, nicht abzählbare Zeit als Begriff in Naturwissenschaft und Philosophie ernsthaft zur Diskussion steht, wie schon lange nicht mehr.

Für die Abbildung von Entwicklungsprozessen in Systemen werden wir daher in unserem Modell die klassische Idee des Automaten verwenden. Entwicklungen im Automaten passieren nicht über ein kontinuierliches Zeitkonzept, sondern sprunghaft über die Berechnung von Folgezuständen. Der angewandten Informatik bleibt ob der Beschränkung der Berechnungsgeschwindigkeit, d.h. des Prozessors, ja gar nichts anderes übrig: In einer bestimmten Zeiteinheit, etwa in einer Sekunde, können nur maximal T Prozessortakte und damit atomare Berechnungen durchgeführt werden. Damit ist es nicht möglich, ein kontinuierliches Zeitkonzept beliebig abzubilden, der Computer muss sich daher auf eine Simulation desselben beschränken. Zu derartigen Simulationen wollen wir die Anwendungen, die auf dem CGST-Modell basieren, aber gar nicht zwingen, sondern vielmehr auf der Ebene der diskreten Schritte im System bleiben – wir werden also keine Abstraktionsebene eines Zeitkonzepts einziehen.

Die Urform des Automaten in der Informatik ist die Turing-Maschine, auf deren spezielle Eigenschaften bezüglich der Berechenbarkeit wir später noch einmal zurückkommen werden, wenn wir uns mit dem Emergenzbegriff in unserem formalen System beschäftigen. Die Turing-Maschine als theoretisches Modell geht auf ein Konzept des Mathematikers Alan Turing zurück, das er zur Repräsentation eines damals noch theoretischen Computers, insbesondere in einer universellen Ausprägung und zur Darstellung von Fragen der theoretischen Berechenbarkeit entworfen hat. Der Ansatz ist so einfach wie genial: Mit Hilfe eines einfachen mathematischen Modells werden ein unendliches Speicherband, ein Lese-/Schreibkopf und ein endliches Set (Programmregister) von Befehlen repräsentiert. Die für die Informatik relevante Aussage in diesem Zusammenhang ist, dass jene Programme, und nur jene Programme, die mittels einer Turingmaschine beschrieben werden können, auch von einem realen Computer (theoretisch) berechenbar sind. Es gibt dabei Modelle für Turingmaschinen mit eingeschränkter Funktionalität und so genannte universelle Turingmaschinen, die als Eingabe ein beliebiges “Programm” in Form einer codierten Turingmaschine erhalten können – eine solche hat Alan Turing ebenfalls beschrieben. Das für die Praxis interessanteste Modell ist jenes einer deterministischen²⁶ Universellen Turingmaschine (UTM). Sie definiert das Fundament der praktischen Informatik und beinhaltet prinzipiell bereits den gesamten Möglichkeitsraum

²⁶Nichtdeterministische Turingmaschinen existieren ausschließlich in der Theorie und dienen der Berechnung von Komplexitätsklassen bzw. lassen sich über ebensolche bestimmte Probleme der Berechenbarkeit leichter eingrenzen. Eine nicht-deterministische Turingmaschine erlaubt Handlungsalternativen bei eindeutigen Zustands-Regel-Kombinationen, die dann – theoretisch – echt zufällig ausgeführt werden.

ebendieser. Ein wesentliches Problem, das hier für die Praxis entsteht, das so genannte Halteproblem, wird in Kapitel 6 noch ausführlicher behandelt.

John von Neumann, ein weiterer Gründungsvater der Informatik, der in erster Linie durch sein abstraktes Modell eines modernen Computers bekannt wurde, hat sich ebenfalls mit Automatentheorie beschäftigt und ist dabei zu einigen interessanten und für unser Modell wesentlichen Schlussfolgerungen gekommen. Neumann hat dabei das Konzept eines Zellulären Automaten (CA) entwickelt, d.h. einer Anordnung von theoretisch parallel arbeitender Automaten mit stark eingeschränkten Verhaltensoptionen. Dieses Modell konnte aber – obgleich eine praktische Umsetzung in Software zum damaligen Zeitpunkt noch gar nicht möglich war – bereits die erstaunliche Komplexität zeigen, die sich bei der Interaktion ganz einfacher Teilkomponenten ergibt. Neumann widmete sich dann einem speziellen Teilproblem im Bereich der Turingmaschinen: Den sogenannten “Universal Constructors” (UC), d.h. Turingmaschinen, die von sich selbst eine Kopie erstellen. Er entwarf dazu einen Zellulären Automaten mit 29 verschiedenen Zuständen, keiner bestimmten Nachbarschaft²⁷ und 84.000 Zellen. Im wesentlichen konzipierte er eine UTM, die dann genau diese Aufgabe, also ihre eigene Reproduktion, ausübte.

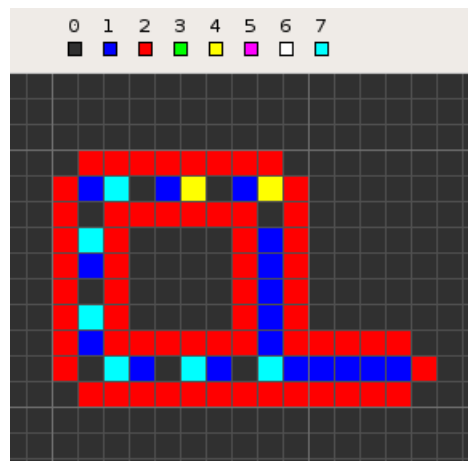


Abbildung 3.4: Die Langton-Schleife als historisches Beispiel für “gesteuerte” Selbstorganisation und Reproduktion in einem Computerprogramm. Bildquelle: Wikimedia Commons.

Das war im Jahr 1952 und damals zu weit weg von der computationalen Wirklichkeit, als dass sich die Wissenschaft eingehender mit diesem Konzept beschäftigte. 1971 beschäftigte sich A. R. Smith erneut mit dieser Problematik und konnte einen Beweis erbringen, dass jede Turing-Maschine auch mittels eines eindimensionalen Zellulären Automaten, d.h. mittels eines CA mit nur eindimensionaler Nachbarschaftsmatrix, abgebildet werden kann und Turing-Maschinen und eindimensionale CAs somit äquivalent sind, was ihre Berechnen-

²⁷Als “Nachbarschaft” wird in der Theorie Zellulärer Automaten jener Raum bezeichnet, der Auswirkungen auf das Verhalten einzelner Zellen hat.

barkeit betrifft. Smith konnte dabei sogar die Konfiguration einschränken: Es braucht theoretisch nur einen CA mit 8 Zuständen und einer 6er-Nachbarschaft, um dieses Resultat zu erzielen. Ein großer Unterschied im Vergleich zum Resultat von Neumann.

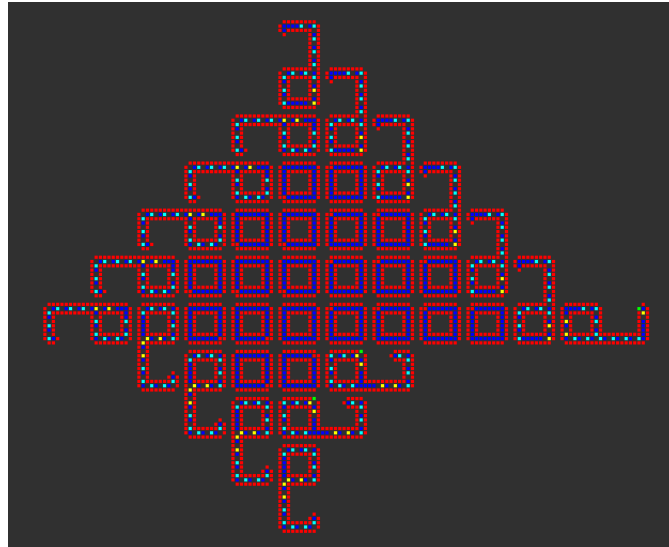


Abbildung 3.5: Ein Langton-Loop-Muster mit aktiven “lebenden” Schleifen außen und inaktiven, “toten” Schleifen innen. Bildquelle: Wikimedia Commons.

Nur wenige Jahre zuvor, 1968, hatte der britische Mathematiker und Informatiker Edgar F. Codd bereits eine Alternative²⁸ zu von Neumanns Modell mit nur 8 Zuständen und einer 5er-Nachbarschaft, allerdings in einem zweidimensionalen System, nachgebaut. Dieser CA-UC war aber immer noch sehr rechenintensiv, da er mehrere Zehntausend Zellen zur Reproduktion benötigte. Der Durchbruch auf diesem Gebiet gelang schlussendlich einem Kollegen von Codd, Christopher Langton, einige Jahre später: 1984 präsentierte²⁹ er der Öffentlichkeit seine so genannten Langton-Schleifen, einen zweidimensionalen CA, der ebenfalls mit 8 Zuständen und einer 5er-Nachbarschaft konzipiert war, aber nur noch wenige hundert (!) Zellen zur Selbstreproduktion benötigte. Langtons Vorgehen erinnert dabei an die Geschichte vom Ei des Kolumbus: In seiner Arbeit argumentierte er, dass es keine UTM brauche, um einen CA-UC zu konstruieren. Auch konnte Langton dann seinen Ansatz bereits visualisieren, und das Ergebnis war beeindruckend: 8 Zustände, wenige hundert Regeln und Zellen sowie eine 5er-Nachbarschaft reichten aus, um eine Schleife (siehe Abbildung 3.4) zu beschreiben, die ohne zentrale Steuerung, nur durch Interaktion der simplen Zellen, sich ständig selbst reproduzierte und dabei ein komplexes Muster hervorbrachte.

²⁸Siehe Edgar F. Codd, “Cellular Automata”, Academic Press, New York 1968.

²⁹Siehe Christopher G. Langton, “Self-reproduction in cellular automata”, Physica D 10, 1984, Seite 135-144.

Die Visualisierung (siehe Abbildung 3.5) erinnert stark an lebende Zellen und der Titel von Langtons Arbeit “Self-reproduction in cellular automata” ist wohl ebenfalls bewusst an biologische Begriffe angelehnt. Tatsächlich beschreibt dieses System bereits einen sehr einfachen Fall von Selbstorganisation, ohne dass dieses Konzept zum damaligen Zeitpunkt in der Biologie bereits eingehender untersucht gewesen wäre. Da sich aber schon unbeabsichtigt derartige Resultate einstellen können und CAs offensichtlich für komplexe Systemsimulationen daher gut geeignet sind, soll das Konzept der Zellulären Automaten in das CGST-Modell zur Abbildung der Systemdynamik integriert werden. Dabei definieren wir, vorerst noch natürlichsprachlich, im nächsten Kapitel dann formal anhand von Übersetzungen der ontologischen Konzepte in Automaten-Syntax, die folgenden Abbildungen unserer Systemdefinition in ein automaten-theoretisches Modell:

- Ein System kann sich durch Zustandsübergänge verändern, d.h. all jene Ausprägungen, die sich auf der Ebene der Instanzierung³⁰ befinden, können durch Regeln mit jedem Zustandsübergang abgeändert werden.
- Elementinstanzen übernehmen dabei die Rolle von Agenten in einem Zellulären Automaten.
- Attribute werden im Zellulären Automaten analog als Eigenschaften der jeweiligen Agenten übernommen.
- Dynamische Regeln für Zustandsübergänge müssen entweder direkt in der Sprache des ZA neu formuliert werden oder können aus einer proprietären oder entsprechend standardisierten Regelsprache³¹ abgeleitet werden.

3.6 System und Umwelt

Niklas Luhmann gründet auf der Differenzierung zwischen System und Umwelt seine gesamte Theorie sozialer Systeme. George Spencer-Brown entwirft aus dieser Differenz heraus ein ganzes Logiksystem. Wir wollen die Beziehung zwischen einem System und seiner Umwelt nehmen, als das, was es ist: Ein Teilaspekt systemischer Perspektive. Auch Hofkirchner³² beschreibt diese Beziehung als Teil der Gesamtbetrachtung:

“Systeme [...] können nämlich in drei verschiedenen Hinsichten betrachtet werden: 1. hinsichtlich der Elemente, die als kleinste Teile gelten, die für das System noch relevant sind [...], 2. hinsichtlich des

³⁰Spontane Änderungen auf der Ebene der Systembeschreibung selbst werden im Anhang behandelt.

³¹Dieser Anwendungsfall wird aufgrund fehlender Standards in diesem Bereich in der Arbeit nicht näher behandelt.

³²Siehe [23], Seite 159.

Zustands, den das System einnimmt und 3. hinsichtlich des Verhaltens, welches das System gegenüber seiner Umwelt (im Netz von Kosystemen oder als Bestandteil eines übergeordneten Systems) an den Tag legt.”

Luhmanns Ansatz, seine Systemtheorie auf der Differenzierung zwischen System, also jenen Teilen, die zum Wesen des Betrachteten gehören und Umwelt, also allen anderen Teilen, ist grundsätzlich nicht so schlecht: Auch oder insbesondere der Logikentwurf von George Spencer-Brown zeigt, dass man damit auch formal sehr weit voranschreiten kann. Man kann sagen “Eine Menge von Elementen Γ und Beziehungen Θ bildet ein System S und die Differenzmenge von S jeweils die Umwelt des Systems” – für das System gelten dann vereinfachte Analyseregeln und die Umwelt wird aus der Kernbeobachtung ausgeklammert. Einflüsse aus der Umwelt, die trotz allem für die Systemanalyse notwendig erscheinen, werden in Form von sogenannten Irritationen in das Konzept der Systemtheorie aufgenommen. Beziehungen zwischen Systemen – zwei unterschiedliche Systeme sind dabei im Regelfall jeweils nur Teil der Umwelt des anderen Systems – können außerdem noch in der Form struktureller Kopplungen bestehen, eine Art höher qualifiziertes Irritationsprotokoll, für dessen Inhalte die beteiligten Systeme, zwischen denen die strukturelle Kopplung besteht, sensibler sind als für herkömmliche Irritationen.

Das Konzept der Irritation, aber vor allem jenes der strukturellen Kopplung ist ein technisch gesehen inkonsistenter Versuch die – in der Realität ja zweifelsohne bestehenden – Beziehungen zwischen Systemen in die Luhmannsche Theorie zu integrieren, obwohl diese ja genau solche Beziehungen eigentlich ausschließt. Aus formaler wie auch aus technischer Perspektive ist dieser Punkt zugleich auch die größte Schwäche der Luhmannschen Systemtheorie, die vielleicht in sozialen Systemen noch teilweise ausgeblendet werden kann, für eine Allgemeine Systemtheorie aber unüberwindbare Probleme in der formalen Abbildung erzeugt.

Es muss eine geregelte und standardisierte Möglichkeit geben, die Kommunikation und Interaktion zwischen unterschiedlichen Systemen bzw. zwischen Systemen und ihren Umwelten abzubilden. Einfach veranschaulichen lässt sich der für das CGST-Modell gewählte Ansatz dafür mit bestehenden Computersystemen: Jeder Rechner hat eine Hardware-Software-Einheit, die über bestimmte Kommunikationsprotokolle, bzw. einen systeminternen Code, die Abläufe in einer Einheit regeln. Die dadurch aufgebaute Kommunikation – etwa einzelne Registerbefehle – sind für außerhalb des Systems stehende Akteure unbrauchbar – d.h. ein Mensch, aber auch ein durch ein Netzwerk verbundener Drucker könnte diese Befehle vielleicht physisch wahrnehmen, versteht aber die dahinter liegende Semantik nicht. Trotzdem ist eine Kommunikation auch mit anderen Systemen möglich: Über wohl definierte Schnittstellen zwischen System und Umwelt – etwa Bildschirm und Tastatur oder aber einen Druckertreiber –

können auch Instanzen unterschiedlicher Systeme miteinander “sinnvoll” interagieren. Das kann wie im Falle der Mensch-Maschine-Kommunikation bewusst so von einem Akteur (dem Menschen) programmiert worden sein; das kann aber auch, wie z.B. zwischen einem Reiter und seinem Pferd oder zwei Kulturen, die das erste Mal aufeinander treffen durch Lernprozesse und eine Trial-Error-Historie Schritt für Schritt entstehen. Soweit einmal zur abstrakten Interpretation derartiger Verhältnisse.

Formal besteht auf der Syntaxebene des CGST-Modells kein Unterschied zu einer abgeschlossenen Systemdarstellung, weil auch eine Beziehung zwischen Systemen als Teil des Wissensgraphen repräsentiert wird. Nur wird in diesem Fall eine spezielle Relation verwendet: *interactsWith*(S_1, S_2) beschreibt generisch eine Beziehung zwischen zwei Systemen. Wie diese Beziehung dann im Detail aussieht, kann ontologisch sehr unterschiedlich implementiert werden: Theoretisch ist von einer einfachen Relation mit einer simplen Semantik bis hin zu komplexen Klassen-Relationen-Verbunden alles denkbar und soll für die Modellierung auch nicht beschränkt sein. Praktisch läuft der Austausch zwischen Systemen im CGST-Modell auf Objekt-Input- und Output-Methoden hinaus, wie sie in der objektorientierten Programmierung üblich sind. D.h. jedes System kann spezifizieren, für welche Inputs aus der Umwelt es “empfänglich” ist und welche Reize es an die Umwelt (bzw. an andere Systeme) abgeben kann. Da die Verarbeitung dieser Schnittstellen dann über Protokolle in den Systemen klar geregelt werden kann, wird hier die Problematik mit den unscharfen Konzepten der Luhmannschen Irritationen und strukturellen Kopplungen überwunden. Ein Beispiel für eine formale Ausprägung des Konzepts in CGST werden wir in Kapitel 5 präsentieren.

3.7 “Avoiding such unnecessary Duplication of Labor”

Neben der standardisierten Abbildung und Simulation von Systemen basierend auf einer allgemeinen Systembeschreibung sind diverse Kalkulationen, um “unnötige doppelte Arbeit” in den Wissenschaften zu vermeiden, das Hauptziel einer Allgemeinen Systemtheorie. Insbesondere wollen wir feststellen können, ob offene Fragen und Probleme in einem analogen System in einem anderen Bereich bereits gelöst wurden und dann die entsprechende Lösung als Hypothese ausprobieren oder aber neue Erkenntnisse, nach denen vielleicht gar nicht explizit gesucht wurde, anhand von Erkenntnissen in anderen Systeminstanzen ableiten. Die Differenzierung der dazu notwendigen Algorithmik nehmen wir nachfolgend anhand der unterschiedlichen Ebenen in unserem CGST-Modell vor.

3.7.1 Ähnliche Systeme

Die einfachste Form des Vergleichs in unserem Modell mit der Absicht, eine Ähnlichkeit festzustellen, ergibt sich auf der Ebene der ontologischen, statischen Beschreibung eines abstrakten Systemmodells. Ohne also auf spezielle Ausprägungen oder gar die dynamische Entwicklung zu sehen, wollen wir nur anhand der Beschaffenheit der Graphen-Repräsentation des abstrakten ontologischen Modells einfache Ähnlichkeiten feststellen. Der erste Schritt dabei ergibt sich logisch aus einem Blick in die Graphentheorie: Ein echter Isomorphismus, d.h. eine mögliche 1:1-Abbildung zweier Ontologien, die sich nur aufgrund der begrifflichen Ausprägungen ihrer Konzepte und Relationen unterscheiden. Das kann natürlich in sehr einfachen Fällen sehr schnell zu schlechten, weil völlig falschen, Ergebnissen führen.

Ab einer gewissen Anzahl von Elementen und Relationen wird die Wahrscheinlichkeit, einen Treffer zu erzielen, aber immer höher – vor allem dann, wenn mathematische Eigenschaften für die Objektrelationen Teil der zu vergleichenden Ontologien sind.

Wir können uns auch an ein Set von Elementen und Relationen annähern, ab dem wir die Empfehlung abgeben, einen solchen Vergleich zu versuchen. Bei zwei verschiedenen Elementen und einer Objektrelation gibt es vier verschiedene mögliche Ausprägungen $o(A, A)$, $o(B, B)$, $o(A, B)$ und $o(B, A)$.

3.7.2 Ähnliche Systeminstanzen

Nun kann es es sein, dass Ähnlichkeiten auf statischer Ebene zwischen Systemen bestehen, ohne dass eine signifikante Ähnlichkeit im abstrakten Systemmodell feststellbar ist. Das kann vor allem dann der Fall sein, wenn Systeminstanzen einzelner Modelle sehr heterogen sind und die zugrunde liegenden Modelle aus nur sehr wenigen Elementen und Relationen bestehen.

Ein Beispiel dafür wäre die Instanz eines sozialen Netzwerks, die nur aus der Relation $kennt(Person1, Person2)$ besteht. Nur anhand der statischen Systeminstanz und noch ohne auf deren Entwicklung einzugehen, lassen sich in der Netzwerktheorie³³ hier Konzepte wie jenes des Gatekeepers beschreiben, denen bestimmte zusätzliche systemische Eigenschaften zugewiesen werden können. Dieses Wissenschaftsfeld ist insbesondere auch deshalb interessant, weil sich in den letzten Jahren sogenannte Social-Networking-Plattformen stark im Internet etabliert haben und dabei in erster Linie auf dieser einen Relation basieren. Auch im Semantic Web ist das Friend-of-a-Friend-Projekt gemessen an der Verteilung und Anzahl der im Web befindlichen Instanzen das erfolgreichste Szenario für eine Semantic-Web-Anwendung. Eine grafische Darstellung eines

³³Siehe www.fas.at

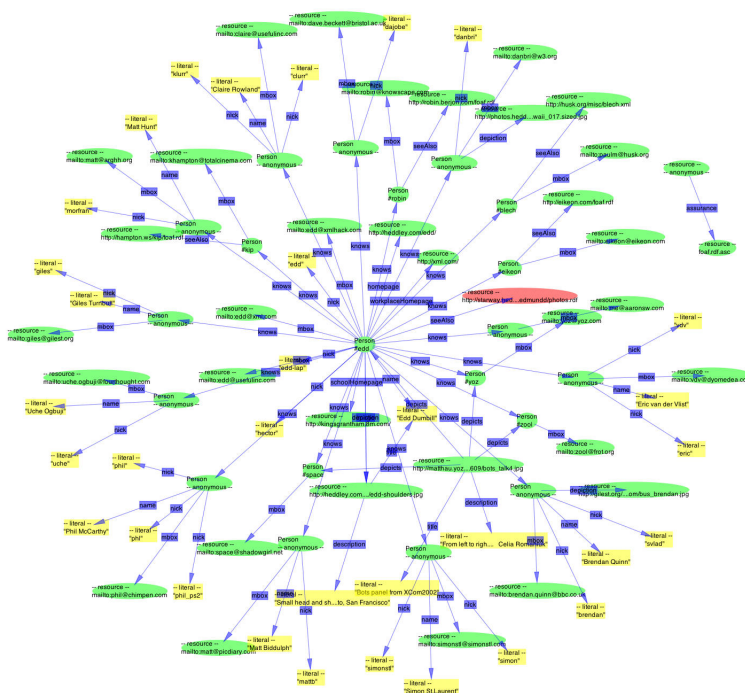


Abbildung 3.6: Auszug aus einer grafischen Repräsentation eines FOAF-Netzwerks.

solchen Netzes ist in Abbildung 3.6 dargestellt. Zahlreiche Arbeiten zum Themengebiet Soziale Netzwerke im Allgemeinen und zu FOAF im Speziellen zeigen, dass schon eine einzelne Relation in einem System ausreichen kann, um eine Vielzahl von Mustern, Verhaltensweisen und Systemanalysen durchführen zu können.

Ein bekanntes Beispiel aus der Literatur ist dabei die oben erwähnte Rolle des “Gatekeepers”, etwa in einem Manager-Netzwerk: Ein Gatekeeper ist eine Person, die aufgrund der Tatsache, dass sie als (Kommunikations-)Schnittstelle, quasi als Flaschenhals, zwischen zwei Teilen eines Netzwerks agiert, maßgeblichen Einfluss auf Kommunikations- und Entscheidungsprozesse ausübt. Der plötzliche Ausfall eines Gatekeepers zieht dabei in der Regel signifikante Reorganisationen des Netzwerks nach sich. Die Kalkulation von Gatekeeper-Elementen ist deshalb ein wichtiges Problem in der Netzwerktheorie.

Ein ähnliches Szenario erhalten wir bei der Betrachtung von Zugverbindungen. Obgleich hier wichtige Verbindungsknoten keinen aktiven Einfluss auf Entscheidungen ausüben, so führt doch auch hier der Ausfall eines solchen Knotens in der Regel zu einer Neuorganisation des Netzwerks. Beobachten lassen

sich derartige Entwicklungen anhand der Modifikation von Zugfahrplänen durch das Öffnen oder Schließen politischer Grenzen und den dadurch gesteuerten Warenverkehr, aber auch durch technisch bedingte kurzfristige Ausfälle wichtiger Bahnhöfe.

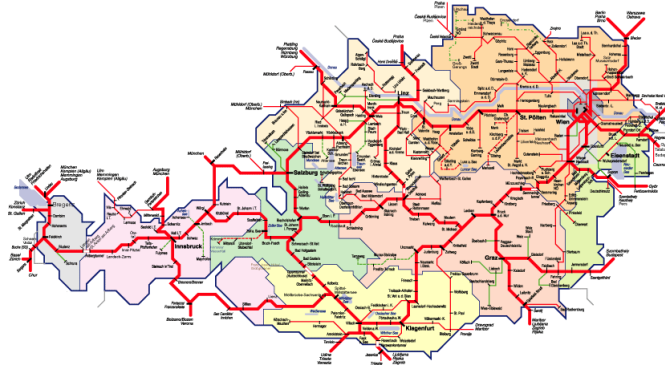


Abbildung 3.7: Streckenübersicht der Österreichischen Bundesbahnen inklusive Störungen. Quelle: <http://www.oebb.at>

Die Kalkulation von derartigen wichtigen Knoten in einem Netzwerk ist umgelegt auf das CGST-Modell ein Problem, das explizit auf der Ebene der statischen Systeminstanzen besteht. Zu jedem beliebigen Zeitpunkt kann die aktuelle Konfiguration einer Instanz analysiert werden und mit Hilfe eines entsprechenden Algorithmus eine Liste von Gatekeeper-Elementen ausgegeben werden. Optimierungen in derartigen Berechnungen oder umfassende Analyse dieser und ähnlicher Rollen in solchen Netzwerken können aber – zumindest in einer experimentellen Phase – mit geringen Modifikationen auf andere Bereiche angewandt werden.

3.7.3 Ähnliches Systemverhalten

Was soll hier unter dem Begriff Systemverhalten verstanden werden? Bisher haben sich die Betrachtungen zur Ähnlichkeit auf statische Charakteristika unterschiedlicher Systeme oder ihrer Instanzen beschränkt. Interessant sind aber natürlich auch Ähnlichkeiten was die Abfolge der Systemzustände und die hier definierbaren Muster betrifft. Eine Abfolge von Zuständen, die sich mitunter sehr oft in Systemprozessen beobachten lässt, ist der Tod eines Systems, d.h. es gibt einen Zeitpunkt t_d in der Abfolge der Systemzustände, ab dem es zu keinen weiteren Änderungen mehr in der Konfiguration des Systems kommt, egal wie viele Folgezustände generiert werden. Das hört sich zuerst einmal trivial an, kann aber, wenn man den Zustandsraum vor t_d analysiert wichtige Erkenntnisse über Systemverhalten im Allgemeinen geben und zu wertvollen Informationen führen – insbesondere ist die Frage, wann ein System einen kritischen Zustand erreicht, der einen Zusammenbruch wahrscheinlich werden lässt, von großer Bedeutung in beinahe allen Wissenschaftsfeldern.

Weitere Beispiele für generische Charakteristika von Systemverhalten wären Aufspaltung, Synthese, Autopoiesis usw. Bevor aber Beispiele formal betrachtet werden können, muss zuerst definiert werden, wie Ähnlichkeit über ein Set von zwei unterschiedlichen Zustandsmengen von durchaus auch sehr unterschiedlichen Systemen definiert werden kann. Im Prinzip geht es ja auch hier um eine Art von Mustererkennung: Wenn wir das Verhalten des Schleimpilzes *Dictyostelium discoideum* beobachten, der sich bei Nahrungsknappheit mit anderen Individuen zu einer Art Wurm zusammenschließt, der sich dann so lange in dieser Formation fortbewegt, bis eine Nahrungsquelle gefunden wird und dann mit der Entstehung und dem Verhalten etwa einer aufgebrachten Menschenmenge vergleichen, so lassen sich oberflächlich gewisse Gemeinsamkeiten feststellen, die auf den ersten Blick nur wenig Gemeinsamkeit auf der strukturellen Ebene besitzen – trotzdem erkennt der menschliche Beobachter ähnliches Verhalten. Ein Algorithmus würde in diesem speziellen Fall etwa dann eine Ähnlichkeit feststellen, wenn er das (rasche) Entstehen und (rasche) Verschwinden einer ganz bestimmten Element-Relation in den jeweiligen Systemen (*chemicalConnectionTo* etwa als Abbildung einer chemischen Verbindung zwischen zwei Molekülen bzw. *isNeighbour* als Abbildung der lokalen Nähe zweier Personen) beobachtet und klassifiziert. Der selbe Algorithmus würde dann wahrscheinlich auch beim Verhalten von Störchen bei der Nestsuche und der Auswahl von Lokalen in der Nachtszene einer Stadt Analogien feststellen können, da auch hier ein bestimmtes gemeinsames Muster der zeitlich beschränkten Element-Relationen entsteht. Untersuchungen zu solchen Parallelen treten in der Wissenschaft immer wieder auf, eine der jüngsten Studien, die als Beispiel für domänenübergreifende Analogiebildung in dynamischen Prozessen genannt werden kann, ist eine Arbeit³⁴ des tschechischen Physikers Petr Seba zur mathematischen Modellierung von Parklücken in Autokolonnen, die aber auch für die Simulation von Abstandsmustern bei Vögeln auf Stromleitungen herangezogen werden kann. Im Detail hat sich Seba dabei die Musterbildung der dabei entstehenden Lücken auch in der Praxis in beiden Fällen angesehen und dokumentiert und war so auf die entsprechende Analogie gestoßen, für deren Erklärung Parallelen im Wahrnehmungssystem von Vögeln und Menschen herangezogen werden (siehe Abbildung 3.8).

Ähnliches Systemverhalten wurde von den Vertretern des Bertalanffy Programms ganz besonders hervorgehoben und mit Beispielen untermauert. Bertalanffy selbst versucht eine Aufzählung dynamischer Systemverhaltensweisen³⁵ zu geben, ohne jedoch eine strukturelle Kategorisierung ebendieser einzuführen. Begriffe, die hier aber immer wieder verwendet werden, sind: Homöostase als Definition eines selbstregulierenden Systemverhaltens; zielwertgesteuertes Systemverhalten, ausgerichtet auf die Erreichung eines definierten oder immanenten Endzustandes (der nicht unbedingt erreicht werden muss); Zerfall des Systems als Sonderfall des zuvor genannten Verhaltens sowie weitere diverse Sonderformen dynamischer Systementwicklungen wie Zentralisierung, Verteilung, Ordnungsbildung, etc. Für die Analogiebildung stützt sich Bertalanffy

³⁴Siehe <http://arxiv.org/abs/0907.1914>.

³⁵Siehe [3], Seite 77 ff.

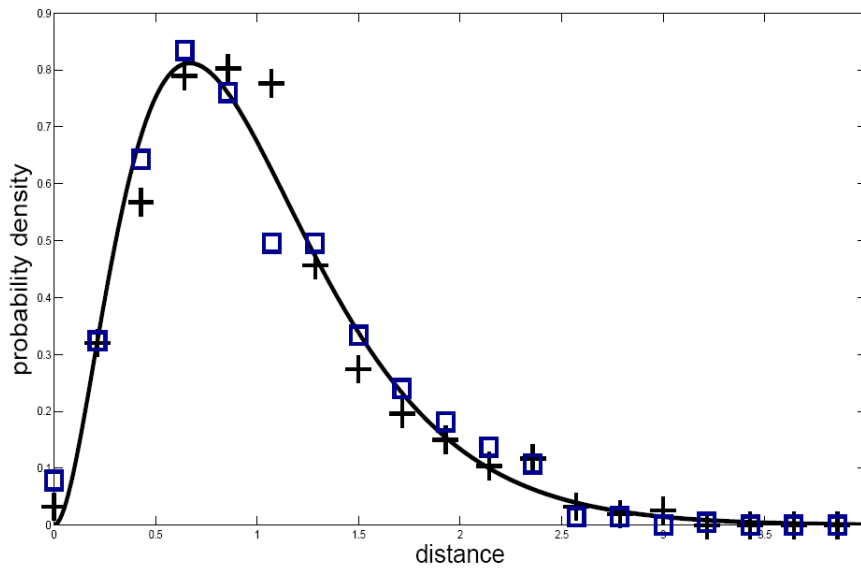


Abbildung 3.8: Grafische Darstellung der Analogie bei der Lückenbildung von geparkten Autos (Kreuze) und Vögeln auf einer Stromleitung (Quadrate). Die Wahrscheinlichkeitsverteilung für die Größe der zu bildenden Lücke ist nach einer Normalisierung des Abstandes im Modell wie im Experiment sehr ähnlich. Quelle: <http://arxiv.org/abs/0907.1914>.

dann eben auf die Verwendung mathematischer Systembeschreibungen. Die exponentielle Entwicklung eines Systemparameters etwa ist prinzipiell ein Differenzierungsmerkmal, das unter Umständen als analog zu einer ebensolchen in einem anderen System betrachtet werden kann. Je spezieller die mathematische Beschreibung der Entwicklung eines Systemparameters, desto höher die Wahrscheinlichkeit, dass eine durch den mathematischen Vergleich vermutete Analogie auch vom wissenschaftlichen Beobachter als solche ausgelegt werden kann.

Was bedeutet das für die praktische Anwendung in einer formalen Repräsentation der Allgemeinen Systemtheorie? Ein erster Ansatz wäre die Entwicklung einer Funktionsbibliothek, die basierend auf einer Kategorisierung, die mit den wesentlichen Begriffen der Allgemeinen Systemtheorie übereinstimmt, eine Grundlage für Vergleiche von Systemparameter-Entwicklungen unterschiedlicher Systeme bildet. Ein Analogie-Paar $AP(s1_x, s2_y)$ wäre dann eine darauf basierende, kalkulierte Gemeinsamkeit der beiden Systeme $S1$ und $S2$ hinsichtlich ihrer beiden dynamischen Parameter $s1_x$ und $s2_y$. Die Relation der Analogie-Paare zur Anzahl der Parameter in den beiden Systemen könnte dann als automatisiert generierte Aussage über die Ähnlichkeit zweier Systeme hinsichtlich deren dynamischen Verhaltens herangezogen werden. Gibt es bei einer kleinen Anzahl von Systemparametern trotzdem sehr viele Analogie-Paare, würde ein entsprechender Algorithmus eine Analogie feststellen.

Da gerade in komplexen Systemen mit vielen verschachtelten Abhängigkeiten aber ein solcher Katalog schnell ausgeschöpft wäre, müsste ein generischer Ansatz weiter gehen. Insbesondere müsste der Möglichkeitsraum für die formale Repräsentation dynamischen Verhaltens ebenfalls dynamisch und konstruktiv berechnet werden, anstatt durch eine Aufzählung bestehender Entwicklungen abgedeckt zu werden. Dazu können etwa Konstruktionsalgorithmen wie sie auch zur Berechnung von Interpolationskurven herangezogen werden, eingesetzt werden, um aus einer Wertekette einer Systemsimulation eine entsprechende mathematische Funktion zu kalkulieren. Da der Programmcode selbst ja die Simulation determiniert, wären auch Verfahren denkbar, die – technisch etwa in Form eines speziellen Compilers – eine Ableitung mathematischer Funktionen aus den Fragmenten der zu den Werten zugehörigen Programmcode-Teilen sowie deren Abhängigkeiten berechnen.

Komplexere Szenarien, die auch im Blickpunkt des Bertalanffy-Programms standen, wie etwa Räuber-Beute-Systeme, bedürfen einer ausgereifteren Algorithmetik, um nicht voreilig Fehlschlüsse zu produzieren. Hier genügt es eben nicht mehr, die Entwicklung der Element-Relationen zu beobachten, weil auch die Entwicklung von Attributwerten der einzelnen Elemente über Status-Abfolgen eine große Rolle spielt. Im Prinzip lassen sich also an dieser Stelle die ursprünglichen Ansätze aus dem Bertalanffy-Programm sehr gut integrieren, obwohl ja für die Wissensrepräsentation und die formale Abbildung ein völlig anderer Ansatz gewählt wurde.

Kapitel 4

Computational General System Theory (CGST)

*“Denn ohne Klarheit in der Sprache
ist der Mensch nur ein Gartenzwerg.”
Element of Crime, “Alle vier Minuten”*

Eine Allgemeine Systemtheorie braucht ein formales Modell, auf das sie sich abbilden lässt und auf dessen Basis praktische Anwendungen entwickelt werden können. Dieses Modell muss es erlauben, Systeme der unterschiedlichen Schichten formal zu beschreiben, Vergleiche mit anderen Systemen darzustellen, neues Wissen abzuleiten und Berechnungen – auch in Form von Simulationen und Prognosen – durchzuführen. Klassische mathematische Ansätze aus dem Bertalanffy-Programm sind wie wir gesehen haben an der fehlenden Möglichkeit der praktischen Umsetzung gescheitert. Auch aktuelle Methoden sind zwar formal vielversprechend, in der Praxis jedoch aufgrund ihres Orchideendaseins praktisch nicht sehr bedeutend.

Gerade für eine Allgemeine Systemtheorie, die ja das Ziel verfolgt, die Erkenntnisse der einzelnen Wissenschaftsdisziplinen stärker zusammenzuführen und das interdisziplinäre Arbeiten zu unterstützen, ist es wichtig, Ansätze zu wählen, die sich bereits allgemein etabliert haben und eine ausreichende Basis bieten können und nicht den Fehler zu begehen, aus dem Nichts heraus ein funktionierendes Modell zu erfinden. Die letzten zwanzig Jahre in den Computerwissenschaften haben erstaunliche Errungenschaften auf diesem Gebiet hervorgebracht, die wir uns bei unserem Vorhaben zu Nutze machen werden.

Wir werden das erfolgreiche Konzept des Ontology-Designs aus dem Semantic Web als statische Basis unseres Modells nutzen und es im Wesentlichen um Konzepte Zellulärer Automaten erweitern. Auf beiden Ebenen, der statischen

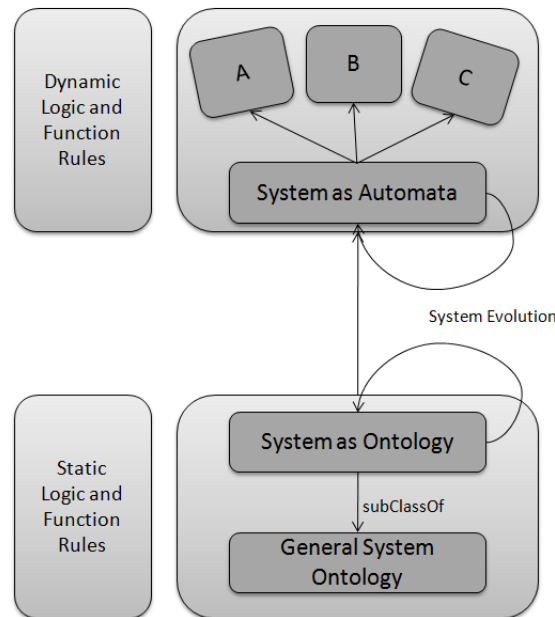


Abbildung 4.1: Die Systemarchitektur der lose gekoppelten CGST-Anwendung, technisch basierend auf Semantic-Web-Technologie und Zellulären Automaten

wie der dynamischen, werden wir bestehende Regelsprachen integrieren, um aus bestehenden Fakten mit Hilfe von systemimmanenten oder aber abgeleiteten Regeln neues Wissen für die Instanzen unseres Modells erzeugen zu können.

Dieser Ansatz hat zwei große Vorteile: Wir ersparen uns viele in den Anfangsstadien derartiger Prozesse übliche Kinderkrankheiten und wir können zu Testzwecken Daten aus bestehenden Anwendungen verwenden. Zur Integration dynamischer Prozesse in das Modell sollen anschließend Zelluläre Automaten verwendet werden, deren Grundgerüste aus den Ontologien abgeleitet und um dynamische Konfigurationen erweitert werden. Da die Implementierung einer eigenständigen Anwendung im Rahmen dieser Arbeit nicht möglich ist, die praktischen Anwendungsmöglichkeiten aber trotzdem veranschaulicht werden sollen, erfolgt die Umsetzung des Modells und der Beispiele auf Basis einer lose gekoppelten Systemarchitektur, wie sie in Abbildung 4.1 veranschaulicht ist.

4.1 Ontologische Systembeschreibung

Die Idee, die Welt formal zu beschreiben, ist so alt wie die Sprache selbst. Umfassende, zusammenhängende begriffliche Beschreibungen finden sich schon bei Aristoteles und finden bei Kant und den Kollegen seiner Zeit einen beeindruckenden Höhepunkt. Aber erst der Siegeszug der Höheren Mathematik und

der Mathematischen Logik motivieren die Wissenschaft zu formal strengeren Versuchen, die Welt in einer abgeschlossenen Menge an Aussagen zu definieren. Ludwig Wittgensteins *Tractatus logico-philosophicus* ist dabei noch ein stark linguistisch orientierter Versuch, während das Hilbert-Programm die mathematische Version vom Versuch, die Welt abgeschlossen zu definieren, darstellt. Auf diesen Punkt werden wir später noch etwas näher eingehen, wenn wir uns die Grenzen der Formalisierbarkeit einer Allgemeinen Systemtheorie, die durch die Grenzen, die uns die Mathematische Logik und für sie dereinst Kurt Gödel gesetzt haben, betrachten.

Mit dem Siegeszug des Computers und der Informatik und den immer komplexeren Anwendungen, die realisiert werden, steigt auch das Bedürfnis, abstrakte Modellierungsmethoden zur Verfügung zu stellen, stetig an. Meilensteine sind hier etwa die Entwicklung von objektorientierten Programmiersprachen und relationalen Datenbankenmodellen. Doch bleiben all diese Entwicklungen der Forderung nach einer allgemein modellierbaren Informatik, die nicht auf die Handhabung durch ebensolche Experten beschränkt ist, das letzte Stück hin zum kontextbeschriebenen Computerprogramm schuldig. Und eben auf dem Gebiet der Computersprachen erleben wir derzeit einen wichtigen Paradigmenwechsel: Das Semantic Web fordert von seinen Teilnehmern nicht nur die Bereitstellung von Information, sondern auch die Strukturierung ebendieser in einer Form, in der sie auch von Maschinen interpretierbar ist. Dazu werden mit Hilfe sogenannter Ontologiesprachen wie der Web Ontology Language OWL¹ des World Wide Web Consortiums Werkzeuge möglich, die es auch Nicht-Informatikern sehr einfach erlauben, Wissen in Form von strukturellen Zusammenhängen von Objekten maschineninterpretierbar abzubilden.

Die Semantic-Web-Sprachen RDF (Ressource Description Framework) und OWL sind dabei heute bereits weit verbreitet und kommen in unterschiedlichsten Domänen zur Anwendung: In der Industrie zur Klassifikation von Bauteilen², in der Wirtschaft, etwa zur Definition von Produktbeschreibungen und Angeboten³, im so genannten "Social Web" zur Beschreibung von Personendaten und Beziehungen⁴ oder im akademischen Bereich zur Repräsentation von Publikationsdatenbanken und Abbildung von Biodaten. Ursprünglich fast ausschließlich im akademischen Umfeld angesiedelt hat sich das Semantic Web dabei vor allem in den letzten drei bis fünf Jahren auch immer stärker als Basistechnologie für kommerzielle Anwendungen etabliert.

Als Basis für alle weiteren Modellierungsschritte wird nun eine sehr einfache Systemontologie definiert und anschließend beschrieben, wie konkrete Systeme aus unterschiedlichen Wissenschaftsdisziplinen an diese Systemontologie

¹<http://www.w3.org/2004/OWL/>

²Siehe <http://www.heppnetz.de/projects/eclassowl/>.

³Siehe <http://www.heppnetz.de/projects/goodrelations/>.

⁴Siehe <http://www.foaf-project.org/>.

anküpfen müssen, um schlussendlich formal vergleichbar und damit von Anwendungen automatisiert verarbeitbar werden.

4.1.1 Eine System-Ontologie basierend auf der Web Ontology Language

Wir benötigen also zunächst ein Konzept für das System selbst. Dem systemischen Paradigma folgend könnte hierzu eigentlich auch die OWL-Metaklasse `owl:Thing` erhalten. Da die Intention zur Einführung dieser Klasse aber eine grundsätzlich andere war⁵, führen wir als Hauptkonzept in unserer Systemontologie eine Klasse `system:System` ein. Ein spezielles System soll dann in der Folge immer `rdf:subClassOf` unserer Klasse `system:System` sein, um entsprechend referenziert werden zu können.

Einzelne Systeme können Teilsysteme übergeordneter Systeme sein. Diese Beziehung wird durch die `system:hasSubSystem` Objektrelation beschrieben.

```
<owl:Class rdf:about="#System"/>

<owl:Class rdf:about="#SubSystem">
  <rdfs:subClassOf rdf:resource="#System"/>
</owl:Class>

<owl:ObjectProperty rdf:about="#hasSubSystem">
  <rdfs:range rdf:resource="#SubSystem"/>
  <rdfs:domain rdf:resource="#System"/>
</owl:ObjectProperty>
```

Eine Teilsystem-Relation ist notwendig, um auch innerhalb großer komplexer Systembeschreibungen Unterteilungen formal beschreiben zu können. Beispiele für die Anwendung dieses Konzepts wären etwa die Unterteilung in bestimmte ausdifferenzierte Systeme in einem Sozialen System, wie Wirtschaftssystem, Politisches System, etc. oder aber die Beschreibung einzelner Organe als Subsysteme im menschlichen Organismus. Die Relation `hasSubSystem` unterliegt keinen weiteren Einschränkungen, da prinzipiell Teilsysteme mehreren übergeordneten Systemen zugeordnet sein können. Wichtig ist außerdem die `subClassOf System` Beziehung, da es sich bei Teilsystemen formal ebenfalls um Systeme im Sinne der CGST-Ontologie handeln muss.

Elemente werden über die Klasse `system:Element` repräsentiert. Die in Kapitel 3.2 definierte Beziehung zum System wird über ein OWL-Objectproperty `system:hasElement` mit dem Systemkonzept als Domain und dem Elementkonzept als Range syntaktisch aufgelöst. Beziehungen zwischen Elementen können in speziellen Systemen über Subproperties des Objectproperties `elementRelation`

⁵<http://www.w3.org/TR/2004/REC-owl-guide-20040210/#DefiningSimpleClasses>

definiert werden, das die syntaktische Auflösung der Element-Relation aus Kapitel 3.2 in OWL realisiert.

```
<owl:Class rdf:about="#Element"/>

<owl:ObjectProperty rdf:about="#hasElement">
  <rdfs:range rdf:resource="#Element"/>
  <rdfs:domain rdf:resource="#System"/>
</owl:ObjectProperty>

<rdf:Description rdf:about="#elementRelation">
  <rdfs:range rdf:resource="#Element"/>
  <rdfs:domain rdf:resource="#Element"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:Description>
```

Element-Relationen wären im Beispiel des Sozialen Systems etwa Beziehungen zwischen Personen wie Verwandtschaftsverhältnisse, Kommunikationsbeziehungen, etc. und in einem Ökosystem etwa Nahrungsketten-Beziehungen, Stoffkreislauf-Element-Beziehungen, etc. Prinzipiell sind auch Sub-Property-Definitionen der `system:hasElement`-Relation selbst denkbar – etwa wenn zur Ausdifferenzierung sehr spezielle Element-Zuordnungen zu Systemen notwendig werden –, diese sollen aber nur dann verwendet werden, wenn sich die Beschreibung der betreffenden Systemeigenschaft nicht durch eine andere Modellierungsform genauso auflösen lässt: Die Element-Zuordnung zu Systemen ist eine (fast) atomare Relation, die im Allgemeinen nicht mehr als die Differenzierung von Zugehörigkeit/Nicht-Zugehörigkeit von Elementen zu Systemen beschreiben soll.

Atomare Daten-Attribute wie sie in Kapitel 3.2 beschrieben sind werden als Subproperties des Datatypeproperties `system:attribute` umgesetzt. Der Datentyp kann dabei frei gewählt werden, sollte sich aber wenn möglich auf gängige Datentyp-Standards, z.B. eingeschränkt auf den XSD-Namespace, beziehen. Die Einführung eines solchen Properties ist nicht unbedingt notwendig, hat sich aber als Teil guten Modellierens von Ontologien-Standards so herausgebildet: existiert ein solches allgemeines Datenproperty im Namespace der Ontologie, können über die Zuordnung dazu später Anfragen an das System effizienter kalkuliert werden, außerdem wird der Austausch von Systeminstanzen bzw. die Suche danach im Web unterstützt.

```
<rdf:Description rdf:about="#attribute">
  <rdfs:domain rdf:resource="#Element" />
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
```

Das `system:attribute`-Property unterliegt keinerlei Einschränkungen, da es nicht nur alle Datentypen erlauben muss, sondern auch sonst keine for-

malen Restriktionen an die zugeordneten Attribute der Elemente in den einzelnen Systembeschreibungen weiter vererben darf. Durch die Deklaration als `owl:DatatypeProperty` ist fixiert, dass alle System-Attribute explizit atomarer Natur sind und keine weitere, komplexere Auflösung mehr möglich ist.

Offen sind jetzt noch systemische Eigenschaften, also jene Attribute in Systemen, die erst durch das Zusammenspiel der Elemente des Systems entstehen und daher das "mehr als die Summe der einzelnen Teile" abbilden. Beispiele dafür sind Temperatur in einem Ökosystem bzw. Körpertemperatur im menschlichen Körper, Geschlecht und Alter von Lebewesen, u.ä. Formal wäre es ausreichend, die Domäne des `system:attribute-Properties` auszuweiten und so auch diese Art von Eigenschaften für Systeme zuzulassen. Da systemische Eigenschaften aber in beinahe allen (verwendeten) theoretischen Arbeiten zu allgemeinen Systembeschreibungen eine besondere Rolle einnehmen, soll diese Rolle auch in der Ontologie über ein differenziertes Property `system:systemAttribute`, das nur Systemen selbst, nicht aber einzelnen Elementen zugeordnet werden kann, berücksichtigt werden.

```
<owl:DatatypeProperty rdf:ID="systemAttribute">
  <rdfs:domain rdf:resource="#System"/>
</owl:DatatypeProperty>
```

Auch dieses Property unterliegt keinen weiteren Einschränkungen und kann prinzipiell beliebige Ausprägungen besitzen. In einem mehrstufigen CGST-Modell, wie es später noch in Kapitel 6 skizziert wird, ist dann auch insbesondere die Abbildung systemischer Eigenschaften auf der Stufe n einer Beobachtung auf eine Element-Eigenschaft der Stufe $n+1$ von Interesse, die diese unneingeschränkte Form der Repräsentation einer systemischen Eigenschaft ebenfalls motiviert.

Bisher wurde die Abbildung der Kernkonzepte System, Element, Relation und Attribut in der OWL-Systemontologie beschrieben. Die wesentlichsten formalen Aspekte sind damit implizit bereits definiert. Eine Reihe weiterer Konstruktionen soll die Systemontologie aber noch mit den in Kapitel 3 beschriebenen Problemstellungen in Verbindung bringen und hilfreiche Annotationen hervorbringen. Zu Beginn wurde die Zuordnung zu Reductive Levels diskutiert. Diese Zuordnung findet sich in der System-Ontologie nun ebenfalls.

```
<owl:FunctionalProperty rdf:ID="belongsToReductiveLevel">
  <rdfs:domain rdf:resource="#System"/>
  <rdfs:range rdf:resource="#ReductiveLevel"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
```

Ein System kann dabei immer nur einem Reductive-Level direkt zugeordnet sein. Das Reductive-Levels-Modell aus Kapitel 3 wurde formal durch Ontolo-

gical Instances in die Systemontologie integriert. Beispielhaft sei eine solche Instanz hier vollständig – also auch mit Kommentaren und Labels – angeführt.

```
<ReductiveLevel rdf:ID="Cells">
  <name xml:lang="de">Zellen</name>
  <name xml:lang="en">Cells</name>
  <description xml:lang="en">
    Level 4 focus on the basic component of
    all higher living systems: The cell.
    Academic disciplines: Biology (cell biology),
    chemistry
  </description>
</ReductiveLevel>
```

Jedes System, das basierend auf der Systemontologie beschrieben wird, soll eine solche Zuordnung wenn möglich enthalten. Die einzelnen Systeminstanzen übernehmen diese Zuordnung dann und können auch keine neuen oder gar andersartigen mehr entwickeln oder ableiten.

Bisher ist die prinzipielle Repräsentation der Konzepte in der Systemontologie sehr eng mit den meisten erwähnten oder im Rahmen von Systemtheorien allgemein definierten Systembeschreibungen konsistent. Ein strittiger Punkt, der aber trotzdem Eingang in die Systemontologie finden soll, ist die Relation zwischen Systemen, jene ontologische Beziehung also, die Luhmann in seiner strukturellen Kopplung unterbringt und die im GST-Diskurs in verschiedenen konstruktiven Definitionen unterschiedlich behandelt wurde. Prinzipiell muss CGST als formale Systembeschreibung, die wissenschaftliches Arbeiten unterstützen soll, so ausgerichtet sein, dass Systembeziehungen implizit sind, d.h. erst durch die Beziehung von Elementen unterschiedlicher Systeme entstehen.

```
<owl:SymmetricProperty rdf:ID="interactsWith">
  <owl:inverseOf rdf:resource="#interactsWith"/>
  <rdfs:range rdf:resource="#System"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#System"/>
</owl:SymmetricProperty>
```

Ontology-Design und Automaten-Design ist aber erfahrungsgemäß weder ein Bottom-Up noch ein Top-Down Prozess, sondern entsteht in der Praxis meist durch die Kombination dieser beiden Verfahren. Dabei kann es hilfreich sein, zu Beginn einer Definition eine System-System-Beziehung explizit zu annotieren, ohne zu diesem Zeitpunkt die Ausprägung genauer zu definieren. Das soll über die `system:interactsWith`-Relation möglich sein. `system:interactsWith`-Relationen können auch über Regeln implizit aus Element-Beziehungen abgeleitet werden, ein Beispiel dafür findet sich im nächsten Abschnitt. Jedenfalls kann ein derart annotiertes Wissen aber sehr hilfreich bei der Auflösung von

Abfragen in Hinblick auf Performance und Berechnungskomplexität sein und ist daher trotz der nicht unumstrittenen Semantik Teil der System-Ontologie.

Ein weiteres Hilfskonstrukt zur expliziten Annotation ist die `hasSystemCategory`-Relation für das System-Konzept. Systeme können hiermit explizit als `ClosedSystem`, `OpenSystem` oder `FormalSystem` beschrieben werden, die Motivation ist die selbe wie bei der `interactsWith`-Relation.

```
<owl:ObjectProperty rdf:ID="hasSystemCategory">
  <rdfs:range rdf:resource="#SystemCategory"/>
  <rdfs:domain rdf:resource="#System"/>
</owl:ObjectProperty>
```

Die Kategorie `FormalSystem` ist notwendig, weil es für diesen Systemtyp keine Entsprechung im Reductive-Levels-Schema gibt. Die Differenzierung zwischen offenen und geschlossenen Systemen ist für die weitere Ableitung in Automaten bzw. für Analysen und Simulationen hilfreich und kann – bei entsprechend detailliertem Wissen über eine System-Instanz – auch aus der Systembeschreibung abgeleitet werden. Solange ein Detailwissen aber nicht hinreichend genau vorhanden ist, ist auch hier die explizite Annotation für den praktischen Prozess äußerst hilfreich.

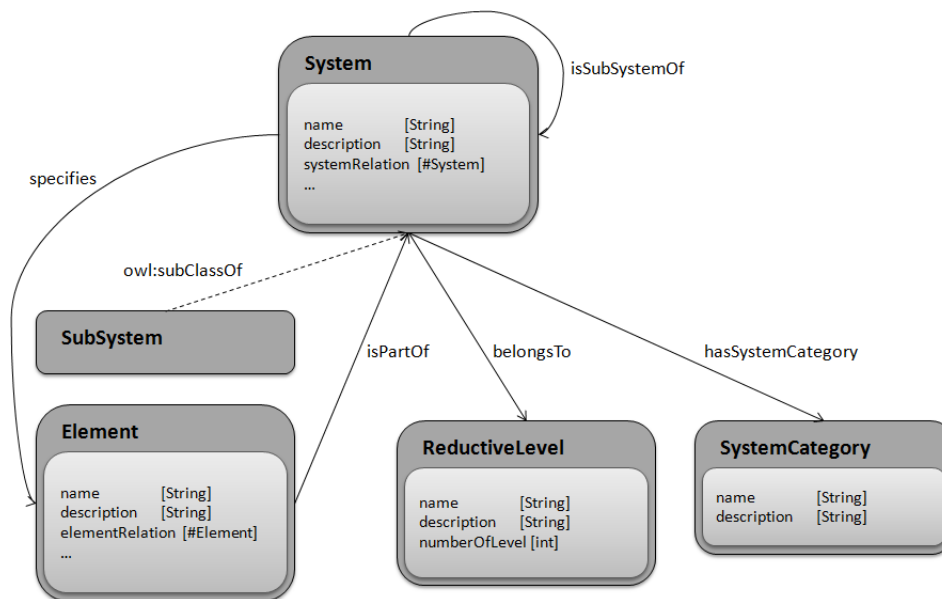


Abbildung 4.2: Konzepte und Relationen der CGST-Systemontologie mit beispielhafter Aufzählung der dazugehörigen Attribute.

Auf rein deskriptive Teile der System-Ontologie wird an dieser Stelle nicht genauer eingegangen. Im Anhang findet sich noch einmal der vollständige OWL-Code der CGST-Systemontologie, eine graphische Übersicht findet sich in Abbildung 4.2.

4.1.2 Komplexität und Ausdrucksstärke

Die Web Ontology Language entstand als notwendige Sprache über dem Resource Description Framework im Semantic-Web-Modell. RDF wurde als Sprache zur semantischen Anreicherung von Webdaten konzipiert, mit dem Ziel, Sachverhalte und Informationen im Web maschineninterpretierbar bzw. leichter automatisierbar zu beschreiben. Schnell wurde jedoch das Bedürfnis nach einer mächtigen Beschreibungssprache laut, die es erlauben sollte, nicht nur einfache, sondern auch komplexe Beziehungen zu beschreiben. Das W3C orientierte sich dabei an den Description Logics, die als vermeintlich guter Kompromiss zwischen menschlicher Interpretation und Maschinenlesbarkeit (und/oder umgekehrt) gesehen wurden.

DLs sind strukturierte Fragmente der Prädikatenlogik mit bestimmten Einschränkungen, auf die hier nicht näher eingegangen werden soll. Einschränkungen bedeutet aber: Einschränkungen bezüglich der Ausdrucksstärke – mit PL1 kann prinzipiell jeder logische Ausdruck erster Stufe beschrieben werden, in DLs sind bestimmte Aussagen ausgenommen – wie auch der Komplexität. PL1 ist nicht entscheidbar, ein Problem, das in dieser Arbeit auch in Hinblick auf CGST noch genauer betrachtet wird.

In der W3C-Community und den für OWL zuständigen Arbeitsgruppen gibt es seit Aufnahme der (Spezifikations-)Tätigkeiten Diskussionen über die Ziele bzw. die daraus abgeleiteten Komplexitäts- und Ausdrucksmerkmale der Sprache. Eine daraus resultierende Folge ist, dass es heute nicht OWL als eindeutige Sprache, sondern drei Dialekte OWL Full, OWL DL und OWL Light sowie zahlreiche proprietäre Zwischen- und Untersprachen, die von der Community und motiviert durch diese Splitting-Strategie entworfen wurden. OWL + RDF sind prinzipiell sehr mächtig und es ist wahrscheinlich, dass basierend auf dem CGST-Ansatz der Großteil der Systembeschreibungen, vielleicht sogar alle Systembeschreibungen, über die OWL-Familie abbildbar sind. Insbesondere bei der Instanzierung sollte es keine Probleme hinsichtlich der Ausdrucksstärke geben.

Eine gute Aufbereitung der Entstehungsgeschichte und ihrer Konsequenzen, die auch für CGST unmittelbar gelten, findet sich in der vielzitierten Arbeit von Ian Horrocks⁶. Dort werden auch schon bereits Komplexitätsmerkmale und Probleme angesprochen. Eine Untersuchung zu Komplexitätsproblemen in logischen Systemen, die auch für Semantic-Web-Anwendungen relevant ist und ebenfalls für CGST unmittelbar interessante Resultate aufzeigt ist die PhD-

⁶Siehe Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen, “From SHIQ and RDF to OWL: The making of a web ontology language”, *Journal of Web Semantics*, 1(1):7-26, 2003.

Arbeit von Stephan Tobies⁷. Zusammengefasst kommen die beiden Arbeiten genauso wie weitere zu folgendem Schluss: Formale Sprachen auf der Ebene von OWL (+ RDF) können problemlos verarbeitet werden, solange bestimmte Definitionen außen vor gelassen werden und eine saubere Modellierung – auch jeweils im Sinne von: unter Kenntnis der Stärken und Schwächen der Sprachen – beachtet wird. Kann aus Gründen der Vollständigkeit oder Notwendigkeit auf gewisse “problematische” Konstruktionen, die zu harten Komplexitätsproblemen führen, nicht verzichtet werden, ist in der Praxis zumeist trotzdem noch eine performante Auflösung möglich, weil Echtdaten in der Regel lückenhaft sind und keine Worst-Case-Szenarien darstellen. Rechnen muss man aber trotzdem mit problematischen Fällen, wenn man OWL in seiner vollen Ausdrucksstärke verwendet (mehr dazu in Kapitel 6).

4.2 Semantic Web Rule Language als Regelkonzept für CGST

Aufgrund der Verwendung von OWL zur Beschreibung der statischen Systemkomponente liegt es nahe, die Semantic Web Rule Language⁸ (SWRL) zur Definition von Regeln in CGST-Systemen heranzuziehen. SWRL ist selbst im Vergleich zu OWL sehr jung und das Ergebnis langer Diskussionen innerhalb der zuständigen W3C-Arbeitsgruppen. SWRL ist bis dato im Gegensatz zu OWL auch noch kein offizieller W3C-Standard, sondern noch immer im Status einer Submission, wird aber aufgrund der nunmehr über vierjährigen Begutachtung und parallelen Verwendung in der Semantic-Web-Community kaum noch maßgeblich adaptiert bzw. durch einen alternativen Ansatz ersetzt werden.

Es handelt sich dabei um eine auf OWL und der Rule Markup Language RuleML, die logische Ausdrücke auf XML-Syntax basierend codiert, aufbauende Kombinationssprache, die zur allgemeinen Kommunikation über Regeln in Semantic-Web-Systemen verwendet werden kann. In CGST kann sie ohne weitere Einschränkungen integriert werden, um auf Basis der Systemontologien bzw. von Instanzen zusätzliche Regeln hinzuzufügen.

Ein einfaches Beispiel ist die Ableitung der Interaktion von Systemen auf Basis von systemübergreifenden Element-Element-Beziehungen. `interactsWith` wurde ja im letzten Abschnitt auch als explizite Möglichkeit der Annotation von System-Interaktionen beschrieben. Die implizite Annotation, die von einem entsprechenden Reasoner aus Element-Beziehungen abgeleitet werden kann, kann etwa über die folgende Regel beschrieben werden ($P1 - P5$ sind Prämissen der

⁷Stephan Tobies, “Complexity Results and Practical Algorithms for Logics in Knowledge Representation”, PhD thesis, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 2001.

⁸Siehe <http://www.w3.org/Submission/SWRL/>.

Regel, $C1$ ist die Konklusion):

- (P1) S_1 type System
 - (P2) S_2 type System
 - (P3) $(E_1 \text{ type Element}) \wedge (S_1 \text{ hasElement } E_1)$
 - (P4) $(E_2 \text{ type Element}) \wedge (S_2 \text{ hasElement } E_2)$
 - (P5) $(e_r \text{ type elementRelation}) \wedge (E_1 e_r E_2)$
- (C1) $\rightarrow S_1 \text{ interactsWith } S_2$

Die Regel erkennt also Elemente unterschiedlicher⁹ Systeme und schließt aus einer Beziehung zwischen diesen Elementen auf eine Interaktion zwischen den betreffenden Elementen. Diese Implikation kann in ihrer praktischen Bedeutung sehr wichtig sein, wenn sie eine vergleichsweise wesentliche Beziehung beinhaltet oder unwesentlich, wenn es sich bei der Beziehung nur um eine unbedeutende Randbeziehung handelt. Auf jeden Fall ist sie aber aus ontologischer Perspektive immer korrekt, weil eine Systeminteraktion dann und nur dann stattfindet, wenn es eine Relation zwischen den Elementen der Systeme gibt. Der Regelsatz ist eine an die formale Syntax von SWRL angelehnte Repräsentation einer SWRL-Regel. Für eine Anwendung bzw. unter Verwendung der SWRL-XML-Syntax muss er wie folgt übersetzt werden:

```
<ruleml:imp>
  <ruleml:_rlabel ruleml:href="#Implicit_SystemInteraction"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasElement">
      <ruleml:var>S1</ruleml:var>
      <ruleml:var>E1</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom swrlx:property="hasElement">
      <ruleml:var>S2</ruleml:var>
      <ruleml:var>E2</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom swrlx:property="elementRelation">
      <ruleml:var>E1</ruleml:var>
      <ruleml:var>E2</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="interactsWith">
      <ruleml:var>S1</ruleml:var>
      <ruleml:var>S2</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

Die Typbeziehungen, dass es sich bei $S1$ und $S2$ um Systeme und $E1$ und

⁹ $S_1 \neq S_2$ ist nicht Teil des Regelsatzes, es könnte sich also auch um Elemente des gleichen Systems handeln. Dann ist die Implikation aber trivial, weil für jedes System S_x ohnedies gelten sollte: $S_x \text{ interactsWith } S_x$.

E2 um Elemente handelt, kann in diesem Fall weggelassen werden, weil ein SWRL-Reasoner, der die CGST-Ontologie im Hintergrund kennt, dieses Wissen implizit ableitet und diese zusätzlichen Prämissen in der XML-Repräsentation daher redundant wären. In der Praxis kann Jess¹⁰ als Reasoning-Engine im Hintergrund verwendet werden. Jess ist prinzipiell als Reasoning-Engine sprach-unabhängig und kann über eine XML-Schnittstelle auch zum Auflösen von SWRL-Konstrukten verwendet werden. Ein entsprechendes Plugin für Protege, JessTab¹¹, ist ebenfalls vorhanden (siehe Abbildung 4.3).

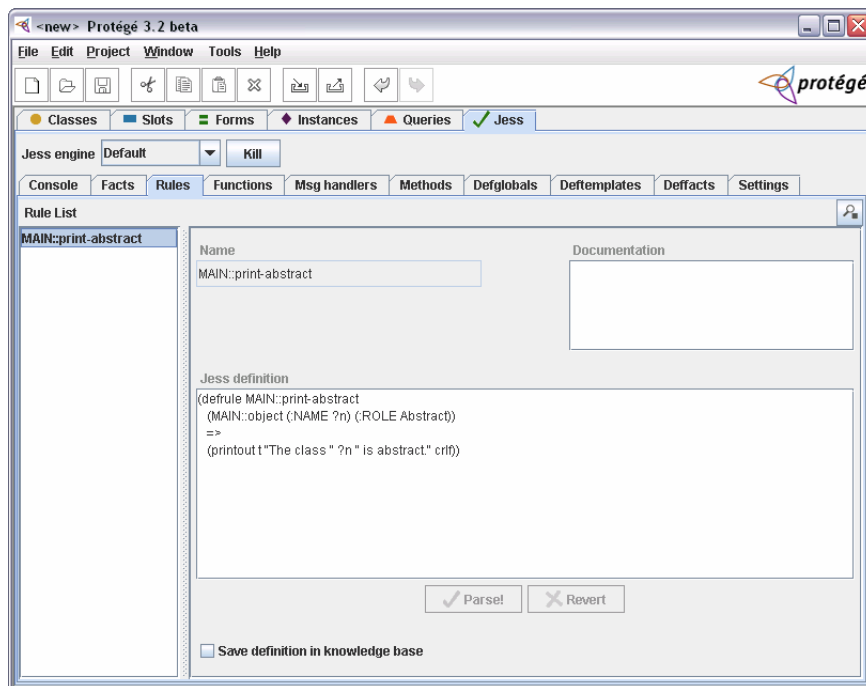


Abbildung 4.3: Es existiert auch ein Jess-Plugin für Protege, in dem die Regeln bearbeitet werden und das Ergebnis zur Laufzeit berechnet werden können.

Jess ist im Zusammenhang mit OWL und RDF sicher nicht die erste Wahl, weil es aufgrund der Ableitung von OWL aus DL heraus naheliegendere Varianten gibt. Das bedeutet, es macht Sinn, DL-Reasoner zum Auflösen von Regeln in Semantic-Web-Systemen zu verwenden. Bekannte Systeme, die DL-Reasoning oder ähnlich ausdrucksstarke Ableitungen erlauben, sind etwa RACER¹², Pellet¹³ oder Fact++¹⁴ sowie die systemgebundene, proprietäre Reasoning-Engine im SW-Speichersystem OWLIM¹⁵. Alle diese Anwendungen sowie die Regelsprachen, auf denen sie basieren, haben jedoch einen Nachteil, den sie aus DL übernehmen und der für das CGST-System nicht akzeptabel ist: sie unterstützen keine arithmetisch-funktionalen Regeln, es sind also in den Prämissen

¹⁰Dokumentation und How-To siehe <http://www.jessrules.com/jess/docs/71/>.

¹¹Siehe <http://www.ida.liu.se/~her/JessTab/>.

¹²Siehe <http://www.racer-systems.com/>.

¹³Siehe <http://clarkparsia.com/pellet/>.

¹⁴Siehe <http://owl.man.ac.uk/factplusplus/>.

¹⁵Siehe <http://www.ontotext.com/owlim/index.html>.

sen und den Konklusionen keine Abfragen bzw. Zuweisungen auf dieser Ebene möglich. SWRL und Jess hingegen erlauben derartige Konstruktionen, weil SWRL nicht nur als OWL-Regelsprache, sondern prinzipiell als Regelsprache für Semantic-Web-Anwendungen entwickelt wurde und dieser Nachteil zum Zeitpunkt der Ausarbeitung bereits bekannt war – eine analoge Diskussion findet sich derzeit auch in den Arbeitsgruppen des W3C zur Ausweitung der Semantic-Web-Querysprache SPARQL, die diese arithmetisch-funktionale Unterstützung derzeit ebenfalls nicht bietet.

Ein einfaches Beispiel für die Notwendigkeit arithmetischer Vergleichsregeln wäre die Ableitung eines booleschen Wertes `thrombocytosis` aufgrund der Anzahl der Thrombozyten, die in einem Integer-Wert `numberOfThrombocytes` abgebildet ist. Eine solche Regel hätte die Form

- (P1) *S₁ type System*
 - (P2) *S₁ hasElement Organism*
 - (P3) *numberOfThrombocytes type attribute*
 - (P4) *numberOfThrombocytes domain Organism*
 - (P5) *numberOfThrombocytes range integer*
 - (P6) *numberOfThrombocytes ≥ 300.000*)
- (C1) *→ thrombocytosis is true*

In DL kann ein solcher Ausdruck nur auf atomarer Ebene behandelt werden, aber nicht über die Logik selber aufgelöst werden. Noch stärker sind die Einschränkungen in der Praxis, wenn es nicht nur um arithmetische Vergleiche, sondern etwa um Aggregatfunktionen wie eine einfache Summenbildung geht. Dafür gibt es dann – im Gegensatz zu SQL – auch in den meisten Querysprachen für Semantic-Web-Anwendungen keine Auflösung. Ist also im Fall der arithmetischen Vergleiche theoretisch noch eine Vorbereitung bzw. ein Workaround in der Querysprache möglich, so scheidet diese Möglichkeit bei Aggregatfunktionen aus – es bleibt SWRL tatsächlich als eine der wenigen Alternativen, die all diesen Ansprüchen genügen.

4.3 Von der Systemontologie zum Zellulären Automaten

Die Systeme sollen jetzt aber nicht nur auf die statischen Ontologien, sondern auch dynamisch abgebildet werden. Das Semantic-Web-Konzept kennt aber keine Standards zur “Ontogenese”, sondern eben nur solche für Ontologien. Die Folge daraus sind zahlreiche proprietäre Lösungen um mit über die Zeit veränderten Ontologien umzugehen – eine davon ist die auf der Plattform myOntology eingesetzte Strategie zur Verwaltung von unterschiedlichen Ver-

sionen mittels klar definierter Freezeps (als konsistent deklarierte, stabile Zwischenversionen), zu deren Zeitpunkten die Ontologien konstant sind, sich aber durch ein maßgebliches Delta von der Vorversion unterscheiden.

Das ist aber nicht das, was wir an dieser Stelle technisch benötigen – wir wollen ja eine Möglichkeit haben, dynamische Regeln zu definieren, die unsere Systeme von einem Zustand s_1 in einen Zustand s_2 überführen und zwar derart, dass dabei die in den Ontologien bereits definierten möglichen Zustände berücksichtigt werden und die dynamischen Regeln auf diesen Konzepten basieren.

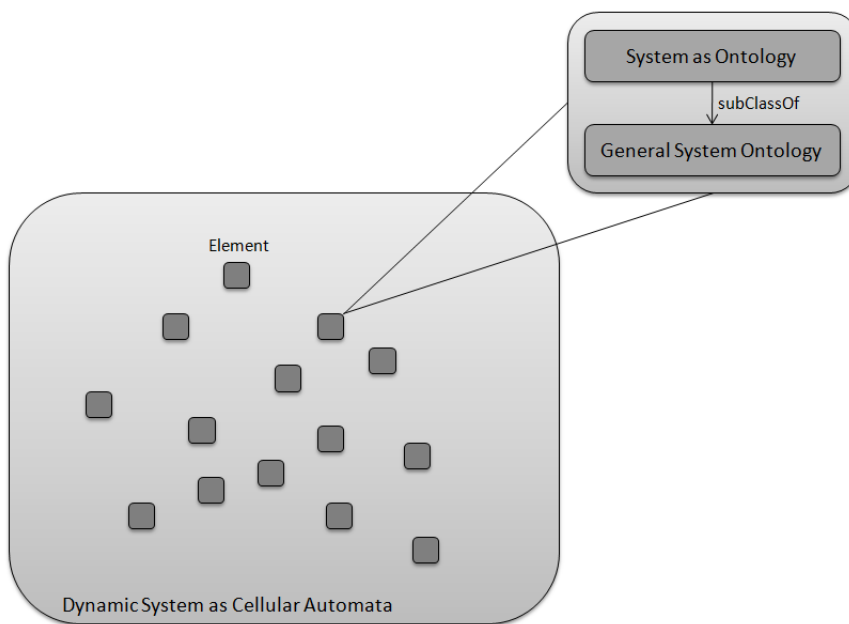


Abbildung 4.4: Aus der ontologischen Systembeschreibung und den einzelnen Instanzen werden Zelluläre Automaten abgeleitet, die in weiterer Folge eine Simulation und dynamische Analyse des Systems erlauben.

Zusätzlich haben wir ja im letzten Kapitel festgestellt, dass wir für den dynamischen Teil des Modells eine Variante benötigen, die Systeme mit vielen autonom agierenden Elementen abbilden kann und auf diese parallelen Prozesse Rücksicht nimmt. In der Informatik hat sich für diesen Teilbereich der Begriff der Zellulären Automaten herausgebildet – er beschreibt eine ganz spezielle Automatenvariante, bei der der Fokus der Abarbeitung des Programms auf der parallelen Aktion vieler autonom agierender, sogenannter Agenten liegt. Im letzten Kapitel wurde die Historie und der Zusammenhang dieses Modells der Informatik mit dem wesentlichen Modell der Turingmaschine bereits beschrieben.

Bevor jetzt die konkrete formale Ausprägung im CGST-Modell definiert

wird, soll noch kurz ganz allgemein der Zusammenhang der beiden Teile – Ontologien und Zelluläre Automaten – erklärt werden: Ontologien werden ja dazu verwendet, formal abstrakte, statische Sachverhalte der Realität in einer maschineninterpretierbaren, gleichzeitig aber auch noch vom menschlichen Entwickler unmittelbar nachvollziehbaren Form zu repräsentieren. Automaten – und das gilt auch für die Sonderform des Zellulären Automaten – enthalten ebenfalls eine statische Sachverhaltsbeschreibung als Teil ihrer Konfiguration, und zwar sowohl betreffend die möglichen Folgezustände in Form einer impliziten Definition des Möglichkeitsraums und zum anderen betreffend die Grundstruktur des Automaten selbst, also welche Zustände es prinzipiell gibt, welche Variablen veränderbar sind, usw. Diese statischen Komponenten sind, was ihre Ausdrucksstärke betrifft, jedenfalls auch durch Beschreibung mittels DL-Semantik oder OWL-Ontologien möglich. Umgekehrt ist es dadurch auch prinzipiell möglich, aus OWL-Ontologien derartige Grundgerüste verlustfrei abzuleiten. Formal werden dazu Transformationsregeln, die die Besonderheiten der CGST-Ontologie beinhalten und daher auf alle Systembeschreibungen die basierend auf dieser Stammontologie definiert werden, erstellt, die CGST-Element-Instanzen in Agenten Zellulärer Automaten überführen, wie Abbildung 4.4 veranschaulicht.

4.3.1 NetLogo – Programmierung Zellulärer Automaten

NetLogo¹⁶ ist eine sehr intuitive Programmiersprache zur Entwicklung von Zellulären-Automaten-Systemen. Die Programmsyntax ist zwar proprietär, aber das Programm ist im Bereich der Zellulären Automaten schon sehr populär, außerdem – was für unsere Sache noch viel wichtiger ist – lässt sich die NetLogo-Syntax aufgrund ihrer Struktur sehr einfach auf die Ontologiesprache OWL abbilden. Bevor wir uns diese Abbildung etwas genauer ansehen und diskutieren, wollen wir aber noch kurz einen Blick auf die derzeitigen Anwendungsgebiete von NetLogo werfen, um die “Fitness” der Applikation in Bezug auf unser Modell hervorzuheben. Eine Einführung in Syntax und Grammatik werden wir an dieser Stelle nicht geben, der interessierte Leser möge also, wie auch schon im vorigen Kapitel betreffend OWL, einen kurzen Blick auf die Einführung in der Online-Dokumentation werfen.

Sehen wir uns zuerst das folgende NetLogo-Programm¹⁷ an, das eine Rebellion in sozialen Systemen, also den Aufstand der Bevölkerung gegen eine herrschende Autorität, simuliert.

In Abbildung 4.5 repräsentieren die ausgefüllten Kreise Bürger als Elemente eines sozialen Systems, die Dreiecke Polizisten als weitere Elemente. Bürger haben die atomare Eigenschaft der “Unzufriedenheit” mit dem politischen Sy-

¹⁶Siehe <http://ccl.northwestern.edu/netlogo>.

¹⁷Eine ausführliche Beschreibung findet sich hier <http://ccl.northwestern.edu/netlogo/models/Rebellion>.

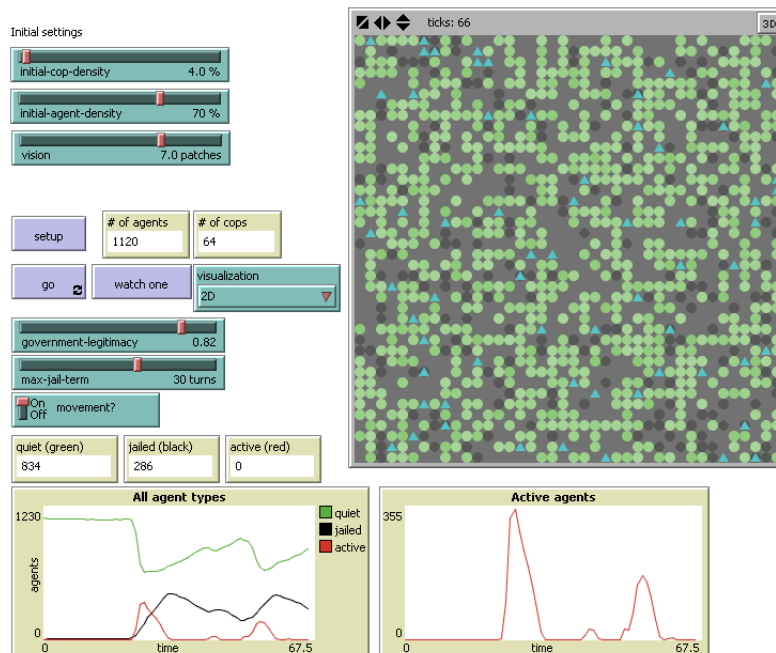


Abbildung 4.5: NetLogo-Simulation einer Rebellion gegen die herrschende Autorität eines politischen Systems.

stem, die sich von der wahrgenommenen Gewalt der Polizei und der Legitimität der Autorität ableitet. Außerdem kalkulieren die einzelnen Bürgerelemente in jedem Status die Gefahr, bei einem potentiellen Gewaltakt festgenommen zu werden. Das System lässt sich anhand diverser einstellbarer Faktoren wie Anzahl der Polizisten, Bevölkerungsdichte, u.ä. konfigurieren.

Die Simulation zeigt dann den Verlauf und den Ausbruchsmoment einer Rebellion, d.h. eines Zustandes im System, in der ein großer Teil der Bevölkerung gewaltbereit ist, weil die einzelnen Bürger ihrerseits ein immer höheres Maß an Gewalt des Staates wahrnehmen. Uns interessiert jetzt gar nicht so sehr die Thematik dieses Beispiels an sich, sondern eigentlich nur die bereits schon festgestellten Repräsentationsarten der einzelnen Teilnehmer: Akteure des politischen Systems sind autonome Agenten des Automaten mit wenigen atomaren Eigenschaften, der Zustand des Systems wird schließlich etwa an der Anzahl der festgenommenen Bürger gemessen.

Ein zweites interessantes Beispiel¹⁸ in der Modell-Bibliothek der aktuellen NetLogo-Simulationen ist eine Implementierung von James Lovelocks Daisyworld-Modell¹⁹. Es zeigt das Systemverhalten eines Planeten, auf dem weiße und schwarze Gänseblümchen mit unterschiedlicher Albedo-Charakteristik leben und sich reproduzieren. Das System ist homöostatisch, d.h. kann störende Entwick-

¹⁸Siehe <http://ccl.northwestern.edu/netlogo/models/Daisyworld>.

¹⁹Siehe [30], Seite 66 f.

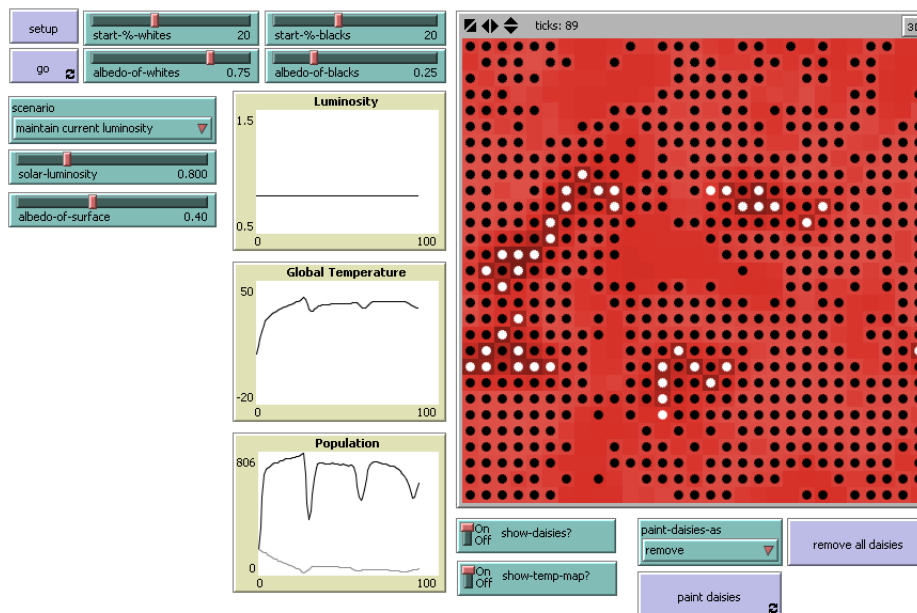


Abbildung 4.6: NetLogo-Simulation von James Lovelocks Daisyworld-Modell.

lungen bis zu einem gewissen Grad kompensieren. Erst wenn durch eine radikale Steigerung oder Minderung der globalen Albedo-Charakteristik die Temperatur zu hoch oder zu niedrig wird, ist kein Reproduzieren der Gänseblümchen mehr möglich und es kommt zu einem totalen Zerfall des Systems. Auch hier kommen wir mit sehr wenig programmatischen Elementen aus, es werden nur weiße und schwarze Gänseblümchen benötigt, die eine konfigurierbare Albedo-Charakteristik und die Fähigkeit zur Reproduktion (ein Standard-Prozess bei Zellulären Automaten und bei NetLogo) besitzen, wie auch Abbildung 4.6 zeigt.

Wenn wir jetzt einen Blick auf die jeweiligen Code-Fragmente werfen, so sehen wir beim ersten Beispiel eine Repräsentation der Agenten durch

```
agents-own [
  risk-aversion      ; R, fixed for the agent's lifetime, ranging from 0-1 (inclusive)
  perceived-hardship ; H, also ranging from 0-1 (inclusive)
  active?           ; if true, then the agent is actively rebelling
  jail-term         ; how many turns in jail remain? (if 0, the agent is not in jail)
]

patches-own [
  neighborhood      ; surrounding patches within the vision radius
]
```

und im zweiten Beispiel eine Repräsentation durch

```
breed [daisies daisy]
```



```

patches-own [temperature] ;; local temperature at this location

daisies-own [
  age      ;; age of the daisy
  albedo   ;; fraction (0-1) of energy absorbed as heat from sunlight
]

```

für die Definition der Agenten bzw. Elemente an sich. Die syntaktische Ähnlichkeit zu den OWL-Fragmenten, in denen Elemente mit ihren Eigenschaften definiert sind, ist offensichtlich. Im Falle von Booleschen Variablen ist überhaupt eine verlustfreie Abbildung möglich, da diese in NetLogo mit einen “?” am Ende der Variablenbezeichnung gekennzeichnet werden. Die Oberfläche (“Patches”) wird bei NetLogo über ein differenziertes Konzept *patches – own* abgebildet, das sich aber grundsätzlich syntaktisch mit den Agentendefinitionen so stark deckt, dass für unser Modell eine analoge Abbildung möglich ist.

Systemeigenschaften können analog dazu definiert werden und werden in NetLogo als globale Variablen gehandelt.

```

globals [
  max-age           ;; maximum age that all daisies live to
  global-temperature ;; the average temperature of the patches in the world
  num-blacks       ;; the number of black daisies
  num-whites       ;; the number of white daisies
  scenario-phase   ;; interval counter used to keep track of what
                  ;; portion of scenario is currently occurring
]

```

Etwas kritisch ist hier sicherlich die Behandlung der maximalen Lebensdauer der Gänseblümchen als globale Variable bzw. Systemeigenschaft, dies kann aber auch anders repräsentiert werden und ist hier sicherlich nur aus Effizienzgründen derartig umgesetzt. Die weiteren Code-Fragmente beziehen sich dann auf regelbasiertes Schließen, die Systemdynamik sowie Monitoringmöglichkeiten. Auf die ersten beiden werden wir später noch etwas genauer eingehen, vorher wollen wir aber noch die Lücke schließen und einen Blick auf die technische Übersetzung von OWL nach NetLogo werfen.

4.3.2 Ableitung von NetLogo-Gerüsten aus OWL mittels XSL

Einer der Gründe, warum wir uns bei der Ontologiebeschreibungssprache für OWL entschieden haben ist der Aufbau als XML-Sprache. Da sich das gesamte Semantic-Web-Schichtenmodell an XML-Syntax orientiert, wird auch die Stufe der ontologischen Beschreibung in dieser Form realisiert. Tatsächlich ist das keine Notwendigkeit für Ontologiebeschreibungen im Allgemeinen, wie die zahl-

reiche Präsenz alternativer Sprachen zur Formulierung von Ontologien zeigt.

Warum ist die XML-Syntax für uns von Vorteil? Zum einen, weil XML ein gängiger Industriestandard ist und weltweit zur Repräsentation von Wissen eingesetzt wird – nur eben auf einer so “schwachen” semantischen Ebene, dass man zuerst Objektbeschreibungen (RDF) und Zusammenhänge zwischen diesen (OWL) darauf aufsetzen muss, um wirklich über Zustände und nicht nur über Daten kommunizieren zu können. Zum anderen, weil die Integration von diversen Regelsystemen, vor allem aber die Transformation in andere Sprachen aufgrund der XML-Übersetzungssprache XSL vergleichsweise einfach und praktisch funktioniert.

Im Prinzip lassen sich ja sehr leicht konzeptuelle Ähnlichkeiten zwischen den Instanzen der Systemontologie und den im letzten Absatz vorgestellten NetLogo-Agenten herstellen: Was wir in der Systemontologie als Unterklasse unseres Element-Konzepts darstellen, wird im Zellulären Automaten zu einem “aktiven” Agenten; die Eigenschaften auf deren Basis der Automat determiniert sind zum Teil Eigenschaften der Agenten und damit ursprünglich atomare Attribute der Element-Unterklassen. Diese Bereiche lassen sich aber mittels XSL im OWL-Graphen sehr einfach identifizieren und praktisch im selben Schritt kann der Prozessor bereits die Abbildung in ein NetLogo-Gerüst der diversen Agenten vornehmen, was im folgenden Codefragment abgebildet ist.

```
<xsl:template match="/">
  <xsl:for-each select="//owl:Class">
    <xsl:if test="current()//rdfs:subClassOf and
      (current()//rdfs:subClassOf[@rdf:resource='&system;Element'])">
      [
        <xsl:value-of select="@*" />
      ]
    </xsl:if>
  </xsl:for-each>
</xsl:template>
```

Die Grenzen der Abbildung sind dabei aufgrund der großen Ausdrucksstärke von OWL (die damit verbundenen Probleme haben wir ja bereits im letzten Kapitel kurz angeschnitten) nicht durch die Sprache gesetzt sondern ergeben sich ganz einfach durch die fehlende Information jenes Teils unseres Modell-Tupels, der mittels einer Ontologiesprache nicht abgebildet werden kann – die Übergangsregeln für den Automaten können erst zum Zeitpunkt der Modellierung mittels NetLogo hinzukommen oder müssen durch eine entsprechende proprietäre Repräsentation – etwa durch ein einfaches XML-Schema oder aber eine OWL-Ontologie für Automaten – hinzukommen.

Der Verarbeitungsprozess ist in Abbildung 4.7 schematisch veranschaulicht. RDF-Instanzen basieren auf einer speziellen Systemontologie, die wiederum auf der CGST-Systemontologie basiert. Das XSL-Transformationskript wurde ba-

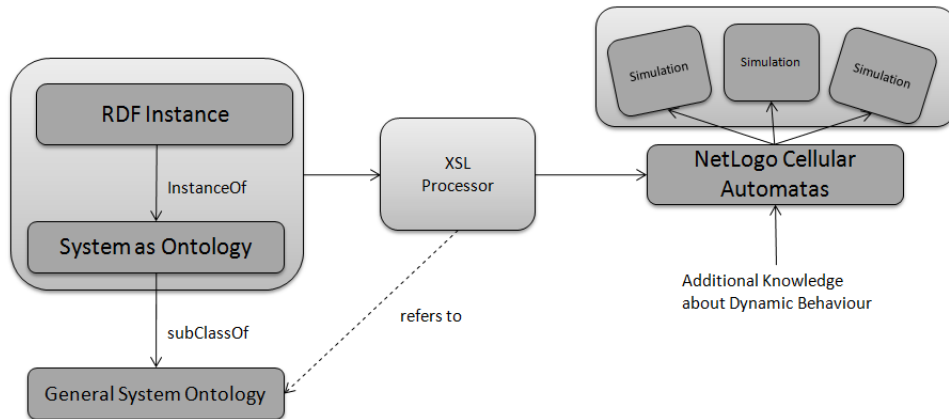


Abbildung 4.7: Verarbeitungsprozess.

sierend auf den speziellen Eigenschaften dieser Ontologie so konfiguriert, dass die notwendigen Informationen aus CGST-basierenden Instanzen und Ontologien in ein NetLogo-Gerüst übersetzt werden können. Erst nachdem das Gerüst erzeugt wurde, können – manuell oder basierend auf weiteren Konfigurationsdateien – zusätzliche Informationen über das dynamische Verhalten der Automaten integriert werden.

4.4 Technische Umsetzung

Langfristig müssen zu einer umfangreichen und effektiven Nutzung des Systems Softwareanwendungen angestrebt werden, die die beteiligten Komponenten nicht in der hier dargestellten losen Form koppeln, sondern einen integrativen Ansatz verfolgen und die einzelnen Schichten möglichst performant und skalierbar vereinen. Dazu müssten Zelluläre Automaten direkt an einen Tripstore gekoppelt werden, der ein entsprechend aussagekräftiges Regelsystem unterstützt oder implementiert. Ein solches System müsste außerdem über entsprechende Frontend- und Backendfunktionalitäten verfügen, die über die einzelnen Schichten abstrahieren und effizient Modellbildung und Analogiesuche in unterschiedlichen Systemen unterstützen. Ein Qualitätsmaßstab für ein solches System wäre dann die tatsächliche Vermeidung von doppelter Arbeit bzw. die gewonnene Zeitersparnis durch die Nutzung eines solchen Systems.

In dieser Arbeit müssen wir uns aber auf eine offene Architektur und die Beschreibung der zur Verfügung stehenden Technologien im Allgemeinen sowie einfache Use-Cases beschränken. Die Abbildung der einzelnen Modellbereiche, die zu Beginn des Kapitels skizziert wurden, auf die in der Praxis verwendeten Technologien, die bisher beschrieben wurden, ist in Abbildung 4.8 veranschau-

licht. “Generic XSL-Transformation” bezeichnet dabei die CGST-Ontologiebasierte Ableitung der NetLogo-Modelle aus den speziellen System-Ontologien, mit “Autopoietic Algorithms” sind Verfahren zur Adaption und Reproduktion der NetLogo-Systeme gemeint, die auf einer Meta-Ebene ansetzen und z.B. eine zielwertgesteuerte Optimierung von speziellen Systemen unterstützen können.

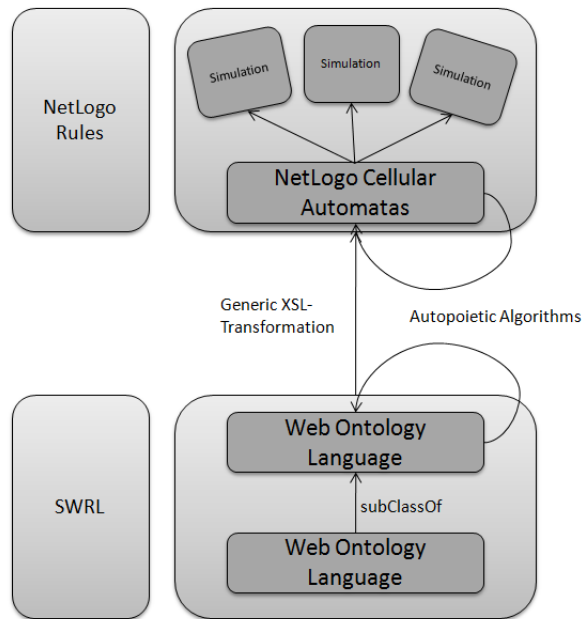


Abbildung 4.8: Darstellung der Hintergrundtechnologien für die einzelnen CGST-Anwendungskomponenten.

Bisher haben wir uns auf das Erstellen und Ausführen von Schematas und Instanzen der einzelnen Bereiche beschränkt, ohne genauer auf die in der Praxis zur Verfügung stehenden Technologien und ihre Eigenschaften einzugehen. Diesem Punkt wollen wir uns nun in den folgenden Absätzen widmen und für jeden der Teilbereiche gängige Frameworks vorstellen und in Hinblick auf ihre Kompatibilität mit CGST evaluieren.

4.4.1 Verwaltung von Γ , Δ und Θ in einem Triplestore

Unter einem Triplestore versteht man einen Datenspeicher, der alle Informationen in Form von Aussagetriples (Subjekt, Prädikat, Objekt), oder anders beschrieben, in Form eines gerichteten Graphen abspeichert. Das kann logisch und physisch sehr unterschiedlich passieren, wodurch mittlerweile eine beachtliche Zahl unterschiedlicher Triplestore-Lösungen verfügbar ist. Für die Anforderungen, die wir aufgrund der Beschaffenheit der System-Ontologie haben, entscheiden wir uns für die Verwendung der Sesame 2.0-Lösung²⁰ zur Veranschaulichung der möglichen Datenverwaltung von CGST-Daten. Eine Untersuchung

²⁰Siehe <http://www.openrdf.org/doc/sesame2/users/>.

über Alternativen und Eigenschaften von Triplestores und den damit verbundenen Performance- und Skalierbarkeitscharakteristika, insbesondere auch in Hinblick auf Reasoning-Tasks bietet Schliefnig²¹.

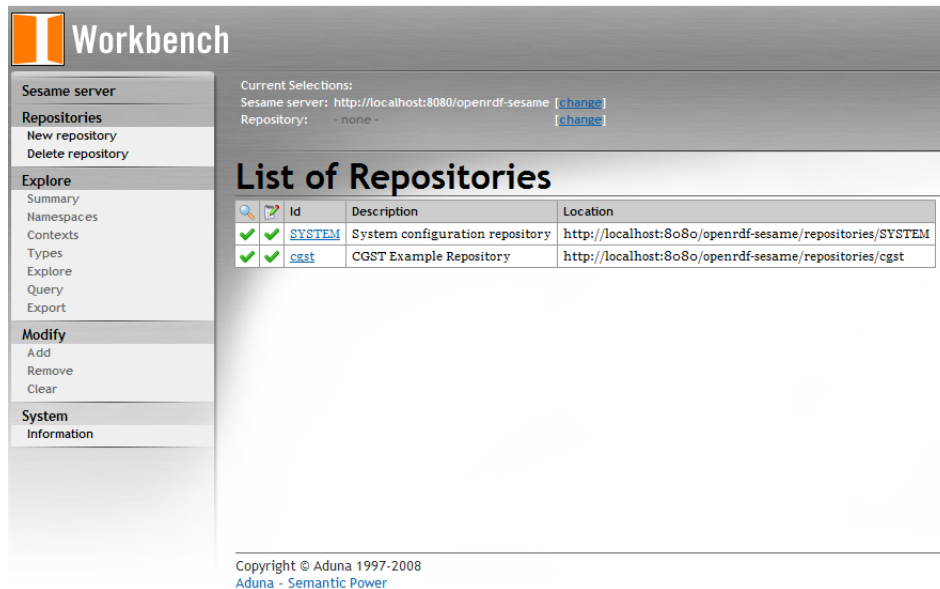


Abbildung 4.9: Repositoryliste im Sesame2 Webinterface.

Bei Speicherinstanzen in Triplestores ist nicht mehr von Datenbanken sondern allgemeiner von so genannten Repositories die Rede. Unter einem Repository versteht man einen abgekoppelten, autonomen Datengraphen, in dem sämtliche Informationen und Metainformationen der Wissensbasis vorhanden sind. Diese Form der Speicherung hat hinsichtlich der Interoperabilität einen entscheidenden Vorteil: Da die Graphen in der Regel ebenfalls basierend auf W3C-Standardsprachen repräsentiert werden, können sie unabhängig von der konkreten darüberliegenden Technologie ausgetauscht werden. Ein Triplestore kann in der Regel auch OWL- und RDF-Dateien direkt laden und speichern, d.h. eine Ontologie sowie darauf basierende Instanzen können ohne Zwischenschritt in einen solchen Triplestore überführt und verwaltet werden.

“Verwaltet werden” heißt dabei in erster Linie “abgefragt werden”. Wie in Relationalen Datenbanken besitzen auch Triplestores Abfragesprachen, um konkrete Informationen aus den Wissensgraphen extrahieren zu können. Diese Abfragesprachen sind in der Regel ebenfalls graphenbasiert und beschreiben dann bei einer konkreten Abfrage einfach nur einen entsprechenden Subgraphen. Auch hier gibt es seitens des W3C bereits einen etablierten Standard: SPARQL wurde explizit zur Arbeit mit OWL- und RDF-Stammdaten konzipiert und ist heute die am meisten verbreitete Triple-Abfragesprache und wird

²¹Martin Schliefnig, “Reasoning in Semantic-Web-Applications”, TU Vienna Diploma Thesis, 2007.

auch von den am häufigsten verwendeten Triplestores unterstützt.

Abbildung 4.9 zeigt das Sesame2-Webinterface mit einer Liste der verfügbaren Repositories. Vorerst wurde nur die CGST-Ontologie in das System geladen.



Abbildung 4.10: Klassenhierarchie der Sample-Wissensbasis.

Darauf basierend können nun allgemeine Abfragen formuliert werden, die bereits a priori Wissensextraktionen über eventuell später hinzukommende Systeminstanzen formulieren bzw. entsprechend allgemein gültig sind. Das können einfache Abfragen der Art

Zeige alle Elemente in der Wissensbasis

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX system:<http://www.systems-everywhere/system#>

SELECT ?element
WHERE
{ ?element rdfs:subClassOf system:Element }
```

oder *Zeige alle Element-Element-Beziehungen in der Wissensbasis*

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX system:<http://www.systems-everywhere/system#>
```

```
SELECT ?element1 ?elementRelation ?element2
WHERE
{ ?element1 rdfs:subClassOf system:Element.
  ?element2 rdfs:subClassOf system:Element.
  ?elementRelation rdfs:domain ?element1.
  ?elementRelation rdfs:range ?element2}
```

sein. Solche Abfragen können vor allem bei großen Wissensbasen sehr schnell gerade benötigtes Wissen über das System zurückliefern und sind sehr flexibel²² und ausdrucksstark. Wichtig ist hier vor allem der Aspekt, dass bei solchen generalisierten Abfragen keinerlei Detailwissen über konkrete Systeme vorhanden sein muss: Das spiegelt auch die ausschließliche Verwendung der allgemeinen Namespaces `rdfs:<http://www.w3.org/2000/01/rdf-schema>` und `system:<http://www.systems-everywhere/system>` wider – der Sample-Namespace oder gar spezielle Konzepte der Sample-Wissensbasis werden hierfür nicht benötigt.

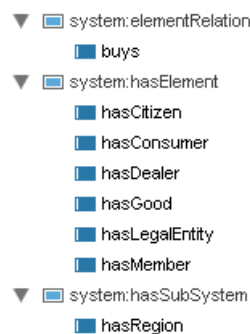


Abbildung 4.11: Hinzugefügte System-Subproperties zur Verknüpfung der unterschiedlichen Domänen-Ontologien.

Zu Test- und Präsentationszwecken soll dieser Ansatz nun mittels allgemein bekannter und verwendeter Semantic-Web-Standardontologien validiert werden. Dazu werden bekannte Ontologien zur Beschreibung von Personeninformationen (FOAF), Geoinformationen (Geonames) sowie E-Commerce Angebotsbeschreibungen (GoodRelations) mit der CGST-Ontologie verknüpft und sinnvoll untereinander kombiniert. Dadurch wird eine Schemastruktur geschaffen, die es etwa erlaubt, individuelle Personen mit Orten oder Kaufhandlungen in Verbindung zu bringen und somit systemische Aspekte aus Sozialen Systemen bzw. Sozialen Netzwerken, Politischen Systemen, Communities und Wirtschaftssystemen abzubilden.

Da für alle verwendeten Ontologien bereits zahlreiche Instanzen im Web verfügbar sind, erhält man auf diesem Weg auch sehr einfach Ausgangsdaten

²²Bei prozessorientiertem Systemdesign besteht diese Möglichkeit prinzipiell nicht, da der Aufbau der Wissensbasis hier nicht konstruktiv stattfindet, sondern modular nach dem Prinzip der Prozessverschachtelung.

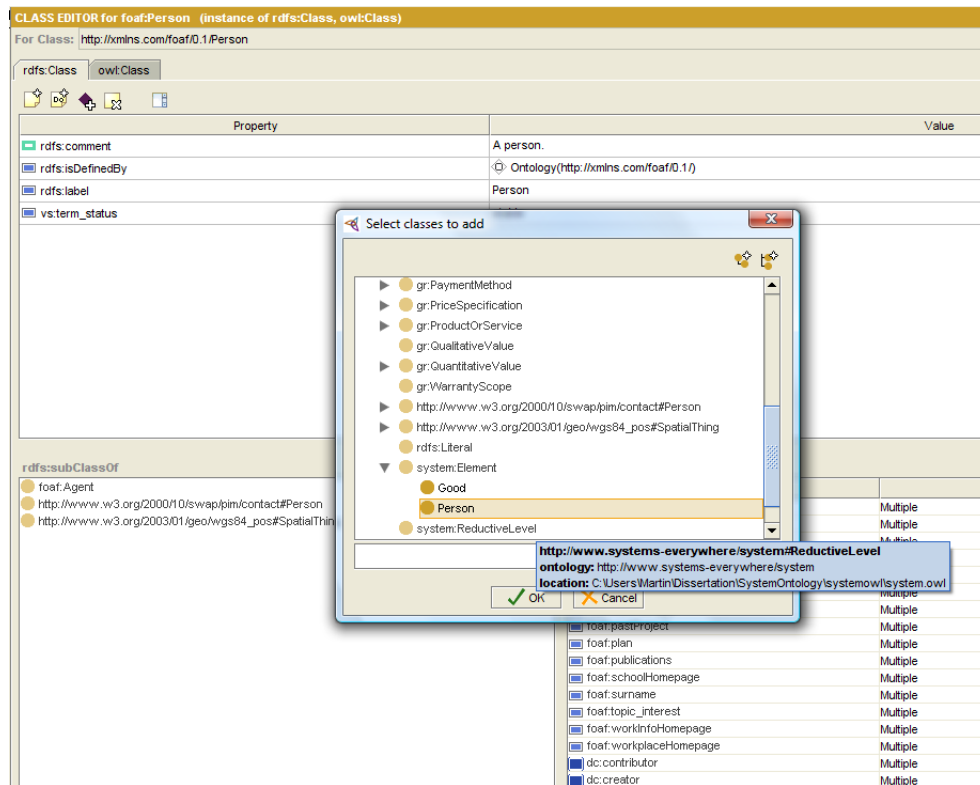


Abbildung 4.12: Verknüpfung der FOAF-Ontologie mit der CGST-Systemontologie.

für eine Testdatenbasis. Die wesentlichen Verknüpfungsaspekte sind in Abbildung 4.10 veranschaulicht, notwendige Verknüpfungen auf der Ebene von Relationen sind in Abbildung 4.11 abgebildet. Beispielhaft sei noch ein Verknüpfungsschritt in Abbildung 4.12 abgebildet: Dieser Auszug stellt die Verknüpfung der FOAF-Ontologie mit der CGST-Ontologie über die Zuweisung des foaf:Person-Konzepts zum system:Element-Konzept dar. Aber auch komplexere Abfragen, die bereits konkrete Systemeigenschaften abbilden, können auf diesem Weg abgesetzt werden.

Zeige alle direkten Wechselbeziehungen und Wechselbeziehungen mit Distanz 2 von Personen im Konzept

```

PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX system:<http://www.systems-everywhere/system#>
PREFIX sample:<http://www.systems-everywhere.com/sample#>

SELECT distinct ?elementRelation1 ?elementX ?elementRelation2
           ?elementY ?elementRelation3
WHERE
{
  ?elementX rdfs:subClassOf system:Element.
  ?elementY rdfs:subClassOf system:Element.
  ?elementRelation1 rdfs:domain sample:Person.

```



```

?elementRelation1 rdfs:range ?elemntX.
?elementRelation2 rdfs:domain ?elemntY.
?elementRelation2 rdfs:range ?elementX.
?elementRelation3 rdfs:domain ?elemntY.
?elementRelation3 rdfs:range sample:Person}

```

Diese Abfrage beschreibt ein ganz besonderes “Beziehungsdreieck”, fokussiert auf das Beispiel-Konzept `sample:Person`: Abgefragt werden all jene Teilgraphen, in denen das Personen-Konzept jeweils eine eingehende und eine ausgehende Relation zu Elementen im System besitzt, wobei diese beiden Elemente (die durchaus auch identisch sein können) wiederum eine gerichtete Beziehung besitzen müssen. Dreiecksbeziehungen in Systemen oder über Systemgrenzen hinweg können in Hinblick auf komplexe Effekte oder versteckte Abhängigkeiten interessant sein und sind gute Beispiele für strukturierte Abfragen, weil sie oft neue Ergebnisse hervorbringen.

4.4.2 Scriptbasierte Erzeugung der OWL-basierten Automaten

Die Integration eines XSL-Prozessors ist relativ unkompliziert und muss nicht im Rahmen einer selbständigen Anwendung passieren sondern kann scriptbasiert erfolgen. Wir wollen hier nicht im Detail auf unterschiedliche Optionen der Verarbeitung eingehen, aber trotzdem kurz einen Auszug aus einem solchen Konsole-Script präsentieren, um die unkomplizierte Handhabung dieses Vorgangs zu verdeutlichen. An dieser Stelle sei nur der Aufruf des Prozessors dargestellt, ein vollständiges, direkt lauffähiges Konsole-Script findet sich im Anhang. Dieses Script kann auf einem beliebigen Windows-Rechner sofort ausgeführt werden und erzeugt auf Basis des ebenfalls im Anhang aufgelisteten XSL-Transformationsfiles generisch NetLogo-Automaten aus vorhandenen OWL-Dateien, die auf die C-GST-Ontologie referenzieren. Der Aufruf des Prozessors, der die Transformation durchführt sieht dabei wie folgt aus

```

for %%E in (%NETLOGO_PATH%\*.owl)
do (call %XSLT_PROCESSOR% -o%INSTANCE_PATH%\%%~nE.nlogo
%%E %STYLESHEET_EVENTS% & SET /A RDF_INDEX+=1 &
echo Creating [!RDF_INDEX! of %RDF_COUNT%] %%~nE.nlogo)

```

und kann auch sehr einfach z.B. aus einer Webanwendung heraus realisiert werden. Bei einer Implementierung der Webanwendung der Triplestoreschnittstelle könnte auf diese Art eine Bearbeitung von Ontologie und Instanzen und eine anschließende Generierung der Automatengerüste sowie die weitere Bearbeitung ebendieser direkt in einer Anwendung über ein Webinterface realisiert werden. Eine solche Anwendung wäre ein interessantes Folgeprojekt für die im Rahmen dieser Arbeit vorgestellten Ansätze.

Auch wurden bisher unterstützende Maßnahmen für die Automatenkonfiguration noch nicht in Betracht gezogen, die ob der expliziten und impliziten

Information in den Ontologien und Instanzen durchaus Sinn machen würde. So können Axiome in der Ontologie Einschränkungen für die Automaten-Entwicklung vorgeben, die unmittelbar in das entsprechende Codefragment übernommen werden können, etwa im Fall von Kardinalitäten bei Element-Element-Beziehungen:

```
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasFather"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Vererbungs-Beziehungen, wie im Beispiel angedeutet, sind in sehr vielen Systemen präsent und müssen zumeist auch in der Simulation von Abläufen entsprechend berücksichtigt werden. Wird das Fragment etwa im Rahmen einer System-Simulation zur Entwicklung von genetischen Defekten verwendet, ist es wesentlich, bei der Rückberechnung der Abhängigkeiten diese Einschränkung zu berücksichtigen bzw. keine Inkonsistenzen diesbezüglich entstehen zu lassen. In den Automaten-Code ließe sich das einfach automatisiert übernehmen, in dem keine Multi-Wert-Variablen sondern einfache Abhängigkeitsbeziehungen, wie sie in NetLogo Teil des Funktionenkatalogs sind, verwendet werden (so ermöglicht es der Befehl `hatch`, aus einer bestehenden Agenten-Instanz heraus weitere Agenten zu erzeugen).

Aber auch das Setting der Instanzdaten in Form der Inhalte der RDF-Daten, kann verwendet werden, um zusätzliche Ausgangsbedingungen für den Automaten-code zu erzeugen oder zumindest in Form von Hilfskommentaren für den Entwickler zu notieren. Etwa wenn es darum geht, Wertausprägungen, statistische Streuungen von Werten in der Instanzmenge oder dynamisch berechnete Grenzen für Attributwerte zu kalkulieren.

Kapitel 5

Das Serendipity System

*“Das größte Glück im Leben ist:
ein bisschen Glück.”
Anita Daniel*

Im folgenden Kapitel wollen wir das vorgestellte Modell anhand eines fiktiven Beispiel-Szenarios in eine konkrete Instanz überführen und dabei auch Lösungsansätze für unterschiedliche Konzepte und Problemstellungen aus der Allgemeinen Systemtheorie skizzieren. Wir nennen unsere Beispiel-Instanz “Serendipity System”¹, weil sie uns die Möglichkeit bieten wird, einige der wesentlichsten theoretischen Konzepte auf sehr einfache Art darzustellen, was tatsächlich wie eine Art glücklicher Zufall wirkt, wenn man bedenkt, wie schwierig solche Prozesse in Systemen aus der Praxis oft zu simulieren sind.

Die Motivation, die hinter diesem Abschnitt steckt, ist also in erster Linie eine praxisnahe Veranschaulichung der Arbeit mit dem CGST-Modell. Fragen nach Designschritten, Abhandlungen von Problemszenarien und der Verknüpfung einzelner wissenschaftlicher Untersuchungen mit dem CGST-Modell sollen dabei Schritt für Schritt beantwortet werden. Die gewählten Beispiele sind dabei zwar fiktiv im Sinne von “nicht unmittelbar in der Natur vorhanden”, stellen aber keinesfalls erfundene Prozesse, sondern vielmehr auf das spezielle Beispiel abstrahiert abgebildete gängige Konzepte und Situationen aus Systemen im Allgemeinen bzw. den theoretischen Arbeiten der Allgemeinen Systemtheorie dar.

¹Wir verweisen hier aus Mangel an Alternativen auf die Online-Enzyklopädie Wikipedia: Dem deutschen Eintrag unter <http://de.wikipedia.org/wiki/Serendipity> folgend, beziehen wir uns hier auf die Bedeutung als “Glücklicher Zufall” und nicht auf die zufällige glückliche Beobachtung.

Um das praktische Arbeiten möglichst gut zu veranschaulichen, werden nicht die einzelnen Bausteine für das System-Modell (Ontologie, Regeln, Automat) als einzelne Teile Stück für Stück entworfen und erklärt, sondern vielmehr aus der praktischen Problemsicht heraus das gesamte Modell mit seinen immer komplexer werdenden Eigenschaften aus einer zu Beginn sehr einfachen Ausgangskonfiguration heraus konstruiert. Wesentlich dabei ist auch die generische Abstraktion so gut es geht aufrecht zu halten, es soll also stets der Blick aus unterschiedlichen Reductive Levels erlaubt werden und das System so nicht einer speziellen Ebene zugeordnet werden. In Gedanken und auch in Form von Kommentaren soll stets die Abbildung auf biologische Systeme, soziale Systeme usw. möglich sein.

5.1 Ein einfaches System

Wir beginnen mit den ersten Bausteinen unserer Serendipity Ontologie. Die einfachste mögliche nicht-triviale Konfiguration wäre ein System mit nur einem einzigen Element, nennen wir es Adam. Weil wir aber schon sehr früh auch so etwas wie Wettbewerb in unserem System erzeugen möchten, stellen wir Adam noch ein zweites Element, Eva, hinzu. Adams und Evas können in unserem System geboren werden, leben und sterben – ontologisch benötigen wir dafür eine Eigenschaft `alive`, die anzeigen soll, ob sich eine Instanz von Adam oder Eva noch am Leben befindet oder bereits gestorben ist. Zusätzlich haben Adams und Evas – das wissen wir, nehmen wir jetzt an, aus vorangegangenen empirischen Studien, können oder wollen es an dieser Stelle aber nicht von einer untergeordneten Ebene heraus determinieren – noch eine mehr oder weniger festgelegte maximale Lebensdauer `age` und einen internen Energiehaushalt, der über die Eigenschaft `energy` abgebildet wird. Energie, die im System vorhanden und von Adams und Evas konsumiert werden kann, bezeichnen wir als “Ambrosia”, ohne dieses Konzept jedoch vorerst als Element zu integrieren – für die Anfangsphase ist Energie, die im System vorhanden ist, nur eine einfache Eigenschaft desselben.

$$\Theta = \{(alive, Adam), (alive, Eva), (age, Adam), (age, Eva), (energy, Adam), (energy, Eva)\}$$

Weil wir zu Beginn sehr wenig über unser System wissen, müssen wir uns auf sehr viel zufällig erzeugte Werte und Instanzen verlassen. Das wird sich im Laufe unserer Systemkonstruktion ein wenig zugunsten der determinierten Werte verschieben. Wir überspringen vorerst den statischen Regelteil und erzeugen aus unserer Serendipity-Ontologie einen ersten Serendipity-Automaten, der die drei Eigenschaften der beiden Elemente abbildet. Nun fügen wir folgende dynamische Regeln hinzu:

DR1 **Sterben.** Jede Instanz von Adam und Eva hat eine vordefinierte maximale

Lebensdauer (Zeitstempel). Zum Todeszeitpunkt wird der Wert von `alive` von `true` auf `false` gesetzt.

DR2 Geboren werden. Neue zusätzliche `Adams` und `Evas` werden regelmäßig “vom Storch gebracht” – gehen also “zufällig” aus dem System hervor, vorerst noch ohne weitere Zusammenhänge.

DR3 Essen. Lebende Instanzen von `Adam` und `Eva` müssen durch Nahrungsaufnahme (Konsumation von `Ambrosia`) ihren Energiehaushalt ständig über einem Wert von 0 halten. Schaffen sie das nicht, sterben sie bereits vor Ablauf ihrer Lebenserwartung.

Für unseren ersten praktischen Testlauf müssen wir noch ein Startset und eine Basiskonfiguration für die Instanzierung definieren. Wir sagen dazu, dass die Geburtswahrscheinlichkeit für `Adams` und `Evas` gleich hoch ist (0.5) und dass vorerst beide, `Adams` und `Evas`, Lebenserwartungen von durchschnittlich 80 Jahren (=Ticks) besitzen, folgend einer randomisierten Verteilung mit einer maximalen Abweichung von 16 Jahren. Außerdem wird das System vorerst so konfiguriert, dass immer ausreichend Ressourcen zur Verfügung stehen, um den Energiebedarf der `Adams` und `Evas` aufrecht zu erhalten – insbesondere gibt es hier vorerst keinerlei Wechselwirkungen, sondern nur eine Konsumation der `Adams` und `Evas` des `Ambrosias` zum Auffüllen des internen Energiebedarfs.

Simulation 1 – Leben und Sterben. Bei Testläufen jeweils über 500 Jahre stellen wir jetzt fest, dass unser System sich bei einer Anfangspopulation von 100 `Adams` und 100 `Evas` nach einer sehr schnell erreichten Bevölkerungsspitze bei ca. 80 Personen, davon jeweils ungefähr die Hälfte `Adams` und `Evas`, “einschwingt”, wie Abbildung 5.1 zeigt.

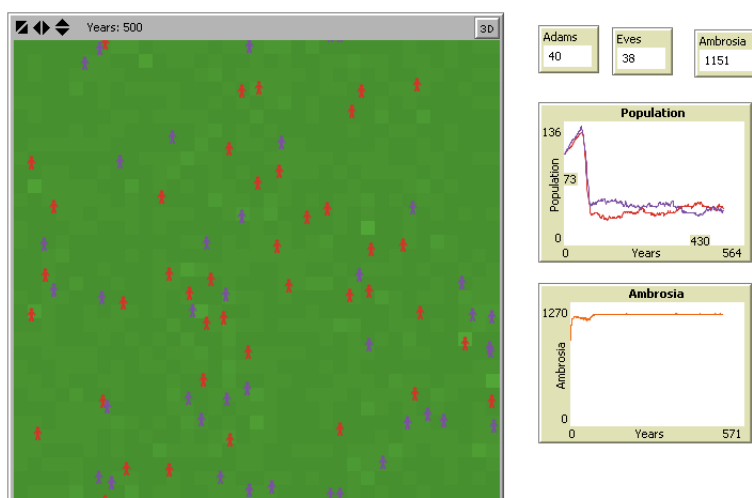


Abbildung 5.1: Adam und Eva im fairen Wettbewerb zueinander.

Das ist ein unspektakuläres, weil zu erwartendes Resultat². Bei einer unendlichen Anzahl an Testläufen ergibt sich eine zu jedem Zeitpunkt t_x gleiche Verteilung von **Adams** und **Evas**. Genau genommen haben wir bisher überhaupt nur eine willkürliche begriffliche Differenzierung dieser beiden Elemente ohne jegliche Unterscheidung auf der Eigenschaftsebene vorgenommen.

Nun vollziehen wir den ersten Schritt weg von einer rein begrifflichen Differenzierung: Wir verleihen den beiden Elementen nun unterschiedliche Ausprägungen ihrer Eigenschaften, sodass sich auch eine spürbare Auswirkung im Gleichgewicht des Systems ergeben sollte. Dazu definieren wir, dass **Evas** eine grundsätzlich höhere Lebenserwartung von durchschnittlich 85 Jahren haben und dass die “Geburtswahrscheinlichkeit”, hier also noch die Wahrscheinlichkeit, mit der der Storch **Adam** oder **Eva** bringt, nicht mehr gleich hoch ist, sondern ab sofort mit $p(\text{Adam}) = 0.4$ und $p(\text{Eva}) = 0.6$ definiert ist.

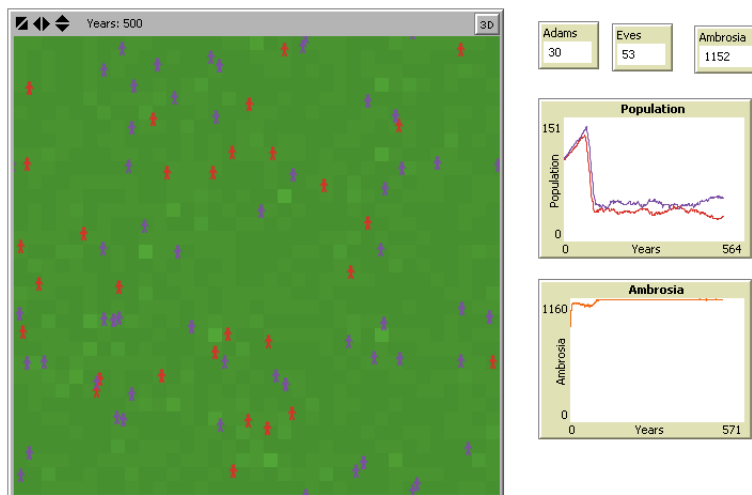


Abbildung 5.2: **Eva** dominiert **Adam** durch höhere Lebenserwartung und Geburtswahrscheinlichkeit.

Bei dieser Konfiguration³ schwingt sich das System ebenfalls nach einer kurzfristigen Spitze ein, allerdings in einem Verhältnis **Adams:Evas**, wie Abbildung 5.2 zeigt. Unser nächstes Ziel ist es jetzt, **Adam** aussterben zu lassen, um in weiterer Folge Strategien untersuchen zu können, mit denen **Adam** das verhindern kann. Dazu müssen wir allerdings zuerst “echte” Fortpflanzung einführen, denn solange ein geheimnisvoller Storch mit einer fest gelegten, wenn auch noch so geringen, Wahrscheinlichkeit neue **Adams** bringt, wird uns das nicht gelingen. Wir ändern hierzu also ab:

DR2 Geboren werden. **Adams** und **Evas** können sich mit einer vordefinierten Wahrscheinlichkeit zu jedem Zeitpunkt, zu dem sie lebendig sind, selbst

²Siehe `./serendipity_adam-and-eve_fair.nlogo`

³Siehe `./serendipity_adam-and-eve_eve-rules.nlogo`

reproduzieren.

Bei gleichbleibenden Eigenschaften zum letzten Experiment geben wir **Eva** nun eine Geburtsquote von 1:60 (d.h. in jedem Jahr ist die Wahrscheinlichkeit, dass eine **Eva** eine neue **Eva** erzeugt, $p(Eva) = 0.017$) und **Adam** eine Geburtsquote von 1:150 ($p(Adam) = 0.007$). **Evas** leben nun also länger und sind auch geburtsfreudiger. Zur formalen Repräsentation wird die Serendipity-Ontologie dabei noch um ein Attribut **fertility** erweitert, dass eben jene Wahrscheinlichkeit enthält.

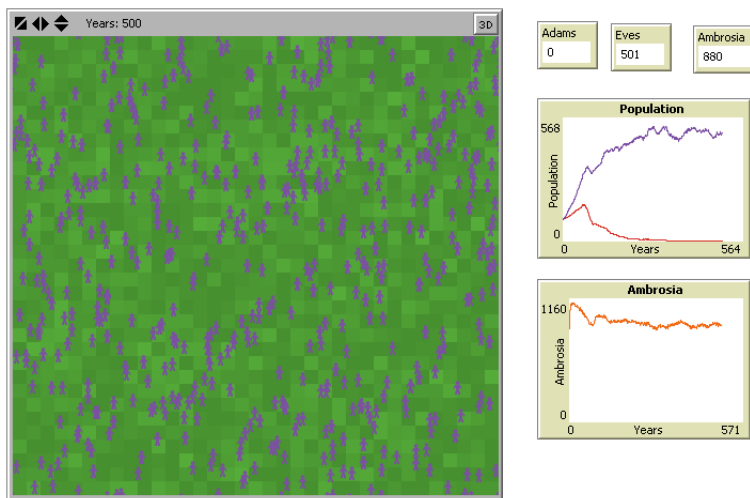


Abbildung 5.3: Nach Einführung der Selbstreproduktion stirbt **Adam** aus.

Das ist zu viel für den armen **Adam**⁴: Nach etwa zwei Drittel der experimentellen Laufzeit stirbt er aus, wie Abbildung 5.3 zeigt. **Eva** hingegen kann ihre Population ständig ausbauen – zumindest so lange, bis sie an die Grenzen der zur Verfügung stehenden Ressourcen stößt. Wir wollen nun versuchen, Konzepte einzuführen, mit deren Hilfe **Adam** sich trotz prinzipieller Unterlegenheit das Überleben sichern kann.

Simulation 2 – Mutation und Selektion. Für einen außenstehenden Betrachter ergibt sich in der Natur das Bild, dass bestimmte Arten sich gut und andere schlecht an die Gegebenheiten ihrer Umwelt anpassen und so entweder bestehen und sich vermehren können oder aber aussterben. Diese Fähigkeit zur Anpassung und ihre Auswirkungen benannte Darwin mit “survival of the fittest”, was oft fälschlicherweise als “Überleben des Stärksten” interpretiert wird. Tatsächlich spielt der Grad der Anpassung an gegebene Umstände eine viel wichtigere Rolle, als die Entwicklung besonderer Stärke. Darwins Konzepte wurden im Laufe der letzten 150 Jahre ständig weiter entwickelt und heute befasst sich die theoretische Biologie im Rahmen der “Evolutionary Development”-Theorie (kurz: Evo-Devo) mit diesen Fragestellungen. Darwins Thesen wurden

⁴Siehe `./serendipity_adam-and-eve_adams-die-off.nlogo`

dabei zwar als noch nicht hinreichend erkannt, um etwa viele Entwicklungsszenarien in der Entwicklung neuer Arten zu erklären, wohl aber sind die wesentlichen Konzepte der Mutation und Selektion auch heute notwendiger und fester Bestandteil in der Erklärung ebendieser.

Mutation und Selektion sind dabei Begriffe aus der Biologie, die aber – wenn auch zumeist unter anderen Rahmenbedingungen – bei Entwicklungen in offenen Nichtgleichgewichtssystemen sehr oft eine wesentliche Rolle spielen. Zwar ist der Begriff des “Sozialdarwinismus” zurecht wissenschaftlich umstritten und im gesellschaftlichen Diskurs oft negativ belegt, doch gibt es zweifelsohne auch in sozialen Prozessen Mutationen in Form von Innovationen in der Technik, in wirtschaftlichen Subsystemen, in gesellschaftlichen Moralvorstellungen, etc. und Selektionen in Form von wettbewerbsgesteuerten Auswahlverfahren dieser abgeänderten Teilsysteme. Findet in einem prinzipiell ausgeglichenen, eingeschwungenem Marktsegment eine “spontane” Innovation statt, kann diese zu einer nachhaltigen Vormachtsstellung des einführenden Unternehmens oder gar zu einer vollständigen Verdrängung der Mitbewerber vom Markt führen. Nicht nur “klassische” GST-Autoren im Wirtschaftssegment wie Kenneth Boulding⁵, sondern auch bekanntere Ökonomen wie Joseph Schumpeter⁶ haben sich mit derartigen Prozessen und Vorgängen eingehender beschäftigt und sind dabei zu solche Szenarien stützende theoretischen Modellen gelangt.

Als Beispiel aus der Praxis sei hier der Umbruchprozess im Segment der Kühlsysteme erwähnt: Bis in die zweite Hälfte des 19. Jahrhunderts hinein war der Markt für Eisblöcke einer der größten Märkte in den USA – zur Konservierung von Lebensmitteln wurden damals noch täglich oder wöchentlich gelieferte Eisblöcke aus der näheren und weiteren Umgebung verwendet. Obwohl die einzelnen Unternehmen stark auf Innovationen setzten und immer bessere Methoden zum Transport der Eisblöcke entwickelten, wurde die aufkommende Erfindung elektrischer Kühlsysteme von allen großen Eis-Unternehmen ignoriert. Innerhalb weniger Jahrzehnte kam es so durch den Erfolg der Kühlschrankfirmen zu einer völligen Ersetzung der Unternehmenslandschaft in diesem Segment, keiner der davor bestehenden großen Industriekonzerne überlebte diese Innovation wirtschaftlich⁷.

Allgemein lässt sich in der Regel feststellen: Eine Neuerung, die durch eine Mutation entsteht, entsteht nicht deshalb, weil sie erfolgreich sein wird, sondern wird als erfolgreich betrachtet, wenn sie überlebt und sich gegenüber anderen Varianten durchsetzt. Die Mutation selbst ist dabei zwar durch die sie verursachenden Effekte determiniert, aber keinesfalls zielgerichtet. Würde et-

⁵Siehe in [11] insbesondere die Kapitel 5, 6 und 7, in denen Boulding die biologische und die soziale Evolution anhand ihrer Prozesse und Meilensteine vergleicht.

⁶Etwa in Joseph Schumpeter, “Theorie der wirtschaftlichen Entwicklung”, 1911.

⁷Siehe Massachusetts Historical Society (ed.), “The Ice King: Frederic Tudor and His Circle”, Boston 2003.

wa durch eine Mutation die Fruchtbarkeit von **Adam** plötzlich signifikant erhöht werden, könnte sich dieser bei gleichbleibenden äußeren Bedingungen plötzlich gegenüber **Eva** durchsetzen und diese sogar dominieren und verdrängen. Unter anderen “Umständen” würde eine höhere Fruchtbarkeit aber sogar zu einem noch stärkeren Misserfolg führen. Das ist ein sehr einfaches Beispiel in unserem Serendipity-System, wie eine Mutation auf der Ebene eines einzelnen Element-Attributes das Systemverhalten maßgeblich beeinflussen kann.

Jetzt stellt sich natürlich folgende Frage: CGST ist ein formales System, dessen Verhalten durch die konfigurierten Parameter determiniert ist und zudem operational geschlossen ist – welchen Sinn macht es da, den Begriff der Mutation abzubilden? Nun, zum einen ist es bei der Definition von Abhängigkeiten im System nicht mehr absehbar, welche Wechselwirkungen sich zu einem bestimmten Simulationszeitpunkt in einer derartigen Form ausprägen. Zum anderen gibt es tatsächlich eine mittelfristig interessante Spielvariante im CGST-System, deren Untersuchung in dieser Arbeit aber hier nur angedeutet werden kann: Die Semantik der Relationen in der Ontologie, aber auch die Auswirkungen der Element-Attribute werden zu einem wesentlichen Teil erst durch die Programmierung im Zellulären Automaten bestimmt. Diese wird aber prinzipiell wie bei jeder Programmiersprache durch eine – hier sehr einfach strukturierte – Syntax repräsentiert. Die einzelnen Programmfragmente lassen sich dabei, im Gegensatz zu komplexeren objektorientierten Programmiersprachen, leicht entfernen oder hinzufügen. Das Programm-Fragment zur Reproduktionsfunktion einer Adam-Zelle sieht etwa wie folgt aus:

```
to reproduce-adam
  if random 150 = 1
    [
      hatch-adams 1
      [ set color red
        set alive? true
        set age 80 + (random 16) - (random 16)
        set energy 100
        setxy random-xcor random-ycor
      ]
    ]
end
```

Wäre der Random-Parameter in der zweiten Zeile niedriger (siehe unten) oder der Basiswert für das Alter höher, würden die Chancen für Adam, sich durchzusetzen, ab einer bestimmten Grenze signifikant steigen. Bei einer entsprechenden Code-Änderung kehrt sich das Systemverhalten um und **Adam** verdrängt nun **Eva** aus dem System, wie Abbildung 5.4 zeigt.

```
to reproduce-adam
  if random 30 = 1
    [
      hatch-adams 1
```

```

    [ set color red
      set alive? true
      set age 80 + (random 16) - (random 16)
      set energy 100
      setxy random-xcor random-ycor
    ]
  ]
end

```

Effektivere Mutationen werden wir uns gleich noch ansehen. Interessant ist jetzt aber folgender Ansatz: Durch eine Markierung von bestimmten Stellen im Code kann ein sinnvoller Rahmen für spontane Veränderungen, die von einem Code-Rekonfigurator randomisiert erzeugt werden, vorgegeben werden. Evaluations-Algorithmen können dann in weiterer Folge jene Rekonfigurationen durch Simulations-Testläufe ermitteln, die zu einem interessanten Systemverhalten geführt haben. Die Konzeption und Implementierung eines derartigen Systems drängt sich als Weiterführung der in dieser Arbeit vorgestellten Ansätze auf.

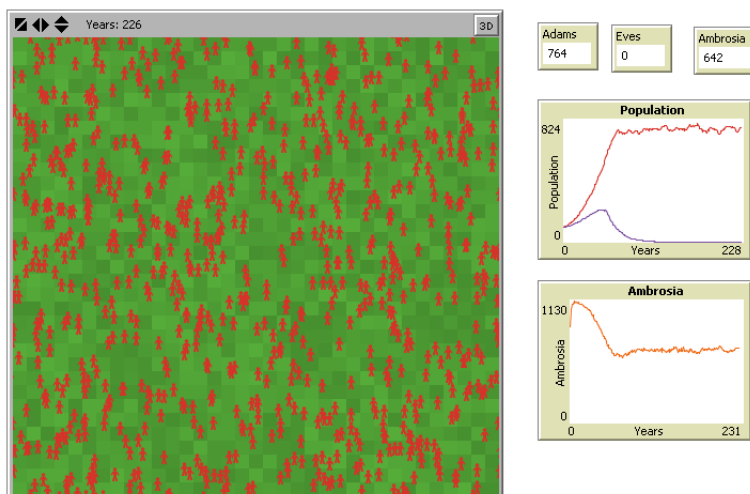


Abbildung 5.4: Eine Mutation in der Fruchtbarkeitsrate von Adam reicht aus, um die Situation umzukehren – nun wird Eva von Adam verdrängt.

In einem langfristig zu verwirklichenden System wäre es möglich, die Vorgabe eines Rahmens stark zu reduzieren, sodass alle möglichen syntaktisch sinnvollen Änderungen im Code durchgeführt werden können und so der Möglichkeitsraum für interessante Mutationskombinationen und die daraus entstehenden Resultate noch einmal vergrößert wird. Die große Herausforderung bei einem solchen Ansatz ist dann aber die effiziente Handhabung in der Algorithmik, da hier eine wesentlich größere Rechenleistung bei der Auswertung der Simulationstestläufe notwendig wäre. Ein Ergebnis, zu dem eine Abfolge von derartigen Änderungen führen könnte, soll im nächsten Szenario behandelt werden.

Simulation 3 – Kooperation. Da wir hier nun komplexere Konzepte behandeln, müssen wir einige unserer Systemparameter modifizieren, um längere Beobachtungszeiträume zu generieren. Den Beobachtungszeitraum selbst erweitern wir auf 2000 Jahre. Zuerst ändern wir die Ursache des Aussterbens von **Adam** ab. In dem vorangegangenen Beispiel hatte er aufgrund der geringen Geburtsrate keine Chance, langfristig zu überleben. Nun wollen wir mit $p(\text{Adam}) = 0.013$ diese Chance an einem Grenzwert aufrecht erhalten. Gleichzeitig bekommt auch **Eva** einen Fruchtbarkeitsschub: $p(\text{Eva}) = 0.024$. In den Testläufen stirbt **Adam** hier trotz allem aus, weil er im Wettbewerb mit **Eva** um die zu Verfügung stehenden Ressourcen irgendwann den Kürzeren zieht. In Summe kann das System nur eine bestimmte Anzahl an **Adams** und **Evas** erhalten, weil **Eva** aber immer stärker dominiert, verdrängt sie **Adam** mit der Zeit aus dem System. Der wichtige Unterschied ist hier aber jetzt, dass **Adam** ohne **Eva** sehrwohl überleben und sich mit der Zeit bis zum Ausschöpfen der Ressourcen vermehren würde.

Adam muss sich also etwas einfallen lassen. Anders gesagt: Durch eine Mutation (im Code von **Adam**) kann **Adam** eine neue Eigenschaft entwickeln, die ihm in irgendeiner Form einen Vorteil bringt. Dazu lassen wir **Adam** tun, was Moleküle, Zellen, Tiere und Menschen tun, um dadurch ihre Eigenschaften zu verändern und neue Eigenschaften auf einer übergeordneten Ebene zu entwickeln (natürlich ohne teleologischer Absichten und Abläufe): Er soll kooperieren, indem er sich mit anderen **Adams** zusammen schließt.

Wir nehmen an, dass sich jede **Adam**-Instanz mit einer Wahrscheinlichkeit von $p = 0.3$ mit einer anderen **Adam**-Instanz, die ihr begegnet, anfreundet. Wir fügen dazu in unserer Serendipity-Ontologie eine Objekt-Relation `isFriendOf` ein, die als systemische Eigenschaft und als Element-zu-Element-Beziehung grundsätzlich zwischen **Adams** und **Adams**, **Evas** und **Evas** und **Adams** und **Evas** bestehen kann. Weil **Adams** mit vielen Freunden Arbeitsteilung vornehmen können, sind sie bei der Abarbeitung der systemischen Ressourcen effizienter. Das bilden wir in unserem Modell jetzt so ab, dass wir eine neue Eigenschaft `power` einfügen, die kennzeichnet, wie gut **Adams** und **Evas** in der Lage sind, Ressourcen abzuarbeiten, d.h. **Ambrosia** aufzunehmen, wenn es vorhanden ist. Wir definieren `power` als Wahrscheinlichkeitswert, der bisher immer 1 war, sich nun aber – für **Adams** und **Evas** gleichermaßen – zwischen 0.2 und 0.4 bewegt.

Für jeden Freund, den **Adam** findet, steigt diese Wahrscheinlichkeit allerdings um 10%, d.h. bei acht oder mehr Freunden ist **Adam** bereits maximal effizient bei der Ressourcenabarbeitung. Dieser Effekt ist, wie er hier modelliert ist, natürlich etwas unmotiviert und entbehrt zunächst noch einer tieferen Argumentation. Diese liese sich aber konstruieren, in dem man etwa die Abarbeitung der Ressourcen auf Teiltätigkeiten aufsplittet und dann Spezialisierungen einführt. D.h. wir hätten zu Beginn folgende Wertetabelle

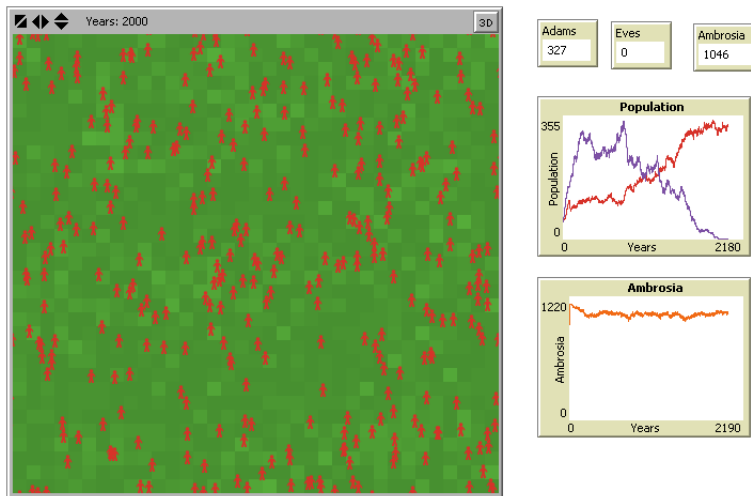


Abbildung 5.5: Adam erhöht seine Chancen durch Kooperation.

Element	Collect	Carry	Prepare
Adam	0.2	0.2	0.2
Eve	0.2	0.2	0.2

die die Fähigkeit zur spezialisierten Arbeit für Sammel-, Transport- und Herstellungstätigkeiten von Adam und Eva zusammen fasst. Für alle Tätigkeiten ist die Effizienz mit 20% sehr beschränkt ausgeprägt, weil diese Generalistenkonfiguration zu Lasten einer potentiellen Spezialisierung geht. Bilden sich Gruppen von Adams, ist eine Ausdifferenzierung möglich, in denen einzelne Instanzen sich auf eine der Teiltätigkeiten spezialisieren und so nicht nur eine weit höhere Einzelleistung erbringen sondern insgesamt die Effizienz in der Gruppe im Vergleich zu früher, aber insbesondere auch im Vergleich zu Eva-Gruppen steigern.

Element	Collect	Carry	Prepare
Adam (F)	0.9	0	0
Adam (C)	0	0.9	0
Adam (P)	0	0	0.9
Eve	0.2	0.2	0.2

Diese Ausdifferenzierung wird in manchen Gesellschaftstheorien tatsächlich auch als maßgeblicher Grund für die Entwicklung komplexer Gesellschaften und deren Erfolg im Vergleich zu einfachen Stammesgruppen beschrieben⁸. In der Zoologie findet eine solche Ausdifferenzierung auf physischer Ebene in Insektenstaaten statt, die durch Spezialisten eine höhere Effizienz erreichen. Diese Prozesse werden heute bereits in der Informatik nachgebaut, tatsächlich findet hier eine Art Schritt-für-Schritt Adaption der evolutionären Entwicklungen statt. So beschäftigen sich jene Teile der Informatik, die die so genannte

⁸Siehe dazu [31] sowie Hannes Wimmer, "Evolution der Politik – Von der Stammesgesellschaft zur modernen Demokratie", Wien 1995.

Schwarm-Intelligenz als Modell für Informationssysteme implementieren, nicht mehr wie zu Beginn ausschließlich mit homogenen Agentensystemen, sondern bilden eine Ausdifferenzierung der Agenten in den Informationssystemen nach⁹.

Insgesamt stehen derartige Resultate jedoch in aktuellen Diskursen genauso zur Diskussion wie die Mutations- und Selektionsketten und Kooperationsmodelle in der Theoretischen Biologie bzw. in ihren jeweiligen Adaptionen in der Theorie Sozialer Systeme. Dabei werden zumeist nicht die Konzepte oder prinzipiellen Muster, die damit in Verbindung gebracht werden, angezweifelt, sondern vielmehr deren Rolle in gesamtheitlichen Entwicklungsprozessen, etwa wenn es um die Entwicklungsschritte von einzelligen Organismen zu höheren Lebewesen oder von Stammesgruppen zu komplexen Gesellschaften geht.

5.2 Ein komplexes System

Das einfache System, wie wir es bisher beschrieben haben, hat es trotz seines überschaubaren Regelsatzes in sich. Obgleich das anhand der Modellierung nicht sofort klar ist, handelt es sich auch bisher schon prinzipiell um ein offenes System: Schuld daran ist die “magische” Regeneration des Ressourcenkonzepts Ambrosia sowie das spukhafte Verschwinden der inaktiven Adam und Eva Elemente im System. Diese Austauschprozesse mit der “Umwelt” geben dem System auch seine Dynamik. Wir werden diesen Effekt jetzt ausnutzen und das System insofern erweitern, als dass wir stärker verschachtelte Abhängigkeiten einbauen und den offenen Charakter des Systems stärker und transparenter herausarbeiten.

Wir modellieren dabei das folgende Szenario: Ambrosia wird von einer reinen Systemeigenschaft zum Element erhoben. Das ist notwendig, um diesem Konzept auch eine entsprechende Abbildung auf statischer Ebene zu ermöglichen, die Behandlung bisher als nicht näher definierbare Ressource wurde nur aufgrund der einfachen Veranschaulichung des ersten Beispielszenarios so gewählt. Ambrosia nimmt nun auch die Stellung als “Schnittstelle” zur Umwelt ein, Adam ist damit nicht mehr ein Konzept, das aus dem Nichts entsteht, sondern geht – um hier einen einfachen Transformationsprozess zu integrieren – direkt aus Ambrosia hervor. Hier wird also eine Art Urmasse-Ausbildung modelliert.

Ein weiteres wichtiges systemisches Konzept, das wir integrieren möchten, ist jenes der Synthese, also der Verschmelzung von zwei oder mehreren Elementen. Hier bilden wir zunächst eine Dreierbeziehung aus, d.h. wenn sich zwei Adams und eine Eva finden, kann sich Adam in dieser Konstellation vermehren:

⁹Siehe dazu Kapitel 3.5 “Differentiation in a Multiagent System and Adaptive Task Allocation” in [8].

(S2.1) $Adam + Adam + Eva \rightarrow Adam + Adam + Adam$

Warum wir gerade diese seltsame Art der Gruppenbildung benötigen, werden wir später noch ausführlicher erklären. Um die Differenzierung zwischen **Adam** und **Eva** auszubauen, führen wir außerdem einen zweiten Grundstoff, **Nektar**, ein. Konsumiert **Adam** Nektar, so entstehen dadurch **Eva** und ein Abfallprodukt, das wir als **Gas** bezeichnen. Stirbt **Adam**, entsteht ebenfalls ein Abfallprodukt, das wir als **Masse** bezeichnen. Gas und Masse verflüchtigen sich und scheiden aus dem System wieder aus – eine weitere Schnittstelle zur Umwelt also, nur in die andere Richtung. Die restlichen Beziehungen formal notiert sehen dann wie folgt aus:

(S2.2) $Ambrosia \rightarrow Adam$

(S2.3) $Adam + Nektar \rightarrow Eva + Gas$

(S2.4) $Adam \rightarrow Masse$

So wie Gas und Masse sofort aus dem System ausgefällt werden, werden dem System aus seiner Umwelt auch ständig Nektar und Ambrosia nachgereicht. Vergleichen kann man so einen Kreislauf etwa mit einem Ausschnitt eines Ökosystems, in dem jetzt die Rückführung von Masse und Gas in Nektar und Ambrosia ausgeblendet wird, weil man nie alles auf einmal beobachten oder modellieren kann, sondern immer irgendwo einen Schnitt zwischen System und Umwelt vornehmen muss. Die einzelnen Zustandsübergänge bedingen außerdem eine “physische” Nachbarschaft der benötigten Elemente sowie Übergangswahrscheinlichkeiten (bzw. Reaktionszeiten).

An dieser Stelle wird es Zeit, einmal einen Blick auf die Ontologie zu werfen (Abbildung 5.6), da diese nun nicht mehr wie in unserem ersten Beispiel nur die notwendigsten Differenzierungen enthält.

Serendipity ist dabei ein System-Konzept in einem eigenen Namensraum <http://www.systems-everywhere.com/serendipity> mit einer Reihe von Elementen **Adam**, **Eve**, **Ambrosia**, **Nektar**, **Masse** und **Gas**. Um die Elemente domainunabhängig dem Serendipity-System zuzuordnen wird eine Objektrelation **serendipity:hasElement** als Sub-Property von **system:hasElement** im Namensraum der Serendipity-Ontologie definiert. Diese Definition ist nicht unbedingt notwendig bei der Erstellung von CGST-System-Ontologien, führt aber zu Vorteilen bei der späteren Verarbeitung von Instanzen in Triplestores. Sämtliche zuvor definierten atomaren Attribute der Serendipity-Elemente wie **age**, etc. werden als Sub-Properties des generischen **system:attribute** Datatype-Property definiert und als **owl:FunctionalProperty** deklariert, um sie eindeutig in das NetLogo-Gerüst überführen zu können.

Die Abhängigkeiten, die in S2.1 - S2.4 beschrieben sind, werden dabei in

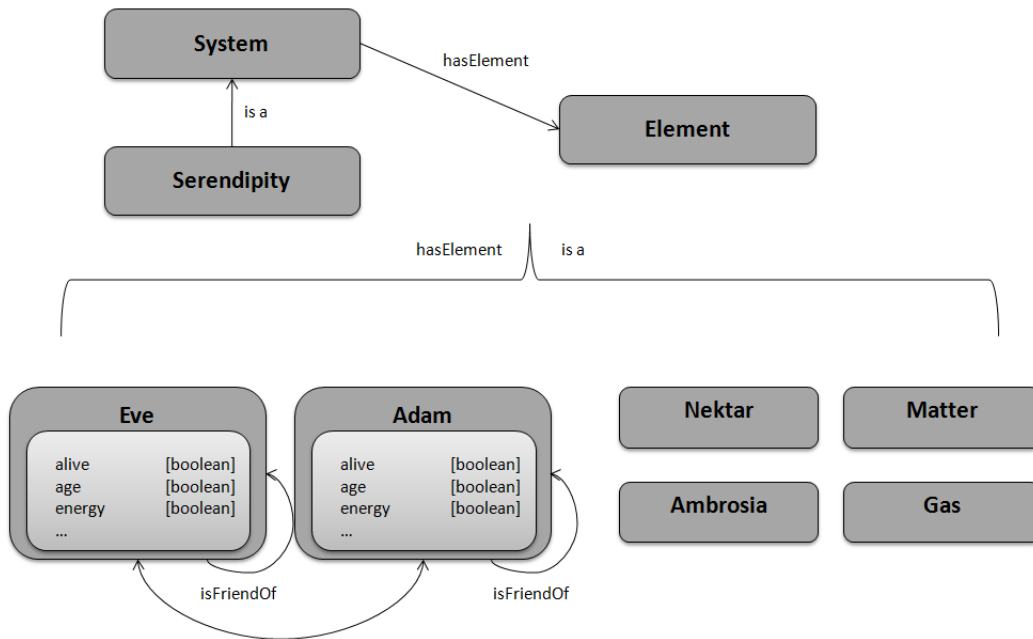


Abbildung 5.6: Die CGST-basierte Serendipity-Ontologie in einem reduzierten Ontologie-Schema.

der aktuellen Ontologie noch gar nicht berücksichtigt. Tatsächlich wäre es von Vorteil, bereits an dieser Stelle entsprechende Objektrelationen zu definieren, die für den nächsten Schritt, die Überführung in ein entsprechendes Automaten-system und die Konfiguration desselben, hilfreich wären. Gemeint ist hier also, prinzipielle Abhängigkeiten zwischen den Elementen, wie sie in den Übergangsregeln definiert werden, auch tatsächlich bereits auf ontologischer Ebene einzuziehen und auch zu benennen. Dafür sollen an dieser Stelle die folgenden Bezeichnungen eingeführt werden, die in Abbildung 5.6 aus Übersichtsgründen nicht nachgezogen werden:

$$\Theta_{Serendipity} = \{affiliates(Adam, Eve), affiliates(Adam, Adam), produces(Ambrosia, Adam), affiliates(Adam, Nektar), produces(Adam, Eve), produces(Nektar, Eve), produces(Adam, Gas), produces(Nektar, Gas), produces(Adam, Matter)\}$$

An der Definition von $\Theta_{Serendipity}$ lässt sich gut erkennen, dass die Objektrelationen nur prinzipielle Domain-Range-Abhängigkeiten, jedoch keine konkreten Kombinationseinschränkungen definieren. D.h. nur aus der Definition der Objektrelationen lässt sich z.B. nicht ableiten, dass eine **Adam**-Instanz gemeinsam mit einer **Nektar**-Instanz in je eine **Eva**-Instanz und eine **Gas**-Instanz übergehen. Solche Einschränkungen könnten noch in zusätzlichen Axiomen de-

finiert werden bzw. sollten bei einer detaillierteren Definition generell erweiterte Modellierungskonzepte zur Abbildung derartiger Axiome gewählt werden, indem etwa ein abstraktes Hilfskonzept **Transformation** eingeführt wird, das auf **TInput** und **TOutput** referenziert und dann an dieser Stelle jeweils Kardinalitäten für die einzelnen Elemente beinhaltet. Eine derart detailreiche Durchexekution des Anschauungsbeispiels soll hier aber jetzt ausgespart werden, wesentlicher ist an dieser Stelle ja jetzt die Konfiguration des Automaten.

Die Grundkonfigurationen sind aus den Beispielen aus Kapitel 5.1 bereits bekannt, allerdings müssen Teile davon aufgrund der geänderten Abbildung der Input- und Output-Stoffe adaptiert werden. Insbesondere betrifft das die zu Beginn angesprochene Umgestaltung zu einem “echten” offenen System, in dem der offene Charakter nicht nur auf der Attributebene simuliert wird, sondern über die eingeführten Input- und Output-Elemente per Definition entsteht.

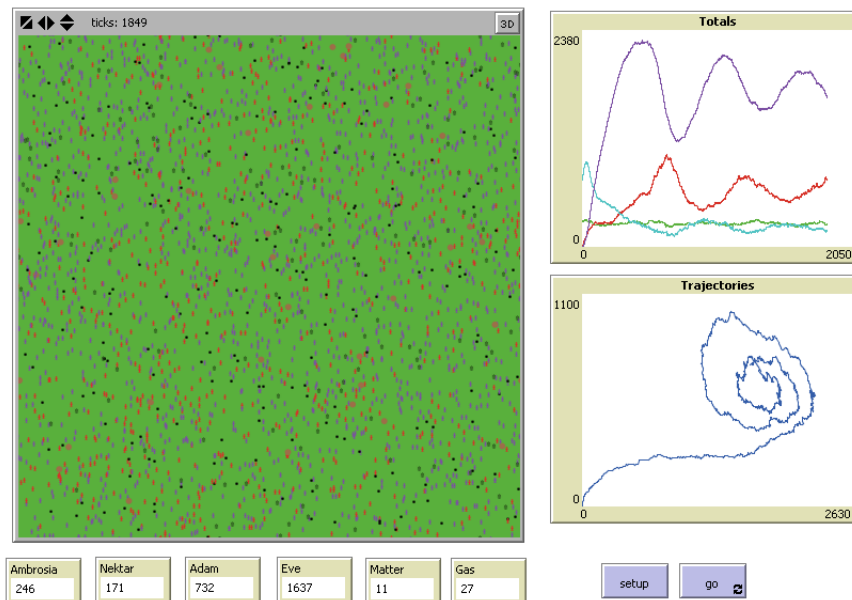


Abbildung 5.7: Die komplexen Zusammenhänge unter den Serendipity Element-Instanzen führen zu einem komplexen Verhalten im System in Form einer Lotka-Volterra-Funktion.

Auf der Ebene der NetLogo-Implementierung müssen jetzt neben den Änderungen, die zu einem offenen System führen, noch die Transformationen selbst realisiert werden. Diese können prinzipiell nur stattfinden, wenn eine “physische Nähe” existiert, d.h. wenn sich zwei oder mehrere Instanzen auf den selben Koordinaten in der Lokalisierungsmatrix befinden. Um das System besser zu steuern bzw. ein nicht-triviales Verhalten zu simulieren, werden für die Transformationen außerdem folgende Übergangswahrscheinlichkeiten definiert: $p(S2.1) = 0.02$, $p(S2.2) = 1.0$, $p(S2.3) = 1.0$ und $p(S2.4) = 0.02$; d.h. die Primär-Element-Bildungen aus den Input-Stoffen wie auch der Übergang zu

den Output-Stoffen kommt relativ selten vor, während die Umwandlungen unter den Primär-Elementen jeweils mit Sicherheit erfolgen, sobald die physische Nähe gegeben ist.

Für das Verhältnis von Adam und Eva sollen jetzt hier – da bekannt ist, dass es sich um ein offenes System handelt – nicht nur der direkte Verlauf sondern auch die Trajektorie dieses Verhältnisses betrachtet werden, was im Ergebnis der Simulation in Abbildung 5.7 veranschaulicht wird. Die Trajektorie entspricht dabei dem Bild eines Räuber-Beute-Schematas, was angesichts der Abhängigkeiten in den Transformationsregeln auch nicht anders zu erwarten war: Zwischen den Input-Elementen, den beiden Akteuren im System und den Output-Elementen besteht ein Räuber-Beute-Verhältnis, das die Ausprägung der Ressourcen und die Entwicklung im System steuert. Derartige Trajektorien, die auch über ein Lotka-Volterra-Gleichungssystem definiert werden können, waren bereits im Bertalanffy-Programm Inhalt zahlreicher Untersuchungen, da sie das Verhalten von Systemen im Nicht-Gleichgewichtsbereich beschreiben.

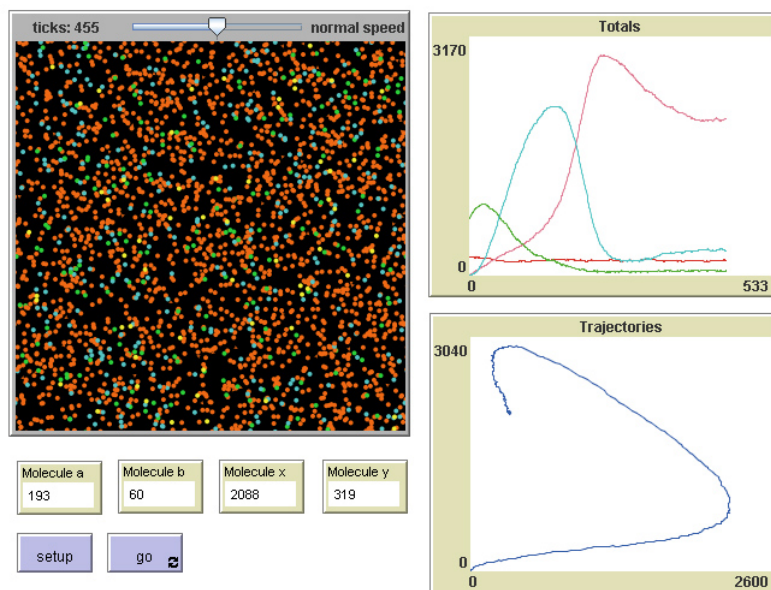
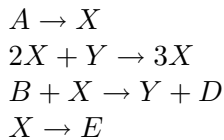


Abbildung 5.8: Prigogines Brüsselator.

Der oben konstruierte Fall ist dabei bewusst auf ein ganz spezielles System, den so genannten Brüsselator, hin konstruiert worden. Dieses einfache Fallbeispiel eines (ursprünglich chemischen) Systems im Nichtgleichgewichtszustand wurde von Ilya Prigogine zur Veranschaulichung seiner Arbeiten im Bereich der Nichtgleichgewichts-Thermodynamik verwendet, für die er 1977 den Nobelpreis für Chemie erhielt. Prigogine beschäftigte sich dabei ursprünglich mit einem anderen chemischen System, für dessen praktisches Verhalten es zuvor noch keine ausreichende theoretische Erklärung gab: So genannte Belousov-Zhabotinsky-Reaktionen, chemischer Oszillatoren, auf die wir im nächsten Kapitel noch eingehen werden. Im Rahmen dieser Arbeit wurde auch ein NetLogo-Modell (siehe

Abbildung 5.8) für Prigogines ursprüngliches System implementiert, das sich im Anhang findet¹⁰.

Die wesentlichen Konfigurationseinstellungen betreffen dabei in erster Linie die Relationen in den Übergangsfunktionen, im Detail ist dabei für den Brüsselator als Spezialfall eines Lotka-Volterra-Systems definiert



wobei auch die Übergangswahrscheinlichkeiten sich in einem bestimmten Rahmen befinden müssen, da sonst das Verhältnis der Primär-Element-Erzeugungen und der Umwandlungen zu einem Stillstand im System oder der Verdrängung eines Elements führen kann. Wie oben zu sehen ist, müssen die Primär-Element-Umwandlungen dabei verhältnismäßig rasch im Vergleich zu den Umwelt-Austausch-Prozessen geschehen. An dieser Stelle kann auch die Frage nach der Wahl der Ausgangskonfiguration für die Menge der Umwelt-Ressourcen beantwortet werden: In den Arbeiten Prigogines findet sich nämlich auch das Resultat, dass der Brüsselator sich für alle Ausgangskonfigurationen, für die

$$B > 1 + A^2$$

gilt, in den auch in Abbildung 5.7 abgebildeten Nichtgleichgewichtszustand übergeht. Dieser Zustand ist dann prinzipiell sehr "stabil", d.h. kann nur schwer durch spontane Störungen von außen beendet werden (etwa durch eine kurzzeitige Erhöhung eines Teils der Umwelt-Ressourcen oder ein verstärktes Ausfällen eines Primärelements).

5.3 CGST-Ähnlichkeits-Algorithmik am Beispiel von Lotka-Volterra-Systemen

Räuber-Beute-Schemata sind weit verbreitet und finden sich nicht nur auf chemischer Ebene, sondern – daher der Name – insbesondere auch in Ökosystemen mit Primär- und Sekundär-Konsumenten sowie in Wirtschaftssystemen. Auch ist die Simulation mittels Zellulärer Automaten für solche Systeme nichts Neues, da eine solche Implementierung, das Wator-Programm, bereits 1984 von Alexander Dewdney und David Wiseman vorgestellt wurde¹¹. Im Wator-Programm

¹⁰Im Rahmen der Implementierung erfolgte auch eine detaillierte Herleitung der Abbildung in Zellulären Automaten auf Basis der ursprünglichen Resultate von Prigogine, die an dieser Stelle ausgespart wurde, allerdings in der Online-Dokumentation nachgelesen werden kann: Martin Schliefnig, "Computational Approaches for Modelling Complex Systems", July 2008.

¹¹Siehe dazu "Scientific American", Dezemberausgabe 1984.

wurde ein Räuber-Beute-Szenario in mariner Umgebung mit Fischen als Beute- und Haien als Raubtieren abgebildet.

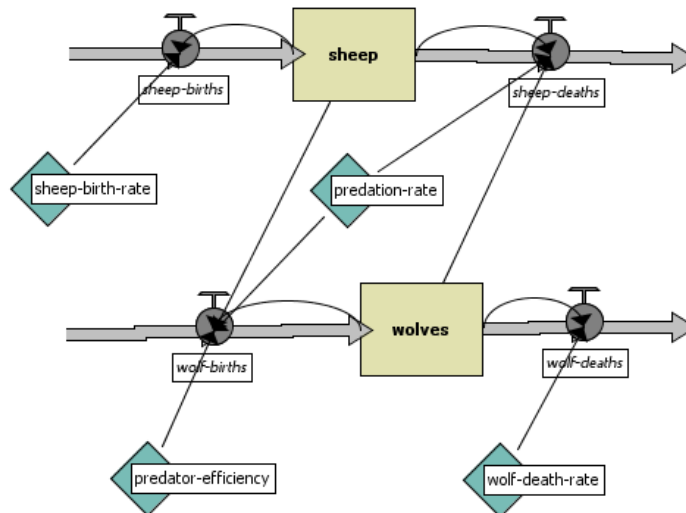


Abbildung 5.9: Ein NetLogo-System-Dynamics-Programm aus der NetLogo-Stammbibliothek zur Simulation eines Räuber-Beute-Schemas.

Auch NetLogo bietet in seiner Stammbibliothek ein Räuber-Beute-Szenario mit Wölfen und Schafen als Akteure und variablen Settings für alle Key-Parameter des Systems. Komplexe, über Differentialgleichungssysteme abgebildete System-szenarien werden auch schwerpunktmäßig in der System-Dynamics-Domäne untersucht und simuliert. An dieser Stelle sei dabei auf die Möglichkeit, derartige Szenarien in NetLogo auch über einen System-Dynamics-Modeller (siehe Abbildung 5.9), ähnlich zu dem zu Beginn der Arbeit vorgestelltem Tool STELLA, zu entwickeln, hingewiesen.

Das Wolf-Schaf-Szenario ist in der NetLogo-Bibliothek nämlich in beiden Varianten verfügbar und zwar auch in einer kombinierten Version, in der sich die Ergebnisse mittels Automaten- und System-Dynamics-Simulation vergleichen lassen, wie Abbildung 5.10 zeigt.

Interessant ist nun natürlich die Frage, wie sich solche ähnlichen Systeme – wie im Beispiel von Lotka-Volterra-Systemen im Allgemeinen oder Brüsselator-systemen im Speziellen –, so sie basierend auf dem CGST-Modell entwickelt wurden, als solche erkennen lassen. In Kapitel 3 wurden prinzipielle Ansätze dazu skizziert, die jetzt anhand einfacher Beispiele anhand mit Hilfe des Serendipity-Szenarios aus 5.2 veranschaulicht werden sollen.

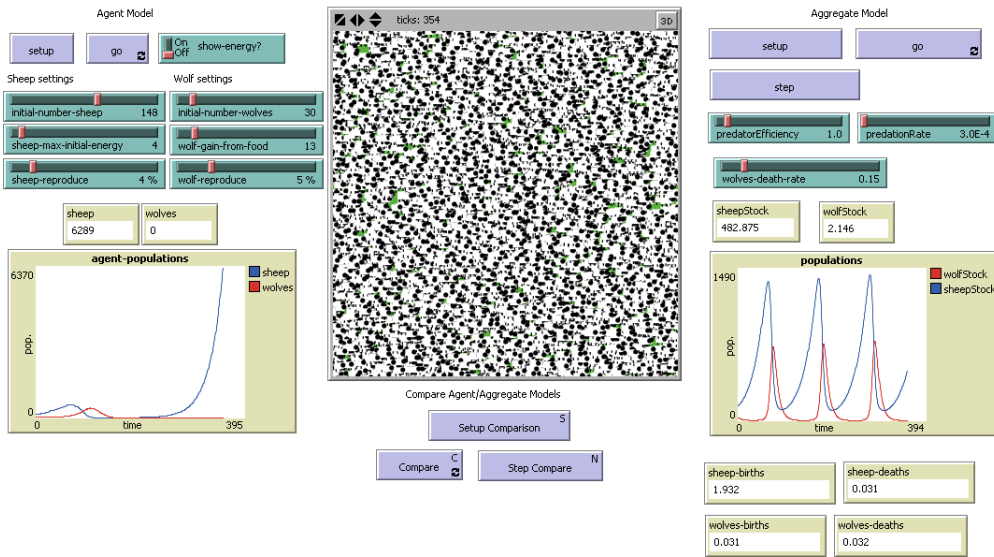


Abbildung 5.10: Programmatischer Vergleich der Automaten- und der System-Dynamics-Variante zur Simulation von Räuber-Beute-Szenarien in NetLogo.

5.3.1 Erkennen von Lotka-Volterra-Abhängigkeitsmustern in CGST-OWL-Subgraphen

Die einem LV-System zugrunde liegenden Abhängigkeiten der Akteure bzw. Elemente erfüllen bestimmte Kriterien bzw. lassen – bei einer entsprechend detaillierten Modellierung – ein doch recht markantes Muster erkennen. Dieses Muster muss nicht unbedingt eindeutig sein bzw. eindeutig auf ein LV-System schließen lassen, aber es wäre zumindest ein guter Kandidat für einen Analogie-Vorschlag einer entsprechenden Anwendung.

Dazu soll der aus $\Theta_{Serendipity}$ abgeleitete gerichtete Teilgraph betrachtet werden. Durch die bestehenden Objektrelationen für die Elemente **Adam** (A), **Eve** (E), **Ambrosia** (Am), **Nektar** (N), **Matter** (M) und **Gas** (G) ergibt sich die folgende Teilgraph-Matrix:

$$\begin{pmatrix}
 AA & AE & AAm & AN & AM & AG \\
 EA & EE & EAm & EN & EM & EG \\
 AmA & AmE & AmAm & AmN & AmM & AmG \\
 NA & NE & NAm & NN & NM & NG \\
 MA & ME & MAm & MN & MM & MG \\
 GA & GE & GAm & GN & GM & GG
 \end{pmatrix} \rightarrow \begin{pmatrix}
 1 & 1 & 0 & 0 & 1 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0
 \end{pmatrix}$$

Um eine Teilgraph-Matrix handelt es sich deshalb, weil sie nur jene Relationen beinhaltet, die für den LV-Prozess relevant sind – so ist etwa die

isFriendOf-Beziehung in der Matrix oben nicht vermerkt. Prinzipiell könnte man eine solche Teilgraph-Matrix aber sehr einfach durch entsprechende Beziehungs-Annotationen der Relationen erreichen. Ein Ansatz, verwandte Systeme zu erkennen, würde nun darin bestehen, nach echten Graph-Isomorphismen in anderen System-Beschreibungen zu suchen, also Teilgraphen mit exakt demselben Matrixmuster bzw. Varianten davon durch Vertauschen der Spalten, weil die Anordnung der Knoten in der Matrix ja prinzipiell willkürlich geschehen ist. Werden derartige Übereinstimmungen gefunden, würden ein Analogievorschlag und eine darauf folgende genauere Analyse der beiden Systeme und eventueller Analogien Sinn machen. Dabei ist zu beachten, dass hier ja nur die Abhängigkeiten selbst, nicht aber eventuelle Kardinalitäten betrachtet wurden, d.h. auch ein LV-System der Form

$$\begin{aligned} 4A &\rightarrow X \\ 3X + 2Y &\rightarrow 5X \\ 2B + X &\rightarrow Y + 2D \\ X &\rightarrow 2E \end{aligned}$$

würde als isomorph und damit analog gefunden werden. Andererseits würden Element-Variationen, etwa das Hinzufügen eines zusätzlichen Input-Elements wie in

$$\begin{aligned} A + C &\rightarrow X \\ 2X + Y &\rightarrow 3X \\ B + X &\rightarrow Y + D \\ X &\rightarrow E \end{aligned}$$

die Analogieprüfung bereits nicht mehr bestehen, obwohl das Verhalten des Systems und dessen prinzipielle Kategorisierung als LV-System weiterhin analog sind. Der Schritt weg von Isomorphismen hin zu Graphen-Matching mit Fehlertoleranz ist aber trotz kleiner Unterschiede ein sehr großer, vor allem was die damit verbundene Rechenkomplexität betrifft. Diese Domäne wird in der Informatik sehr gründlich analysiert, weshalb es bereits viele – zum Teil ernüchternde – Resultate in diesem Bereich gibt¹².

5.3.2 Erkennen von Lotka-Volterra-Verhaltensmustern durch Trajektorienanalyse

Das Verhältnis von Räuber- und Beute-Element in einem LV-System bildet bei Inter-Spezies-Konkurrenz und bestimmten Ausgangskonfigurationen, die dazu führen, dass das System sich in einem Fließgleichgewicht einpendelt, ebenfalls ein markantes Muster, das sowohl bei der Brüsselator-Simulation als auch beim

¹²Siehe hierzu etwa Adel Hlaoui und Sengrui Wang, “A New Algorithm for Inexact Graph Matching”, in ICPR (4), Seiten 180-183, 2002.

komplexen Serendipity-Szenario erkennbar war. Es handelt sich dabei um eine Art “Schneckenform” als Ergebnis immer kleinerer Ungleichgewichte zwischen den beiden Akteuren im Laufe der Zeit (siehe Abbildung 5.11). Idealerweise kommt es dabei zu keinen Überschneidungen im Verlauf der Trajektorie, was aber nicht garantiert werden kann.

Für die Analogie-Berechnung bieten sich nun prinzipiell unterschiedliche Strategien an, Profile zu erzeugen und auf Ähnlichkeit zu prüfen, die in Kapitel 3 nur kurz skizziert wurden und nun anhand des konkreten Szenarios veranschaulicht werden sollen:

- **Graphenmuster Profile.** Ausgehend von einem Funktionsgraphen können heute bereits mitunter sehr effektive Verfahren eingesetzt werden, um ähnliche Graphen in einer Bilder-Bibliothek zu erkennen und so durch Rückschlüsse Ähnlichkeiten zwischen Systemen herstellen. Bild-Matching-Verfahren können dabei ähnlich wie im letzten Unterkapitel auf Graph-Matching-Verfahren basieren¹³ oder durch Sketch-Drawing-Ansätze gestützt sein¹⁴. Ein derartiger Ansatz, der rein auf dem Systemoutput als “Muster” basiert, ist in erster Linie für besonders komplexe und ausgeprägte Funktionen mit markanter Systemantwort interessant.
- **Deskriptive Graphenprofile.** Funktionsgraphen können auch bezüglich ihrer Eigenschaften über logische Aussagen beschrieben bzw. eingeschränkt werden. Im Beispiel der LV-Graphen wären solche Aussagen etwa “Für einen Wert auf der x-Achse kann es auch mehrere Werte auf der y-Achse geben”, “Nicht für jeden Wert auf der x-Achse muss es einen Wert auf der y-Achse geben”, “Der Funktionsgraph besitzt keine Schnittpunkte”, etc. Für jeden Graphen kann so ein deskriptives Profil bestehend aus mathematisch-logischen Aussagen generiert werden, das dann mit Profilen anderer Graphen auf Übereinstimmung geprüft werden kann.
- **Codebasierte Profile.** Eine deutlichere Beschreibung des Funktionsprofils kann über die Verarbeitung des NetLogo-Codes selbst erreicht werden. Durch die verschiedenen Programmfragmente entstehen Abhängigkeitsbeziehungen, die von einem speziellen Compiler in entsprechende Gleichungssysteme übersetzt werden können. Bibliotheken für solche Verfahren sind in Mathematikprogrammen wie MATLAB oder Mathematica bereits seit vielen Jahren Standard und liefern zuverlässige Resultate. Prinzipiell landet man durch diesen Ansatz wieder bei den Auseinandersetzungen des ursprünglichen Bertalanffy-Programms, d.h. bei einem Set von (Differential-)Gleichungssystemen und der Problematik, diese anhand ihrer Eigenschaften auf Ähnlichkeiten zu prüfen – was, wenn es als Teil

¹³Siehe etwa Adel Hlaoui und Shengrui Wang, “A new algorithm for graph matching with application to contentbased image retrieval”, in: Terry Caelli et al. (ed.), SSPP/SPR, Volume 2396 of Lecture Notes in Computer Science, Seiten 291-300, Springer 2002.

¹⁴Siehe <http://labs.systemone.at/retrievr>

eines umfangreicheren Analyseansatzes zum Einsatz kommt, nicht unbedingt ein Nachteil sein muss.

- **Speicherbasierte Profile.** Führen die graphenbasierten Ansätze zu keinem Erfolg und ist die codebasierte Variante zu rechenintensiv, weil die Verschachtelungen im Code sehr komplex sind, bietet sich als Heuristik auch noch die Kalkulation eines speicherbasierten Profils an. Da die Codezeilen im NetLogo-Automaten ja ebenfalls wieder nur von einem Compiler übersetzt und vom Betriebssystem ausgeführt werden, hinterlässt jedes Programm ein mehr oder weniger spezielles Muster in Bezug auf die Abfolge der Speicherallokationen und die Abfolge von rechenaufwändigen Schritten. Derartige Muster können ebenfalls codiert und so zu einem Profil zusammengefasst werden, auch wenn die Erfolgswahrscheinlichkeit bei diesem Ansatz aufgrund der betriebssystemabhängigen Parameter eher gering sein dürfte.

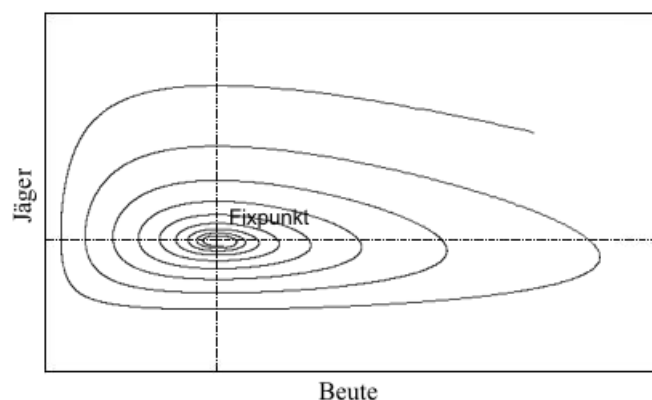


Abbildung 5.11: Typisches Muster einer LV-Trajektorie. Quelle: Wikimedia Commons.

Für das LV-Beispiel würde sich die codebasierte Variante wohl am besten eignen, da der Abhängigkeitsgraph und damit das Differentialgleichungssystem hier noch relativ einfach erzeugt werden könnte und eine Analogiebildung anhand der mathematischen Funktionsbeschreibungen gerade bei LV-Systemen gut möglich ist¹⁵.

¹⁵Tatsächlich gibt es für diesen Gleichungssystem-Typ sogar eine relativ einfache Beschreibung, siehe dazu Alfred Lotka, "Analytical Theory of Biological Populations", Springer, New York 1998.

Kapitel 6

Selbstorganisation und Emergenz

*“Dass ich nicht mehr, mit sauerm Schweiß,
zu sagen brauche was ich nicht weiß;
Dass ich erkenne was die Welt
im Innersten zusammenhält.”
Johann Wolfgang von Goethe, “Faust”*

2009 wird in den Naturwissenschaften als Darwin-Jahr geführt, weil Charles Darwin seinen 200. Geburtstag feiert und vor 150 Jahren sein Werk über die Entstehung der Arten erschienen ist. Tatsächlich hat Darwin Antworten auf bis dahin scheinbar spukhaft ablaufende Prozesse in der Natur gefunden, selbst wenn der Streit mit der Kirche über seine Theorie auch heute noch nicht völlig beigelegt ist. Generell wurden seit dem Beginn der Aufklärung viele Antworten auf große Fragen der Menschheit in vielen verschiedenen Wissenschaftsbereichen gegeben, die allesamt den Horizont unserer Gesellschaft erweitert und unsere Lebensqualität meist nachhaltig verbessert haben.

Große Fragen wie die nach der Entstehung des Lebens selbst, der Beschaffenheit des Universums oder neue Fragen über zufälliges Verhalten bei Elementarteilchen, wie sie seit dem Beginn der Quantenphysik entstanden sind, geben heute aber noch immer Rätsel auf. In vielen Systemen spielen zwei zentrale, jedoch gemessen an ihrer Bedeutung noch viel zu wenig untersuchte Konzepte, eine derart große Rolle, dass sie vielleicht irgendwann auch einmal zur Beantwortung der oben angeführten Fragen ihren Teil beitragen werden: Emergenz, also das Konzept für die Entstehung von Neuem in der Welt und Selbstorganisation, als oft simpler aber wirkungsvoller Gestaltungsprozess von der atomaren bis hin zur gesellschaftlichen Ebene und weiter. Diesen beiden wesentlichen Ideen wollen wir in der Arbeit in Bezug auf das CGST-Modell einen ganz besonderen

Stellenwert verleihen und sie in den folgenden Absätzen genauer untersuchen.

6.1 Emergenz: Wie kommt das Neue in die Welt?

Wenn wir die Entwicklung von Systemen im Allgemeinen und die Entstehung “neuer” Eigenschaften, insbesondere auf höheren Ebenen, im Speziellen betrachten, gibt es metaphysisch gesehen prinzipiell drei Möglichkeiten, um derartiges Verhalten zu erklären:

- Den klassischen Determinismus, d.h. alle nachfolgenden Zustände in einem System sind durch die vergangenen und gegenwärtigen Zustände des Systems bereits determiniert. Eine Eigenschaft wie etwa das menschliche Bewusstsein ergibt sich demnach als logische Folge der zuvor stattgefundenen ebenfalls determinierten Entwicklungen und wäre daher – zumindest theoretisch – berechen- und voraussagbar, wenn nur sämtliche Fakten bekannt und verarbeitbar wären.
- Den Indeterminismus, d.h. das Konzept eines echten Zufalls, wie es etwa von Vertretern der Quantenphysik propagiert wird. Demnach gibt es auf einer Mikroebene echte zufällige, also nicht mehr weiter zerlegbare und erklärbare, Ereignisse, die sich mehr oder weniger auch auf die Entwicklung von Makrozuständen auswirken.
- Die Emergenztheorie, d.h. eine nicht näher definierbare Kraft oder Regel, die, ohne dabei determiniert zu sein, bei einer hohen Komplexität in einem System sprunghaft neue Eigenschaften in ebendiesem erzeugt. Emergente Prozesse haben dabei gewisse notwendige, jedoch keine hinreichenden Ausgangszustände, sodass sich insbesondere Zeitpunkt und Ausprägung von emergenten Sprüngen nicht voraussagen lassen. Sie sind dabei aber auch keineswegs zufällig, weil sie bestimmten, wenn auch nicht näher erkennbaren Gesetzen folgen.

Keine dieser drei Theorien konnte bisher bewiesen werden und es ist auch für die nahe Zukunft kein entsprechender Beweis zu erwarten. Wissenschaftler haben sich bei der Konzeption von Erklärungsmodellen in der Regel für eine bestimmte Hypothese aus diesem Rahmen zu entscheiden oder ihre Ausprägungen zumindest einzuschränken, um notwendige Ausgangssituationen für darauf basierende Theorien zu schaffen. Für unser CGST-Modell treffen wir dabei folgende Annahme, die wir nachfolgend so gut es geht argumentieren werden:

Entwicklungen in Systemen, d.h. zukünftige Zustände von Systemen, wie sie im CGST-Modell abgebildet werden, sind entweder determiniert oder emergent. Zufall wird, so er verwendet wird, nur als unechter Zufall in Simulationen

verwendet, um komplexe Entwicklungen auf untergeordneten Ebenen, für die statistische Werte bekannt sind, zu generieren. Das Konzept eines echten Zufalls, wie er etwa in Interpretationen der Quantenphysik vorkommt, ist nicht Teil des CGST-Modells.

Jetzt stellen sich auf Anhieb mehrere Fragen: Wieso wird an dieser Stelle der Indeterminismus so leichtfertig verworfen? Wie kann in einem formalen System Emergenz modelliert werden oder auftreten? Welche Einschränkungen oder Auswirkungen entstehen durch diese Grundannahmen in unserem Modell?

Die pragmatische Antwort zur Frage des Indeterminismus lautet: Er muss als Konzept verworfen werden, weil sich ein echter Zufall formal nicht abbilden lässt. Diesen kann es, wenn es ihn denn tatsächlich gibt, nur in der Natur, niemals aber in einem formalen System geben. Diese Gegebenheit wäre deshalb auch ein harter Bruch zwischen der Realität und ihrer Abbildbarkeit in formalen Modellen. Echter Zufall lässt sich weder berechnen noch logisch herleiten, denn genau so ist er ja definiert. Dieser pragmatische Schritt ist aber auch aus wissenschaftlicher und erkenntnistheoretischer Perspektive der erfolgsversprechendere, denn ein echter Zufall bedeutet auch eine echte Grenze menschlicher Erkenntnis und Kontrolle, die zwar von den meisten Vertretern der Quantenphysik verteidigt¹ und zuweilen als schön oder elegant bezeichnet wird, dem Logiker aber gegen sein Streben nach absoluter Formalisierbarkeit wirkt.

6.1.1 Formalisierbarkeit von Emergenz

Die Antwort auf die Frage der Formalisierbarkeit der Emergenz bedarf einer tieferen Argumentation: Prinzipiell wurde bis zum heutigen Tag kein Beweis für auch nur einen einzigen echten emergenten Prozess in der Realität erbracht. Es gibt Annahmen, dass es sich bei den molekularen Eigenschaften von Wasser, die tatsächlich der Physik und Chemie noch immer große Probleme bereiten, um echte Emergenz handeln könnte. Auch das menschliche Bewusstsein oder das Entstehen von Leben stehen auf der Liste der möglichen Kandidaten für emergente Sprünge in Systementwicklungen. Bewiesen ist das aber alles nicht.

¹Tatsächlich ist entgegen der weit verbreiteten Meinung und Darstellung in der Fachliteratur der Zufallscharakter in der Quantenmechanik keinesfalls außer Streit gestellt oder gar bereits bewiesen. Zum einen ist eine mögliche Interpretation des in diesem Zusammenhang immer wieder zitierten Problems der Verletzung der Bell'schen Ungleichung ein absoluter Determinismus, bei dem auch der Beobachter selbst Teil einer universalen determinierten Kette ist; zum anderen ist auch das Vorhandensein lokaler verborgener Variablen durchaus möglich, wenn man gleichzeitig die Verschränkung der Quanten über einen mit Überlichtgeschwindigkeit funktionierenden Quantenkanal oder Nichtlokalität der Welt akzeptiert, was einer klassischen deterministischen Interpretation der Quantenmechanik wieder Raum verschafft. Und nicht zuletzt gibt es alternative Denkrichtungen zur Interpretation der Quantenmechanik, wie etwa die De-Broglie-Bohm-Theorie, die ebenfalls ein konsistentes Modell von Determinismus und Quantenmechanik erzeugen.

Es gibt auch bisher nur sehr wenige wirklich fokussierte wissenschaftliche Auseinandersetzungen mit dem Emergenzbegriff. Eine vor allem begriffliche und wissenschaftshistorische Analyse des Emergenz-Konzepts und seiner Entstehungsgeschichte hat Achim Stephan in [46] beschrieben. Dabei zeigt er auch die oft unterschiedliche Verwendung und Bedeutung dieses Begriffs in der Literatur auf und führt als Folge dieser Diskrepanz ein Klassifikationsschema unterschiedlicher Emergenzvarianten ein, das in dieser Arbeit ebenfalls verwendet werden soll.

Stephan differenziert die Emergenzvarianten eines **schwachen Emergentismus**, **schwachen diachronen Emergentismus**, **synchronen Emergentismus**, **diachronen Strukturemergentismus**, **starken diachronen Emergentismus** und des **starken diachronen Strukturemergentismus**. Im Prinzip ergibt sich die Notwendigkeit einer solchen Differenzierung durch die folgenden Eigenschaften, die in der Literatur dem Emergenzbegriff zugeschrieben werden: Systemische Eigenschaften, Irreduzibilität, Neuartigkeit und Struktur-Unvorhersagbarkeit. Im vergangenen Abschnitt ist ohne Verweis auf eine spezielle Variante einfach nur die Rede von “Emergenz”, gemeint ist aber dort wie auch überall anders in dieser Arbeit – so nicht eine andere spezielle Variante begrifflich verwendet wird – die von Stephan beschriebene Variante des starken diachronen Emergentismus bzw. dort wo explizit darauf hingewiesen wird eines Strukturemergentismus, der alle zuvor erwähnten Charakteristika in seinem Emergenzbegriff vereint:

- **Systemeigenschaften.** Es existieren Eigenschaften in Systemen, die nicht Teil der einzelnen Elemente des Systems sind, sondern erst durch den Zusammenschluss der Elemente zum System entstehen.
- **(Unvorhersagbare) Neuartigkeit.** Durch die Entwicklung von Systemen entstehen “genuin neue” Eigenschaften. Einige dieser Systemeigenschaften sind dabei prinzipiell nicht vorhersagbar.
- **Irreduzibilität.** Es existieren Systemeigenschaften, die sich durch Analyse nicht auf die Eigenschaften und das Verhalten der Elemente des Systems reduzieren lassen.
- **Struktur-Unvorhersagbarkeit.** Das Entstehen von einigen (neuen) Strukturen in Systemen ist prinzipiell nicht vorhersagbar.

Der erste Punkt ist bereits hinreichend für den schwachen Emergentismus² und in dieser Form auch bereits Teil und Grundannahme des CGST-Modells (“systemische Eigenschaft”). Der zweite Punkt, die genuin neuen Eigenschaften, ist etwas schwierig, weil es sich hier ja oft auch nur um ein begriffliches

²Siehe [46], Seite 66 f.

Phänomen handelt. Funktionale Neuartigkeit auf der Makroebene festzustellen ist eine schwierige formale Angelegenheit, auf die wir – auch ob der geringeren Wichtigkeit – hier nicht den Fokus legen wollen. Der dritte Punkt, die Irreduzibilität bestimmter Systemeigenschaften, ist der klassische und eindeutige Widerspruch zum Determinismus und der entscheidende Punkt – in einem vollständig determinierten System kann es keine nicht reduzierbaren Eigenschaften oder Entwicklungen geben. Die weiteren Spielarten des Emergenzbegriffs sind für die Differenzierung in dieser Arbeit nicht weiter interessant, ein kurzer Auszug aus der Begriffserklärung der Zusammenhänge aus Stephans Arbeit soll das verdeutlichen³:

“Der schwache diachrone Emergentismus entsteht aus dem schwachen Emergentismus durch Hinzufügen der zeitlichen Dimension in Form der Neuartigkeitsthese. Beide Versionen sind mit dem reduktiven Physikalismus kompatibel. Schwache Emergenztheorien finden derzeit vor allem in den Kognitionswissenschaften, besonders zur Charakterisierung systemischer Eigenschaften in konnektionistischen Netzen sowie in Theorien der Selbstorganisation Verwendung. Der synchrone Emergentismus, der aus dem schwachen Emergentismus durch Hinzufügen der Irreduzibilitätsthese hervorgeht, feiert in der zeitgenössischen Debatte um den nichtreduktiven Physikalismus eine Renaissance. Diese Theorie ist nicht mehr mit dem reduktiven Physikalismus verträglich. Der starke diachrone Emergentismus unterscheidet sich nur unwesentlich durch die Berücksichtigung der zeitlichen Dimension in Form der Neuartigkeitsthese vom synchronen Emergentismus.”

Praktische Beispiele, die diese Differenzierung veranschaulichen, sind in der aktuellen Literatur zum Emergenzbegriff nur selten vorhanden. An dieser Stelle soll daher versucht werden, den Emergenzbegriff anhand ausgewählter Beispiele, die in besonderem Zusammenhang mit den Inhalten einer Allgemeinen Systemtheorie stehen, zu erklären. Der starke diachrone Emergentismus angewandt auf die potentiell emergenten molekularen physikalischen Eigenschaften von Wasser müsste demnach etwa folgendes bedeuten:

H1 Systemische Wassereigenschaften. Wasser besitzt zweifelsohne systemische Eigenschaften, die weder Teil der Wasserstoff- oder Sauerstoffmoleküle noch Teil der einzelnen Wassermoleküle sind. Erst auf der Makroebene des “Wasser-Systems” finden sich Charakteristika wie der jeweilige Aggregatzustand, Dichte- oder Kohäsionseigenschaften. Auch die Anomalie des Wassers (siehe Abbildung 6.1), die es von anderen Stoffen unterscheidet, ist als Dichtefunktion eine solche systemische Eigenschaft.

³Siehe [46], Seite 71.

H2 Unvorhersagbarkeit der Wassereigenschaften. Dass man, ohne es zu wissen und nur unter Kenntnis der atomaren und molekularen Eigenschaften, voraussagen könne, dass Wasser bei einer bestimmten Temperatur fest und bei einer anderen gasförmig wird, klingt aufs erste unglaubwürdig. Tatsächlich wäre es aber mit heutigem (mathematischen) Wissen bei einer exakten Definition der atomaren und molekularen Eigenschaften und der Abbildung der Aggregatzustände auf eben diese formalisierten Eigenschaften durchaus möglich, solche und auch weitergehende Aussagen, wie etwa über Kohäsion oder Dichte, formal zu berechnen und zu simulieren.

H3 Irreduzibilität von Wassereigenschaften. Wasser wird, trotz seiner Anomalien und besonderen Eigenschaften, heute bereits viel besser wissenschaftlich verstanden als noch vor 100 Jahren. Einige Eigenschaften, insbesondere die unter bestimmten Bedingungen extreme Kohäsion oder etwa so genanntes “hochdichtes Wasser”⁴, können heute noch nicht vollständig erklärt werden.

Obwohl es bezüglich einiger Eigenschaften von Wasser und auch seiner grundsätzlichen molekularen Struktur noch offene Fragen gibt, ist damit nicht gesagt, dass diese eines Tages nicht doch vollständig analytisch beantwortet werden können. Stephan spricht⁵ bereits davon, dass

“wir heute – nach den bahnbrechenden Erfolgen der Chemie und Physik – die Eigenschaften und Dispositionen chemischer Verbindungen nicht mehr zu den synchron emergenten Phänomenen rechnen”

und bezieht sich dabei auf die ursprünglichen Arbeiten eines Wegbereiters der Emergenztheorie, Lloyd Morgan, der Eigenschaften chemischer Verbindungen noch als Beispiele in seiner Theorie verwendete.

Um die einzelnen Charakteristika des Emergenzbegriffs noch deutlicher herauszustreichen und zu veranschaulichen und die Problematik allgemeiner darzustellen, soll noch ein zweites Beispiel nach dem gleichen Muster betrachtet werden. Obwohl die Abgrenzung zur Metaphysik hier schon sehr schwierig ist, sollte eine entsprechende Abbildung der Bewusstseinsproblematik, nicht zu verwechseln mit dem verwandten Leib-Seele-Problem, ein geeignetes weiteres Beispiel geben.

H1 Systemische Bewusstseins-eigenschaften. Bewusstsein ist eine systemische Eigenschaft des Gehirns. Im Laufe der Evolution entwickelte sich –

⁴Siehe Simon Engemann, Harald Reichert, Helmut Dosch, Jörg Bilgram, Veijo Honkimäki, Anatoly Snigirev: “Interfacial melting of ice in contact with SiO₂”, Physical Review Letters

⁵Siehe [46], Seite 245.

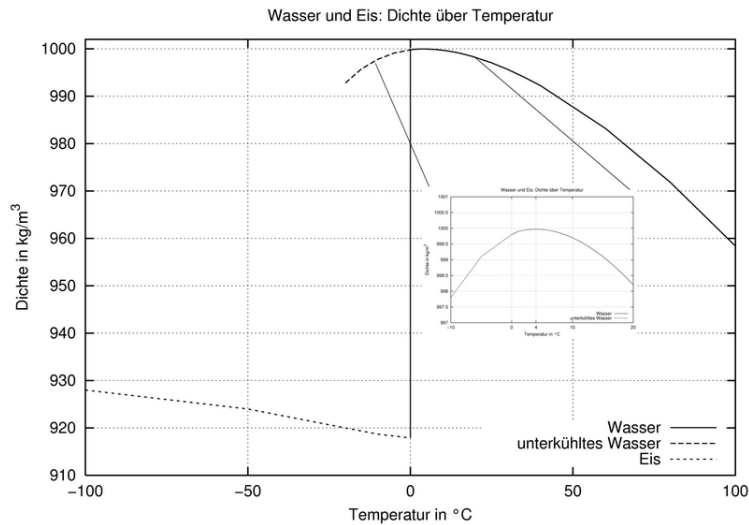


Abbildung 6.1: Temperatur/Dichte-Funktionsgraph zur Veranschaulichung der Anomalie des Wassers. Quelle: Wikimedia.

in qualitativ sehr hochwertiger Ausprägung beim Menschen, aber auch bei Tieren – eine Form der Wahrnehmung, die aufgrund der damit verbundenen Denkprozesse und Handlungen einen Vorteil gegenüber anderen verschafft. An dieser Stelle soll das Bewusstsein beim Menschen betrachtet werden: Neurobiologen können heute die physischen Prozesse, die zur Bewusstseinsbildung beitragen, bereits relativ gut beschreiben⁶, wenngleich auch ein vollständiges und zufriedenstellendes wissenschaftliches Modell noch aussteht. Jedenfalls aber ist Bewusstsein eine Eigenschaft, die nur durch das Zusammenspiel vieler einzelner Neuronen in unterschiedlichen Gehirnhälften entsteht. Die für die sozialen Bindungen wesentliche Kontrolle der Emotionen (und auch die vermeintliche Willensbildung) hat ihren Sitz nach heutigen Erkenntnissen im präfrontalen Cortex, der maßgeblich für das Berechnen von Handlungsalternativen, die Entscheidungsfindung und das Überwachen von Handlungen verantwortlich ist. Es kann auch aufgrund von Knochenfunden gezeigt werden, dass die Ausbildung des Schädels und jener Region, die den präfrontalen Cortex beheimatet, Hand in Hand geht mit der Entwicklung des Menschen zum sozialen Wesen⁷.

H2 Unvorhersagbarkeit von Bewusstseinsbildung. Die Vorhersagbarkeit bzw. Unvorhersagbarkeit von Bewusstseinsbildung ist eine schwierige und zum heutigen Zeitpunkt sicherlich noch ungelöste Frage. Außer Streit steht: Der Mensch in seiner heutigen Ausformung hat ein Bewusstsein, dass es ihm erlaubt, über sich selbst zu reflektieren, sich selbst wahr zu nehmen und sich auch in andere hineinzusetzen. Tieren wird außerdem

⁶Siehe dazu etwa [39], Seite 196 ff.

⁷Siehe [39], Seite 81 ff.

ein Teilbewusstsein zugesprochen, das aber – auch bei Schimpansen und ähnlich intelligenten und weit entwickelten Arten – bei weitem nicht an die Qualität des Bewusstseins des Menschen herankommt. Eine der dringendsten Fragen, die im Laufe der letzten 60 Jahre entstanden ist, ist jene über die Möglichkeit eines Bewusstseins bei hochkomplexen Maschinen, also einer echten Künstlichen Intelligenz. Befürworter dieser These sagen ein derartig zu erwartendes Verhalten bei einer theoretisch durchaus möglichen erreichbaren Komplexität und Eigenentwicklung maschineller Prozesse heute bereits voraus.

H3 Irreduzibilität von Bewusstsein. Bewusstsein scheint auf den ersten Blick ein guter Kandidat für Irreduzibilität zu sein, da es kaum vorstellbar ist, dass sich diese Eigenschaft auf ganz bestimmte Neuronenverbindungen oder Gruppen davon festmachen lässt. Die primitiven Formen von Bewusstsein bei Tieren und die etwas höher entwickelte Form etwa bei Schimpansen ist aber ein Indiz dafür, dass Bewusstsein prinzipiell eine fließende Eigenschaft ist, die sich offensichtlich mit bestimmten Mustern von Gehirnkonfigurationen stärker oder schwächer ausprägen kann. “Fließende” Eigenschaften sind aber wiederum schlechte Kandidaten für Irreduzibilität, da sie prinzipiell das Ergebnis einer kontinuierlich fortschreitenden Entwicklung und somit auch nicht irreduzibel, sondern nur in einem langfristigen und damit schwer nachvollziehbaren Prozess entstanden sind.

Insbesondere die Arbeiten von Gerhard Roth und seinem Kollegen Wolf Singer⁸ zeigen die Möglichkeit einer determinierten Beschaffenheit und Entwicklung von Bewusstsein auf, die sich durchaus im Laufe der nächsten Jahrzehnte als gängige Wissenschaftsmeinung etablieren könnte. Und mehr noch: In diesen Erkenntnissen liegt freilich auch die Antwort auf die Frage nach einer “echten Künstlichen Intelligenz”, also nach der Möglichkeit eines maschinenbasierten Bewusstseins, verborgen. Demnach könnte diese “alte” Frage der Informatik nicht nur positiv beantwortet werden, die Antwort könnte auch weit weniger spektakulär ausfallen, als das von den bisherigen Befürwortern der These skizziert wurde. Eine schwierige Frage in diesem Zusammenhang, die oft die Stellvertreterdiskussion für das gesamte Generalthema ausmacht, wurde bisher noch außen vor gelassen: Laut den Arbeiten von Roth und Singer ist auch der Freie Wille nicht mehr länger ein unlösbares Mysterium, sondern ebenfalls determiniert. Trifft das zu, fällt auch dieser Pfeiler für die Verfechter einer nicht möglichen Künstlichen Intelligenz weg, denn wenn der Mensch tatsächlich keine absolute Entscheidungsfreiheit bzw. einen abstrakten Freien Willen besitzt, muss man auch nicht länger erklären, wie denn eine determinierte Maschine eine solche Eigenschaft entwickeln können sollte.

⁸Siehe etwa [42].

Roth und Singer sind dabei in ihren Arbeiten sehr überzeugend. Die Indizien und aus Sicht von Roth und Singer bereits für eine derartige Schlussfolgerung ausreichenden Erkenntnisse auf dem Gebiet der Neurobiologie und Hirnforschung, die diese These stützen, verdichten sich und setzen sich – trotz der kontraintuitiven Stossrichtung dieser Lehrmeinung – auch immer stärker im wissenschaftlichen Umfeld durch. Wesentliche Beiträge zum aktuellen Diskurs und umfassende wissenschaftliche Erläuterungen dazu finden sich etwa in einer Deutung der aktuellen Experimente in diesem Bereich in [20], einer Auseinandersetzung von Gerhard Roth mit seinem Kollegen Klaus-Jürgen Grün zum Thema Willensfreiheit[40] und seinem dazu passenden biologischen Grundwerk[39]. Zu den gesellschaftlichen Implikationen dieses vermeintlichen wissenschaftlichen Paradigmenwechsels sei auf die Argumentationen von Singer, etwa in [42] verwiesen.

All das hilft uns bei der Formalisierbarkeit von Emergenz aber bisher nur sehr eingeschränkt weiter – denn ohne tatsächliche Anwendungsfälle ist es schwierig, ein entsprechendes Konzept in ein formales Modell zu integrieren.

Emergenz soll daher nun dort gesucht werden, wo man sie vielleicht am aller wenigsten vermuten würde: In der Mathematik, genauer gesagt in der mathematischen Logik. Die Idee dabei ist die folgende: Wenn wir Zustände und Prozesse von Systemen formal auf unser CGST-Modell abbilden und dabei hinreichend genau sind und es dabei keinen echten Zufall gibt; wenn wir nun bei der logischen Konsistenzprüfung oder der Berechnung von Folgezuständen einen Zustand erreichen, der nicht entscheidbar ist; wenn wir daher auch keine Möglichkeit haben, aus den Ergebnissen der Berechnung Rückschlüsse auf das System in der Realität zu ziehen; dann gibt es nur zwei Möglichkeiten: Die Mathematik und die mathematische Logik mit all ihren Eigenschaften wie wir sie heute kennen und damit auch die Informatik sind als formales System insofern fehlerkonzipiert, als dass sie nicht die tatsächlichen Gegebenheiten der Natur formal beschreiben können. Oder aber nicht entscheidbare Zustände sind als echte Emergenz der mathematischen Logik anzusehen, die sich aus der Konsequenz, dass die Mathematik und die mathematische Logik tatsächlich korrekte formale Abbildungen der Realität erlauben, ergeben.

Zunächst: Was bedeutet hier “nicht entscheidbar”? Die Mathematik und die Informatik kennen zwei verwandte Probleme, die beide im Laufe des 20. Jahrhunderts beschrieben wurden. In den 1920-Jahren startete die Mathematik unter der Schirmherrschaft des deutschen Mathematikers David Hilbert den Versuch, eine vollständige formale Beschreibung auszuarbeiten, die künftig als Grundlage für die Mathematik selbst, aber auch andere Wissenschaften gelten sollte. 23 Probleme wurden dazu formuliert, die es in der Folge zu lösen galt, außerdem wurde Schritt für Schritt die Arithmetik axiomatisiert mit dem Ziel, als Abschluss ein vollständiges und konsistentes Werk zu schaffen. An die-

sem als Hilbert-Programm bezeichneten Versuch beteiligten sich die führenden Mathematiker unterschiedlicher Länder und hatten bereits ein gutes Stück der im Rahmen des Programms anzufertigenden Schriften fertig gestellt. Zeitgleich arbeitete ein junger Logiker an der Universität Wien, Kurt Gödel, an seiner Dissertation und Habilitation – und war dabei stets bemüht, die Arbeiten Hilberts zu verfolgen.

Gödels Arbeiten sollten zu einem für das Hilbert-Programm tragischen Schluss⁹ kommen: In seiner Publikation¹⁰ “Über formal unentscheidbare Sätze der Principia mathematica und verwandter Systeme” widerlegte Gödel Hilberts Versuch, die Arithmetik widerspruchsfrei zu axiomatisieren, indem er in einem formalen Beweis zeigte, dass dies für das formale System der Arithmetik und alle anderen ähnlich aussagekräftigen formalen Systeme nicht möglich ist.

Alan Turing sollte Jahre später zu einem ähnlichen Schluss für die theoretische Informatik kommen: Für sein Modell der Turing-Maschine stellte er selbst die Frage auf, ob es denn Programme geben kann, deren Terminierbarkeit man nicht feststellen kann, d.h. deren Eigenschaft der Berechenbarkeit nicht entscheidbar ist. So gelangte er zum Halteproblem: Eine Maschine, die feststellen soll, ob ein ihr übergebener Code terminiert, kann bei der Übergabe des eigenen Codes nicht allgemein zu einem Ergebnis kommen, da jede hier gewählte (theoretische) Strategie zu einem unauflösbaren Widerspruch führt. Auf die Analogie der beiden Probleme, insbesondere in Hinblick auf die Notwendigkeit der Selbstreferenz, wird später noch näher eingegangen.

H1 Systemische Eigenschaften der Arithmetik. Ohne Zweifel bildet die Arithmetik ein formales System mit systemischen Eigenschaften. Durch die Konstruktion der Natürlichen Zahlen ergeben sich eine Reihe von Eigenschaften, die sich nur im System (gerade und ungerade Zahlen, Teilungseigenschaften, Primzahlen, etc.) ergeben¹¹. Eine ganz besondere Eigenschaft, anhand derer auch die folgenden Charakteristika untersucht werden sollen, ist die Primzahl-Eigenschaft: Sie bereitet den Mathematikern seit Jahrhunderten Kopfzerbrechen und ist auch heute noch für eine beachtliche Zahl von ungelösten Fragen in der Mathematik verantwortlich – und das, obwohl die Grundidee eine relativ einfache zu sein scheint. Jedenfalls ist es aber eine Eigenschaft, die erst im System der Zahlentheorie, nicht aber als Eigenschaft ihrer Elemente ohne diesen Kontext

⁹Die Meinungen über die Bewertung eines Scheiterns des Hilbert-Programms gehen hier jedoch auseinander, da es eigentlich Hilberts ursprüngliche Motivation war, festzustellen, ob eine vollständige und widerspruchsfreie Definition der Arithmetik möglich ist.

¹⁰Nachzulesen in [18], Seite 144 ff.

¹¹So lautet etwa ein beliebter Mathematikerwitz, der die Bedeutung der einfachen Natürlichen Zahlen hervorheben soll: “Jede natürliche Zahl ist interessant. Denn angenommen, es gäbe eine uninteressante natürliche Zahl, dann gäbe es auch eine kleinste uninteressante Zahl: dies macht diese Zahl aber wirklich interessant! Also ist dies doch eine interessante Zahl. Der Widerspruch zeigt: Es gibt keine uninteressante Zahl.”

besteht.

H2 Unvorhersagbarkeit von Nichtentscheidbarkeit. Historisch gesehen handelt es sich bei der Unentscheidbarkeit von PL1-Aussagen natürlich um eine unvorhergesagte Eigenschaft. Wie oben bereits erwähnt, war die allgemeine Annahme ja gegensätzlicher Natur, dass also die Mathematik im Allgemeinen bzw. die Arithmetik im Speziellen vollständig entscheidbar ist. Dies ist aber nicht die Art von Vorhersagbarkeit, die wesentlich ist, da die grundsätzliche Frage lauten muss: Kann man zu dem Zeitpunkt, zu dem man dieses formale System konstruiert, bereits “vorhersagen”, dass es unentscheidbare Aussagen beinhalten kann?

H3 Irreduzibilität von Nichtentscheidbarkeit. Kann man die Unentscheidbarkeit analytisch erklären? Auf einer formalen Ebene lässt sie sich, wie später noch einmal genauer betrachtet werden soll, auf die Einführung der Selbstreferenz in einem sonst hinreichend mächtigen System “reduzieren”. Das Erkennen von notwendigen Eigenschaften ist aber noch kein Ausschlussgrund für starke Emergenz. Vielmehr bedeutet Irreduzibilität in diesem Zusammenhang ja, wie auch in den bisherigen Beispielen gezeigt wurde, eine vollständige analytische Erklärung des speziellen systemischen Verhaltens. Ob das im Falle von nicht entscheidbaren mathematischen Aussagen der Fall ist, darüber wird viel diskutiert.

Die formale Unentscheidbarkeit der Mathematik und verwandter Systeme scheint also tatsächlich ein guter Kandidat für synchron emergentes Verhalten in einem, wengleich auch formalen, System zu sein. Welche Aussagekraft dieses Resultat für die Natur und die Beschaffenheit der Welt hat, hängt stark von der Interpretation der Rolle der Mathematik im Weltbild ab. Dazu müssen Abbildungen von naturwissenschaftlichen Problemen auf formale Systeme betrachtet und anhand der dadurch entstehenden Ergebnisse analysiert und bewertet werden – an dieser Stelle freilich insbesondere die Beziehung zwischen dem “menschlichen Rechenapparat” und dem auf den oben beschriebenen formalen Systemen basierenden künstlichen.

Die entscheidende Frage, die sich auch Gödel schon gestellt hat¹², ist also: Entspricht das menschliche Denken ebenfalls einem System, das ähnlich ist zu den gerade untersuchten formalen Systemen oder aber ist das menschliche Denken in der Lage, seine eigenen Grenzen immer wieder aufs Neue zu überschreiten und unterliegt deshalb nicht der Einschränkung dieser Systeme, dass es Aussagen gibt, die nicht vom menschlichen Denken feststellbar – also bewiesen oder falsifiziert – sind? Anders gefragt: Funktioniert das menschliche Gehirn

¹²Gödels metaphysische Auseinandersetzungen mit seinen Resultaten sind unter anderem in [47] dokumentiert.

wie eine Maschine bzw. gibt es echte Künstliche Intelligenz? Nun lässt sich zeigen, dass unabhängig wie diese Frage zu beantworten ist, eine stark emergente Eigenschaft vorliegt.

Angenommen, das menschliche Gehirn und das menschliche Denken sind formal abbildbar und erfüllen aufgrund der Tatsache, dass sie zumindest in der Lage sind, auch Prädikatenlogik-Aussagen (PL1) zu konstruieren, die Voraussetzungen für ein PM-verwandtes (wobei PM hier für eine von Gödel verwendete vereinfachte Version der Principia Mathematica steht) System. Dann gibt es Aussagen, Tatsachen der Realität, die im menschlichen Denken formal abgebildet werden, von denen nicht entschieden werden kann, ob sie zutreffen oder nicht. D.h. es gibt Dinge im Denken des Menschen und damit auch in der dadurch konstruierten Realität, die ob ihrer Unentscheidbarkeit nicht reduzibel, nicht vorhersagbar und damit auch stark emergent **erscheinen** und im menschlichen Denken und der Wissenschaft dieses Kriterium daher auch erfüllen – denn in diesem Fall kann auch der Mensch das System seines eigenen Denkens nicht verlassen.

Angenommen, das menschliche Gehirn erfüllt diese Eigenschaft nicht, d.h. es ist im Gegensatz zur an der formalen Logik orientierten Maschine in der Lage, durch “Kreativität” (so nannte Gödel diese entscheidende Eigenschaft) immer wieder das System seines Denkens zu verlassen und Lösungen für alle Probleme zu finden.

Aufgrund der (zumindest theoretisch) formal abbildbaren physischen Zustände im Gehirn und deren prinzipieller Zuordnung zu einem PM-verwandten System zu einem Zustand s_x , in dem eine Aussage A nicht entscheidbar ist und zu einem Folgezustand s_{x+t} , in dem die Aussage A dann entscheidbar ist (für alle Aussagen A), muss es eine außerhalb der Regelung des formalen Systems stehende Komponente, die aber auf eben dieser Ebene wirkt, geben, die ein solches, durch die Eigenschaft des formalen Systems prinzipiell nicht mögliches Verhalten doch ermöglicht, dadurch aber nicht vorhersagbar und nicht analytisch erfassbar, also irreduzibel ist, geben – diese Komponente ist aber aufgrund ihrer Eigenschaften im Kontext des Systems des menschlichen Denkens stark emergent und überträgt diese Eigenschaft auch auf ebendieses System.

In beiden Fällen findet sich also starke Emergenz. D.h. entweder gibt es Dinge in unserer Umwelt, die wir aufgrund unseres Denkens nicht erfassen können und uns daher stark emergent erscheinen (und immer erscheinen werden). Oder aber das menschliche Denken besitzt eine Komponente, die sich nicht durch das menschliche Denken, also die Möglichkeiten der wissenschaftlichen Formalisierbarkeit beschreiben lässt und daher als stark emergent bezeichnet werden muss. Diese Tatsache kann wohl als Beweis für die Existenz starker Emergenz – zumindest auf der Ebene menschlicher Erfassbarkeit, die in wissenschaftlichen

Arbeiten auch nicht verlassen werden kann – gesehen werden.

Zur Vermeidung der beiden Probleme in CGST gäbe es zwei simple Einschränkungen: Um die statische Entscheidbarkeit zu gewährleisten, würde eine Einschränkung der Ausdrucksstärke von OWL Full auf OWL DL (mit ein paar zusätzlichen zu beachtenden Regeln) ausreichen, da OWL DL im Gegensatz zu OWL Full entscheidbar ist. Um nicht in das Halteproblem-Dilemma zu steuern, müsste man CGST der Einschränkung unterwerfen, dass keine Methode ihren eigenen Code als Input erhalten darf – dies müsste zwar ständig vor der Ausführung einer Methode geprüft werden, ist aber ansonsten verfahrensmäßig unbedenklich.

Die Einschränkung von OWL Full auf OWL DL können wir nur implizit einführen. Das bedeutet, dass wir für jene Systeme, für die eine Abbildung in OWL DL ausreichend ist, sagen können, dass wir die Entscheidbarkeit behalten und somit in diesen speziellen Systeminstanzen keine problematischen statischen Zustände erhalten. Sollten aber Konzepte aus OWL Full, die zu einer Unentscheidbarkeit führen, verwendet werden müssen, weil sonst eine vollständige Abbildung nicht möglich ist, können und dürfen wir auf diese Konzepte nicht verzichten. Wir dürfen an dieser Stelle nicht vergessen, dass Description Logics ja bewusst als strukturierte Fragmente von PL1 mit entsprechenden Vorteilen konzipiert wurden, dass die Prädikatenlogik erster Stufe aber die für die Mathematische Logik notwendige Form zur Abbildung der Realität in ein formales System mit all ihren Nachteilen darstellt.

Ähnliches gilt für das Halteproblem. Sollte in einer Systeminstanz – etwa durch die Simulation von Selbstorganisation und ohne Überwachung und Steuerung derselben – eine Zustandsabfolge entwickelt werden, die zum Halteproblem analog ist, so wird in der computationalen Praxis der dahinterliegende ausführende Prozess scheitern, in dem er einen Stack-Overflow produziert, weil die entstehenden konditionalen Bedingungen nicht aufgelöst werden können. Eine Feststellung dieser speziellen Situation ist in bestimmten Fällen möglich (etwa könnte die Einschränkung, dass keine Funktion ihren eigenen Code als Parameter erhalten darf, getroffen werden), würde aber prinzipiell die Simulation ungewollt beeinträchtigen und dabei trotzdem nicht alle möglichen Problemfälle ausschließen.

Jedes ähnliche formale System, das gewisse Grundvoraussetzungen erfüllt, enthält diese Eigenschaft. Deshalb ist eine der großen Fragen des 20. Jahrhunderts, die heute noch immer nicht beantwortet und aufgrund der Thesen von Alan Turing, die vorhin kurz präsentiert wurden, allgemein als das Turing-Problem bekannt ist: Ist das menschliche Gehirn, das ja im wesentlichen eine Art Neuronen-Computer darstellt, äquivalent zu einem Computer, der den Grenzen der Mathematik und der Logik unterliegt oder schafft es das Gehirn, aufgrund

noch nicht erklärbarer Eigenschaften, das, was ein formales System und seine technische Verwirklichung, der Computer, nicht schafft: Den “Sprung” heraus aus jeder Unentscheidbarkeit durch eine besondere Form oder Fähigkeit des kreativen Problemlösens und damit – so könnte man dann sagen – stark emergentes Verhalten?

Bereits Gödel selbst hat sich mit dieser Frage auseinandergesetzt und sie unter anderem mit von Neumann diskutiert, wie Wang in [47] zusammenfasst.

“In other words, the theorem does not exclude the possibility that there is a machine M , which in fact is equivalent to the mathematical mind. Assume given such a machine M . Gödel infers then, by his theorem, two consequences from the supposition: (1) M ‘cannot be proved to be so’, (2) ‘nor even proved to yield only correct theorems’.”

(1) geht direkt aus dem Gödelschen Unvollständigkeitssatz und seinen Implikationen hervor. Wichtig an dieser Stelle ist der Hinweis, dass mit “mathematical mind” nicht die Mathematik an sich oder eine Maschine, die diese abbildet, gemeint ist, sondern der menschliche Geist, der diese Mathematik erfunden und geprägt hat und sie auch ständig weiter entwickelt. Es geht Gödel hier also tatsächlich um die “echte” Künstliche Intelligenz. Die zweite Teilaussage argumentiert er dabei wie folgt:

“(2) Suppose it could be proved that M yields only correct theorems. In that case, the consistency of M would also be provable by us and, therefore, by the hypothesis on M , a theorem provable by M . But this contradicts Gödel’s theorem. Therefore, (2) is true.”

Eine solche Maschine, die also eigentlich nach den Regeln der Mathematik geformt und gebaut wurde und irgendwann dem menschlichen Geist ebenbürtig wird, könnte demnach auch falsche Aussagen ableiten oder zumindest Aussagen, von denen sie selbst nicht prüfen kann, ob sie wahr oder falsch sind. Aus einem theoretisch deterministischen Apparat würden auf diese Weise quasi-indeterministische Zustände entstehen – diese Zustände wären dann tatsächlich weder vorhersagbar noch reduzibel und demnach synchron bzw. strukturemergent.

Stephan hat in seinem Emergenz-Werk ebenfalls versucht, ein Beispiel für “echtes”, d.h. synchrones und sogar strukturemergentes Verhalten in der Mathematik zu suchen, nachdem er in den Naturwissenschaften ebenfalls auf kein zuverlässiges, unbestrittenes Beispiel gestoßen war. Seine Argumentation¹³, ein

¹³Siehe [46], Seite 242 ff.

solches Verhalten in der Eigenschaft mathematischer Funktionen, die deterministisches Chaos beschreiben, festzumachen, darf aber auch auf Basis der von ihm geleisteten Vorarbeit angezweifelt werden. Denn so beeindruckend die Eigenschaften und Auswirkungen derartiger Funktionen – wie etwa beim Lorenz-Attraktor zur Wetterbeschreibung, aus dessen Repräsentation auch der Name “Schmetterlingseffekt” entstand – auch sein mögen, so sind sie sicher weder neuartig noch irreduzibel. Zu jedem Zeitpunkt, d.h. Status, einer solchen Funktion kann der nächste Zustand eindeutig berechnet werden und die Werte der Zielmenge können auch immer durch die Analyse der jeweiligen Funktionen auf den Einsatz der Werte aus der Definitionsmenge zugeordnet werden. Die besondere Beschaffenheit oder Kombination der Funktionen, die zu stark unterschiedlichem Verhalten bei kleinen Änderungen führen können, sind zweifelsohne interessant, aber keinesfalls emergent. So fundiert und begrifflich exakt die Arbeit Stephans sonst sein mag, dieser Schluss ist etwas voreilig und vielleicht das Resultat des Versuchs, der Begriffsanalyse am Ende doch noch ein hoffnungstragendes Beispiel zur Festmachung echter Emergenz hinzuzufügen.

Zumindest eine bestehende interessante Arbeit, die sich mit Emergenz in mathematischen Systemen beschäftigt, lässt sich dann aber doch noch finden: Alex Ryans¹⁴ Abbildung des Emergenzbegriffs auf geometrische Formen, insbesondere auf die Eigenschaften des Möbiusbands und der Zuordnung emergenter Charakteristika zu den damit verbundenen Begriffen aus der Geometrie ist tatsächlich von großer Relevanz, auch für die in dieser Arbeit getroffenen Hypothesen. Im Abstract heißt es bei Ryan dazu:

“Since its application to systems, emergence has been explained in terms of levels of observation. This approach has led to confusion, contradiction, incoherence and at times mysticism. When the idea of level is replaced by a framework of scope, resolution and state, this confusion is dissolved. We find that emergent properties are determined by the relationship between the scope of macrostate and microstate descriptions. This establishes a normative definition of emergent properties and emergence that makes sense of previous descriptive definitions of emergence.”

Ryans Arbeit trägt den Titel “Emergence is coupled to scope, not level” und seine Argumentation geht in die Richtung, dass Emergenz in der Mathematik an der “Informationsauflösung” und dem Betrachtungsfeld festgemacht wird und nicht – im Sinne einer allgemeinen Objektivität – an unterschiedlichen Stufen zwischen denen es emergente Sprünge geben kann. Emergenz wäre so gesehen einfach nur eine Frage der Definition von Mikro- und Makroeigenschaften. So unterscheidet Ryan dann – wie auch sonst durchaus üblich – zwischen zwei verschiedenen Definitionen für emergente Eigenschaften¹⁵:

¹⁴Siehe Alex Ryan, “Emergence is coupled to scope, not level”, 2006.

¹⁵Siehe ebenda, Seite 8.

“Definition – Weak Emergent Property. A property is weakly emergent iff it is present in a macrostate but it is not apparent in the microstate, and this macrostate differs from the microstate only in resolution. A weak emergent property is a limitation of the observer, not a property of the system.”

Eine schwach emergente Eigenschaft wäre hier also verglichen mit dem Diskurs in der Arbeit von Stephan “noch weniger”, weil sie auf die Rolle des Beobachters, genauer: auf dessen Verfügbarkeit an Information über das System, reduziert wäre und nicht einmal eine systemische Eigenschaft im klassischen Sinn darstellt¹⁶.

“Definition – Novel Emergent Property. A property is a novel emergent property iff it is present in a macrostate but it is not present in any microstate, where the microstates differ from the macrostate only in scope.”

Mit diesem Problem des “Zoomens” in Systemen beschäftigt sich auch Stephan, wenngleich er dieses Problem vom Emergenzbegriff prinzipiell getrennt sieht. Ein Beispiel, das Ryan für diese Art von neuartigen emergenten Eigenschaften bringt, ist die Zerlegung eines Möbiusbands in Dreiecke und die damit verbundenen unterschiedlichen geometrischen Eigenschaften: Hat man nur einzelne Dreiecke “im Blickfeld”, können die geometrischen Eigenschaften des Möbiusbands selbst freilich nicht erkannt werden. Dass es sich dabei aber um Emergenz im eigentlichen Sinn handelt, scheint Ryan nicht annehmen zu wollen, weshalb er schlussendlich auch den Emergenzbegriff mathematisch wie folgt definiert¹⁷:

“Definition – Emergence. Emergence is the process whereby the assembly, breakdown or restructuring of a system results in one or more novel emergent properties.”

Das Beispiel mit emergenten Eigenschaften des Möbiusbands aus mathematischer Sicht ist interessant, die abgeleiteten mathematischen Definitionen zum Emergenzbegriff für den hier vorgestellten Ansatz und in Hinblick auf die prinzipielle Intention dieses Begriffs weniger. Wenn Emergenz nur vom Blickwinkel des Betrachters und dem Informationsgrad, der diesem zur Verfügung steht, abhängt, verschwindet der Kern der Emergenz, die Möglichkeit eines emergenten Sprungs mit allen daraus vermuteten Konsequenzen. Auch eine Anwendung der Definitionen von Ryan auf das vorhin beschriebene Szenario mit dem Gödelschen Unvollständigkeitssatz führt zu Problemen, da der emergente Sprung hier

¹⁶Siehe ebenda, Seite 10.

¹⁷Siehe ebenda, Seite 10.

an der Struktur des Systems selbst ansetzt und nicht an Objekten die sich innerhalb dieser Struktur bewegen und definieren lassen. Eine saubere mathematische Definition des Emergenzbegriffs, die auch allgemeine Zustimmung in einer breiten wissenschaftlichen Öffentlichkeit findet, steht also bis dato noch aus und ist – auch ob der vielen meta-physischen Verknüpfungen – sicherlich nicht leicht zustande zu bringen.

6.2 Die Selbstorganisation des Universums

In seinem Buch über selbstorganisierende Prozesse [24], das 1992 in einer Neuauflage erschien, findet sich bei Erich Jantsch folgender Satz im Vorwort von Peter Kafka: “Ein Sachbuch mit etwa 250 Zitaten aus über 15 Jahre alten wissenschaftlichen Originalarbeiten und Fachbüchern? Welchen Sinn kann es haben, ein solches Buch heute neu aufzulegen?” Wer dieses Buch, das eine Übersicht von Prozessen der Selbstorganisation in unterschiedenen Wissenschaftsgebieten – in seinem Aufbau prinzipiell der Struktur unserer Reductive Levels folgend – heute, 2009, liest, wird sich sofort selbst die Antwort geben können: Die Erkenntnisse und Thesen sind aktuell wie vor 17, wie schon vor 56 Jahren. Die Selbstorganisation als wissenschaftlicher Untersuchungsgegenstand hat das Potential dazu, die wichtigsten Erkenntnisse des 21. Jahrhunderts hervorzubringen.

Den Ursprung der mathematischen Untersuchungen dieser Prozesse, dessen weitreichende Bedeutung Bertalanffy in [4, Seite 39] bereits vorausahnt

“Erst in jüngster Zeit ist eine Verallgemeinerung der Thermodynamik erfolgt mit dem Ziel, Nichtgleichgewichtszustände, offene Systeme und irreversible Prozesse zu erfassen (Prigogine, 1947; Meixner 1949; Denbigh, 1951; de Groot, 1951). Das verspricht, eine beachtliche Entwicklung innerhalb der heutigen Physik zu werden, und eröffnet neue Ausblicke biologischer Phänomene.”

und auch die später erfolgreichen Akteure benennt, liefert Ilya Prigogine mit seiner Beweisführung zur Thermodynamik von offenen Systemen im Fließgleichgewicht, für die er 1977 auch den Nobelpreis für Chemie erhielt. Da ist es kein Zufall, dass Jantsch ihm als “Katalysator des Paradigmas der Selbstorganisation” sein oben erwähntes Buch widmet. Seit damals hat das Konzept – oder an manchen Stellen wirklich das Paradigma – der Selbstorganisation nahezu alle Wissenschaftsbereiche erfasst und um neue Thesen und Erkenntnisse erweitert. Trotzdem kann man sagen, dass es noch nicht gelungen ist, die – vor allem mathematischen – Ideen selbstorganisierender Prozesse in den allgemeinen wissenschaftlichen Diskurs auf der Ebene der universitären Bildung zu integrieren.

Dass Jantsch für Prigogine den Begriff des Katalysators bemüht, ist dabei kein Zufall: Katalytische bzw. Autokatalytische Prozesse sind der Grundbaustein der Selbstorganisation. In Prigogines Nobelpreis-Werk über die Untersuchung der Nichtgleichgewichtsthermodynamik gelingt der Durchbruch zur Auflösung des vermeintlichen Entropie-Widerspruchs durch die Einführung eines chemischen Reaktionssystems mit autokatalytischen Elementen, d.h. mit Elementen, die im System ihre eigene Produktion fördern. Auch in [35, Seite 32 ff.] geht Prigogine auf diese Historie und die Rolle der Autokatalyse ein, Stuart Kauffman ergründet auf diesen Kernprozessen in [25, Seite 287 ff.] sogar eine neue Theorie zur Entstehung des Lebens, die vom mathematischen bzw. wahrscheinlichkeitstheoretischen Standpunkt aus betrachtet den kreationistischen bzw. Intelligent-Design-Argumentationsfiguren wesentliche Erkenntnisse entgegenzusetzen hat (für einen populärwissenschaftlichen Diskurs heute aber wohl noch zu komplex zu sein scheint).

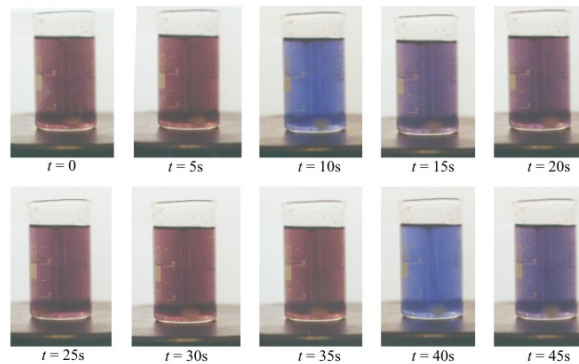


Abbildung 6.2: Belousov-Zhabotinsky-Reaktion. Quelle: Wikimedia.

Auch der in Kapitel 5 vorgestellte Brüsselator ist ein Beispiel für ein autokatalytisches bzw. selbstorganisierendes System: ohne die autokatalytische Transformation $2X + Y \rightarrow 3X$ würde sich sowohl die Entwicklung der Systemkomponenten als auch die Entropieentwicklung im System völlig anders gestalten; und obwohl keines der beteiligten Moleküle über ein ‘höheres Wissen’ bezüglich des Einschwingens des Systems besitzt, so entsteht doch nach kurzer Zeit ein Fließgleichgewicht, das sich bei konstanter Zufuhr der Ausgangsstoffe in einem bestimmten und vorhersagbaren Grenzbereich hält. In der chemischen Praxis lässt sich dieses Konzept sehr gut anhand der Belousov-Zhabotinsky-Reaktion veranschaulichen (siehe Abbildung 6.2): Durch die oben beschriebenen, hier ebenfalls bestehenden Eigenschaften, verhält sich das chemische System als chemischer Oszillator, was für den Beobachter gut aufgrund ständiger Übergänge von einer Rot- zu einer Blaufärbung der Reaktionsflüssigkeit zu sehen ist.

Nun soll Selbstorganisation aber anhand von Beispielen außerhalb der Ursprungsthematik von chemischen Nichtgleichgewichtssystemen veranschaulicht werden, um von der abstrakten Ebene mathematischer Modelle für chemische Reaktionen zu einer die Rolle als naturwissenschaftliches Grundkonzept besser

repräsentierbaren Betrachtungsperspektive zu gelangen. Für eine wissenschaftliche Einführung in das Thema basierend auf aktuellen Arbeiten sei an dieser Stelle insbesondere [12] empfohlen.

Fischschwärme und Soziale Insekten. Bestimmte Fischarten, darunter auch der Atlantische Hering, organisieren sich in Gruppen, was in der Biologie als Schwarmbildung¹⁸ bezeichnet wird. Die Tiere bewegen sich dabei derart synchron und geordnet fort, dass es aussieht, als würden sie in dieser Einheit ein größeres Tier bilden wollen (siehe Abbildung 6.3). Erstaunlich dabei ist auch insbesondere das Verhalten bei einer Störung, etwa wenn ein großer Fisch durch den Schwarm hindurch schwimmt und die Tiere dabei zum Ausweichen zwingt: ebenfalls synchron wird ein Ausweichmanöver vollzogen, um kurze Zeit später wieder den bisherigen Fluss im Schwarm herstellen zu können.



Abbildung 6.3: Photographie eines Fischschwarms. Quelle: sciencephoto.com.

Dieses Verhalten hat durchaus seine Vorteile: Der Schwarm hat insgesamt eine bessere Orientierungsleistung und bietet eine Schutzfunktion gegenüber Feinden. Im Laufe der Evolution haben sich solche Fischarten sogar so stark in der Gruppe entwickelt, dass ihnen Fähigkeiten zum Überleben außerhalb der Gruppe abhanden gekommen sind – so kann sich der Atlantische Hering etwa nur noch begrenzt in seiner Umgebung orientieren, wenn er den Anschluss an die Gruppe verliert. Beim Verhalten des Heringschwarms spricht man von Selbstorganisation, weil jeder Fisch für sich autonom agiert und trotzdem ein übergeordnetes, komplexes System – der Schwarm – mit systemischen Eigenschaften und einem scheinbar wohl überlegten Verhalten entsteht. Die systemischen Parameter sind hier noch nicht so augenscheinlich und markant wie bei anderen Beispielen, es handelt sich daher um eine vergleichsweise “schwache” Variante der Selbstorganisation.

¹⁸Siehe dazu [12], Seite 168 ff.

Auch soziale Insekten, zu denen unter anderem Ameisen und Bienen gezählt werden, organisieren sich in Gruppen – hier ist dann bereits von Insektenstaaten die Rede – und erreichen dadurch als soziales System im Tierreich bestimmte systemische Eigenschaften. Im Gegensatz zu den Fischschwärmen kommt es hier aber bereits zu einer klassischen Ausdifferenzierung der Elemente, was auch in diesem Fall bereits als Arbeitsteilung bezeichnet wird. Bei vielen Ameisenarten werden unterschiedliche Funktionen im Kollektiv auch durch auffällige phenotypische Ausprägungen gestützt. Einzelne Arten, die nur in kleinen Gruppen und ohne diese phenotypischen Differenzierungen bestehen, sind aber ein Indiz dafür, dass sich diese differenzierten Ausprägungen im Laufe der Evolution – wahrscheinlich durch die gesamt effizientere Arbeitsleistung des Kollektivs – entwickelt haben.

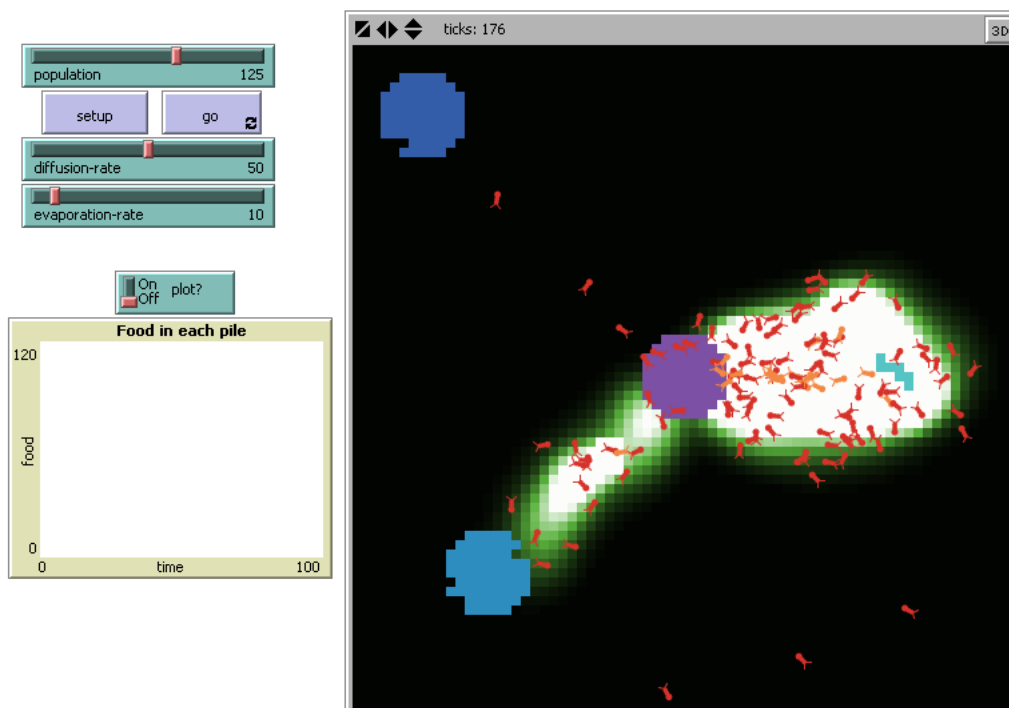


Abbildung 6.4: NetLogo-Simulation zur Erschließung von Nahrungsquellen bei Ameisen.

Das Vorgehen der Ameisen bei der Suche und Erschließung von Nahrungsquellen sowie weitere Arbeitsteilungsszenarien sind in der Informatik wichtige Vorbilder für verteilte Algorithmen, von denen man sich unter anderem Lösungen für harte, komplexe Probleme verspricht. Eine gute Zusammenfassung zu diesem Thema und Beispiele für solche angelehnten Algorithmen finden sich in [8]. Eine einfache Simulation der auf Pheromonen basierenden Entwicklung von Ameisenstraßen findet sich auch in der NetLogo-Model-Library¹⁹, siehe dazu auch Abbildung 6.4.

¹⁹Siehe <http://ccl.northwestern.edu/netlogo/models/Ants>.

Dictyostelium discoideum. Eines der populärsten Beispiele für Selbstorganisation findet sich eine Ebene weiter unten in der Welt der Biologie. Es wird in einigen der im Laufe dieser Arbeit erwähnten Referenzwerke bemüht, unter anderem auch in [12] und [46]. Eine empfehlenswerte Einführung im Web mit Videoaufnahmen findet sich außerdem unter http://www.biologie.uni-hamburg.de/b-online/d27_10/27_10.htm.

Dieses Beispiel ist durch die häufige Referenzierung in der einschlägigen Fachliteratur zum Thema Selbstorganisation geradezu schon klassisch. Es handelt sich hier im Prinzip um einen beobachtbaren Übergang in der Natur von einem einzelligen zu einem vielzelligen Lebewesen. Die Schleimpilzart *Dictyostelium discoideum* besteht aus einzelnen Amöben, die, wenn sie sich in einer Nährstofflösung bewegen, unabhängig voneinander agieren und sich durch einfache Zellteilung vermehren.

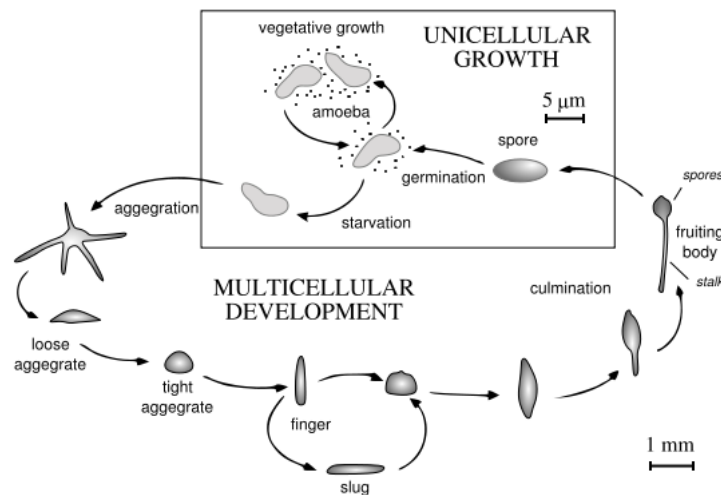


Abbildung 6.5: Verhaltenskreislauf des *Dictyostelium discoideum*. Quelle: Wikimedia.

Wird die Nahrung knapp, verändern sich die Amöben chemisch so, dass sie sich wellenartig zu einer Art Wurm zusammenfinden, der sich dann – von außen betrachtet wie ein neues, selbständiges Lebewesen agierend – lichtgesteuert über eine gewisse Strecke fortbewegen kann. Stößt er dabei auf eine neue Nahrungsquelle, kehrt sich der Prozess um. Wird keine Nahrung gefunden, entsteht ein Schleimpilz mit Sporen, der sich so auch in einer weiter entfernten Umgebung wieder fortpflanzen kann.

Dieser Prozess ist tatsächlich höchst erstaunlich: Einzelne Zellen erreichen nur über chemische Kommunikation, dass sie quasi nach Belieben individuell agieren oder aber einen komplexeren Organismus inklusive Ausdifferenzierung ihrer Rollen darin bilden können. Freilich kann man jetzt sagen, dass es sich hierbei nicht um ein neues komplexes Lebewesen sondern nur um eine, wenn

auch besondere, Organisationsform dieser Einzeller handelt. Tatsächlich ist das Verhalten des *Dictyostelium discoideum* aber eines der besten Erklärungsmodelle und Beispiele bisher um den Übergang einzelligen Lebens zu komplexen Organismen zu veranschaulichen und wird in der Biologie auch tatsächlich so interpretiert. Interessant in diesem Zusammenhang sind auch Analogiebildungen zur Ausdifferenzierung bei der Bildung sozialer Gruppen – die ja ebenfalls auch durch Ressourcenknappheit gefördert werden können – und komplexer Gesellschaftsformen. Obwohl bis heute noch kein vollständiges wissenschaftliches Bild vom Verhalten dieses Schleimpilzes gezeichnet werden kann, so scheinen die chemischen Prozesse dahinter eben auch für die Ausprägung dieses Schleimpilzstatus mit Fruchtkörper und Sporen verantwortlich zu sein: Über Selbstorganisation und nicht über eine “Blaupause” oder eine zentrale Steuerung.

Selbstorganisation in Systemen kann also neue, (schwach) emergente Eigenschaften in (durch Selbstorganisation neu entstehenden) Systemen hervorbringen. Neben bereits genannten sind weitere Beispiele für solche Eigenschaften etwa:

- Die **regulativen Eigenschaften des Marktes** in einer freien Marktwirtschaft. Erste frühe Wirtschaftssysteme (siehe dazu auch Luhmanns “Die Wirtschaft der Gesellschaft”) entwickelten sich über reziprozitäre Prozesse, d.h. über einen durch Erwartungshaltungen motivierten Tauschprozess, der mit der aufkommenden Strukturierung von Stammesgesellschaften immer stärkeren Regulativen und Protokollen unterworfen wurde. Daraus entstanden Zahlungsmittel als Abbilder gesellschaftlicher bzw. wirtschaftlicher Werte, die die wirtschaftliche Kommunikation in der Folge vereinfachten und neue, komplexere Wirtschaftssysteme hervorbrachten. Dass solche Systeme immer komplexere Ausprägungen und in einer ökonomischen Evolution enorme abstrakte Ausprägungen erfuhren, deutet auf homöostatische Eigenschaften und damit auf selbstregulierende Prozesse in der freien Marktwirtschaft abseits staatlich bzw. politisch gesetzter Regulierungen hin. Zusammenfassend könnte man sagen: Die vielzitierte “unsichtbare Hand” in Adam Smiths ökonomischem Hauptwerk ist eine schwach emergente Eigenschaft freier bzw. schwach regulierter Wirtschaftssysteme.
- Der **anthropogene Treibhauseffekt** in seiner seit Beginn der Industrie bestehenden Form als Konsequenz der Organisation der Menschen zu sozialen Gruppen und komplexen, leistungsfähigen Gesellschaften. Auch der Treibhauseffekt selbst und die Herausbildung einer Atmosphäre können als schwach emergente Eigenschaften systemischer Prozesse betrachtet werden: Durch das Wechselspiel von physikalischen und chemischen Prozessen mit dem aufkeimenden pflanzlichen Leben auf der Oberfläche des Planeten entstand eine neue Schicht, die die Bedingungen ebendafür stetig verbesserte und sich heute in einem homöostatischen Zustand befin-

det. Da sowohl das Leben selbst voraussichtlich durch selbstorganisierende Prozesse entstand als auch die ökologischen Stoffkreisläufe (Kohlenstoff-Kreislauf, Stickstoff-Kreislauf, etc.) Konsequenzen selbstorganisierender Teilsysteme (Gletscherbildung, Bakterienaktivitäten und Atmung, industrielle Entwicklung getragen von wirtschaftlicher und gesellschaftlicher Organisation) sind, sind die meisten Eigenschaften des “Ökosystems Erde” direkte oder indirekte Konsequenzen von selbstorganisierenden Prozessen. Diese Tatsache ist insbesondere in Hinblick auf Wahrscheinlichkeitsrechnungen zur Möglichkeit extraterrestrischen Lebens und möglichen Szenarien zur Herausbildung alternativer planetarer Systeme interessant und bisher noch kaum beachtet.

- Die **Bildung von protoplanetarischen Scheiben** bzw. die Bildung von Sonnensystemen mit Sonnen und Planeten als physikalischer Prozess folgt zu Beginn großteils der Gravitationswirkung kleiner Staubpartikel und führt über eine Verdichtung schließlich zur über Druck erzeugten Kanalisierung von Energie und weiters über die dadurch entstehende Gravitation zur Ausbildung der uns bekannten Systeme. Die Wechselwirkung zwischen den Staubpartikeln im All und die daraus resultierenden Prozesse sind ebenfalls ein Beispiel von Selbstorganisation – auf physikalischer Ebene²⁰. Gravitation kann in diesem Zustand der Systementwicklung noch über einfache physikalische Relationen zwischen den Staubelementen modelliert werden. Vielleicht liegt in dieser Herangehensweise auch die Lösung der schwierigen und umstrittenen Interpretation der Gravitationskraft und ihres Ursprungs.

Diese Beispiele sind wichtig, um zu zeigen, dass Selbstorganisation tatsächlich ein Konzept zur Herausbildung und Verhaltensweise von Systemen auf allen Ebenen ist und nicht auf chemische oder biologische Systeme beschränkt ist. Selbstorganisation formal zu beschreiben ist trotzdem oder gerade deshalb kein einfaches Unterfangen. Da wir für den eingangs erwähnten Brüsselator auch eine Simulation basierend auf dem CGST-Modell präsentiert haben, müsste es eigentlich möglich sein, das Konzept der Selbstorganisation in dieses Modell zu integrieren – und zwar so, dass es mittels einfacher Prozeduren möglich ist, zu entscheiden, ob ein System selbstorganisatorische Prozesse und Eigenschaften besitzt, oder nicht.

Eine formale Definition des Konzepts bzw. dieser Idee für unser Modell zu finden ist dabei gar nicht so einfach. Einen guten Ausgangspunkt stellt aber die folgende Formulierung in [23, Seite 169] dar:

“Information in selbstorganisierenden Systemen bedeutet dann: Das System gibt im Prozess der Selbstorganisation der Wirkung ihre

²⁰Siehe [24], Seite 119 ff.

Form, indem es sich selbst formt, sich umformt, neu formt. [...] und daher kann gesagt werden: es informiert sich.”

Tatsächlich scheint die Erzeugung von Information quasi “aus dem Nichts heraus” der springende Punkt und das große Geheimnis selbstorganisierender Systeme zu sein. Masse und Energie können wie wir wissen nicht aus dem Nichts heraus entstehen. Mit Information verhält es sich anders, und die Kraft, mit der sich das System hier münchhausenartig selbst informiert und dadurch den Informationsgehalt stetig steigern kann, scheint Selbstorganisation zu heißen.

6.2.1 Selbstorganisation im CGST-Modell

Wie kann das Konzept der Selbstorganisation jetzt in das CGST-Modell explizit integriert werden – implizit ist es das ja bereits, wie die Beispiele aus Kapitel 5 gezeigt haben? Bei einem gegebenen speziellen Systemmodell S_x oder einer Systeminstanz SI_y wollen wir also einen Algorithmus anwenden können, der das Regelsystem oder das Verhalten der Simulation der Systeminstanz als Input und eine Entscheidung über die Eigenschaft der Selbstorganisation als Output (ja oder nein) besitzt.

Dabei nutzen wir die Eigenschaft aus, dass (positive oder negative) Feedback-Schleifen notwendige Gegebenheiten für Selbstorganisation sind. D.h. wenn es keine Abläufe im System gibt, die als (Auto-)Katalysatoren für (ebendiese) Vorgänge dienen, kann es auch nicht zu einer Selbstorganisation kommen. Für die System-Ontologie ist die notwendige Bedingung daher trivial: Es muss zumindest ein Property mit der Eigenschaft `subPropertyOf elementRelation` geben, da es ansonsten auch keine Abhängigkeit von Elementen voneinander und damit auch kein Feedback geben kann.

Die weitere Eingrenzung im dynamischen Teil ist nicht so einfach: Wie lässt sich positives oder negatives Feedbackverhalten zwischen Elementen formal in einem NetLogo-Programm feststellen, wenn wir aus dem ersten Teil des Tests bereits wissen, dass es Beziehungen zwischen Elementen gibt? Im Prinzip entspricht Feedback hier ja einer Änderung von Parametern oder Aktionen eines Agenten im System durch eine Abhängigkeit oder Kommunikation mit einem anderen Agenten oder indirekt durch Abhängigkeit von einer Systemvariable, die wiederum durch das Agentenverhalten determiniert ist. Es muss also (1) an zumindest einer Stelle im Programm zu einer Änderung von Agenteneigenschaften kommen oder Agentenaktionen ausgeübt werden und dabei (2) für ebendiesen Agenten eine Abhängigkeit bezüglich der Beschaffenheit anderer Agenten, einer Relation oder einer Systemvariable bestehen. Auch hier bieten sich – analog zur Kalkulation von Verhaltensanalogien – wieder unterschiedliche Ansätze an, um ein selbstorganisierendes Set zellulärer Automaten festzustellen bzw. zumindest vorzuschlagen:

- Codebasiert kann ein Compiler, der ähnlich konfiguriert ist wie im Ansatz in Kapitel 5.3, einen Abhängigkeitsgraphen mit den einzelnen Agenten als Knoten konstruieren und feststellen, ob Teilgraphen, die eine direkte oder indirekte Selbstreferenz für einen oder mehrere Agenten bilden, bestehen. Dabei muss auch der Umweg über globale Systemvariablen berücksichtigt werden. Können derartige Teilgraphen identifiziert werden, ist die prinzipielle Voraussetzung für Selbstorganisation durch das Bestehen einer selbstreferenziellen Feedback-Kette gegeben. Das ist aber nur die wichtigste notwendige Bedingung, hinreichend für Selbstorganisation ist sie noch nicht – der Prozess an sich kann ja in ein völlig anderes Verhalten übergehen, wenn die Ausgangsparameter entsprechend eingestellt sind: Denken wir nur an eine zu geringe Konzentration von Eingangsstoffen eines der Moleküle im Brüsselator – in so einem Fall würde ein Wechselprozess wie er im Idealfall entsteht, nie zustande kommen. Die Berechnung der notwendigen Konzentrationen bzw. Systemparameter, um bei einer bestimmten Abhängigkeitskonfiguration auch selbstorganisierendes Verhalten zu erzeugen, ist bis heute in erster Linie eine analytische Vorgehensweise, für die es erst wenige Ansätze zur Automatisierung gibt. Ein codebasiertes Ausschließen explizit nicht-selbstorganisierender Prozesse wäre also möglich; eine Ableitung explizit selbstorganisierender Prozesse aus der Codekonfiguration heraus erfordert jedoch noch weitergehende algorithmische und mathematische Arbeiten zu dem Thema.
- Ein guter Indikator für ein selbstorganisierendes System ist ein Einpendeln des Systems in einem Fließgleichgewicht, d.h. ein sich in einem bestimmten Rahmen haltendes Verhältnis der Systemelemente nach einer gewissen Anfangsphase. Auch durch kurzzeitige, leichte bis mittelfristige Störungen sollte dieses Verhältnis nicht nachhaltig gestört werden können (Homöostase). Mittels entsprechend konzipierter Schnittstellen zu Net-Logo würde sich ein Meta-Simulator, der diese Verhaltenseigenschaften generisch für Zelluläre Automaten prüfen kann, ohne größere ersichtliche Probleme, allerdings mit dem entsprechenden Implementierungsaufwand realisieren lassen: Der Simulator müsste dabei in mehreren aufeinanderfolgenden Schritten (1) den Abhängigkeitsgraphen der Agenten konstruieren, um nur in oben beschriebenen Kandidatenfällen fortzuschreiten, (2) die Systemelemente und den Relationsrahmen (d.h. die Koordinatengrenzen der Trajektorie(n)) berechnen und über einen signifikanten Zustands-Zeitraum hinweg das Verhalten des Systems innerhalb dieses Rahmens überprüfen und (3) “Störaktionen” konstruieren und in das System einspielen, um das Verhalten des Systems (Homöostase-Check) zu überprüfen. Werden alle Meta-Simulationstests erfolgreich von der Automaten-Instanz bestanden, ist sie ein guter Kandidat für ein selbstorganisierendes System.

Freilich würden hier prinzipiell auch noch die Möglichkeiten einer Graphen- und Speicherbasierten Implementierung in Frage kommen, doch gibt es im Zu-

sammenhang mit Selbstorganisation für diese Ansätze noch keine Vorarbeiten, was ein unmittelbares Ansetzen im Rahmen einer praktischen Anwendung sehr schwierig macht. Ganz allgemein ist die Thematik der Selbstorganisation auf dem Gebiet der Informatik bisher noch kaum untersucht worden, obwohl sich hier mit der Möglichkeit, selbst-adaptive und selbstorganisierende Informationssysteme zu entwickeln, ganz neue Möglichkeiten auftun, was die zu erwartende Qualität und Intelligenz derartiger Anwendungen betrifft.

6.2.2 Selbstorganisation und Emergenz in Multi-Level-CGST-Instanzen?

CGST ist in der Form, wie das Modell beschrieben wurde, auch stark beschränkt, was die Simulation der Entwicklung komplexer Systeme betrifft. Übergänge von einem Reductive Level zum nächst höheren können immer nur skizziert bzw. in Ansätzen simuliert werden, nicht jedoch vollständig dargestellt werden: Das liegt in erster Linie am “Platzmangel” des Modells, also an der Beschränkung der Zellulären Automaten bzw. des dahinter befindlichen Speichers und der benötigten Rechenleistung für die Simulationen.

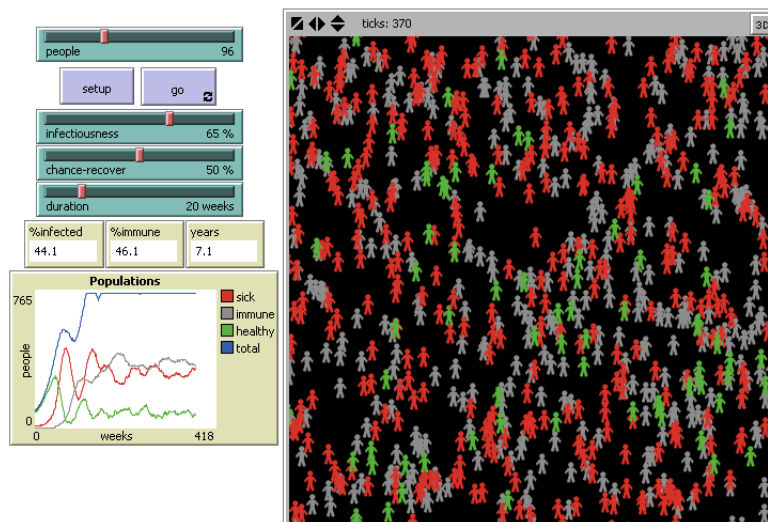


Abbildung 6.6: Implementierung einer Virusepidemie in der NetLogo-Stammbibliothek.

Es kann daher an dieser Stelle nur in Fragmenten gezeigt werden, was in einem Projekt mit entsprechend zur Verfügung stehenden Rechenressourcen umgesetzt werden kann: Eine einzelne dynamische CGST-Instanz, also eine konkrete NetLogo-Simulation, wird als Systemsimulation betrachtet. Gleichzeitig können die Systemeigenschaften dieser Instanz aber auf Attributsausprägungen einer CGST-Instanz, die sich auf einer höheren Reductive-Level-Stufe befindet, abgebildet werden. Die Abbildungsfunktion $f(S, Q, s_x)$ basiert dabei auf einem beliebigen Set an System-Eigenschaften Q des Systems S auf der Ebene n und

bildet einen Zahlenwert (u.U. auch als Codierung für einen boolschen Wert) auf eine einzelne Eigenschaft q_a eines einzelnen Agenten eines Systems auf der Ebene $n+1$ ab.

$$f(S_n, Q, s_x) \rightarrow q_a(S_{n+1}, s_y)$$

Auf der Applikationsebene müssen also Abbildungsrelationen zwischen einer Anzahl von N CGST-Instanzen auf der Ebene n und N Agenten einer Systeminstanz auf der Ebene $n+1$ erzeugt und verwaltet werden. Um einen solchen Vorgang anhand eines Beispiels zu veranschaulichen, soll ein Beispiel aus der Biologie betrachtet werden: In der NetLogo-Stammbibliothek findet sich eine Simulation zur Verbreitung einer Virusepidemie²¹ innerhalb einer Gesellschaft (siehe Abbildung 6.6). Die Ausbreitung bzw. die Übertragung der Viruserkrankung wird dabei sehr einfach über ein Boolean-Wert `sick?` und einen entsprechend randomisierten Ansteckungsprozess bei physischer Nähe realisiert. Einzelne Agenten-Instanzen können außerdem `immune?` sein. Die Implementierung stützt sich dabei auf theoretische Arbeiten zu dem Thema²². Die durch historische Daten und medizinische Erkenntnisse mögliche Berechnung von Parametern, auf deren Grundlage sich Ausbreitungsszenarien berechnen lassen, lässt mitunter sehr präzise Prognosen für etwaige Szenarien unter verschiedenen Umweltbedingungen zu. In der Praxis kommt es trotzdem immer wieder zu überraschenden Entwicklungen, wie zuletzt auch die in den letzten Jahren von Experten oft fehleingeschätzten Ausbreitungen der Vogel- (2007) und Schweinegrippe (2009) gezeigt haben.

Detailliertere Prognosen, die auch die biologischen Eigenschaften des Erregers selbst direkt in die Berechnung bzw. in die Simulation mit einbeziehen könnten dabei zu noch exakteren Prognosemöglichkeiten führen. Für das ausgewählte Szenario würde das etwa bedeuten, die Simulation in einer Multi-CGST-Instanz, die auf der Ebene unterhalb der mehrzelligen Organismen, also auf Zellebene, beginnt, zu realisieren. Abbildung 6.7 zeigt eine CGST-Simulation für die Ausbreitung einer bakteriellen Infektion in einem Organismus auf der Ebene n und die Ausbreitung innerhalb einer Gesellschaft auf der Ebene $n+1$, wobei in dieser ersten Abbildung die fokussierte Agenteninstanz auf der Ebene $n+1$ noch gesund (`sick = false`) ist.

Die Ausbreitung hängt von verschiedenen Parametern ab und basiert auf einer sehr einfachen Organismus-Ontologie, in der krankheitserregende Bakterien wie auch Leukozyten als Abwehrzellen des Organismus enthalten sind und einzelne Zellen vom Erreger befallen werden können. Erreicht der Befall innerhalb des Organismus eine kritische Grenze, gilt der Organismus als krank (`sick = true`). Die Körpertemperatur des Organismus wird vereinfacht über

²¹Siehe <http://ccl.northwestern.edu/netlogo/models/Virus>.

²²Siehe Yorke, et al., "Seasonality and the requirements for perpetuation and eradication of viruses in populations", *Journal of Epidemiology*, Volume 109, Seite 103-123.

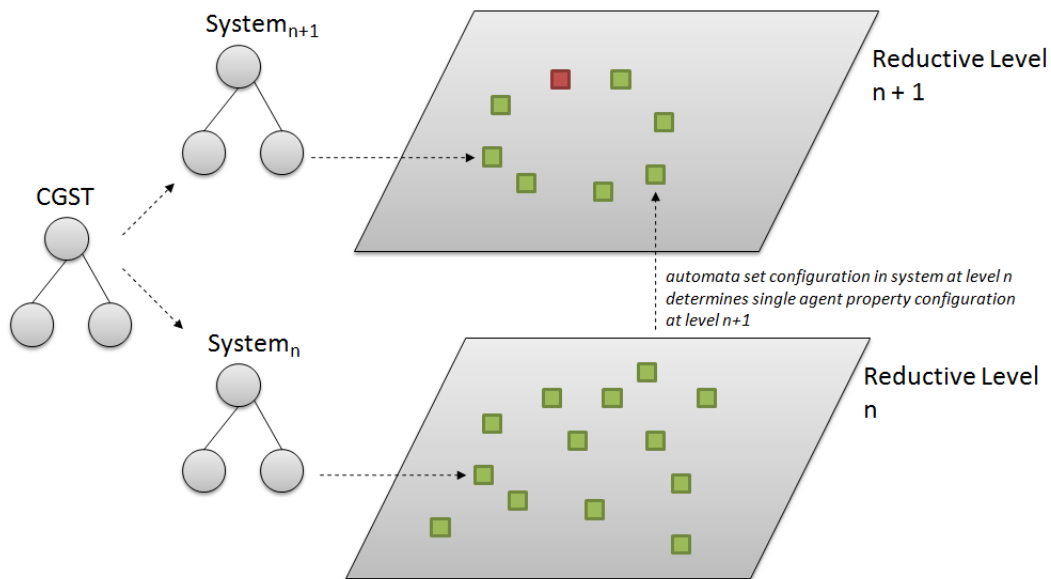


Abbildung 6.7: Abbildung einer Multi-CGST-Instanz. Das System auf Level n beeinflusst den Status einzelner Agenten auf der Ebene $n+1$.

eine Infektionsfunktion aus der Anzahl der Krankheitserreger berechnet und bildet den Gesundheitszustand des Organismussystems ab – ab einer gewissen Temperatur T_{sick} wird die in der Applikation zugeordnete Agenteninstanz auf der Ebene $n+1$ als krank gekennzeichnet (siehe Abbildung 6.8).

Was passiert also in einer solchen Simulation, wenn die in Kapitel 4 erwähnten “spontanen” Code-Mutationen auf allen Ebenen eingeführt werden? Eine Code-Mutation auf einer sehr niedrigen Ebene könnte dann tatsächlich signifikante Auswirkungen auf einer sehr hohen Ebene haben – etwa könnte die Mutation einer Eigenschaft einer Bakterien-Zelle das Aussterben einer Gesellschaft nach sich ziehen oder durch veränderte biologische Grundstrukturen die Verhaltensmuster derart beeinflussen, dass sich eine Gesellschaft neu organisiert.

Dass es sich bei der Eigenschaft `sick?` um eine systemische, d.h. schwach emergente Eigenschaft handelt, ist dabei nicht nur intuitiv, sondern auch formal augenscheinlich. Die Entstehung solcher schwach emergenten Eigenschaften in einem Multi-CGST-Szenario lässt sich also recht einfach skizzieren: Per Definition handelt es sich bei jeder Ebenen-überschreitenden Abbildung auf ein Agenten-Attribut um eine solche.

Da die systemische Eigenschaft prinzipiell auch über einen Selbstorganisationsprozess entstehen könnte (etwa im Falle anthropogener Effekte in Ökosyste-

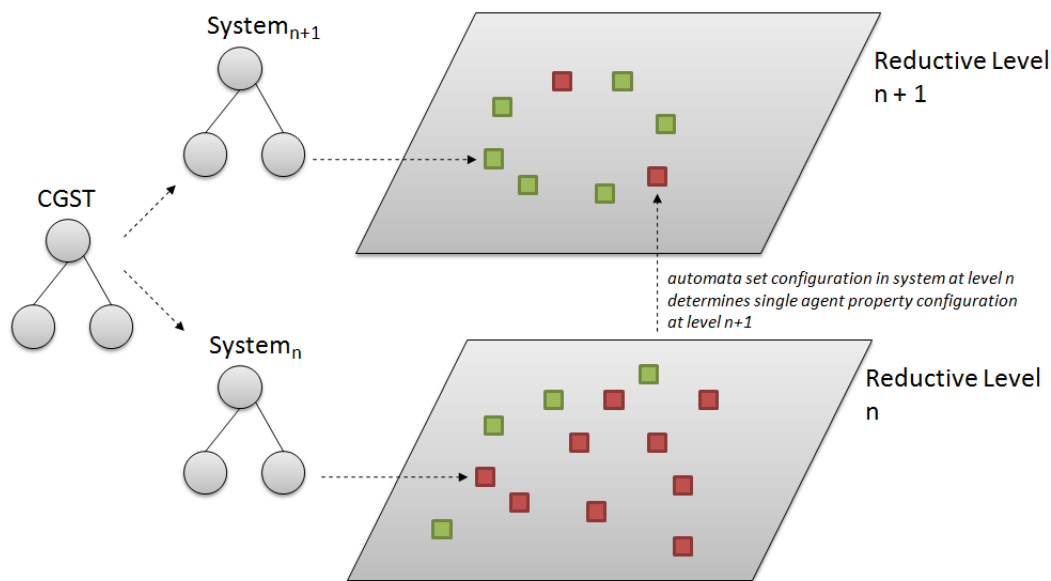


Abbildung 6.8: Anschauungsszenario für eine derartige Abhängigkeit und die stufenübergreifende Kausalität wäre z.B. eine sich ausbreitende Virusinfektion in einem Organismus (Level n) und innerhalb einer Bevölkerung (Level $n+1$).

men und deren Auswirkung auf bestimmte Parameter im Ökosystem), wären über mehrere Reductive-Levels konstruierte Multi-CGST-Instanzen mit Sicherheit interessante Applikationen, um Experimente bzw. Simulationen zur Untersuchung komplexer Systeme durchzuführen und daraus neue Erkenntnisse, vor allem auch auf einer formalen Ebene, zu gewinnen. Noch interessanter wird es jetzt allerdings, wenn man fragt, ob ein Szenario folgender Art möglich ist bzw. wie das damit verbundene Verhalten interpretiert werden kann:

Ausgangsbasis ist eine C-GST-Instanz $\langle S, Q, s_0 \rangle$ wobei S das statische Systemmodell inklusive statischer Regeln, Q die Ausgangsbasis und Dynamik des Zellulären Automaten und s_0 den Anfangszustand der Instanz beschreibt. $\langle S, Q, s_0 \rangle$ sei widerspruchsfrei und entscheidbar. Was kann oder muss jetzt passieren, um einen Zustand $\langle S, Q, s_t \rangle$ zu erreichen, der nicht mehr entscheidbar ist, d.h. der zumindest ein nicht entscheidbares Teilproblem und damit einen Konflikt in der Determinationskette beinhaltet, enthält?

CGST verwendet Full-OWL, damit kann es zu TBox-ABox-Kombinationen kommen, die nicht entscheidbar sind. Dazu muss angemerkt werden, dass das W3C bei der Konzeption von OWL und der Zerlegung in die drei Untersprachen OWL Full, OWL DL und OWL Lite explizit darauf Rücksicht genommen hat, bekannte Paradoxien aus der Logik zu vermeiden. Ganz ist das aber, vorausge-

setzt man möchte eine gewisse Ausdrucksstärke bewahren, nicht möglich. Ein Beispiel, das unentscheidbar ist und ob seiner Charakteristika durchaus auch aus GST-Sicht interessant sein kann, ist das *Postsche Korrespondenzproblem*, das auch als *Domino-Tiling-Problem* in einer etwas anschaulicheren Variante in der Theoretischen Informatik behandelt wird²³. In einer Arbeit von Boris Motik²⁴ findet sich eine Repräsentation dieses Problems in OWL Full, die somit ein praktisches Beispiel für ein unentscheidbares Problem in OWL Full darstellt.

Das Domino-Tiling-Problem beschreibt im Wesentlichen eine $N \times N$ -Matrix von Dominosteinen, die aus unterschiedlichen Farben bestehen und einer Reihe von Regeln, welche Farben aneinandergereiht werden dürfen und welche nicht. In der Matrix stellt sich nun folgendes Problem: Unter der Annahme, dass es für jede Farbe unendlich viele Steine gibt, gibt es für ein spezielles Set an Regeln und ein bestimmtes Set an Farben eine Lösung für das Problem? Tatsächlich gibt es Sets, für die das Problem nicht entscheidbar ist. Da das Problem mittels OWL Full beschrieben werden kann, würde ein Reasoner, der versucht, diesen Zustand aufzulösen, immer scheitern. Ein Kritikpunkt am Beispiel von Motik ist jedoch, dass die gewählte Repräsentation Ausdrücke enthält, die ein Ontology-Designer wohl niemals in der Realität so verwenden würde, wie etwa

(11.7) `GRID rdfs:subClassOf owl:allValuesFrom`

und

(11.10) `rdf:type owl:sameAs owl:onProperty`

11.7. beschreibt, dass `GRID` eine Unterklasse von `owl:allValuesFrom` ist, `owl:allValuesFrom` ist aber nur aufgrund der Eigenheiten von OWL Full als Klasse repräsentiert und nicht dazu gedacht, sie auch in dieser Form so zu verwenden – diese Aussage bedeutet ja: `GRID` ist eine Unterklasse des Konzepts, das die Einschränkung auf eine Liste von Werten definiert. 11.10 ist noch seltsamer und beschreibt, dass `rdf:type` und `owl:onProperty` äquivalent sind – auch diese Aussage kann wohl designtechnisch als “Unfug” bezeichnet werden, syntaktisch und im Rahmen von OWL Full ist sie aber prinzipiell erlaubt.

Nun ist es schwer vorstellbar, dass derartige Aussagen in einer ausgearbeiteten Ontologie auftreten. Sehrwohl vorstellbar wäre aber, dass sie das Produkt einer zufälligen Code-Mutation auf der TBox-Ebene einer Systeminstanz sind.

²³Bei Hao Wang, auf den die ursprüngliche Beschreibung dieses Problems zurückzuführen ist, handelt es sich um jenen Logiker, der auch in [47] die metaphysische Auseinandersetzung Gödels mit dessen Resultaten zum Unvollständigkeitssatz dokumentiert hat.

²⁴Siehe Boris Motik, “Reasoning in Description Logics using Resolution and Deductive Databases”, 2006, Seite 186 ff.

D.h. im Prinzip wurde hier gezeigt, dass bei der Möglichkeit, “zufällige” Codemutationen auf der Ebene der Ontologie zuzulassen, durchaus eine Situation entstehen kann, die unentscheidbar ist. Würde der dynamische Teil jetzt eine Zustandsregel

Setze $a = 0$ wenn X und $a = 1$ wenn $\neg X$

beinhalten und X der Query “Ist P lösbar?” wobei P das oben beschriebene Dominoproblem beschreibt, entsprechen, würde der dahinterstehende Reasoner vor einem unlösbaren Problem stehen und es nie zu einer Fortführung des Automaten kommen. Interessant aus GST-Sicht sind hier jetzt noch die Arbeiten von Wang und Winfree. Hao Wang hat ein analoges Problem, die so genannte Wang-Parkettierung²⁵, beschrieben, die oft für Turingmaschinen-Tests verwendet wird. Hier werden die Dominosteine durch quadratische “Parketteinlagen”, die aus vierfärbigen gleichseitigen Dreiecken bestehen, ersetzt. Erik Winfree hat eine Portierung des Problems zur Konstruktion von Molekülen aus DNA vorgenommen²⁶. Das Problem hat also durchaus auch einen Bezug zu wissenschaftlichen Anwendungen außerhalb der theoretischen Informatik.

Angenommen $\langle S, Q, s_0 \rangle$ beschreibt ein System von Zellen mit einer DNA, wobei auch Codemutationen auf S zulässig sind. Das System enthält außerdem folgende Regel zur Beschreibung eines Genotype-Phenotype-Verhältnisses: Wenn ein Muster wie oben beschrieben für ein bestimmtes Teilset erzeugbar ist, erhält der Organismus die Eigenschaft P_a , wenn ein solches Muster nicht erzeugbar ist, erhält er die Eigenschaft $P_{\neg a}$. Der Organismus wird dabei über ein System in der nächst höheren Stufe einer Multi-Level-CGST-Instanz beschrieben und P_a bzw. $P_{\neg a}$ beschreiben einander ausschließende Eigenschaften des Systems $\langle S_{n+1}, Q_{n+1}, s_t \rangle$. An dieser Stelle muss CGST aus den oben genannten Gründen “scheitern”. Für eine weiterführende Interpretation dieses Ablaufs stellt sich nun aber eben jene entscheidende Frage: So es solche Vorgänge in der Natur tatsächlich geben sollte (d.h. diese Resultate aus der Mathematik auch tatsächlich eine Relevanz in der Natur besitzen), wie sieht dann das Verhalten ebendieser in einer solchen Situation aus? Und tatsächlich wäre hier eine mögliche Antwort ein stark emergenter Sprung.

6.3 Die Münchhausen-Vermutung der Selbstreferenz

In den letzten Abschnitten wurden zwei wesentliche Thematiken – die der (starken) Emergenz und die der Selbstorganisation – aus einem ganz bestimmten

²⁵Siehe Hao Wang, “Proving theorems by pattern recognition-II”, Bell System Tech Journal 40(1), 1961.

²⁶Siehe Erik Winfree et al., “Design and Self-Assembly of Two-Dimensional DNA Crystals”, Nature 394, Seite 539-544, 1998.

Blickwinkel und mit bestimmten Annahmen, zu denen es durchaus gegenteilige geben mag, betrachtet. Dabei lässt sich eine interessante Gemeinsamkeit feststellen, die an dieser Stelle noch einmal gesondert hervorgehoben werden soll.

Es scheint so, als ob als einziger “echter Kandidat” für ein stark emergentes Verhalten zum heutigen Zeitpunkt die Mathematik und andere formale Systeme beschrieben werden können. Nur in der Mathematik erreichen wir einen offensichtlich bewiesenen Qualitätsanspruch für starke Emergenz: In der traditionellen Zahlentheorie, die sich durch einfache Regeln erzeugen lässt, gibt es in Verbindung mit logischen Aussagen innerhalb dieses Systems Zustände, von denen basierend auf eben jener Logik nicht festgestellt werden kann, ob es sich dabei um wahre oder falsche Aussagen handelt. Verantwortlich dafür ist – im Gödelschen Beweis – die Möglichkeit der Selbstreferenz in solchen Aussagen. Ebenso benötigt das Halteproblem die Selbstreferenz – das universelle Entscheidungsprogramm, das an seinem eigenen Code scheitert – um zu einem echten Problem zu werden.

Auf der anderen Seite finden wir die Autokatalyse als zentrales Element der Selbstorganisation und selbstorganisierender Systeme. Diese Form der selbstreferenzierenden Unterstützung im Aufbau von Systemen scheint – wenngleich mathematisch zum heutigen Zeitpunkt noch ungleich weniger untersucht wie Gödels Unvollständigkeitssatz – maßgeblich für die Entstehung von immer komplexeren Systemen und die damit verbundene “spukhafte Vermehrung” von Information zu sein. Auch hier ist aber die Selbstreferenz ein entscheidendes und notwendiges Konzept: Ohne sie würde es für Prigogines Nichtgleichgewichtsthermodynamik kein erfolgreiches Modell geben – keinen Brüsselator und auch sonst keine Lösung für Systeme, die einen durch Energieaustausch vollzogenen Entropieexport durchführen und so nicht dem Wärmetod²⁷ ausgeliefert sind.

Zusammengefasst liegt deshalb die Vermutung nahe: Die Welt, wie wir sie kennen, die immer komplexere Systeme ausbildet und in der es Zustände gibt, die in ihrer formalen Abbildung nicht entscheidbar sind, wäre ohne die prinzipielle Möglichkeit der Selbstreferenz nicht möglich. Ohne Selbstreferenz gibt es keine Vermehrung von Information auf der einen und keine Unentscheidbarkeit auf der anderen Seite und damit nur eine theoretisch mögliche Sammlung von Elementen, die sich zwar in Zustandsübergängen neu ordnen, dabei aber keine neue Information erzeugen können. Diese Vermutung wollen wir ob der Analogie

²⁷Als Wärmetod wird der absolute Gleichgewichtszustand, d.h. der Zustand maximaler Entropie, von dem aus es keine Weiterentwicklung mehr geben kann, bezeichnet. Geschlossene thermodynamische Systeme unterliegen dem Zweiten Hauptsatz der Wärmelehre, wonach ein System in Folgezuständen immer nur eine gleiche oder höhere Entropie, jedoch keine Verminderung derselben, erreichen kann. Beim Stillstand des Systems, dem Erreichen des maximalen Entropie-Zustands, spricht man aufgrund der Verknüpfung mit der Wärmelehre vom Wärmetod. Nur offene Systeme, die einen Energieaustausch mit ihrer Umwelt vornehmen, können dadurch eine Entropiereduktion erreichen und damit “dem Wärmetod entfliehen”.

zum Aufziehen des Baron Münchhausen am eigenen Schopf die “Münchhausen-Vermutung der Selbstreferenz” nennen. Eine wichtige Schlussfolgerung dieser Vermutung ist dann: Ein Universum ohne Selbstreferenz stirbt den Wärmetod, bei einem Universum mit Selbstreferenz darf nach heutigem Wissensstand darüber gestritten werden, ob es den Wärmetod stirbt.

Dieser Punkt ist eine der heute wesentlichen Argumentationsfiguren und Diskussionsinhalte der modernen theoretischen Physik und gilt als höchst umstritten. Obwohl der Mensch auch seine weitere Umgebung, also das Universum, bereits ganz gut analytisch beobachten und erklären kann (siehe Abbildung 6.9), ist über die Ebene des Universums hinaus keine weitere systemische Ebene bekannt oder zumindest nachweisbar. Daher fokussieren sich elementare physikalische wie auch systemische Fragen auf diesen Punkt, ohne jedoch eine zufriedenstellende Antwort in Aussicht zu haben.

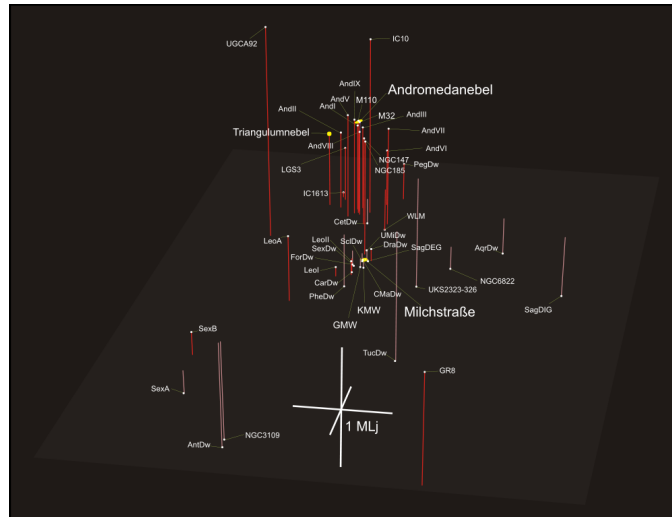


Abbildung 6.9: Mit heutigen technischen Messgeräten können sogar Modelle für Galaxien, die ebenfalls systemisch beschrieben werden können, erstellt werden. Auch Simulationen von Galaxien-Kollisionen und die der dahinter stehenden Dynamik sind möglich. Eine Antwort auf die Frage nach den Charakteristika des Universums als offenes oder geschlossenes System geben freilich auch diese Arbeiten nicht. Quelle: Wikipedia.

So die Vertreter der Selbstorganisation auf dem richtigen Weg sind und dieses neue wissenschaftliche Paradigma tatsächlich Antworten auf ganz wesentlichen Fragen unserer Zeit – Entstehung von Leben, Entstehung von Bewusstsein, Evolutionary Development, Künstliche Intelligenz, u.v.a. – liefert; so der Gödelsche Beweis tatsächlich die metaphysische Bedeutung einer echten Unentscheidbarkeit in allen entsprechenden Systemen besitzt und nicht nur ein interessantes Resultat innerhalb der Mathematik darstellt; dann scheint das Konzept der Selbstreferenz tatsächlich eines der wichtigsten, wenn nicht überhaupt das *wichtigste wissenschaftliche Konzept überhaupt* darzustellen.

Und zwar in vielen unterschiedlichen Wissenschaftsbereichen und damit natürlich auch in einer Allgemeinen Systemtheorie. Das bedeutet dann zum einen, dass Systeme, die durch selbstreferenzielle Komponenten das Potential zur Komplexitätsausbildung besitzen, ganz besonderes Interesse verdienen sollten, zum anderen, dass eine Allgemeine Systemtheorie der Untersuchung der Selbstreferenz eine ganz besondere Rolle zuschreiben sollte und in der vergleichenden Analyse und Beschreibung dieses Problems vielleicht nicht nur eine wichtige Aufgabe, sondern auch das notwendige existentielle Kernelement zur Rechtfertigung einer solchen Meta-Wissenschaft findet.

Denn wie zu Beginn dieser Arbeit ausgeführt wurde, war neben den fehlenden Möglichkeiten zur praktischen Anwendung die fehlende Perspektive eines Potentials, wesentliche Beiträge zu den großen Fragen und Problemen der Wissenschaft im Speziellen und der Gesellschaft im Allgemeinen zu liefern, ein Hauptgrund für das Verschwinden einer globalen Bewegung zur Ausarbeitung einer Allgemeinen Systemtheorie. Die Fokussierung auf das Paradigma der Selbstorganisation in allen Bereichen unseres Lebens und der Wissenschaft sowie die damit verbundene Aufklärung der Emergenzhypothese könnten eine solche Perspektive bilden und als Motivation zur Wiederaufnahme und Weiterverfolgung des Bertalanffy-Programms dienen.

Kapitel 7

Integration bestehender Wissensbasen

*“Alle Sätze der Logik sagen aber dasselbe.
Nämlich nichts.”
Ludwig Wittgenstein, “Tractatus logico-philosophicus”*

Eine der Anforderungen an das CGST-Modell war zu Beginn, dass sich auch bereits bestehende formale Wissensbasen mit vertretbarem Aufwand so auf das Modell abbilden lassen, dass Simulationen und Berechnungen, die darauf basieren, möglich sind. In diesem Kapitel wollen wir abschließend noch anhand von drei Beispielen zeigen, dass die Auswahl der entsprechenden formalen Sprachen und die Art und Weise, wie diese im CGST-Modell kombiniert wurden, tatsächlich auch die Integration bestehender Wissensbasen erlauben. Wir werden dazu drei unterschiedliche Use-Cases so aufbereiten, dass wir den Integrationsweg und die Ergebniskalkulation für ausgewählte Teilprozesse detailliert genug veranschaulichen, um dem Leser die Abarbeitung weiterer Fälle zu ermöglichen.

Wir haben uns dabei für folgende Szenarien entschieden: (1) Die Abbildung einer bestehenden Ontologie im Bereich der Molekularbiologie auf die Systemontologie und (2) den Import der System-Ontologie und die anschließende Verknüpfung mit nachbearbeiteten Ontologien auf kollaborativen Ontologieentwicklungsplattformen. Dadurch sollen unterschiedliche Anknüpfungspunkte bestehender Anwendungen und Daten an das vorgestellte Modell skizziert werden.

7.1 Proteine als Systeme

Im Web gibt es bereits eine beachtliche Anzahl an Ontologien zur Beschreibung unterschiedlicher Domänen. Ein Weg, um gezielt nach Ontologien und Instanzdaten für bestimmte Bereiche zu suchen ist die Semantic-Web-Suchmaschine Swoogle¹. Daten, die sich aufgrund ihrer Modellierung und der Art und Weise, wie sie der Semantic-Web-Community zur Verfügung gestellt werden, an bestimmte Richtlinien halten, werden außerdem in der Linked Open Data (LOD)² Initiative gesammelt und untereinander verknüpft. LOD ist ein wichtiger Schritt innerhalb des Semantic Web zur ursprünglichen Vision einer großen, weltweit verknüpften und verteilten Datenbasis, die aufgrund der integrierten Semantik weit intelligentere Anwendungen erlaubt, als dies derzeit der Fall ist.

In diesem bestehenden Datensortiment finden sich auch viele potentielle Kandidaten, die als Repräsentationen von Systemen interpretiert werden können und demnach eine Abbildung auf unser CGST-Modell möglich machen – der größere Teil dabei, das muss hier auch erwähnt werden, eignet sich allerdings nicht dazu, weil z.B. eine RDF-Repräsentation von Bilddaten (flickr.org) oder Benutzerbewertungen zu Produkten (revyu.org) keine ontologischen Beschreibungen von Systemen in unserem Sinn darstellen.

Eine jener Ontologien, die hinreichende Anknüpfungspunkte bietet, ist die Protein-Ontologie³ des “Protein Ontology Consortium”. Das Ziel und die Beschaffenheit der seit 2002 entwickelten und ständig erweiterten Ontologie wird auf der Webseite der Ontologie beschrieben:

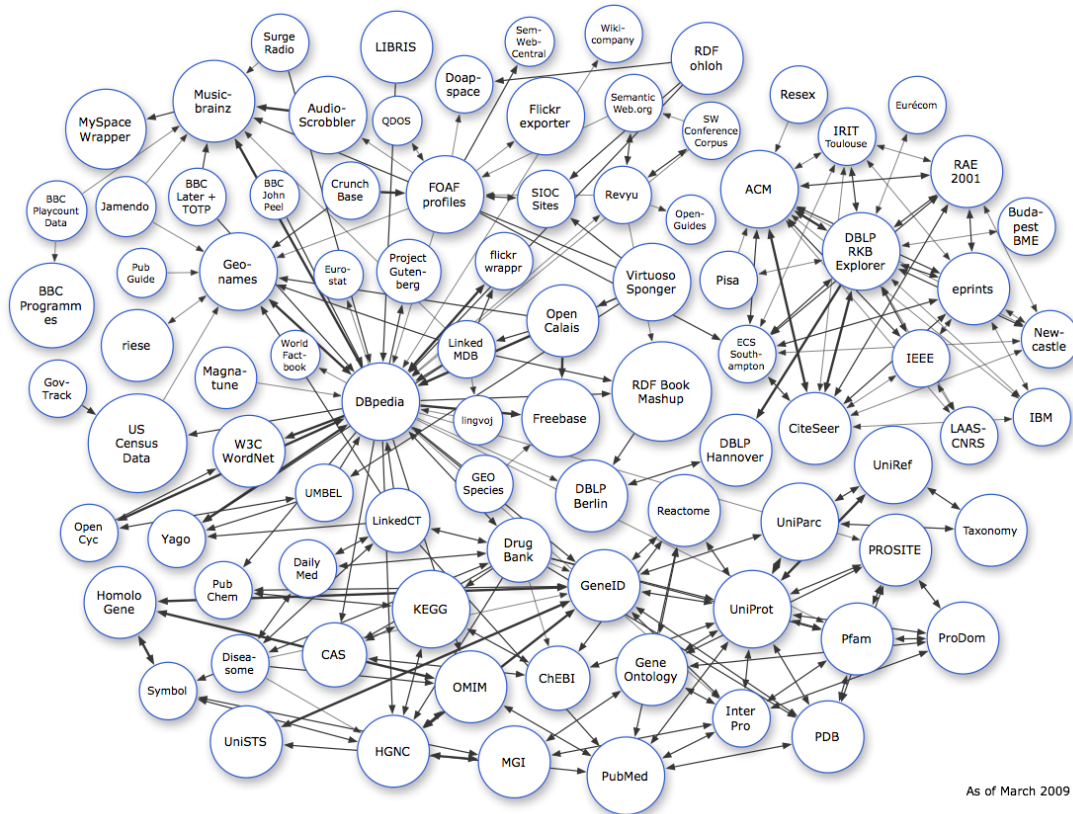
“Protein Ontology (PO) is a means of formalizing protein data and knowledge; protein ontology includes concepts or terms relevant to the domain, definitions of concepts, and defined relationships between the concepts. PO integrates protein data formats and provides a structured and unified vocabulary to represent protein synthesis concepts. PO provides integration of heterogeneous protein and biological data sources. PO converts the enormous amounts of data collected by geneticists and molecular biologists into information that scientists, physicians and other health care professionals and researchers can use to easily understand the mapping of relationships inside protein molecules, interaction between two protein molecules and interactions between protein and other macromolecules at cellular level. PO also helps to codify proteomics data for analysis by researchers. ”

Anhand dieser Beschreibung lässt sich schon vermuten, dass durch die ähn-

¹Siehe <http://swoogle.umbc.edu/>.

²Siehe <http://linkeddata.org/>.

³Siehe <http://proteinontology.org.au/proteins.php>



As of March 2009

Abbildung 7.1: Schematische Darstellung von Linked Open Data (LOD) mit unterschiedlichen Quelldaten und Verknüpfungen. Quelle: linkeddata.org

liche Intention, nämlich den systemischen Ansatz, eine Abbildungsmöglichkeit gegeben ist. Sehen wir uns dazu zuerst die Proteinontologie gemeinsam mit der CGST-Ontologie in einem Ontologieeditor an (Abbildung 7.2).

Sie beinhaltet Konzepte für Atome, Verbindungen, Moleküle, Proteine und Klassifikationen ebendieser, wobei die den Konzepten zugeordneten Attribute in erster Linie Annotationen und natürlichsprachliche Beschreibungen und nur vereinzelt detailliertere Eigenschaften zur Ausdifferenzierung auf einer Wertebene beinhalten. Im nächsten Schritt müssen wir einen Blick auf die Dokumentation⁴ der Ontologie werfen, um den Aufbau und die Zusammenhänge der einzelnen Konzepte auch richtig zu interpretieren – das ist natürlich kein einfacher Schritt, da für diese Arbeit ein gewisses Domänenwissen notwendig ist. Ausnahmsweise versuchen wir als Informatiker uns trotzdem in einer vernünftigen Zuordnung, obwohl in späterer Folge bei Verwendung des CGST-Modells dieser Schritt in der Regel immer von einem Domänenexperten vorgenommen wird. Außerdem können wir nicht die gesamte Ontologie behandeln – diese umfasst viele Millionen Triple –, sondern müssen uns auf ein kleines repräsentatives Subset beschränken. Abbildung 7.2 zeigt unterschiedliche Konzepte in der

⁴Siehe <http://proteinontology.org.au/documentation.htm>.

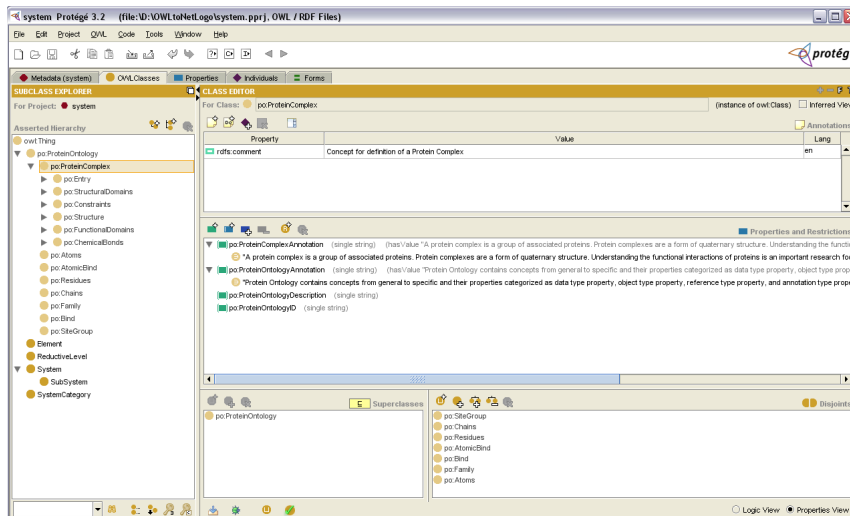


Abbildung 7.2: System- und Proteinontologie im Ontologieeditor Protege.

Protein-Ontologie, insbesondere unterschiedliche Arten von Proteinkonzepten. Wir wollen uns aber in unserem Showcase auf ein leichter zu differenzierendes Konzept fokussieren und werden uns im Detail mit der Abbildung der Klasse `protein:Residues` beschäftigen.

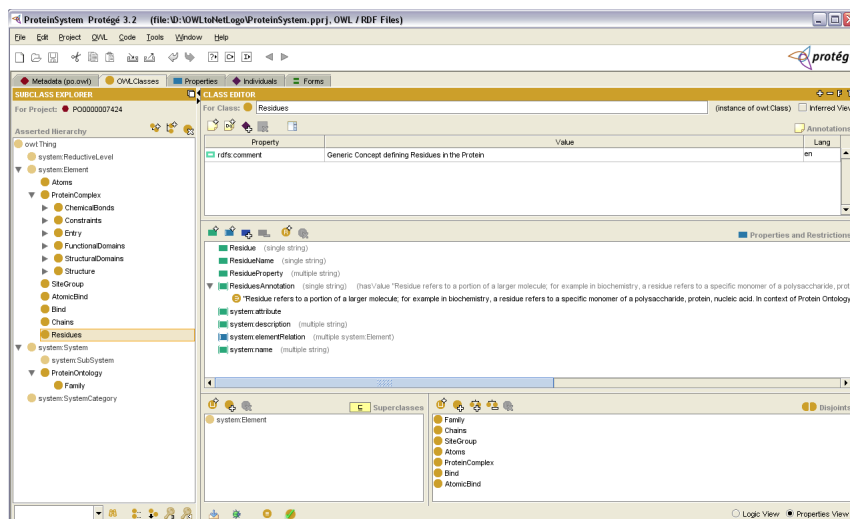


Abbildung 7.3: Abbildung der Konzepte der Proteinontologie auf die Konzepte der Systemontologie.

Vorher können wir aber auf jeden Fall eine Zuordnung der einzelnen Klassen der Protein-Ontologie zu unseren System-Konzepten durchführen. Wir verfolgen dabei den Ansatz, dass wir alle generischen Konzepte⁵ der Ontologie als Elemente eines Protein- bzw. Zellsystems interpretieren. Diese Elemente besit-

⁵http://proteionontology.org.au/documentation.htm#Protein_Ontology_Generic_Concepts

zen in weiterer Folge auch Eigenschaften wie Koordinatenangaben, mit denen wir uns aber an dieser Stelle nicht weiter beschäftigen werden. Eine einzige Ausnahme, für die eine Zuordnung zum Element-Konzept nicht korrekt wäre, ist das `protein:Family`-Konzept⁶, das eine qualitative Klassifikation, allerdings kein eigenständiges, diversifiziertes Element darstellt. Die dadurch entstehende Abbildung der Konzepte auf die Konzepte der System-Ontologie ist in Abbildung 7.3 dargestellt.

Interpretiert werden kann das am Beispiel des `protein:Residues`-Elements dann so: Dieses Konzept, das laut der Dokumentation

“Residue refers to a portion of a larger molecule; for example in biochemistry and molecular biology, a residue refers to a specific monomer of a polysaccharide, protein or nucleic acid. In context of Protein Ontology residue represents one of the 20 amino acids that build protein sequence represented by a three-alphabet code like: CYS.”

Rückstände chemischer Prozesse in Form ausgewählter Moleküle repräsentiert, ist in einem System auf der Ebene von Molekülen oder – da in der Protein-Ontologie auch Zellen vorkommen – Zellen als autonom agierendes Element zu referenzieren. Dadurch ergeben sich dann auf der Ebene der einzelnen Rückstands-Instanzen zusätzliche Annotationen wie sie in dem Auszug in Abbildung 7.4 dargestellt sind.

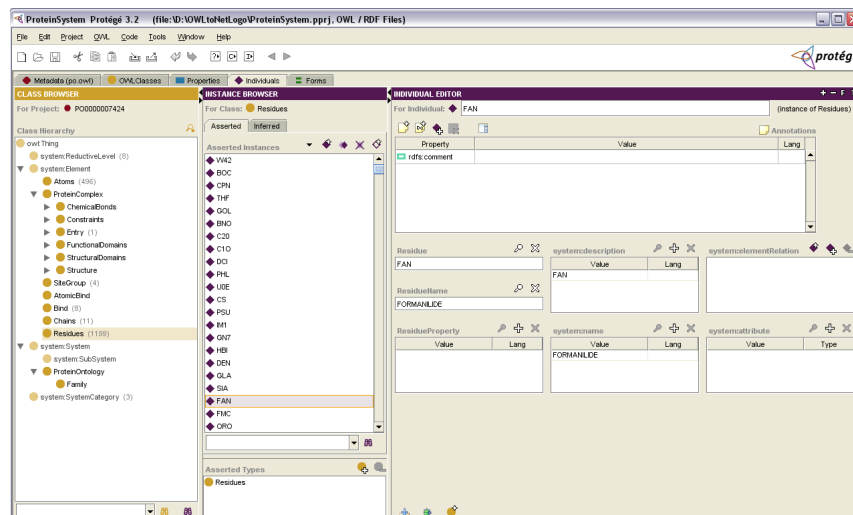


Abbildung 7.4: Auswirkungen der System-Abbildung auf Instanzebene.

Die Protein-Ontologie ist nicht auf Basis des CGST-Modells entstanden. Umgekehrt wurde beim Design der CGST-Ontologie auch keine Rücksicht auf

⁶http://proteionontology.org.au/documentation.htm#Family_Concept

die speziellen Parameter der Protein-Ontologie genommen. Trotzdem ist wie wir sehen eine hinreichend sinnvolle Abbildung möglich. Diese “hinreichend sinnvolle Abbildung” ist aber natürlich aufgrund der prinzipiellen Unabhängigkeit der beiden Designvorgänge nicht optimal und auch nicht völlig unproblematisch. Für diesen speziellen Fall können etwa die folgenden kritischen Punkte vermerkt werden, die aber die grundsätzlich erfolgreiche Abbildung nicht unpraktikabel machen:

- Die Protein-Ontologie hat durch die oben beschriebene Ausrichtung zwar einen impliziten systemischen Charakter, verletzt dieses systemische Paradigma aber an mehreren Stellen (Beispiel: `protein:Family`-Konzept wird auf der selben Ebene definiert wie elementare Konzepte wie `protein:Residues`). Dadurch ist ein Mapping auf die System-Ontologie nur partiell möglich.
- Der systemische Ansatz bezieht sich auf die Ordnung der Konzepte und die Beziehungen zwischen ebendiesen, hat aber keine Übertragung in dynamische Simulationen in Ausblick, wodurch die Abdeckung atomarer Element-Attribute sehr gering ausfällt. Dadurch ergibt sich auch ein sehr eingeschränktes Attributfeld bei der automatischen Erzeugung eines Zellulären Automaten.

Diese Problematiken werden umso deutlicher, wenn wir uns zwei weitere Ontologien aus ähnlichen Domänen ansehen, für die sich ein Mapping schon schwieriger gestaltet. Die GeneOntology⁷ ist ein klassisches Beispiel für den Umgang mit Ontologien in den Bio-Wissenschaften. Der Unterschied zur Protein-Ontologie wird schon bei der einführenden Beschreibung deutlich:

“The Gene Ontology project provides a controlled vocabulary to describe gene and gene product attributes in any organism.”

Durch die Schwerpunktsetzung auf ein strukturiertes Vokabular ohne jeglichen systemischen Hintergrund ist das Produkt eine Klassifikation die ein intelligentes Biologie-Wörterbuch realisiert, nicht aber jene Ziele, die das Semantic Web mit seinen Ontologien genauso anvisiert wie CGST, nämlich die maschineninterpretierbare ontologische Beschreibung von Domäneigenschaften, erreicht.

7.2 Kollaborative Ontologieentwicklung

Ein großer Vorteil von CGST gegenüber anderen Ansätzen zur Realisierung einer vergleichenden Allgemeinen Systemtheorie ist die bereits weite Verbreitung der dahinter liegenden Datenstrukturen. Darüber hinaus haben sich aber

⁷Siehe <http://www.geneontology.org/index.shtml>.

in den letzten Jahren im Semantic-Web-Bereich auch verstärkt Ansätze zur kollaborativen Entwicklung von Schemata herausgebildet: “Avoiding unnecessary Duplication of Labor” bedeutet ja auch, dass man in der Praxis möglichst effizient mit anderen Personen aus dem selben oder ähnlichen Wissenschaftsdomänen formal Probleme bzw. Systeme und ihre Eigenschaften modellieren kann.

Kollaboratives Ontology-Design ist ein wichtiges Thema im Semantic Web, da mittelfristig Ontologien für Domänen als weltweite oder zumindest regionale Standards, die dann über Mappings mit anderen Standards der gleichen Domäne verknüpft werden, von Expertengruppen, Entwicklern und Benutzern in gemeinsamen Prozessen entwickelt und gewartet werden sollen. Für den Ansatz der praktischen Realisierung einer Allgemeinen Systemtheorie sind diese Prozesse essentiell, da sie eine verteilte und umfangreiche Aufbereitung von Wissen auf Basis einer entsprechend ausdrucksstarken Sprache und maschineninterpretierbar gewährleisten. Diese Möglichkeiten bieten bestehende Systemdesign-Tools nur in einfachen Ansätzen, während im Semantic-Web-Bereich heute bereits praxiserprobte und benutzerfreundliche Tools mit entsprechenden Supportfunktionen für Benutzer ohne Wissen im Bereich von Semantic-Web-Sprachen existieren. In der Folge seien auszugsweise einige dieser Anwendungen vorgestellt.

7.2.1 Collaborative Protege – Protege Plugin

Collaborative Protege⁸ ist eine der aktuellsten Erweiterungen des in dieser Arbeit bereits vorgestellten Protege-Ontologieeditors. Über ein Plugin ermöglicht es diese Anwendung, simultan oder in Abfolge in Gruppen an einer gemeinsamen Ontologie zu arbeiten. Die Anwendung (siehe Abbildung 7.5) unterstützt diesen Prozess dabei über ein Vorschlagsystem, Kommentarfunktionen, Versioning und weitere Funktionalitäten, die den kollaborativen Prozess direkt über die vertraute Protege-Umgebung unterstützen.

Dieses Tool eignet sich daher sehr gut, um einer Gruppe von Ontology-Designern ein verteiltes, vernetztes gemeinsames Arbeiten an Ontologien möglichst effizient zu ermöglichen. Da auch alle Grundfunktionen von Protege unterstützt werden, besteht auf einer technisch-formalen Ebene kein Nachteil zur Ontologie-Modellierung gegenüber dem Single-User-Modus.

7.2.2 myOntology

Protege und Collaborative Protege wurden zwar über viele Jahre hinweg als mächtige Tools zum professionellen Ontology-Design weiterentwickelt, eignen sich aber nur für Entwickler, die mit OWL und RDF als Semantic-Web-Sprachen

⁸Siehe http://protegewiki.stanford.edu/index.php/Collaborative_Protege.

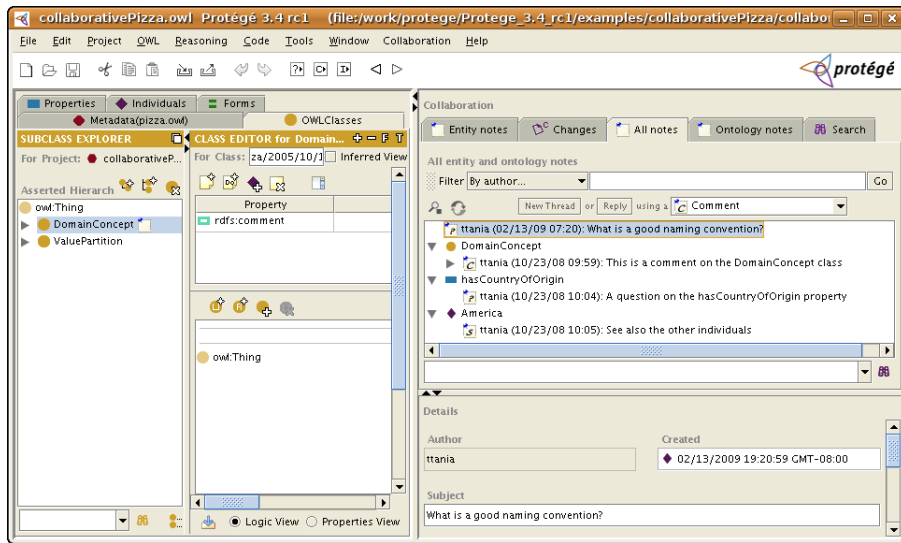


Abbildung 7.5: Collaborative Protege erlaubt das verteilte, kollaborative Arbeiten an gemeinsamen (System-)Beschreibungen.

vertraut sind und über die Bedeutung der einzelnen Modellierungskonzepte im Detail Bescheid wissen. Insbesondere OWL und die Klassen-Systematik sind nicht immer intuitiv, was bei unerfahrenen Benutzern oft zu unbeabsichtigten Fehlkonstruktionen beim Ontologie-Bau mit Protege führt.

Dieser Umstand ist in der Semantic-Web-Community wohl bekannt. Deshalb entstanden in den letzten Jahren zahlreiche Anwendungen, die über bessere Hilfskonstruktionen und einfachere Benutzeroberflächen auch unerfahrenen Benutzern die Möglichkeit geben sollen, korrekte Ontologien intuitiv zu entwickeln. Eine solche Plattform, die diesen Prozess auch mit dem kollaborativen Ansatz verbindet, ist myOntology⁹. Die Plattform (siehe Abbildung 7.6) ist von der Benutzeroberfläche her stark an ein Wiki angelehnt und bietet auch Usability-mäßig sehr ähnliche Funktionen. So können registrierte Nutzer sehr einfach Elemente bearbeiten, hinzufügen oder wieder löschen – jedoch nur in begrenztem Ausmaß und so, dass – dem Wiki-Prinzip folgend – einzelne User nur sehr wenig “Schaden” anrichten können während die Masse der Nutzer gemeinsam an den wichtigen Konzepten der einzelnen Domänen arbeiten und so ein qualitativ hochwertiges Ergebnis hervorbringen kann. myOntology unterstützt auch Ontologie-Import und ermöglicht es so, die CGST-System-Ontologie zu importieren und auf Basis dieser Grundkonzepte Systemontologien für Domänen auf unterschiedlichen Reductive-Levels anzulegen.

Konzepten und Relationen können dabei auch Meta-Attribute zugewiesen werden, um den Diskurs durch eine bessere Repräsentation des gesammelten

⁹Siehe <http://www.myontology.org>. myOntology ist ein gemeinsames Projekt von STI Innsbruck, der Uni der Bundeswehr in München, Smart Information Systems, System.One und dem E-Commerce-Standardisierungsverein der WKÖ, Austriapro, und wurde vom BMVIT im Rahmen des FIT-IT-Programms gefördert.

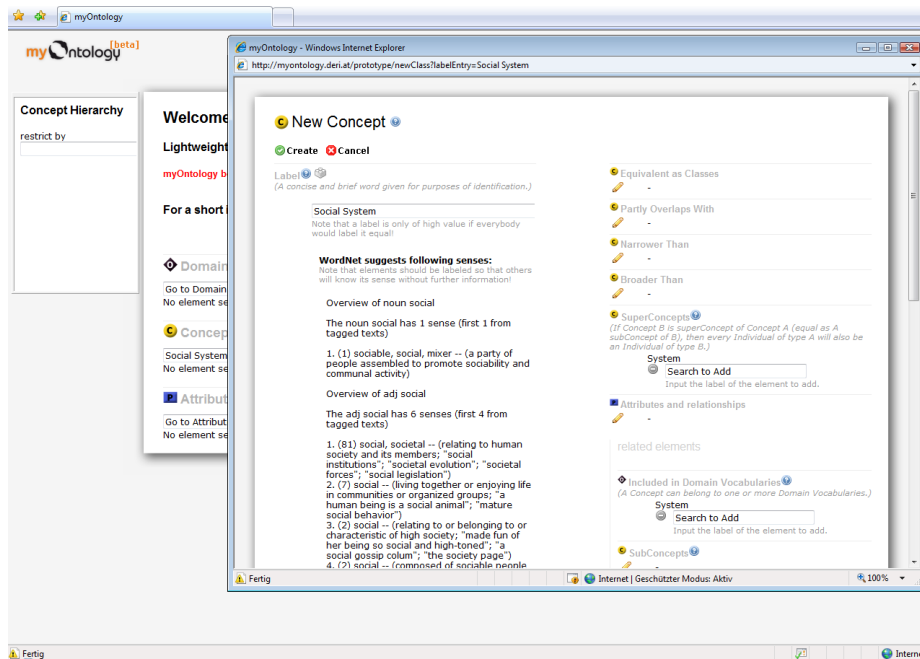


Abbildung 7.6: Bearbeitungsmodus zur Erstellung eines Konzepts “Soziales System” als Unterklasse des CGST-System-Konzepts.

Wissens zu unterstützen. In Abbildung 7.7 wird die Repräsentation eines Konzepts “Person” als Element eines Sozialen Systems realisiert, wobei es sich beim Sozialen System um eine Unterklasse des CGST-System-Konzepts und beim Konzept Person um eine Unterklasse des CGST-Element-Konzepts handelt. Diese Relationen werden dem Benutzer auch entsprechend übersichtlich dargestellt. Auf diesem Weg ist es möglich, einer breiten Benutzergruppe ein einfaches Tool zum kollaborativen, webbasierten Arbeiten an unterschiedlichen System-Beschreibungen zu ermöglichen. myOntology ist als OpenSource-Software lizenziert und kann daher auch beliebig in abgeschlossenen Communities oder themenspezifisch eingesetzt werden, wobei die entstehenden Schemata jederzeit und unterstützt durch ein Versioning-System in Form einer OWL-Ontologie exportiert und in anderen Anwendungen (z.B. Ableitung von NetLogo-Frameworks) eingesetzt werden können.

7.2.3 Ontoverse

Zwischen der einfachen Oberfläche und Funktionalität von myOntology und dem klassischen Ontologie-Expertentool Protege finden sich einige weitere Anwendungen, die ebenfalls kollaborative Ontologieentwicklung unterstützen – ein Beleg für den Charakter der Semantic-Web-Sprachen einer verteilten, offenen und verbreiteten Technologie, die bereits heute weltweit breite Anerkennung gefunden hat. Da im letzten Abschnitt ein Beispiel aus der Biochemie vorgestellt wurde, soll hier nun noch einmal kurz gezeigt werden, dass auch für solche kom-

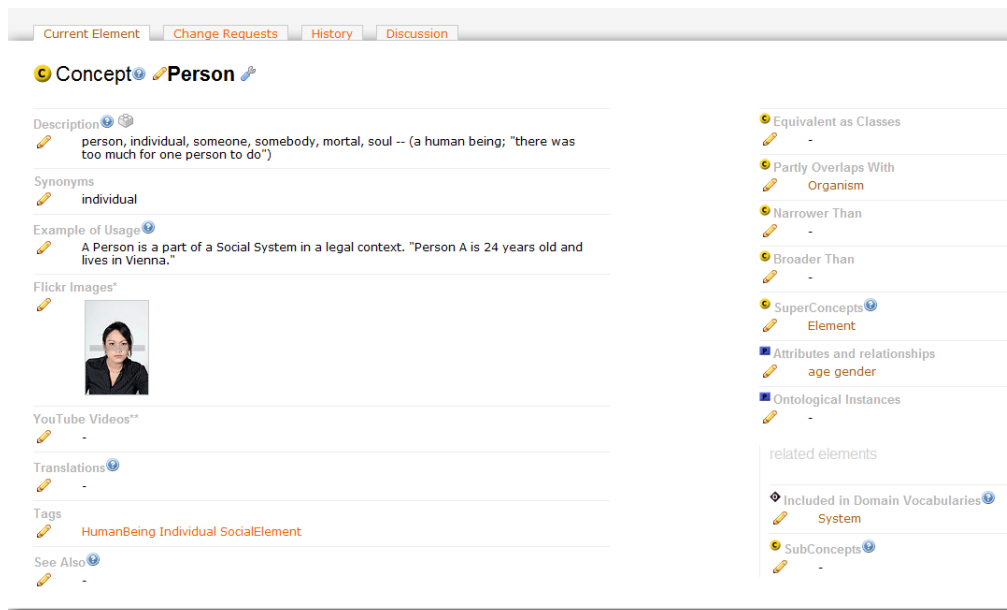


Abbildung 7.7: Detailansicht des Konzepts für “Personen” als Teil Sozialer Systeme auf myOntology.

plexeren Domänen kollaborative Ansätze bereits in der Praxis funktionieren.

Ontoverse¹⁰, eine Art “Schwesterprojekt” von myOntology, gefördert von der deutschen Bundesregierung, ist ebenfalls als Webanwendung konzipiert und verfolgt wie auch myOntology das Ziel, auf diesem Weg OWL-basierte kollaborative Ontologieentwicklung zu unterstützen. Der Ansatz geht dabei stärker in Richtung bekannter Ontologieeditoren und erinnert von Bearbeitungs-Funktionalität und Unterstützung auch teilweise an Protege. Das erlaubt prinzipiell ein professionelleres weil ausdrucksstärkeres Vorgehen, erschwert ungeübten Benutzern aber den Zugang. Als Prototyp-Ontologie wurde eine Domänenbeschreibung für den Bioinformatikbereich, BIO2Me¹¹, erstellt, die zeigen soll, wie Ontoverse genutzt werden kann, um auch tiefer gehendes Expertenwissen in einem kollaborativen Beschreibungsprozess zu strukturieren und diese Information dann in Form einer OWL-Ontologien an Anwendungen weiter zu geben. Sehr hilfreich ist dabei auch vor allem die dynamische und generische graphische Repräsentation (siehe Abbildung 7.8) der jeweils aktuellen Ontologie bzw. einzelner Teilgraphen.

Welche der vorgestellten Anwendungen (oder anderer weiterer) für Problemstellungen, bei denen es darum geht, von einer kleinen oder großen Gruppe kollaborativ Systeme beschreiben zu lassen, verwendet werden soll, hängt stark vom Wissen der Mitglieder über Ontologiesprachen und den darauf aufbauenden Tools ab – jedenfalls gibt es aber für jede Zielgruppe, vom Anfänger bis

¹⁰Siehe <http://www.ontoverse.org>

¹¹BioInformatics Ontology for Tools and Methods.

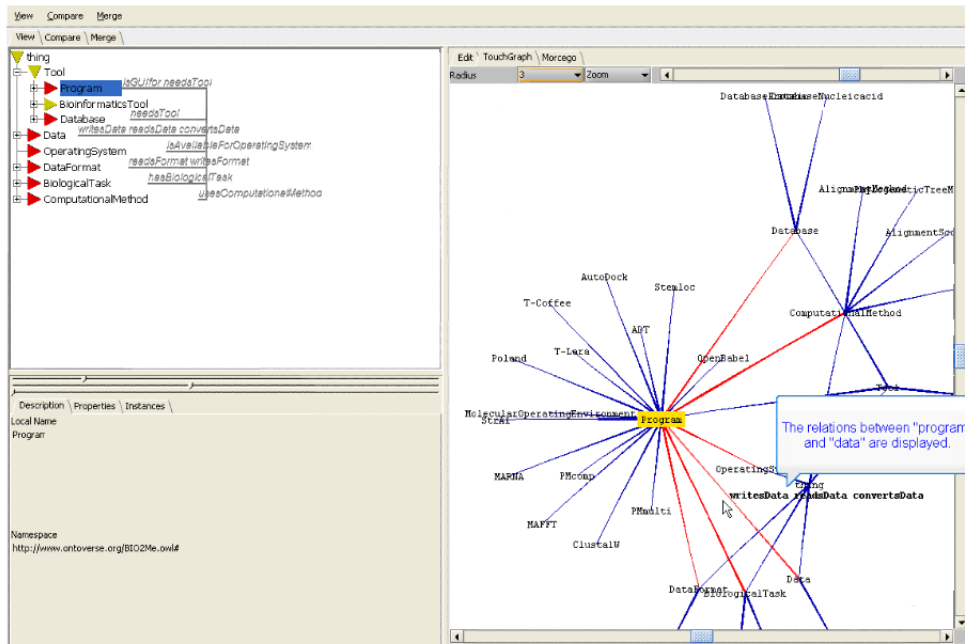


Abbildung 7.8: Graphische Repräsentation eines Auszugs der BIO2Me-Wissensbasis in Ontoverse.

zum Profi-Ontology-Engineer – passende Werkzeuge.

Kapitel 8

Zusammenfassung

*“Nur die Fragen, die prinzipiell unentscheidbar sind,
können wir entscheiden.”
Heinz von Förster*

In der Arbeit wurden wesentliche Ergebnisse des Bertalanffy-Programms mit Hilfe heute aktueller und verbreiteter Standards formalisiert, sodass mit dem daraus entstehenden CGST-Modell sehr einfach System-Modellierungen, Simulationen und Vergleiche von unterschiedlichen Systemen möglich sind. Das CGST-Modell ist dabei als ein auf modernen Methoden der Informatik basierendes formales Modell zur Repräsentation einer Allgemeinen Systemtheorie einzigartig und bietet im Vergleich zu den ursprünglichen Ansätzen aus dem Bertalanffy-Programm große Vorteile in Hinblick auf die praktischen Nutzungsmöglichkeiten und die Integration bereits bestehenden und formalisierten Wissens. Aufbauend auf diesem Modell wurden außerdem Anwendungsmöglichkeiten skizziert, die über diese einfachen Fälle hinausgehen und komplexe, über mehrere Stufen hinweggehende Applikationen zur Repräsentation, Simulation und Analyse von Systemen ermöglichen können.

Darüber hinaus wurde aufbauend auf dem Bertalanffy-Programm, dessen Überführung in ein formales System und basierend auf den Ergebnissen der theoretischen Informatik eine zentrale metaphysische These zur Interpretation einer Allgemeinen Systemtheorie konstruiert. Diese These beschreibt die Beziehung des Selbstorganisations- und Emergenz-Begriffs in Hinblick auf die formale Repräsentation eines allgemeinen Systembegriffs und kommt zum prinzipiellen Schluss, dass basierend auf den Charakteristika der Mathematik und der Mathematischen Logik nichtentscheidbare Zustände in ebendiesen Systemen bei ihrer Interpretation in der Realität auf emergente Zustände hindeuten und damit Alternativen zu einem rein deterministischen Konstrukt des CGST-Modells – und damit einer Allgemeinen Systemtheorie bzw. der Wissenschaft als solches –

darstellen.

Dies gilt aber, wie viele andere wissenschaftliche Erkenntnisse auch, nur dann, wenn man die Rolle der Mathematik als abstrakte, jedoch tatsächliche und gültige Abbildung der Realität akzeptiert und nicht einen Fehler im System der Mathematik an sich sieht oder vermutet. Wesentliche Punkte der Arbeit seien in den folgenden Abschnitten noch einmal kurz zusammengefasst.

8.1 Motivation

Seit der Mensch strukturiert wissenschaftliche Untersuchungen anstellt, ist es eines der Hauptziele, die Welt und die Vorgänge, die sich in ihr abspielen, gesamtheitlich in ein wissenschaftliches Weltbild zu fassen und zu erklären. Dieses Ziel wurde über viele Jahrhunderte und Jahrtausende immer wieder phasenweise stärker und schwächer verfolgt, wobei wir uns heute aufgrund der technisch gestützten Ausdifferenzierung in den westlichen Gesellschaften wieder verstärkt in einer Phase der wissenschaftlichen Spezialdisziplinen befinden. Einer der letzten Anläufe zur Etablierung eines ganzheitlichen wissenschaftlichen Weltbildes war das Bertalanffy-Programm, das zwar wesentliche Erkenntnisse auf viele unterschiedliche Wissenschaften übertrug, das Ziel, eine Allgemeine Systemtheorie als Wissenschaftsdisziplin nachhaltig einzuführen, aber nicht erreichen konnte.

Die vorliegende Arbeit geht davon aus, dass eine derartige Disziplin wertvolle neue Erkenntnisse in einer immer komplexer werdenden Welt liefern könnte und das Verstehen schwieriger Prozesse maßgeblich unterstützen könnte und versucht daher, die notwendigen Grundbausteine für eine moderne Variante der Allgemeinen Systemtheorie bereit zu stellen. Um die zukünftige Rolle einer Allgemeinen Systemtheorie zu gestalten, ist es aber wichtig, die vergangenen Bemühungen zu kennen; diese Bemühungen und ihre wichtigsten Vertreter und Resultate werden zu Beginn der Arbeit kurz vorgestellt.

Auf etwas sei jetzt am Ende in dieser Zusammenfassung außerdem noch dezidiert hingewiesen, nachdem die beiden Personen in unterschiedlichen Kapiteln der Arbeit eine zentrale Rolle eingenommen haben: Kurt Gödel und Ludwig von Bertalanffy, deren Leben erstaunliche Parallelen aufweist, sind wohl aufgrund der Innovationskraft ihrer Arbeiten zwei der bedeutendsten Persönlichkeiten, die jemals in Österreich gewirkt haben. Beide wurden in ehemaligen Kronländern der Monarchie Anfang des 20. Jahrhunderts geboren; beide begannen ihre wissenschaftliche Karriere an der Universität Wien und wohnten Sitzungen des Wiener Kreises bei, zählten sich aber nie zu diesem, weil sie dem logischen Positivismus kritisch gegenüber standen; beide emigrierten in

die USA¹, wo sie den wissenschaftlichen Diskurs in ihrem Bereich maßgeblich beeinflussten und die faszinierende Neuartigkeit ihrer Thesen entsprechend rezipiert und gewürdigt wurde². Beide, hier aber Bertalanffy noch mehr als Gödel, sind heute im akademischen Umfeld im Inland außerhalb ihrer Domäne weitestgehend unbekannt und erhielten posthum nie die Würdigung und den Status als große heimische Wissenschaftler, die ihnen ob ihrer Leistungen zustehen würde. Es soll daher auch ein ganz besonderes Anliegen sein, im Rahmen dieser Dissertation noch einmal die Bedeutung und Tragweite der Kombination der Thesen dieser beiden Persönlichkeiten dargestellt zu haben.

Im Rahmen der Arbeit wurden auch unterschiedliche aktuelle wissenschaftliche Probleme aus unterschiedlichen Wissenschaftsbereichen angeführt und betrachtet, um den Bezug zu einer Allgemeinen Systemtheorie zu veranschaulichen und den Beitrag, den eine solche Theorie dazu leisten könnte, zu verdeutlichen. Dabei wurden bewusst Fragestellungen ausgewählt, die entscheidende Rollen auf gesellschaftlicher Ebene einnehmen und deren Beantwortung jedenfalls große gesellschaftliche Konsequenzen mit sich bringen.

8.2 Das CGST-Modell

Die Konstruktion eines computationalen Modells für die Allgemeine Systemtheorie (“Computational General System Theory”, kurz CGST) war das wesentliche praktische Ziel dieser Arbeit. Im Laufe des Bertalanffy-Programms und auch schon davor wurden zahlreiche Versuche unternommen, eine allgemeine formale Systembeschreibung zu konstruieren, die auf der einen Seite als Meta-Beschreibung für Systeme im Allgemeinen dienen kann, auf der anderen Seite aber speziell genug ist, um über bloße triviale Aussagen hinaus zu gehen.

Eine solche formale Beschreibung hat sich bisher nicht etablieren können, jedoch gab es bisher auch noch kaum Versuche, eine solche Form aus dem Kontext der aktuellen Beschreibungs-Standards der Informatik heraus zu konstruieren. Der hier vorgestellte Ansatz basiert auf den Semantic-Web-Standardsprachen OWL und RDF und ermöglicht daher eine große Kompatibilität mit bereits bestehenden Wissensbasen und eine rasche Integration bereits bestehender Daten. Die Wahl, Zelluläre Automaten, die aus den Semantic-Web-Beschreibungen abgeleitet werden können, für den dynamischen Systemteil bzw. die Simulation von Systemen zu verwenden, begründet sich ebenfalls auf der zunehmenden Verwendung dieses Konzepts in der Informatik und der aktuellen tragenden Rolle solcher Ansätze, etwa in der 2009 vorgestellten mathematischen Suchmaschine

¹Ein interessantes Detail am Rande: Für die gegenseitige Vorstellung von Gödel und Einstein, die sich zu einer langen und fruchtbaren Freundschaft entwickeln sollte, zeichnet sich der Erfinder des Reductive-Levels-Modells aus Kapitel 3, Paul Oppenheim, verantwortlich. Siehe dazu [50], Seite 112.

²Siehe dazu Bertalanffy’s Biographie in [14] und Gödel’s Biographie in [50].

wolframalpha.com.

Um die praktische Verwendung des Modells zu veranschaulichen, wurde ein fiktives Beispiel konstruiert, das durch die unterschiedlichen Modellierungsphasen begleitet und dabei aus der Systemtheorie bekannte Konzepte auf das CGST-Modell bzw. verschiedene Instanzen abbildet. Weiters wurden ausgewählte bestehende Wissensbasen mit der CGST-Ontologie verknüpft, um den Anspruch, auch bereits bestehende, unabhängig von CGST entwickelte Wissensbasen integrieren zu können, zu validieren.

8.3 Selbstorganisation und Emergenz

Die Betrachtung zweier in der Systemtheorie heute ganz wesentlicher Konzepte, von Selbstorganisation und Emergenz, und deren Integration in das CGST-Modell führt schlussendlich zu dem wesentlichen metaphysischen Resultat der Arbeit: Emergente Systementwicklungen im Sinne einer dritten Alternative zu deterministischen und indeterministischen Erklärungsmodellen für die Ausbildung von neuen Eigenschaften auf unterschiedlichen Ebenen können heute in der wissenschaftlichen Praxis nicht unmittelbar nachgewiesen werden. Bildet man jedoch die Eigenschaften und Entwicklungen von Systemen unter Verwendung der Mathematik und der Mathematischen Logik formal auf das CGST-Modell (oder ähnlich ausdrucksstarke Modelle) ab, entstehen nach den Resultaten des Österreichischen Mathematikers Kurt Gödel Zustände, die formal innerhalb dieses Systems nicht mehr entscheidbar sind.

Bei der Rückführung dieser Resultate in die wissenschaftliche Praxis ist das Auftreten starker Emergenz in ebendiesen Situationen eine mögliche Erklärung zur Interpretation dieser Umstände. Selbstorganisation als Prozess, der Information in einem System erzeugt, wird dabei als zentrales Konzept in diesem Zusammenhang und als wichtiges, wenn nicht sogar wichtigstes wissenschaftliches Konzept bzw. Paradigma im neuen Jahrhundert betrachtet. Beide Konzepte eint und verbindet in dem in der Arbeit aufgezeigten Kontext die Eigenschaft der Selbstreferenz, die daher als wesentliches Element einer Allgemeinen Systemtheorie beschrieben und als Motivation und möglicher Fokus für ebendiese vorgeschlagen wird.

8.4 CGST-Weltbild und Ausblick

Die Welt ist im Großen und Ganzen determiniert, formal abbildbar und berechenbar – so wird sie heute zumindest nicht zuletzt aufgrund der Möglichkeiten der Informatik immer stärker gesehen. Die Unentscheidbarkeit derselben und die daraus konstruierbaren Modelle werden dabei oft ignoriert oder vergessen,

lassen aber Spielraum offen für irreduzible, prinzipiell neue Eigenschaften und Entwicklungen – also für echte Emergenz. Eine systemische Sichtweise auf allen Ebenen der Wissenschaft und eine Verknüpfung der aktuellen Wissenschaftsdomänen mit einer Allgemeinen Systemtheorie können nicht nur dabei helfen, große Fragen und Probleme unserer Zeit zu lösen, sondern haben tatsächlich auch das Potential, Grundlage für ein neues Weltbild zu sein.

Ein solches Weltbild wäre daher beides: Ein stark rationales und wissenschaftliches, das durch den ganzheitlichen Zugang auch vermeintlich unerklärbare oder transzendente Vorgänge zu erklären vermag sowie ein humanistisches, das trotz der Eingliederung des Menschen in ein umfassendes Modell dem menschlichen Denken eine besondere Qualität zuspricht und damit dem Menschen als Lebewesen auf dem Planeten (und im Universum) eine tragende und verantwortungsvolle Rolle zuspricht, aus der auch besondere Rechte und Pflichten entstehen.

Anhang A

C-GST Ontologie im OWL-Format

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY owl2 "http://www.w3.org/2006/12/owl2#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY system "http://www.systems-everywhere.com/system#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://www.systems-everywhere.com/system#"
  xml:base="http://www.systems-everywhere.com/system"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:system="http://www.systems-everywhere.com/system#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl2="http://www.w3.org/2006/12/owl2#">
  <owl:Ontology rdf:about=""/>

  <!--
  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  //
  // Object Properties
  //
  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  -->

  <!-- http://www.systems-everywhere.com/system#belongsToReductiveLevel -->
  <owl:ObjectProperty rdf:about="#belongsToReductiveLevel">
    <rdf:type rdf:resource="#owl:FunctionalProperty"/>
    <rdfs:range rdf:resource="#ReductiveLevel"/>
  </owl:ObjectProperty>
```

```

    <rdfs:domain rdf:resource="#System"/>
</owl:ObjectProperty>

<!-- http://www.systems-everywhere.com/system#hasElement -->
<owl:ObjectProperty rdf:about="#hasElement">
    <rdfs:range rdf:resource="#Element"/>
    <rdfs:domain rdf:resource="#System"/>
</owl:ObjectProperty>

<!-- http://www.systems-everywhere.com/system#hasSubSystem -->
<owl:ObjectProperty rdf:about="#hasSubSystem">
    <rdfs:range rdf:resource="#SubSystem"/>
    <rdfs:domain rdf:resource="#System"/>
</owl:ObjectProperty>

<!-- http://www.systems-everywhere.com/system#hasSystemCategory -->
<owl:ObjectProperty rdf:about="#hasSystemCategory">
    <rdfs:domain rdf:resource="#System"/>
    <rdfs:range rdf:resource="#SystemCategory"/>
</owl:ObjectProperty>

<!--
////////////////////////////////////
//
// Data properties
//
////////////////////////////////////
-->

<!-- http://www.systems-everywhere.com/system#description -->
<owl:DatatypeProperty rdf:about="#description">
    <rdfs:range rdf:resource="&xsd:string"/>
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdfs:Description rdf:about="#Element"/>
                <rdfs:Description rdf:about="#ReductiveLevel"/>
                <rdfs:Description rdf:about="#System"/>
                <rdfs:Description rdf:about="#SystemCategory"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
</owl:DatatypeProperty>

<!-- http://www.systems-everywhere.com/system#name -->
<owl:DatatypeProperty rdf:about="#name">
    <rdfs:range rdf:resource="&xsd:string"/>
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdfs:Description rdf:about="#Element"/>
                <rdfs:Description rdf:about="#ReductiveLevel"/>
                <rdfs:Description rdf:about="#System"/>
                <rdfs:Description rdf:about="#SystemCategory"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
</owl:DatatypeProperty>

```

```

    </rdfs:domain>
</owl:DatatypeProperty>

<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->

<!-- http://www.systems-everywhere.com/system#Element -->
<owl:Class rdf:about="#Element"/>

<!-- http://www.systems-everywhere.com/system#ReductiveLevel -->
<owl:Class rdf:about="#ReductiveLevel"/>

<!-- http://www.systems-everywhere.com/system#SubSystem -->
<owl:Class rdf:about="#SubSystem">
  <rdfs:subClassOf rdf:resource="#System"/>
</owl:Class>

<!-- http://www.systems-everywhere.com/system#System -->
<owl:Class rdf:about="#System"/>

<!-- http://www.systems-everywhere.com/system#SystemCategory -->
<owl:Class rdf:about="#SystemCategory"/>

<!--
////////////////////////////////////
//
// Individuals
//
////////////////////////////////////
-->

<!-- http://www.systems-everywhere.com/system#AstronomicSystems -->
<ReductiveLevel rdf:about="#AstronomicSystems"/>

<!-- http://www.systems-everywhere.com/system#Atoms -->
<ReductiveLevel rdf:about="#Atoms"/>

<!-- http://www.systems-everywhere.com/system#Cells -->
<ReductiveLevel rdf:about="#Cells"/>

<!-- http://www.systems-everywhere.com/system#ClosedSystem -->
<SystemCategory rdf:about="#ClosedSystem"/>

<!-- http://www.systems-everywhere.com/system#EcoSystems -->
<ReductiveLevel rdf:about="#EcoSystems"/>

<!-- http://www.systems-everywhere.com/system#ElementaryParticles -->
<ReductiveLevel rdf:about="#ElementaryParticles"/>

```

```
<!-- http://www.systems-everywhere.com/system#FormalSystem -->
<SystemCategory rdf:about="#FormalSystem"/>

<!-- http://www.systems-everywhere.com/system#LivingOrganisms -->
<ReductiveLevel rdf:about="#LivingOrganisms"/>

<!-- http://www.systems-everywhere.com/system#Molecules -->
<ReductiveLevel rdf:about="#Molecules"/>

<!-- http://www.systems-everywhere.com/system#OpenSystem -->
<SystemCategory rdf:about="#OpenSystem"/>

<!-- http://www.systems-everywhere.com/system#SocialSystems -->
<ReductiveLevel rdf:about="#SocialSystems"/>

</rdf:RDF>
```

Anhang B

CGST Transformationscript Aufruf

```
@echo off
title OWLtoNetLogo Converter

setLocal EnableDelayedExpansion

rem ***** Change following values *****

SET GLOBAL_PATH=D:\OWLtoNetLogo

SET OWL_PATH=%GLOBAL_PATH%\OWL

SET INSTANCE_PATH=%GLOBAL_PATH%\NetLogo

SET STYLESHEET_NETLOGO=%GLOBAL_PATH%\owl_to_netlogo.xsl

rem ***** Path to xslt-transformer *****

SET XSLT_PROCESSOR=C:\Programme\Python24\Scripts\4xslt.exe

rem ***** Internal variables *****
SET RDF_COUNT=0
SET RDF_INDEX=0
SET RDF_BASICS_INDEX=0

rem ***** File count calculation *****
rem Calculate XML files Count
for %%E in (%OWL_PATH%\*.xml) do ( SET /A RDF_COUNT+=1 )

rem ***** Removing files *****

echo.
echo Deleting files in '%INSTANCE_PATH%'...
```

```
for %%E in (%INSTANCE_PATH%\*.nlogo) do (del %%E & echo Deleting %%~nxE)

rem ***** Creating Automata Instances *****

echo.
echo Creating NetLogo Automata...
for %%E in (%OWL_PATH%\*.owl) do (call %XSLT_PROCESSOR% -o%INSTANCE_PATH%\%%~nE.nlogo
%%E %STYLESHEET_NETLOGO% & SET /A RDF_INDEX+=1 &
echo Creating [!RDF_INDEX! of %RDF_COUNT%] %%~nE.nlogo)

time /T

echo.
echo Press any key to finish...
@pause

endLocal
```

Anhang C

NetLogo-Serendipity- Implementierung (Kooperation)

```
;; adams and eves are elements of an abstract system

breed [ adams adam ]
breed [ eves eve ]
undirected-link-breed [isFriendOf-links isFriendOf-link]

;; adams and eves both own properties alive, age and energy

adams-own
[
  alive?          ;; is the element active?
  age             ;; for how many ticks is the element active?
  energy         ;; adams and eves need energy to keep on being alive/active
  power
]

eves-own
[
  alive?
  age
  energy
  power
]

patches-own
[
  ambrosia        ;; ambrosia is a source of energy
]

;; setup the start state of the system
;; at the beginning, there are as many adams (red) as there are eves (violet)
;; adams have an average life time of 80 ticks (years)
```



```

;; eves have an average life time of 80 ticks (years)
;; to make it more interesting we add some kind of randomized "variance":
;; life time is distributed between 64 and 101 years

to setup
  ca
  clear-all
  set-default-shape adams "person"
  set-default-shape eves "person"
  ask patches
  [
    set ambrosia random 30
    set pcolor green - (ambrosia / 30)
  ]
  create-adams 50
  [ set color red
    set alive? true
    set age 80 + (random 16) - (random 16)
    set energy 100
    set power 40
    setxy random-xcor random-ycor
  ]
  create-eves 50
  [ set color violet
    set alive? true
    set age 85 + (random 16) - (random 16)
    set energy 100
    set power 40
    setxy random-xcor random-ycor
  ]
end

;; stop development after 500 years

to go
  if ticks >= 2000 [ stop ]
  tick
  update-patches
  move-people
  update-people
  do-plots-pop
  do-plots-amb
end

to update-patches
  ask patches
  [
    regrow
    set pcolor green - (ambrosia / 30)
  ]
end

;; adams and eves move around and lose energy, because they have nothing else to do

to move-people
  ask adams [
    if age > 0 [

```

```

        right random 360
        forward 2
        set age age - 1
        set energy energy - 10
        talk
    ]
]
ask eves [
    if age > 0 [
        right random 360
        forward 2
        set age age - 1
        set energy energy - 10
    ]
]
end

;; if they are still alive, they try to get energy by eating amrosia
;; if they cannot find enough ambrosia they starve

to update-people
ask adams [
    ifelse age < 1 or energy <= 0 [
        set alive? false
        set color black
        die
    ]
    [
        eat
        power-upgrade
        reproduce-adam
    ]
]
ask eves [
    ifelse age < 1 or energy <= 0 [
        set alive? false
        set color black
        die
    ]
    [
        eat
        reproduce-eve
    ]
]
end

;; eating is consuming ambrosia

to eat
    if ambrosia >= 10 and power > random 47 [
        set ambrosia ambrosia - 10
        set energy energy + 10
    ]
end

;; ambrosia is regrowing continuously for some magical reason

```

```

to regrow
  if ambrosia <= 25
    [
      set ambrosia ambrosia + 5
    ]
  end
end

to reproduce-adam
  if random 78 = 1 ;;
    [
      hatch-adams 1
      [ set color red
        set alive? true
        set age 80 + (random 16) - (random 16)
        set energy 100
        set power 40
        setxy random-xcor random-ycor
      ]
    ]
  end

to reproduce-eve
  if random 42 = 1 ;;
    [
      hatch-eves 1
      [ set color violet
        set alive? true
        set age 85 + (random 16) - (random 16)
        set energy 100
        set power 40
        setxy random-xcor random-ycor
      ]
    ]
  end

; communication can lead to freidnship and cooperation
to talk
  if ( random 10 < 6 ) and ( count isFriendOf-link-neighbors < 10 )
    [
      create-isFriendOf-links-with adams-on neighbors
    ]
  end

; power is featured by the number of friends someone has
to power-upgrade
  set power 40 + ( count isFriendOf-link-neighbors ) * 2
  if power > 59 [ set power 59 ]
end

to do-plots-pop
  set-current-plot "Population"
  set-current-plot-pen "Adams"
  plot count adams with [alive? = true]
  set-current-plot-pen "Eves"
  plot count eves with [alive? = true]
end

```

```
to do-plots-amb
  set-current-plot "Ambrosia"
  set-current-plot-pen "Ambrosia"
  plot count patches with [ambrosia >= 10]
end
```

Anhang D

NetLogo-Implementierung für Prigogine's Brüsselator

```
turtles-own [state]

; molecule to states mapping
; a = 0, b = 1 - 3, x = 4 - 5, y = 6 - 7, d = 8, e = 9

to setup
  clear-all
  create-turtles 1000
  set-default-shape turtles "circle"
  ask turtles [
    setxy random-pxcor random-ycor
    set state random 4
    set-turtle-color
  ]
  do-plots-trajectories
  do-plots
end

to go
  if ticks >= 2000 [ stop ]
  move-turtles
  change-turtles
  manage-molecules
  tick
  do-plots
  do-plots-trajectories
end

to move-turtles
  ask turtles [
    right random 360
    forward 2
  ]
end
```

```

to set-turtle-color
  if state = 0 [ set color 65 ]
  if state > 0 and state < 4 [ set color 45 ]
  if state > 3 and state < 6 [ set color 25 ]
  if state > 5 and state < 8 [ set color 85 ]
  if state = 9 [ set color 105 ]
end

to change-turtles
  ask turtles [
    if state = 0 ; a molecule to compute a --> x, k= 0.025
    [
      set state random 40
      ifelse state = 1
        [ set state 4 ]
        [ set state 0 ]
    ]
    if state > 0 and state < 4 ; b molecule to compute b + x --> y + d, k=1
    [
      if count turtles-here with [ state > 3 and state < 6 ] > 0
      [
        set state random 100
        ifelse state > 50
          [ set state 6 ]
          [ set state 8 ]
        ]
      ]
    if state > 5 and state < 8 ; y molecule to compute 2x + y --> 3x, k=1
    [
      if count turtles-here with [ state > 3 and state < 6 ] > 1
      [ set state 4 ]
    ]
    if state > 3 and state < 6 ; x molecule to compute x --> e, k=0.01
    [
      set state random 100
      ifelse state = 1
        [ set state 9 ]
        [ set state 4 ]
    ]
    if state = 8 ;; precipitate d with p=0.3
    [
      set state random 100
      ifelse state < 30
        [ die ]
        [ set state 8 ]
    ]
    if state = 9 ;; precipitate e with p=0.3
    [
      set state random 100
      ifelse state < 30
        [ die ]
        [ set state 9 ]
    ]
    set-turtle-color
  ]
end

```

```

; open system: a and b are put into the system continuous
to manage-molecules
  create-turtles 20 [
    setxy random-xcor random-ycor
    set state random 4
    set-turtle-color
  ]
end

; standard plotting methods
to do-plots
  set-current-plot "Totals"
  set-current-plot-pen "molecule-a"
  plot count turtles with [ state = 0 ]
  set-current-plot-pen "molecule-b"
  plot count turtles with [ state > 0 and state < 4 ]
  set-current-plot-pen "molecule-x"
  plot count turtles with [ state > 3 and state < 6 ]
  set-current-plot-pen "molecule-y"
  plot count turtles with [ state > 5 and state < 8 ]
end

; plotting dynamic trajectory for x/y
to do-plots-trajectories
  set-current-plot "Trajectories"
  set-current-plot-pen "traj-x"
  plotxy count turtles with [ state > 5 and state < 8 ]
  count turtles with [ state > 3 and state < 6 ]
end

```

Literaturverzeichnis

- [1] Martin Balluch, “Die Kontinuität von Bewusstsein”, Wien 2005.
- [2] Ludwig von Bertalanffy, “Nikolaus von Kues”, München bei Georg Müller, 1928.
- [3] Ludwig von Bertalanffy, “General System Theory”, New York 1969.
- [4] Ludwig von Bertalanffy, “Biophysik des Fließgleichgewichts”, 1953.
- [5] Dirk Baecker, “Kalkül der Form”, Frankfurt am Main 1993.
- [6] Dirk Baecker (Hg.), “Schlüsselwerke der Systemtheorie”, Wiesbaden 2005.
- [7] Julian Barbour, “The End of Time: The Next Revolution in Physics”, Oxford University Press, USA 2000.
- [8] Eric Bonabeau et al., “Swarm Intelligence”, Oxford University Press 1999.
- [9] Hartmut Bossel, “Systeme, Dynamik, Simulation”, Books on Demand GmbH, Norderstedt 2004.
- [10] Kenneth Boulding, “General Systems Theory – The Skeleton of Science”, in: Ludwig von Bertalanffy, Anatol Rapoport (ed.), “General Systems – Yearbook of the Society for General Systems Research Volume 1”, 1956.
- [11] Kenneth Boulding, “Ecodynamics”, Sage Publications, Kalifornien 1978.
- [12] Scott Camazine et al., “Self-Organization in Biological Systems”, Princeton University Press 2001.
- [13] F. Stuart Chapin III et al., “Principles of Terrestrial Ecosystem Ecology”, Springer 2002.
- [14] Mark Davidson, “Querdenken! Leben und Werk Ludwig von Bertalanffys”, Frankfurt am Main 2005.
- [15] Johann Dieckmann, “Schlüsselbegriffe der Systemtheorie”, München 2003.
- [16] Bernhard Dotzler (Hg.), “Futurum Exactum – Ausgewählte Schriften zur Kybernetik und Kommunikationstheorie”, Springer Verlag, Wien 2002.

- [17] Manfred Drack, "Is Paul Weiss' and Ludwig von Bertalanffy's System Thinking still valid today?", Proceedings of the 50th Annual Meeting of the International Society for the Systems, ISSS 2005.
- [18] Solomon Feferman (Hg.), "Kurt Gödel / Collected Works ", New York 1986.
- [19] Kurt Flasch, "Nikolas von Kues in seiner Zeit", Stuttgart 2004.
- [20] Christian Geyer (Hg.), "Hirnforschung und Willensfreiheit", Frankfurt am Main 2004.
- [21] A. D. Hall, R. E. Fagen, "Definition of System", in: Ludwig von Bertalanffy, Anatol Rapoport (ed.), "General Systems – Yearbook of the Society for General Systems Research Volume 1", 1956.
- [22] Veronika Hofer, "Organismus und Ordnung", Wien 1996.
- [23] Wolfgang Hofkirchner, "Projekt Eine Welt: Kognition – Kommunikation – Kooperation", Lit Verlag 2001.
- [24] Erich Jantsch, "Die Selbstorganisation des Universums", Erweiterte Neuauflage, München 1992.
- [25] Stuart A. Kauffman, "The Origins of Order", Oxford University Press 1993.
- [26] Hans Kelsen, "General Theory of Law and State", Cambridge 1949.
- [27] Nikolaus von Kues, "De ludo globi", in: Gerda von Bredow (Hg.), "Schriften des Nikolaus von Kues – Heft 21", Felix-Meiner-Verlag, Hamburg 1999.
- [28] Ruprecht Kurzrock Hg., "Forschung und Information Band 12, Systemtheorie", Berlin 1972.
- [29] Ervin Laszlo, "Systemtheorie als Weltanschauung", München 1998.
- [30] James Lovelock, "Gaia: the practical science of planetary medicine", Gaia Books, London 1991.
- [31] Niklas Luhmann, "Soziale Systeme", Frankfurt am Main 1984.
- [32] Niklas Luhmann, "Einführung in die Systemtheorie", Heidelberg 2002.
- [33] Humberto R. Maturana, "Biologie der Realität", Frankfurt am Main 1998.
- [34] Klaus Müller, "Allgemeine Systemtheorie", 1996.
- [35] Gregoire Nicolis, Ilya Prigogine, "Die Erforschung des Komplexen", München 1987.
- [36] Paul Oppenheim, Hilary Putnam, "Unity of Science as a Working Hypothesis", in: H. Feigl et al. (ed.), "Concepts, Theories and the Mind-Body-Problem. Minnesota Studies in the Philosophy of Sciences, Vol. 2", Minnesota 1958.

- [37] Talcott Parsons, “Das System moderner Gesellschaften”, München 1972.
- [38] Anatol Rapoport, “General System Theory”, Abacus Press, Cambridge 1986.
- [39] Gerhard Roth, “Fühlen, Denken, Handeln”, Frankfurt am Main 2003.
- [40] Gerhard Roth, Klaus-Jürgen Grün (Hg.), “Das Gehirn und seine Freiheit”, Göttingen 2006.
- [41] Stanley N. Salthe, “Evolving Hierarchical Systems”, Columbia University Press, 1985.
- [42] Wolf Singer, “Ein neues Menschenbild? – Gespräche über Hirnforschung”, Frankfurt am Main 2003.
- [43] Lars Skyttner, “General Systems Theory”, World Scientific Publishing 2005.
- [44] Adam Smith, “An Inquiry into the Nature and Causes of the Wealth of Nations”, 1776.
- [45] George Spencer-Brown, “Laws of form”, New York 1969.
- [46] Achim Stephan, “Emergenz – Von der Unvorhersagbarkeit zur Selbstorganisation”, Dresden University Press 1999.
- [47] Hao Wang, “Mind, Brain, Machine”, in: “Jahrbuch 1989 der Kurt-Gödel-Gesellschaft”, Salzburg 1989.
- [48] Norbert Wiener, “Cybernetics or Control and Communication in the Animal and the Machine”, MIT Press 1948.
- [49] Norbert Wiener, “Mensch und Menschmaschine”, Frankfurt am Main 1964.
- [50] Palle Yourgau, “Gödel, Einstein und die Folgen”, München 2005.

LEBENS LAUF

M. SCHLIEFNIG

ZUR PERSON

Name Martin Schliefnig
Geboren 17.11.1982 in Klagenfurt
Staatsbürger- Österreich
schaft

SCHULAU SBILDUNG

09/1993 – 06/2001 Alpen-Adria-Gymnasium Völkermarkt
09/1989 – 06/1993 Volksschule St. Kanzian

HOCHSCHULAU SBILDUNG

seit 03/2007 Doktoratsstudium der Informatik am Institut für Gestaltungs- und Wirkungsforschung der TU Wien
03/2005 – 02/2007 Masterstudium Computational Intelligence an der TU Wien, Diplomarbeit zum Thema “Reasoning in Semantic-Web-Applications”
10/2002 – 02/2005 Bakkalaureatsstudium Technische Informatik an der TU Wien
10/2001 – 09/2002 Studium der Angewandten Informatik an der Universität Klagenfurt

BERUFLICHER WERDEGANG

seit 07/2005 Mitgründer, Member of Board und Forschungsleiter der Smart Information Systems GmbH (www.smart-infosys.com)