



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

DISSERTATIONEN DER LMU



60

DANIEL KURZawe

Die Dynamik von Forschung und Gesellschaft

Simulationen von Wissenschaftsprozessen

OLMS

Eine Simulation von Wissenschaftsprozessen

Inaugural-Dissertation

zur Erlangung des Doktorgrades der Philosophie
der Ludwig-Maximilians-Universität
München

vorgelegt von
Daniel Kurzawe
aus Göttingen
2023

Referent: Prof. Dr. Wolfgang Balzer
Korreferent: Prof. Dr. Stephan Hartmann
Datum der mündlichen Prüfung: 06.11.2020

Daniel Kurzawe

Die Dynamik von Forschung und Gesellschaft
Simulationen von Wissenschaftsprozessen

Dissertationen der LMU München

Band 60

Die Dynamik von Forschung und Gesellschaft

Simulationen von Wissenschaftsprozessen

von
Daniel Kurzawe

Eine Publikation in Zusammenarbeit zwischen dem **Georg Olms Verlag** und
der **Universitätsbibliothek der LMU München**

Gefördert von der Ludwig-Maximilians-Universität München

Georg Olms Verlag AG
Hagentorwall 7
31134 Hildesheim
<https://www.olms.de>

Text © Daniel Kurzawe 2023

Diese Arbeit ist veröffentlicht unter Creative Commons Licence BY 4.0.
(<http://creativecommons.org/licenses/by/4.0/>). Abbildungen unterliegen ggf.
eigenen Lizenzen, die jeweils angegeben und gesondert zu berücksichtigen sind.

Erstveröffentlichung 2023

Zugleich Dissertation der LMU München 2020

Bibliografische Information der Deutschen Nationalbibliothek

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

Open-Access-Version dieser Publikation verfügbar unter:

<http://nbn-resolving.de/urn:nbn:de:bvb:19-296871>

<https://doi.org/10.5282/edoc.29687>

ISBN 978-3-487-16307-9

Inhaltsverzeichnis

Danksagung	IX
Vorwort	XI
Teil I Grundlagen	1
1 Eine Simulation von Wissenschaftsgemeinschaften	3
1.1 Perspektiven auf die Wissenschaft	5
1.2 Computersimulationen als Forschungsmethode	15
1.3 Die Anwendung von Simulationen in der Wissenschaftstheorie	17
1.4 Die Theoriefindung in simulierten Umgebungen.....	21
1.5 Prolog als Simulationsumgebung	23
2 Die Entwicklung und Darstellung wissenschaftlicher Theorien	25
2.1 Perspektiven auf Methoden und Theorien	26
2.1.1 Wissenschaftstheoretische Beschreibung von Theorien.....	28
2.2 Theorien und Modelle	29
2.2.1 Strukturalistische Theorien in Prolog.....	29
2.2.2 Theorien in der Simulation.....	32
2.3 Theoriefindung in der Simulation	33
2.3.1 Simulierte Welten auf Grundlage von Ereignissen und Gesetzen.....	34
2.3.2 Aufbau von Hypothesen in der Simulation.....	38
3 Soziale Prozesse in der Wissenschaft.....	41
3.1 Forschungs- und Wissenschaftshandlungen	42
3.1.1 Forschung und Lehre	44
3.1.2 Drittmittelförderung: Forschende im Wettbewerb.....	46
3.2 Akteure und Gemeinschaften in der Wissenschaft	47
3.2.1 Wissenschaftsgemeinschaften.....	49
3.3 Wissenschaftliche Handlungen auf Grundlage von Wissen, Überzeugungen und Intentionen	50
3.4 Kommunikation in der Wissenschaft.....	52
3.4.1 Publikationen.....	53
3.4.2 Publikationsprozesse.....	55

3.4.3	Gutachter in Reviewprozessen	56
3.5	Empirische Datengrundlage für Publikationsnetzwerke	58
3.5.1	Datenerhebung und Analyse von arXiv	59
4	Wissenschaft im Kontext der Gesellschaft	63
4.1	Die deutsche und europäische Forschungslandschaft	65
4.1.1	Drittmittelgeförderte Forschung als Prozess am Beispiel der DFG-Förderung	67
4.2	Die Dynamik von Forschung und Lehre	70
4.3	Die gesellschaftliche Wahrnehmung der Wissenschaft	72
Teil II	Simulation	75
5	Die Simulationsumgebung	77
5.1	Aufbau der Simulationsumgebung	77
5.1.1	Simulationskern	78
5.1.2	Simulationszeit	84
5.1.3	Die Planung von Aktionen	85
5.1.4	Eindeutige Namen durch UUIDs	86
5.2	Generierung der simulierten Umgebung	87
5.2.1	Variante 1a: Ereignisse und Typen	88
5.2.2	Variante 1b: Ereignisraum	90
5.2.3	Variante 2: Objekte, Relationen und Eigenschaften	92
5.3	Aktionen der Agenten im Detail	92
5.3.1	Aktionen zu Täuschungen in der Wissenschaft	106
6	Analyse von Simulationsläufen	109
6.1	Basis Simulation	109
6.1.1	Basiskonfiguration	109
6.1.2	Entwicklung der Agenten	110
6.1.3	Aktionen in den Simulationsbeispielen	111
6.2	Simulationsbeispiele zu Publikationsformen	112
6.2.1	Experiment 1: Reviews und Open Access	114
6.2.2	Experiment 2: Publikationen und Zitationen	116
6.3	Simulationsbeispiel zu Ressourcen als Grundlage von Handlungen	118
6.3.1	Experiment 3: Forschungsbudgets und Wissen	118
6.4	Einordnung	119

Teil III Implementierung	121
7 Die kommentierte Simulationsumgebung.....	123
7.1 Modul: simulation.pl.....	124
7.2 Modul: configuration.pl	133
7.3 Modul: utils.pl.....	138
7.4 Modul: create_world_event.pl	151
7.5 Modul: agents.pl.....	160
7.6 Modul: actions.pl.....	165
7.7 Modul: action_definition.pl.....	174
Literatur.....	199
Stichwortverzeichnis	207
Abbildungsverzeichnis.....	209
Tabellenverzeichnis	211

Danksagung

An dieser Stelle will ich den vielen Menschen danken, die mich während der Arbeit an diesem Projekt begleitet haben und mir Halt und Kraft gaben. Ich danke Prof. (i.R.) Dr. Wolfgang Balzer für seine Inspiration, die vielen Gespräche und insbesondere seine unermessliche Geduld. Ebenso danke ich Prof. Dr. Stephan Hartmann und Prof. (i.R.) Dr. Klaus G. Troitzsch für ihre Kommentare und Unterstützung bei der Finalisierung der Arbeit.

Ohne den unglaublichen Rückhalt und die grandiose Unterstützung meiner Frau Dr. Ruth Reiche hätte dieses Projekt wohl nie ein Ende gefunden. Ebenso danke ich meiner Familie, für die Unterstützung und Geduld mit mir. Meine Freunde bitte ich hier auch um Verzeihung für die vielen Abende und Wochenenden, an denen ich nicht anwesend war. Meine Kolleginnen und Kollegen mögen mir die Arbeitstage verzeihen, an denen mir die Spuren der Nacht ins Gesicht geschrieben standen. Und ich danke allen, die mit mir über meine Arbeit diskutiert, meinen Ideen gelauscht und mich auf dem Weg begleitet haben. Ich widme diese Arbeit Kasimir Frederik Kurzawe, welcher mir schlussendlich gezeigt hat, was im Leben wirklich wichtig ist.

Vorwort

Die Arbeit an diesem Projekt hat mich über viele Jahre begleitet und war dabei stets Teil der vielen Stationen und Windungen, die einem das Leben bereitet. Entsprechend zeigen die folgenden Kapitel nicht nur die theoretischen und praktischen Beschreibungen einer Simulation zum Wissenschaftssystem, sondern sind ebenfalls ein Zeugnis meines persönlichen Weges durch die Wirrungen dieser Forschungslandschaft. Auf dieser Reise hat sich mein Verständnis der Wissenschaft von dem Bild einer deskriptiven Wissenschaftstheorie hin zu einer lebendigen wissenschaftlichen Praxis gewandelt. Es war mir ein Anliegen dies auch im Text zu zeigen und auf die doch oft üblichen historisierenden Beschreibungen weitestgehend zu verzichten. Stattdessen sollten konkrete und aktuelle Anwendungsszenarien eine Grundlage für die Modelle sein und so mitunter auch die Aktualität und Relevanz der Wissenschaftstheorie aufzeigen. Und die Notwendigkeit dazu ist aktuell an vielen Stellen zu sehen: In dem gesellschaftlichen Diskurs zur Rolle der Forschung und Bildung, in der Wissenschaftspolitik und nicht zuletzt ganz konkret auch in den Informationswissenschaften. Besonders hier bedarf es neuer Konzepte und Denkweisen, um der steigenden Komplexität und Informationsvielfalt in der Wissenschaft zu begegnen. Die Informationswelt als Ganzes gewinnt zunehmend an Komplexität und Inhalt und mit ihr insbesondere auch die Wissenschaft. Die darin bestehenden Systeme wachsen heran, werden globaler und differenzieren sich in immer weiter verzweigten Armen aus. Neue Untersuchungsgegenstände werden definiert, Disziplinen entstehen und neue Communities gründen sich. Auch wenn die technischen Möglichkeiten zur Informationsverarbeitung und einhergehend die Vernetzung von Informationssystemen zunehmen, so bleibt die Wissenschaft doch letztendlich ein menschlicher Prozess und damit an all die damit einhergehenden Möglichkeiten und Grenzen gebunden. Es sollte bedacht werden, dass im Zentrum dieser Prozesse letztendlich dann auch die menschliche Erkenntnis steht, nicht die Ansammlung von Daten und Fakten. Je komplexer die Informationswelten werden, um so wichtiger ist es auch die Rückführung in die sozialen Systeme zu betrachten.

Diese Arbeit versucht diesen Kreislauf aufzuzeigen. Sie beschreibt ein technisches Konstrukt, eine Reflexion über die Wissenschaft und eine Auswahl dazu geschaffener Simulationsmodelle. Strukturelle Beschreibungen werden in den Kontext sozialer Systeme gestellt und dann in Programmzeilen zu einem Simulationsmodell geformt. Dabei sind die Simulationen teils pragmatischer als ich es mir zu Beginn des Projekts gewünscht hätte, doch immer mit dem konkreten Ziel, reale Systeme in Teilaspekten zu repräsentieren. Aber die Arbeit zeigt eben auch die Verbindung von theoretischen und angewandten Betrachtungen der Wissenschaft und bietet damit einen neuen Blick auf diese Welten. Ich hoffe mit meiner Arbeit einen Beitrag und Inspi-

ration dazu zu bieten, sich dieser zentralen Herausforderung der Wissenschaft zu stellen.

Das Projekt startete dabei mit einer recht allgemeinen Vorstellung zu einer *Simulation der Wissenschaft*. Darin sollte vieles explizit gezeigt werden, was in der Literatur gerne in wenigen blumigen Worten erwähnt wird. Ich wollte offenlegen, was oft in Nebensätzen oder Fußnoten über die konkrete Anwendung wissenschaftstheoretischer Überlegungen zu finden war. Es sollte gezeigt werden, wie denn nun die vielen Stränge der Wissenschaftsforschung ineinandergreifen können und damit auch aufgezeigt wird, wie in unserer Gesellschaft Erkenntnisse entstehen und sich verbreiten. Ich wollte genau die Lücken zwischen Kuhn, Knorr-Cetina, der formalen Wissenschaftstheorie und Wissenschaftssoziologie füllen, oder zumindest die Verbindungen aufzeigen. Balzer zeigte hier in seinen vielen Arbeiten genau diese Relevanz, betonte die sozialen Einflüsse in der Wissenschaftstheorie und prägte ebenso die formalen Ansätze in den sozialen Simulationen. Aufbauend auf dieser Basis, konnte die Reise starten. Wenn auch die teils weit gegriffenen Vorstellungen sich über die Zeit in konkretere und damit auch realistischere Ziele wandelten, blieb aber die Vorstellung eine Brücke zwischen Theorie und Praxis zu schlagen. Diese Brücke sollte schlussendlich eine Verbindung zwischen den vielen Perspektiven auf die Wissenschaft bilden. Im Kern dieser Überlegungen stand eine übergreifende Simulation. Methodische Konkurrenz fürchtete ich durch das Aufblühen der Forschung zum *Maschinellen Lernen* und der *Künstlichen Intelligenz*. Dabei unterscheiden sich die Ansätze recht klar. In den hier besprochenen Simulationen werden ja gerade konkrete Modelle geschaffen und durch explizite Beschreibungen von Prozessen gängige Beschreibungsmodelle zusammengeführt. Dem gegenüber steht die Abstraktion von Modellen in neuronalen Netzen durch komplexe Algorithmen. So erfolgreich dieser Ansatz sich in der Anwendung zeigt, so wenig Erkenntnisse zeigt dieser über die ablaufenden Prozesse und Strukturen. Entsprechend beginnt diese Arbeit mit einem theoretischen Grundgerüst. Dies wird dann über die Kapitel schrittweise um konkrete und reale Systeme ergänzt, um dann daraus Modelle abzuleiten. Von der anderen Seite hatte O'Connor kurz vor Vollendung meiner Arbeit Simulationen zu ähnlichen Themenkomplexen vorgestellt. Methodisch wählte sie jedoch andere Wege und wählte weniger detailliertere Modelle.

Hier zeigt sich die Reise zwischen Methoden, Themen und Begrifflichkeiten aus unterschiedlichen Disziplinen mit den zugehörigen Begriffswelten und Fragestellungen. Passenderweise wurde in den letzten Jahren zunehmend auch in der Öffentlichkeit über die Wissenschaft diskutiert. Diverse Plagiatsaffären, Monopolstellungen von wissenschaftlichen Verlagen, die prekären Arbeitsbedingungen, etwa unter dem Slogan *Ich bin Hanna*, und starre Strukturen an Instituten rückten in den öffentlichen Diskurs und fanden ihren Weg in die Medien. Neben den theoretischen Reflexionen über die Wissenschaft, habe ich aber auch meine praktischen Erfahrun-

gen im Wissenschaftssystem einfließen lassen können: Über die Jahre habe ich selbst über viele Drittmittelanträge oder die Beteiligung an Gutachterverfahren praktische Einblicke sammeln können. Ich konnte Artikel publizieren, lehren und auf Konferenzen vortragen. Ebenso habe ich Projekte und Ressourcen verwaltet. Ich wurde Teil des Systems, welches ich hier betrachte. Diese Erfahrungen haben mich bei der Wahl der Simulationsbeispiele inspiriert. Hinzu kamen die viele Gespräche mit Forschenden unterschiedlichster Disziplinen. Genau diese Einblicke und der Austausch haben mein Verständnis von dem, was wir alle unter *Wissenschaft* verstehen, noch einmal deutlich erweitert. Gerade die vielen Eindrücke von Gesprächen mit Forschenden unterschiedlichster Disziplinen, etwa mit Forschenden im indonesischen Dschungel, Literaturwissenschaftler*innen oder Physiker*innen haben die Theorie in einen anderen Kontext gesetzt. Aber ich sprach auch mit der Verwaltung und Administration, den Techniker*innen und auch Entscheidungsträger*innen und Politiker*innen und habe die weiteren Arme des Wissenschaftsbetriebs kennengelernt. All diese Stimmen haben mir geholfen ein Bild zu zeichnen und dies nicht auf Laborstudien zu beschränken. Aber dies war eben nur der eine Teil der Betrachtung. Der Aufbau einer Simulation war mitunter eine technische Fragestellung, die teils formaler Natur war und teils der Informatik zuzuordnen wäre.

Um in diesem thematischen Geflecht bestehen zu können und zwischen thematischer Tiefe und inhaltlicher Breite zu entscheiden, verlangt es nach Kompromissen. Je mehr Perspektiven ergänzt wurden, um so mehr Theorie galt es zu berücksichtigen. Da das Ziel aber ja gerade keine neue Theorie sein sollte, sondern eine lauffähige Simulation, wurden hier Grenzen gesetzt. Die Arbeit sollte eine möglichst umfangliche Spannbreite zeigen, mit einer Verbindung von stark idealisierten wissenschaftstheoretischen Perspektiven und realen soziologischen und wissenschaftshistorischen Gegebenheiten.

Teil I

Grundlagen

1 Eine Simulation von Wissenschaftsgemeinschaften

Lebten wir in einer für die Wissenschaft perfekten Welt, würden wir die Ereignisse, die uns umgeben, durch unsere Beobachtung in ihrer Gesamtheit erfahren können. Wir würden all ihre Eigenschaften und Relationen erkennen und die dahinter verborgenen Gesetze verstehen. Gemeinschaftlich würden wir unsere Perspektiven ergänzen und so schrittweise ein immer vollkommeneres Bild der Realität schaffen. Erklärungen würden mit jeder weiteren Beobachtung und jedem Gedanken präziser werden. In dieser perfekten Welt gäbe es eine vollkommene, widerspruchsfreie und vollständige Sprache, in der alles eindeutig beschrieben würde. Es ständen genug Ressourcen bereit, um allen Fragen nachzugehen. Jedem Individuum stände genug Zeit zur Verfügung, um das gesamte, bis dahin gesammelte Wissen zu erfassen und um dies weiter ergänzen zu können. Die Forschung wäre eine gesamtgesellschaftliche Kooperation mit der gemeinsamen Zielsetzung, Wissen zu schaffen, welches allen Interessierten im vollen Umfang zur Verfügung steht. Doch die Realität ist leider eine andere.

Mit jedem Blick werden Probleme sichtbar, die neue Grenzen und Herausforderungen aufzeigen. Je tiefer wir in die Welt blicken, umso mehr sind wir auf die Hilfe von Instrumenten angewiesen. Je präziser wir versuchen zu schauen, um so komplexer scheint die Welt. Dabei nehmen wir die Welt auf unterschiedliche Weise wahr und Widersprüche entstehen. Sinne geben nur ein verzerrtes Bild und auch die Sprache selbst, in der wir das Gesehene und Gedachte formulieren, ist nicht absolut exakt und enthält mindestens subjektive Anteile. Approximationen sind notwendig, um mit den Ungenauigkeiten zu arbeiten. Aus unseren individuellen Erfahrungen, Erlebnissen und unserem Vorwissen ergeben sich spezifische Perspektiven und Ansichten. Diese sind nicht immer klar ersichtlich und miteinander vereinbar. Erfahrungen und Vorannahmen können neutrale Blicke färben. Kommunizieren Forschende, so geschieht dies auf verschiedensten Wegen: Im direkten Austausch auf Konferenzen oder über Publikationen werden Standpunkte dargestellt und für die eigenen Ansichten geworben. Selbst wenn die Kommunikation in präzisen und möglicherweise stark formalisierten Sprachen erfolgt, bleiben dennoch Perspektiven, Interpretationsräume und Ansichten offen. Der Austausch ist auf der einen Seite insbesondere in den letzten Jahren durch die Digitalisierung von Informationen sowie Open Access und auf der anderen Seite durch Patente und Kommerzialisierung von wissenschaftlichen Zeitschriften in einem Umbruch. Die Privatisierung von Bildung beeinflusst Inhalte und die Zugänglichkeit zu Wissensressourcen.

Soziale Prozesse finden an vielen Stellen der vergleichsweise stark reglementierten und nüchternen Forschung statt. Soziale Gefüge, wie Gutachterkreise, sind Teil der

Wissenschaft und haben ihre Hürden und Querbeziehungen und damit einen Einfluss auf die Forschung. Reputation, Macht, das Geschlecht oder die soziale Herkunft und Stellung können Einflussfaktoren sein. Um Forschung betreiben zu können, sind Ressourcen notwendig. Im einfachsten Fall sind es die Zeit, um die Forschung durchzuführen, und die finanziellen Mittel, um die notwendige Ausstattung zu finanzieren. Fördergelder werden auch über soziale Prozesse organisiert. Institutionen organisieren mit der Hilfe von Gutachtersystemen die Verteilung der Mittel. Gesellschaft und Wirtschaft üben über die Politik einen Einfluss auf die Schwerpunktsetzung von Förderprogrammen aus. Um eine Forschungsförderung zu erhalten, muss also die Notwendigkeit und Machbarkeit gegenüber Gutachter*innen dargestellt werden. Doch auch die Gesellschaft muss die Forschung legitimieren, insofern über staatliche Instrumente Mittel für diese bereitgestellt werden oder kommerzielle Produkte erworben werden, welche die Produkte der Forschung verwertet. Hier treten soziale Prozesse auf einer gesamtgesellschaftlichen Ebene in Erscheinung. Neben der eigentlichen Forschungsleistung rücken zunehmend auch die Reputation und die gesellschaftliche Relevanz in den Fokus der Wissenschaft. Aus diesem Themenkomplex ergeben sich eine Reihe von Fragestellungen in Bezug auf die Wissenschaft. Diese Arbeit beschäftigt sich damit, wie Forschende unter welchen Einflussfaktoren Wissen generieren und verbreiten können.

Beginnend bei der Definition der Welt, in der die Forschenden agieren, über die sozialen Gruppen bis hin zur gesamtgesellschaftlichen Betrachtung, zeigt sich, dass Wissen eine zentrale Stellung in der Forschung hat und entsprechend komplexe Verbindungen zwischen den beschriebenen Perspektiven bestehen und wie mannigfaltig die Einflussfaktoren sind. Doch auch die Varietät der Disziplinen, Methoden und Perspektiven mit den teils spezifischen sozialen Schichten bieten weitere Betrachtungsdimensionen mit Abhängigkeiten und hoher Komplexität. Computersimulationen erlauben es, solche komplexen Prozesse darzustellen und zu untersuchen, in der die Dynamik zwischen einzelnen Parametern und Ebenen in einer kontrollierten Umgebung dargestellt und untersucht werden kann. Die darin geschaffenen künstlichen Gesellschaften¹ erlauben es, Forschende in idealisierten Zuständen zu beobachten. Die Konstruktion einer solchen Simulation stellt allerdings bereits die erste Herausforderung dar und liegt im Fokus dieser Arbeit. Um die Fragestellungen umfassend zu bearbeiten, sollte eine darauf ausgelegte Simulationsumgebung konzipiert werden.² Es beginnt damit, dass bei einer Verbindung dieser unterschiedlichen Perspektiven ein Rahmen geschaffen werden muss, in dem sowohl die theoretischen als auch sozialen und gesellschaftlichen Prozesse ineinandergreifen können. Wird dann darauf aufbauend ein Simulationsmodell geschaffen, erlangt dies schnell eine hohe

1 Vgl. Balzer, Kurzawe und Manhart 2014.

2 Vgl. Ihrig und Klaus G Troitzsch 2013, S. 8.

Komplexität und dadurch eine hohe Anzahl von Parametern. Um hier beispielhafte Prozesse zu beschreiben, ist es notwendig Schwerpunkte für die Betrachtung zu wählen. Die Prozesse werden dazu stark idealisiert und in Extrembedingungen gesetzt, um beobachtbare Effekte zu provozieren und Abhängigkeiten zwischen Parametern und Einflussfaktoren aufzuzeigen. Dafür werden auch exemplarische empirische Daten zusammengetragen und erhoben, um Kenngrößen für Simulationsexperimente zu gewinnen. Folglich sind die Schaffung einer Simulationsumgebung für eine modellhafte Abbildung und Untersuchung von prototypischen Wissenschaftsprozessen mit entsprechenden Rahmenhandlungen der Wissenschaft als auch konkrete Forschungshandlungen in einer Multiagentensimulation das Ziel dieser Arbeit.

1.1 Perspektiven auf die Wissenschaft

Die Verwendung von Simulationen zur Erforschung der Wissenschaft erlaubt zwei Arten von Untersuchungen: Zum einen lassen sich Hypothesen über den Wissenschaftsprozess aus unterschiedlichen Perspektiven erforschen und zum anderen lassen sich konkrete Systemzustände untersuchen, indem die konstruierten Systeme und erzeugten Gesellschaften in definierte Zustände versetzt und dann beobachtet werden. Unterschiedlichste ineinandergreifende Prozesse und interagierende Akteure bilden aus wissenschaftssoziologischer Perspektive das Konstrukt der Wissenschaft.³ Der Wissenschaftsprozess ist entsprechend facettenreich und viele Forschungsfelder haben Bestandteile des Wissenschaftsprozesses selbst zum Forschungsgegenstand. Die Struktur und Dynamik von Theorien, Methoden und Prozessen steht etwa im Fokus der Wissenschaftstheorie. Methodisch ordnet sich diese zwischen der Philosophie, Soziologie, Wissenschaftsgeschichte, aber auch der formalen Logik ein. Während die empirische Soziologie die real gelebte Wissenschaft in der Interaktion zwischen Personen und Gruppen betrachtet, konstruiert die Wissenschaftstheorie einen Idealtyp der Wissenschaft. Oder wie Giere schreibt: „Philosophical theories of science are generally theories of scientific rationality. The scientist of philosophical theory is an ideal type, the ideally rational scientist“⁴. Paul Thagard betont noch konkreter die soziale Komponente: „In science, however, there is also an important social dimension, involving interaction among scientists and social regulatory forces such as peer review“⁵. Eine Verschränkung dieser Perspektiven ist jedoch weder Teil der Wissenschaftstheorie mit den deskriptiven Betrachtungen der Theoriestruktur und Historie mit idealisierten Forschungsprozessen noch der Soziologie mit entsprechend empirisch betonten Betrachtungen, welche die Wissenschaft als ein System, bestehend

3 Vgl. Nowotny, Scott und Gibbons 2003.

4 Giere 1997, S. 2.

5 Thagard 1988, S. 88.

aus Akteuren, Handlungen und deren Interaktionen, beschreibt, jedoch die Struktur und Vielschichtigkeit der Inhalte selbst ausblendet.

Wissenschaftstheoretische Betrachtungen nehmen insbesondere gerne den Theorieaufbau in der Physik mit entsprechender Historie als Beispiel. Der Fokus auf die Physik und andere empirische Naturwissenschaften (etwa der Chemie oder Biologie) mag zum einen daran liegen, dass sich hier klare Entwicklungsstränge und Diskurse nachvollziehen lassen, aber möglicherweise mag es auch auf den Unterschieden in der sprachlichen Definition des Begriffs der Wissenschaft mit all ihren Facetten (Geisteswissenschaften, Lebenswissenschaften, Naturwissenschaften, Sozialwissenschaften, Kulturwissenschaften etc.) liegen. Dies wird jedoch der Vielfalt nicht gerecht. Theoriebegriffe, Methoden und Forschungsfragen in methodisch weniger formal geprägten Disziplinen sind möglicherweise diffuser und die Forschung findet etwa in Diskursen mit vielen parallelen Strängen statt.

Aus einer Kombination der hier genannten Perspektiven bildet sich ein umfassenderes Bild der Wissenschaft, welches nicht nur Theorien, Wissensformen, Methoden oder Sozialstrukturen getrennt in den Blick nimmt. So lassen sich die Stärken der unterschiedlichen Disziplinen verbinden. Auch weitere Aspekte anderer Forschungsbereiche lassen sich ergänzen: Beispielsweise betrachtet die Philosophie die grundlegenden Fragestellungen zur Art und dem Wesen des Wissens, Denkens und Handelns, wie etwa Fragestellungen in der Erkenntnistheorie, Logik, Sprachphilosophie, Ethik und Entscheidungstheorie. Durch eine historische Betrachtung und Aufarbeitung der Wissenschaftsgeschichte, können diese Perspektiven in Bezug zum realen Wissenschaftsgeschehen gesetzt werden.

In diese Aufzählungen von Forschungsperspektiven auf das Wissenschaftsgeschehen können sich noch weitere Forschungsfelder einreihen, welche sich direkt mit Aspekten des Gebildes der Wissenschaft befassen oder sich in zweiter Ordnung mit Teilsystemen und Komponenten der Wissenschaft beschäftigen. Das wären etwa die Psychologie, welche innere Prozesse der Akteure beschreibt oder die Linguistik, welche sich mit der Sprache befasst, die das Wissen transportiert. Das hier im folgenden gezeichnete Bild wird allerdings nur einen kleinen Teil der denkbaren Perspektiven abdecken können und in diesem Aspekt leider kompromissbehaftet bleiben. So werden insbesondere Konzepte der Wissenschaftstheorie und Soziologie ihre Anwendung finden wohingegen etwa politik- oder wirtschaftswissenschaftliche Aspekte nur erwähnt werden. Dennoch, so ist das Ziel, werden die zentralen Komponenten herausgestellt und in die Betrachtung einbezogen.

In der Konzeption der Simulation und entsprechender Simulationsmodelle stellen die Unterschiede der beschriebenen Betrachtungsweisen auf die Wissenschaft die erste Herausforderung dar. Sowohl die Begrifflichkeiten als auch die verwendeten Methoden unterscheiden sich. Deutlich wird dies bereits bei der Betrachtung der Methoden und Beschreibungen in den erwähnten Disziplinen. Während in der Wis-

senschaftssoziologie die Wissenschaft als soziologisches Konstrukt beschrieben wird und die Akteure der Wissenschaft somit im Fokus der Betrachtung stehen, wird dieser Aspekt in der Wissenschaftstheorie nur im begrenzten Rahmen thematisiert. Die Struktur von Theorien und spezifische Fragen, etwa zur Inkommensurabilität, der Theoretizität von Begriffen oder konkreten Entwicklungen von Theorienetzen werden dabei dann wieder nicht in der Wissenschaftssoziologie betrachtet, genauso wie die Gruppendynamik von Wissenschaftsgemeinschaften nicht Teil der Wissenschaftstheorie ist. Eine Schnittstelle für diese Perspektiven hat Kuhn 1962 geschaffen, indem er eine Verbindung zwischen sozialen Prozessen der Wissenschaft und den Entwicklungen wissenschaftlicher Theorien gezogen hat. Der wissenschaftstheoretische Strukturalismus hat diese Verbindung aufgenommen und mit der präzisen Theoriebeschreibung untermauert.⁶

Jedoch bietet die Verbindung zwischen Wissenschaftssoziologie und Politikforschung eine weitere Sichtweise: Die Wissenschaft lässt sich etwa als Wissensproduzent verstehen.⁷ Die systematische Betrachtung von Forschungsorganisationen, Finanzierungen und den politischen Entscheidungsprozessen ergänzt dieses Bild. Indikatoren innerhalb von sozialen Systemen versuchen die Wissenschaft in ihrer Struktur messbar zu machen.⁸ Die definierten Wissenschaftsindikatoren beziehen sich dabei alleinig auf die sozialen Konstrukte, etwa dem Erfolg bei der Einwerbung von Drittmitteln. Hierbei werden jedoch Begriffe der Wissenschaftstheorie außer Acht gelassen und keine Verbindungen zwischen der inhaltlichen und strukturellen Entwicklung eigentlicher Theorien und den sozialen Prozessen gezogen.

In anderen wissenschaftssoziologischen Untersuchungen wird der Sozialraum untersucht, in dem die Forschung erfolgt. Forschende selbst sind dabei Untersuchungsgegenstand und werden in der direkten Interaktion anhand ihrer Handlungen betrachtet und beschrieben.⁹ Hier werden ebenfalls nicht die eigentlichen Theorien einbezogen.

1. Auf der Theorieebene werden Theorien in ihrer Struktur und die Dynamik zwischen Theorien abgebildet. Auf dieser Ebene wird zum größten Teil die wissenschaftstheoretische Perspektive abgebildet. Theorien werden nicht als geschlossene Konstrukte in Form von einfachen Satzmengen oder atomaren Ausdrücken verstanden, sondern werden als komplexe Strukturen, bestehend aus Regelsystemen und den

⁶ Das Konzept wurde zunächst in Sneed 1971 beschrieben und über die Zeit weiterentwickelt. In Balzer, Moulines und Sneed 1987 findet sich eine umfassende Darstellung des formalen Apparats mit Beispielen. Balzer und Brendel 2018 zeigt eine aktualisierte Sichtweise und fokussiert sich dabei nicht auf formale und wissenschaftstheoretische Aspekte, sondern beschreibt auch andere Perspektiven.

⁷ Vgl. etwa Balzer 2003 oder Knorr-Cetina und Harré 2012.

⁸ Hornbostel 1997 entwickelt entsprechende Indikatoren.

⁹ Vgl. Knorr-Cetina und Harré 2012.

Anwendungskontexten, dargestellt.¹⁰ Es wird genau definiert, welchen Regeln Modelle einer Theorie unterliegen. Theorien können dabei miteinander in Beziehung stehen. Durch die Betrachtung von Beziehungen können größere Kontexte in Form von Theoriennetzen aufgebaut und analysiert werden. Diese Konstrukte sind über die Zeit variabel. Mit dem Fortschreiten der Forschung verändern sich Theorien und intertheoretische Beziehungen. Somit hat der Forschungsprozess Einfluss auf Theoriennetze und die darin enthaltenen Theorieelemente. Durch die Entwicklung von Hypothesen, Theorien und Netzen ergeben sich umfangreiche Verzweigungen, wobei sowohl Theorien innerhalb eines Netzes als auch ganze Theoriennetze in Konkurrenz stehen können.¹¹ Nicht nur Theoriennetze sind einer zeitlichen Dynamik unterworfen, sondern auch die intendierten Anwendungen der Theorien. Diese verändern sich über die Zeit aufgrund von Forschungsprozessen, doch auch durch weitere Einflussfaktoren wie beispielsweise ökonomische und gesellschaftliche Entwicklungen.

4. Simulationsebene	Die Abbildung der inhaltlichen Ebenen in einer allumfassenden Simulation.	Wissenschaftliche Entdeckung, technische Entwicklungen und genauere Beobachtung bieten weitere Anwendungsszenarien. Doch ebenso können ursprünglich intendierte Anwendungen durch diese Entwicklungen wegfallen oder es können sich andere Theorien als geeigneter erweisen. Daten, welche durch empirische Beobachtung, Vorhersage oder Anwendung der Hypothesen entstehen, ermöglichen es, Theorien zu bewerten. Je stärker die durch die Theorie postulierten Vorhersagen abweichen, desto mehr
3. Gesellschaftsebene	Wissenschaft in Kontext zu anderen Bezugssystemen (etwa Ökonomie oder Politik).	
2. Sozialebene	Soziale Prozesse in der Wissenschaft und deren Einfluss auf die Theorieentwicklung.	
1. Theorieebene	Wissenschaftstheoretische Betrachtung von Theorien und deren Entwicklung.	

Abbildung 1.1: Überblick über die Ebenen

Zweifel entstehen. Diese können sowohl die Daten, die Messmethoden, aber auch die Theorie betreffen. Durch die Weiterentwicklung der Messmechanismen und der Erweiterung von intendierten Anwendungen ist somit auch der Passungsgrad zwischen den Hypothesen und Daten einer zeitlichen Dynamik unterworfen. Je nach Passung von Hypothesen zu Daten, ergeben sich Vergleichbarkeitskriterien zwischen unterschiedlichen Theorien und Hypothesen. Dadurch treten indirekt verschiedene Hypothesen und Theorien in eine Konkurrenz zueinander. Empirische Theorien entstehen in der Simulation aus Beobachtungen. Beobachtungen werden über die Wahrnehmung der Agenten in Daten überführt und aus diesen Hypothesen und Theorien abgeleitet. Jeder dieser Schritte bietet eine prinzipienbedingte Unschärfe in der Beobachtung, Sprache und Kommunikation.

¹⁰ Vgl. Balzer, Moulines und Sneed 1987.

¹¹ Vgl. Moulines 2014. zeigt etwa vier Typen von intertheoretischen Beziehungen auf.

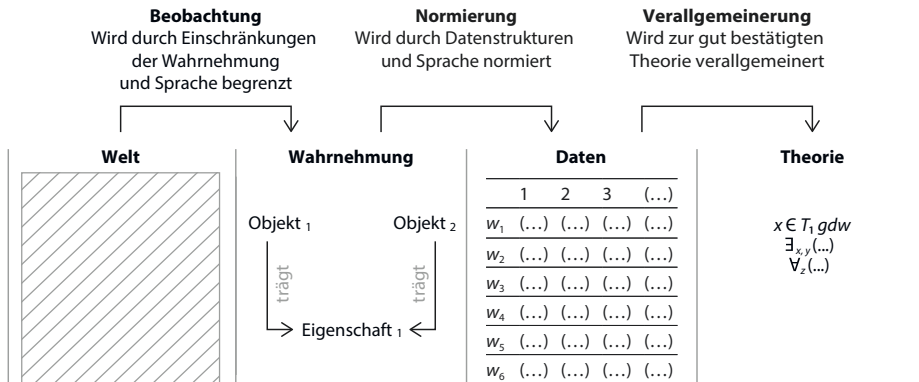


Abbildung 1.2: Mögliche Varianten der Theoriebildung durch empirische Beobachtung

2. Nicht nur die Theorien stehen in Konkurrenz zueinander, sondern auch Forschende um Reputation oder finanzielle Ausstattung. Dies gilt auch, wenn sie in unterschiedlichen Forschungskontexten aktiv sind. Setzen sich Forscher auf sozialer Ebene durch, können Theorien auch jenseits der formalen Passung zu Beobachtungen, der Qualität des Erklärungsgehalts oder ähnlicher Faktoren in ihrer Entwicklung beeinflusst werden. Soziale Phänomene haben somit auch Einflüsse auf die Theorieentwicklung.¹² Parallel zu der Betrachtung der Theorieebene lässt sich das Wissenschaftsgeschehen auch auf der sozialpsychologischen Ebene mit dem Fokus auf gesellschaftliche Prozesse und der Interaktion von Individuen betrachten. Diese Prozesse werden in der Simulation von Agenten, autonom agierenden Programmbestandteilen mit eigenen Überzeugungsbasen und Handlungen, repräsentiert.

Dabei sind insbesondere drei Typen von Agenten in der Forschung hervorzuheben: Dozent*innen, Studierende und Förder*innen. Jeder Agententyp hat spezifische Eigenschaften, insbesondere in der Form von Handlungen und Intentionen. Dozent*innen forschen und geben Wissen weiter. Dies können Überzeugungen, Fakten, spezifische Theorien und Methoden sein. Studierende erlernen Wissen, bilden dazu Überzeugungen auf Grundlage des Erlernten und erlangen neue Fähigkeiten. Die Fördernden stützen das System mit Ressourcen, sind aber selbst davon abhängig, dass sie legitimiert werden. Forschungsförderer sind von politischen Entscheidungen abhängig, da sie die zugewiesenen öffentlichen Gelder verteilen oder sind selbst Teil des politischen Systems. Dadurch unterliegen sie dem Einfluss von Wahlen und werden in ihrem Handeln somit direkt oder indirekt durch die Gesellschaft legitimiert.

¹² Vgl. Balzer und Manhart 2014.

Ändert sich eine öffentliche Meinung zu bestimmten Forschungsthemen, etwa der Forschung mit praktischen Versuchen in der Gentechnik, der embryonalen Stammzellenforschung an Menschen oder der Forschungsreaktoren in der Kernphysik, so ändert sich auch der Mittelfluss durch Forschungsförderer.

Die Überzeugungen können sich auf Theorien, Daten oder auf das Vertrauen zu den Arbeiten anderer Agenten des Systems beziehen. Aus Beobachtungen oder Annahmen werden Überzeugungen in Forschungshandlungen gebildet, welche selbst wieder aus der Auswertung von Überzeugungen und der Wissensbasis entstehen. Auch werden sie aus Überzeugungen anderer Agenten gebildet. Dies kann im Austausch, etwa durch Publikationen geschehen. Dabei sind Überzeugungen von Faktoren, wie der Güte der Daten, der individuellen Befähigung diese zu beurteilen oder der Ausstattung, um mit hinreichender Qualität zu forschen, abhängig. Die Überzeugungen und Handlungen sind verschränkt. Die Verbindung zwischen Überzeugungen und Handlungen lässt sich über das *Belief-Desire-Intention Model (BDI)*¹³ beschreiben. Das BDI beschreibt die Verbindung zwischen Überzeugungen in einer Wissensbasis (Belief), den Wunschzuständen (Desire) sowie Zielen zu Planhandlungen (Intention). Forschende haben zum Ziel, Erklärungen zu Phänomenen zu finden und diese von andere Erklärungsmodellen abzugrenzen und zu verteidigen. Daneben bestehen noch eine Vielzahl weiterer individueller Ziele: Beispielsweise die Medienpräsenz, um die Reputation zu stärken.¹⁴

Je nach Komplexität und Art der Problemstellung handelt es sich bei der Theoriefindung um eine schöpferische Tätigkeit. Diese verlangt nach Kreativität oder Phantasie, wie Heinrich Parthey schreibt: „Für Forscher ist dabei Phantasie wichtiger als bisheriges Wissen. Unter Phantasie wird dabei die produktive-schöpferische Fähigkeit des Bewußtseins verstanden, Elemente des Gedächtnisses sinnlich-anschaulich neu zu kombinieren beziehungsweise zu neuen Vorstellungen und Gedankenverknüpfungen weiterzuentwickeln.“¹⁵ Diese Beschreibung verführt jedoch zu einem romantisierten oder auch mystifizierten Verständnis der Theoriefindung. In diesem Kontext wird auch von einem *Eureka-Moment* gesprochen: Einer plötzlichen Eingebung, in welcher Hypothesen spontan entstehen. Dem spontanen und kreativen Verständnis steht ein stark reglementierter Prozess von Handlungen gegenüber, welche explizit die exakt wiederholte Ausführung von genau elaborierten Methoden verlangen. Detaillierte Beispiele dieser Prozesse finden sich etwa bei Knorr-Cetina und Harré 2012. Die Entdeckung sei dem zufolge ein sozialer Prozess, in welchem jedoch wieder die „Innovationen [...] ihren Ursprung in der Kreativität des individuellen Wissenschaftlers“¹⁶ haben. Jantzen 2016 argumentiert, dass die Theoriefindung auf Grundlage von

13 Vgl. Wooldridge 2000, S. 21ff und Rao und Georgeff 1995.

14 Fuhrin 2013 beschreibt das Verhältnis zwischen Wissenschaft, Politik, Medien und Öffentlichkeit.

15 Vgl. Parthey 2013, S. 9.

16 Ebd., S. 36ff.

der Logik steht und eine Art *discovery function* wissenschaftliche Entdeckungen und Erklärungen ermöglicht.

Durch die Anwendung erprobter Methoden, etwa die Durchführung von Experimenten mit variierenden Variablen, werden vergleichbare Ergebnisse erzielt und das wissenschaftliche Fundament der Reproduzierbarkeit, sowie der Möglichkeit zur Bewertung der Forschungsergebnisse geschaffen. Dadurch ergeben sich zwei Effekte: Zum einen bleibt den Forschenden bei der Verwendung einer etablierten Methode hier eine Entwicklung der Methode erspart und es stehen mehr Ressourcen für die Durchführung von Forschung ohne Methodenentwicklung zur Verfügung. Zum anderen hängt die Validität der Forschung auch an der Akzeptanz der Methode. Ist die Methode etabliert und entspricht sie dem aktuellen Wissensstand, können sich die Forschenden darauf berufen. Wenn methodisch korrekt gearbeitet wird, so stehen nur die Wahl der Methode, das Ergebnis und die daraus abgeleiteten Folgerungen zur Diskussion. Das letzte Argument kann natürlich auch ein Risiko darstellen, denn prinzipielle Kritik an einer Methode vererbt sich in der Folge auch auf die darauf aufbauende Forschung. Insbesondere kann eine stark etablierte Methode auch zu einer Überschätzung oder falschen Anwendung führen. Ein bekanntes Beispiel ist der Signifikanztest, welcher zur Legitimation von statistischen Analysen verwendet wird, jedoch häufig in seiner Aussagekraft überschätzt oder falsch gedeutet wird.¹⁷

Publikationen der Agenten können Fakten verbreiten. Diese werden mit den bestehenden Überzeugungen und etwa auf der Grundlage weiterer Beobachtungen reflektiert und darauffolgend, entsprechend der Vereinbarkeit mit eigenen Hypothesen und Überzeugungen, bewertet. Die Wissenschaft befindet sich durch die Etablierung digital gestützter Forschung und immer stärker vernetzter Infrastrukturen mit einer wachsenden Menge an zugänglichem Wissen in einem Umbruch. Durch Open Science Bewegungen werden die Rollen von Verlagen, Gutachtersystemen und Infrastruktureinrichtungen, wie Bibliotheken und Datenzentren, neu definiert. Forschungsförderer verlangen die Offenlegung der Forschung zugrundeliegenden Daten und Methoden und der vollständigen Forschungsergebnisse. Dadurch entwickeln sich zunehmend auch Forschungsbereiche, welche sich vollständig auf den Aspekt der Weiterverwendung von Daten konzentrieren, *data driven science* genannt. Auch werden weitere Arten der Wissensgenerierung ermöglicht. Forschung geschieht nicht mehr unbedingt nur durch abgegrenzte disziplinär oder methodisch definierte Gemeinschaften, sondern wird in einem gesellschaftlichen Wandel aufgebrochen.¹⁸ Sie findet zunehmend disziplinenübergreifend und in verschiedenen Ebenen der Gesellschaft statt. Die traditionelle und disziplinäre Forschung wird nicht durch diese als

17 Der Artikel von Wasserstein und Lazar 2016 beschreibt die problematische Situation in der Anwendung und wurde etwa auch in *Nature* thematisiert, vgl. Baker 2016.

18 Vgl. Gibbons u. a. 1994, S. 15ff.

*mode 2*¹⁹ beschriebene Öffnung der Forschung verdrängt, sondern existiert weiterhin parallel dazu. Jedoch birgt das Wachstum von Vielfalt und Verfügbarkeit der Informationsquellen auch Risiken: Klassische Gutachtersysteme, in denen die Prozesse für Dritte teils undurchsichtig sind, greifen hier nicht mehr.²⁰ Bewährte Strukturen werden so geschwächt und mit der möglicherweise gewonnenen Transparenz müssen sich erst wieder neue Verfahren etablieren. Neue Zeitschriften entstehen, die sich bewähren müssen. Dabei entstehen ebenso *fraud journals*, wissenschaftliche Zeitschriften, welche sich an etablierten Zeitschriften orientieren und so den Anschein von Seriosität wecken. Jedoch werden hier nicht die gleichen qualitativen Standards angesetzt und Publikationen zweifelhafter Qualität gegen Bezahlung akzeptiert. So entsteht ein Markt, welcher darauf ausgerichtet ist, Forschung, welche sich aufgrund von Schwächen nicht in die etablierten wissenschaftlichen Systeme einfügt, durch Vortäuschung zu legitimieren.²¹

Auf Grundlage von Überzeugungen betreffend Theorien und Projektförderungen bilden sich Wissenschaftsgemeinschaften. Forschende gruppieren sich um eine Theorie und versuchen diese auszubauen, die intendierten Anwendungen zu erweitern oder den Erklärungsgehalt zu stärken. Je nach Vorgehen der Agenten entwickeln sich dabei Theorienetze.

3. Doch die Agenten und ihre Handlungen sind nicht nur direkt von systeminhärenten Faktoren abhängig, auch äußere Systeme, wie die Ökonomie, Politik oder die Konkurrenz auf wissenschaftspolitischer Ebene haben Einfluss auf die Handlungen und damit auch auf die Theorieentwicklung. Die Verbreitung von Überzeugungen und Theorien ist ein maßgeblicher Faktor, um die eigene Position im System zu stärken. Die Reputation beeinflusst etwa die Bindung der Agenten an Wissenschaftsgemeinschaften. Oder um ein weiteres Beispiel zu nennen: Die Ausstattung der Forschung in Form von Forschungsgeldern kann möglicherweise Einfluss auf den Umfang und die Durchführung von Forschungshandlungen und Publikation nehmen. Dabei ist die Dynamik zwischen den angrenzenden Systemen, der Forschung, Politik und Wirtschaft in einem wissenschaftlichen Gesamtprozess in einer weiteren Abbildungsebene repräsentiert. Forschungsstrategien bestimmen den Fluss von Forschungsgeldern. Die Wissenschaftspolitik steht in einer Wechselbeziehung zur Ökonomie und Wählerschaft. Dabei müssen insbesondere Bedürfnisse dieser Gruppen,

19 Vgl. Gibbons u. a. 1994.

20 Herb 2015 beschreibt allgemein Open Science und auch Spezifika der Soziologie. In Hagenhoff u. a. 2007 werden konkrete Beispiele und Repositorien beschrieben. N. C. Taubert u. a. 2018 zeigt weitere aktuelle und kommende Paradigmen in der Wissenschaftskommunikation auf.

21 Dieses Thema wurde insbesondere in der Medizin und angrenzenden Forschungsbereichen bereits breit diskutiert. Hier gibt es neben der Reputation auch starke kommerzielle Einflussfaktoren. Dies macht etwa die Publikation von falschen oder qualitativ schlechten Studien besonders attraktiv. Beschrieben wird das etwa in C. R. Triggle und D. J. Triggle 2007, S. 39-51 oder Haug 2015, S. 3293f.

der Bevölkerung, berücksichtigt werden. Der gesellschaftliche Wunsch nach Innovationen von Konsumgütern oder militärischen Entwicklungen verlangt nach neuen Waren. Aber auch die Wissenschaft selbst kann als Ware verstanden und gehandelt werden.²² Kooperationsvereinbarungen in der internationalen Politik fördern Kooperationen oder Wettbewerbe zwischen bestimmten Nationen auf Grundlage nichtwissenschaftlicher Strategien, andere Bereiche werden aufgrund ethischer oder anderer Maßgaben reglementiert. Alle diese Prozesse haben wieder Einfluss auf die Strukturierung der Wissenschaftsgemeinschaften und somit auf die Theoriedynamik.

4. Die drei skizzierten Ebenen des Wissenschaftsprozesses – Theorieebene, sozialpsychologische und Gesellschaftsebene – fußen auf einer gemeinsamen Simulationsebene. Diese Ebene bildet den Rahmen, in dem die drei Betrachtungsebenen in der eigentlichen Simulation durch Simulationsmodelle abgebildet werden. Je nach Simulationsmodell wird der Fokus der Untersuchung auf Teilaspekte gelegt. Dabei gilt es auch die Brücken zwischen den Ebenen darzustellen. Theorien, welche auf der Theorieebene dargestellt werden, können von sozialen Agenten aufgestellt, diskutiert und verbreitet werden. Dies geschieht über Forschungsprozesse, die über soziale Handlungen definiert sind und in einem gesellschaftlichen Rahmen ablaufen. In einigen Beispielen wird die Welt, in der die Agenten handeln, dynamisch erzeugt und ändert sich über die Zeit. Diese Änderungen folgen Gesetzen, welche die Agenten erforschen können. Die Gesetze werden ebenfalls für jeden Simulationslauf erzeugt und die Komplexität der Welt kann zum Start einer Simulation definiert werden. Die Simulation unterschiedlicher Welten erlaubt es, Verbindungen zwischen der Definition der Welt und unterschiedlichen Paradigmen der Wissenschaft zu identifizieren und aufzuzeigen.

Der Aufbau dieser Arbeit orientiert sich an den skizzierten Ebenen. Begonnen wird mit der Beschreibung auf der Theorieebene und der Einführung des formalen Apparates im zweiten Kapitel. Im Kern steht die Definition eines hinreichend präzisen Theoriebegriffs und der Beschreibung der Rahmensysteme. Diese werden dann in Prolog überführt. Ziele dieses Kapitels sind es, die Abbildungen von Theorien in der Simulation zu beschreiben und die Notation, wie sie in dem folgenden Kapitel verwendet wird, einzuführen.

Das dritte Kapitel beschreibt die Sozialebene und reichert den formalen Begriffsapparat aus dem zweiten Kapitel um soziale Begriffe an. Diese bilden den Rahmen für die sozialen Multiagentensimulationen. Es werden die Handlungen und Prozesse der Agenten im Detail beschrieben, welche auf der Sozialebene abgebildet werden. Auch werden die Wechselbeziehungen zu den anderen Ebenen beschrieben. Dabei

22 Vgl. Balzer 2003, S. 87ff.

werden die sozialen Prozesse aus realen Beispielen der Forschungspraxis entnommen. Aufbauend auf den Handlungen und Prozessen, werden dann Modelle entwickelt, welche den Simulationen im fünften Kapitel als Grundlage dienen.

Im dem vierten Kapitel werden die sozialen Prozesse auf der Gesellschaftsebene, in einen größeren Kontext gebracht. In diesem Kapitel werden gesellschaftliche Faktoren besprochen und weitere Beispiele und Kennzahlen aus der Forschungspraxis angegeben. Dazu werden Zahlen vom Statistischem Bundesamt verwendet. Mit diesen Daten werden Simulationen konstruiert, welche auf existierendem Datenmaterial aus der Forschungspraxis beruhen.

Die eigentliche Implementierung der Simulation mit den Experimenten wird in dem fünften Kapitel gezeigt. Es werden drei Simulationen in ihrem Aufbau beschrieben: (1) Die erste Simulation konzentriert sich auf die Mikroebene: Es wird die Interaktion kleiner Gruppen von Agenten simuliert. Dabei liegt das Augenmerk auf einer Beobachtung der Wechselbeziehung zwischen der Theorie- und Sozialebene. Es werden unterschiedliche Phänomene mit Variationen der Parameter simuliert. So wird etwa der Effekt des bewussten Fälschens von Forschungsdaten und der möglichen Effekte auf die Theorieentwicklung und die Überzeugungsbasen untersucht. Dieses Beispiel wurde aufgrund von prominenten Fällen gewählt, in denen zum einen bewusst Daten zur Stärkung von Hypothesen gefälscht und zum anderen Arbeiten zur Erlangung von Titeln plagiiert wurden. (2) Die zweite Simulation betrachtet die Interaktion zwischen Wissenschaftsgemeinschaften im Hinblick auf die Verbreitung von Fakten. Dabei wird untersucht, welche Einflussfaktoren eine Evolution auf Theorie- und Sozialebene beeinflussen. (3) In der dritten Simulation werden dann Gesamtsysteme und deren Wechselwirkungen untersucht. Dabei wird die Projektforschung als Beispiel gewählt. Dieses Beispiel wurde aufgrund der sich verändernden Forschungsförderung durch einen Zuwachs von zeit- und projektgebundenen Fördermitteln gewählt.

Das sechste Kapitel analysiert die Ergebnisse von ausgewählten Simulationsläufen. Die Simulationen bieten eine Vielzahl von Parametern, welche über Testreihen variiert werden und erlauben verschiedenste Beobachtungen. Bei der Betrachtung dieser Veränderung der Parameter und den daraus entstehenden Effekten auf die simulierten Systeme zeigen sich tiefgreifende Abhängigkeiten. Zum Teil orientieren sich die Parameter an den vorgestellten realen Systemen, welche in den vorangehenden Kapiteln vorgestellt werden oder sind komplett konstruierte Systemzustände. Signifikante Simulationsläufe werden im Detail vorgestellt, die gewählten Konstanten und dynamischen Parameter beschrieben und in den Kontext gebracht. Dieses Kapitel bietet auch Perspektiven auf weitere oder vertiefte Anwendungsmöglichkeiten der entwickelten Methode und Software.

Doch bevor die Simulation und deren Hintergründe besprochen werden, gilt es die Methode der Simulation und insbesondere deren Anwendung in der Wissenschaftstheorie und verwandten Bereichen zu beleuchten.

1.2 Computersimulationen als Forschungsmethode

Die Grundlage zu Computerexperimenten und Simulationen in der Forschung lassen sich auf die Entwicklung der Differenzialrechnung durch Newton im 17. Jahrhundert zurückführen.²³ Experimentelle Aufbauten und präzise mathematische Beschreibungen haben systematische Experimente ermöglicht, welche sich technisch und methodisch weiterentwickelten. Aber auch in der Philosophie finden sich Grundlagen zum Verständnis von Simulationen. Gedankenexperimente, wie sie in der Philosophie eine lange Tradition haben und auch in der Wissenschaftstheorie ihre Anwendung finden, sind bereits als eine Art von Vorüberlegung einer Simulation zu verstehen. Gewöhnlich werden Gedankenexperimente genutzt, um auf einen Sachverhalt aufmerksam zu machen, eine Argumentation zu untermauern oder ein bestimmtes Argument zu verdeutlichen. Dabei sind Gedankenexperimente meist so gewählt, dass diese leicht vermittelbar und nachvollziehbar sind.²⁴ Jedoch gehen Simulationen über diesen Aspekt hinaus. Auch wenn die Simulationsmodelle mit entsprechendem Regelwerk nachvollziehbar sein sollten, erzeugt der Ablauf eine hohe Komplexität, welche eine manuelle Auswertung in einer annehmbaren Zeit mit vertretbarem Aufwand von Menschen verhindert. In Simulationen werden intendierte Prozesse durch kontrollierte und genau beschriebene Prozesse in definierten Umgebungen nachgebildet. Entsprechend Hartmann 1996 liegt einer Simulation ein *dynamisches Modell* zugrunde, welches sich mit Ausnahme einiger definierter statischer Komponenten über die Laufzeit entwickelt. Computersimulationen sind hierbei Simulationen, bei denen die Modelle und Systeme durch Algorithmen und Funktionen repräsentiert und die Simulationsschritte durch Berechnungen im Computer durchgeführt werden. Der Computer wird als physikalisches System mit Modellinformationen in Form von Programmen und Daten konfiguriert und durchläuft das Experiment anhand der physikalisch vorgegebenen Architektur und des Programmcodes. Beides weist in modernen Computern eine Vielzahl von Abstraktionsebenen auf, welche sich grob in die Hochsprachen mit den Simulationsprogrammen, den Verwaltungsebenen im Betriebssystem, dem erzeugten Maschinencode und der physikalischen Hardware unterteilen lassen. Der semantische Inhalt, etwa der Simulationsmodelle und Daten, ist durch die Kodierung vollständig von den mechanischen Prozessen getrennt.

²³ Vgl. Gramelsberger 2010, S. 204f.

²⁴ Klaus G. Troitzsch 1997, S. 46 oder Saam 2015.

In der Astrophysik werden solche Simulationen beispielsweise eingesetzt, um frühe Phasen in der Entstehung des Universums oder etwa dem Verhalten des Plasmas auf der Oberfläche der Sonne zu untersuchen oder etwa in der physikalischen Chemie, um Molekülstrukturen zu untersuchen. Zur Formulierung der Modelle können Beobachtungsdaten, Hypothesen, Theorien oder statistische Verfahren genutzt werden. Theoretische und empirische Forschung wird mittels der Verwendung von Simulationen um eine weitere Perspektive ergänzt. In der Wissenschaftstheorie bieten sich analoge Anwendungsmöglichkeiten an. Auch wenn der Vergleich zwischen der traditionell epistemisch geprägten Wissenschaftstheorie zu simulationsgetriebenen Forschungsgebieten zunächst künstlich wirkt, sind die möglichen Einsatzarten von simulationsgestützter Forschung durchaus vergleichbar. Sowohl die Simulation von modellierten Systemen, aufbauend auf wissenschaftsgeschichtlichen und soziologischen Modellen, als auch die vollständige Konstruktion von möglichen Welten bieten neue Perspektiven. Neben der philosophischen Begriffsarbeit, werden Modelle zur Theoriebildung, dem Forschungssystem und der Entwicklung der Wissenschaften auf Grundlagen von soziologischen und wissenschaftsgeschichtlichen Erkenntnissen gebildet. Gerade im Grenzbereich zur Wissenschaftssoziologie bieten Simulationen eine Verbindung zwischen theoretischen und empirischen Methoden.

In der Beschreibung gibt es eine Vielzahl von Methoden: Von statistischen Modellen, neuronalen Netzwerken über numerische Simulationen bis zu Multiagentensimulationen gibt es eine große Vielfalt für unterschiedliche Anwendungszwecke. Mit diesen Methoden werden die zu beobachtenden Systeme künstlich geschaffen und mit Werkzeugen untersucht. Der Zweck ist vielschichtig und spezifisch zur Disziplin und Fragestellung. Simulationen werden in der Forschung aber wohl am prominentesten eingesetzt, um ein System in nicht beobachtbaren Zuständen zu untersuchen und die Entwicklung von Modellen und Theorien²⁵ zu ermöglichen oder in der Optimierung von Prozessen und Verfahren (etwa die Aerodynamik von Flugzeugen). Simulationen können auch ethisch problematische Forschung ermöglichen, etwa um Tierversuche und im allgemeinen nicht vertretbaren Aufwänden zu vermeiden, Risiken in Medikamentenstudien zu minimieren oder auch Risiken in die Kontaminationen der Umwelt ausschließen.

Es gibt eine Vielzahl von Anwendungsbereichen, in denen Simulationen auf unterschiedliche Weise in der Forschung, aber natürlich auch in der Entwicklung und Industrie als Methode zur Untersuchung eingesetzt werden.²⁶

Zur Anwendung von Computersimulationen stellt sich insbesondere im Vergleich zu empirischen Experimenten die Frage, ob der epistemische Gehalt beider Methoden vergleichbar ist. Morrison 2009 sieht hier durchaus eine Vergleichbarkeit zwi-

25 Vgl. Hartmann 1996.

26 Vgl. Gilbert und Klaus G. Troitzsch 2005 oder Hofmann 2009.

schen Simulationen und Laborexperimenten. Er begründet dies durch die vergleichbare Rolle und Funktion von Modellen, welche sowohl als Grundlage für Simulationen, als auch andere Arten von Experimenten dienen. Die Gestaltung von Simulationsmodellen und Experimenten lässt sich zumindest bei einer abstrakten Betrachtung vergleichen und die Ergebnisse von Simulationsexperimenten könnten so selbst als Messung verstanden werden. Diesem Schluss widerspricht Giere 2009 in Bezug zu genuinen Simulationsmodellen. Computersimulationen, welche in keiner Verbindung zu empirischen Systemen stehen, unterliegen gerade darin, dass diese sich nicht an realen Systemen messen lassen können. Diese Eigenschaft unterscheidet Simulationen so maßgeblich von empirischen Experimenten.

1.3 Die Anwendung von Simulationen in der Wissenschaftstheorie

Konkrete Pionierwerke der Simulation in der Wissenschaftstheorie sind Thagard 1988 oder Langley u. a. 1987. Hier wurden durch Simulationen und entsprechenden theoretischen Überlegungen aufgezeigt wie sich Aspekte des Forschungsprozesses in einer Simulation abbilden lassen. Thagard beschreibt einen ansatztheoriebildende Prozesse zu simulieren. Theorien werden in Anlehnung zur strukturalistischen Wissenschaftstheorie als Datenstrukturen, bestehend aus Regeln, Konzepten und Daten beschrieben. Die Theoriefindung wird durch Problemlösungsstrategien dargestellt. Balzer 2015 beschreibt die Verbindung zwischen sozialen Simulationen und der strukturalistischen Wissenschaftstheorie, wie sie auch in den folgenden Kapiteln aufgegriffen wird. Im Kontext der künstlichen Gesellschaften wird dies auch in Gilbert und Klaus G. Troitzsch 2005 oder Balzer, Kurzawe und Manhart 2014 detaillierter beschrieben.

Soziale Simulationen von Prozessen in der Wissenschaft werden beispielsweise in O'Connor und Weatherall 2017, O'Connor und Bruner 2017, Zollman 2011, Ahrweiler 2011 gezeigt. Hier wurden in unterschiedlichen Ansätzen ebenfalls Aspekte zum Forschungsprozess modelliert und zum Teil auch implementiert und simuliert. Insbesondere legt Ahrweiler in ihrem Positionspapier einige Grundsteine, welche in dieser Arbeit berücksichtigt werden: Das Konzept von Agenten, welche sich zu Gemeinschaften um Theorien bilden, wird hier bereits angedacht und geht in dem Konzept in den Aspekten der Theoriepräsentation deutlich weiter. Natürlichsprachliche Sätze sollen analysiert und so die Semantik stärker erfasst werden. Ahrweiler beschreibt ebenfalls Prozessebenen, doch betont hier die sozialen Prozesse nur in Bezug auf Gruppen und benennt keine gesellschaftlichen oder weiteren Dimensionen. Pluchino, Biondo und Rapisarda 2018 untersuchen die Einflüsse von Talent und Glück auf den Erfolg bzw. Wohlstand. In Pluchino, Burgio u. a. 2019 wird die Methode auf den spezifischen Erfolg für eine Karriere in der physikalischen Forschung adaptiert und betrachtet in dem Kontext ebenfalls den Einfluss der Interdisziplina-

rität auf den Karriereerfolg. Das Modell basiert auf der Analyse statistischer Daten. Agenten agieren entsprechend des statistischen Modells in einem zweidimensionalen Raum (NetLogo). Insbesondere die Arbeiten von O'Connor beschäftigen sich mit der Modellierung von Wissenschaftsgemeinschaften und zeigen so, auf welche Weise Simulationen auch in der Wissenschaftstheorie ihre Anwendung finden können. Jedoch verwendet O'Connor in Abgrenzung zu den hier gezeigten Simulationen lokale Modelle, welche sich auf ausgewählte Teilbereiche des Forschungsprozesses beschränken. Auch in der Methode unterscheiden sich die Simulationen. O'Connor verwendet oft spieltheoretische Ansätze und bildet so nicht die direkten Interaktionen zwischen einzelnen Agenten ab und abstrahiert deutlicher bei den Handlungen, Motivationen und Prozessen. O'Connor geht auch nicht auf die spezifisch gewählte Methode ein und beschränkt sich auf die Modellbeschreibung. Wajnberg u. a. 2004 beschreiben einen strukturalistischen Ansatz für ein Discovery System zum Aufbau von Theorienetzen. Dies ähnelt Thagards Ansatz, jedoch mit einem deutlich formaleren Fokus auf die strukturalistische Repräsentation und den theoretischen Unterbau. Der Theoriebildungsprozess wird in Ansätzen modelliert. Soziale und gesellschaftliche Zusammenhänge werden jedoch nicht betrachtet und der Forschungsprozess, ebenfalls ähnlich zu Thagard, auf eine Problemlösungsstrategie reduziert. Dies vernachlässigt, wie zu zeigen ist, eine weitreichende Dynamik der Forschung, welche nicht nur aus rationalen Ableitungen besteht. Klaus G. Troitzsch 2017 zeigt, wie Schellings Segregation-Modell durch eine strukturalistische Rekonstruktion klarer dargestellt werden kann und wie der strukturalistische Ansatz bei der Formulierung von Simulationsmodellen beiträgt. Die Arbeit Hofmann 2009 modelliert Entscheidungsprozesse und implementiert diese in ein Simulationsmodell in der SMASS Umgebung (Prolog). Dazu wird das *Belief-Desire-Intention Model* verwendet. Agenten verfolgen individuelle Ziele und verhandeln diese, um eine gemeinsame Aktivität durchzuführen. Ein Ansatz für die Beschreibung von Gruppenentscheidungen auf Grundlage von Emotionen wird von Thagard und Kroon 2006 beschrieben und in der Simulationsumgebung *HOTCO 3* simuliert. Doch auch in anderen Teilbereichen der Philosophie existieren Beispiele und Anwendungen, wie etwa die Modellierung und Simulation von Abstimmungen am Beispiel des EU-Ministerrats in Beisbart und Hartmann 2011. Diese Auswahl von Arbeiten soll einen Überblick über die Methoden geben und erhebt keinesfalls den Anspruch der Vollständigkeit.

Trotz der großen Verbreitung von Simulationen in natur- und sozialwissenschaftlichen Disziplinen und den vielfältigen Anwendungsbereichen ist die Verwendung von Computersimulationen als Methodik in den Geisteswissenschaften bisher wenig thematisiert und liegt auch nicht im Forschungsinteresse der meisten geisteswissenschaftlichen Disziplinen. Doch zeichnet sich in den letzten Jahren auch in den Geisteswissenschaften ein Trend zur Anwendung von computergestützten For-

schungsmethoden²⁷ ab: Mit den digitalen Geisteswissenschaften (oder auch *Digital Humanities*) ist eine Strömung entstanden, welche computergestützte Forschungsmethoden zur Beantwortung von geisteswissenschaftlichen Fragestellungen etabliert. Dabei sind Computersimulationen jedoch bisher noch vergleichsweise unterrepräsentiert. In der Soziologie findet die Anwendung von *social simulation* durchaus Einsatz.²⁸

Mit der Evolution der Computertechnologie und deren wachsender Verfügbarkeit sind sowohl die Anwendungsbereiche als auch die Verwendung in technischen Gebieten stetig gewachsen.²⁹ Durch jede Technikgeneration wächst das Spektrum der Möglichkeiten technologisch anspruchsvollere Simulationen zu gestalten. Fortschritte in der Forschung bringen neue Fragestellungen auf, welche ebenfalls die Entwicklung der Simulationstechnik beflügelt, wie auch die Möglichkeiten, welche durch Computersimulationen gegeben sind, die Forschung mit neuen Methoden bereichern. Dabei birgt die Verwendung von Simulationen in der Forschung nicht mehr Herausforderungen als die eigentliche Modellbildung – oder wie es Frigg und Reiss herausstellen: „[...] the epistemological problems presented to us by simulations have much in common with the ones that arise in connection with models [...]“³⁰ Die Herausforderungen bei der Gestaltung einer Simulation sind dagegen mannigfaltig: Basieren Simulationen auf komplexen mathematischen Modellen, erfordern diese große Rechenkraft. Klimasimulationen sind hier ein gutes Beispiel: Aus einer Vielzahl von komplexen Daten und teils unterschiedlichen lokalen Klimamodellen wird eine Gesamtsimulation erstellt. Aber auch der Schienen- oder Flugverkehr wird aufgrund von simulierten Streckenauslastungen und Auslastungstests geplant. Maschinen, Gebäude, aber auch Computerprogramme werden simuliert und auf ihre Belastbarkeit geprüft, um Materialschwächen oder Probleme in der Programmierung ausfindig zu machen. Die Marktwirtschaft nutzt Simulationen, um Prognosen und Effekte von Ereignissen möglichst zielsicher zu deuten. Auch in der Unterhaltungsindustrie werden verhältnismäßig komplexe Computersimulationen eingesetzt. Im Bereich der Computerspiele simulieren Wirtschaftssimulationen etwa ökonomische Systeme oder Flugsimulatoren das Verhalten von Flugzeugen. Doch unterscheidet sich hier stark die Zielsetzung von denen, welche in der Forschung oder der Industrie ihre Anwendung finden. Während in manchen Forschungsbereichen, etwa der Simulation zum Verhalten von Plasma auf der Sonne,³¹ die Effizienz und Reduktion der Komplexität maßgeblich für die Durchführung der Untersuchung ist, liegt

27 Hier hat sich auch der Begriff *Digitale Methoden* etabliert.

28 Takahashi, Sallach und Rouchier 2007 gibt hier Beispiele.

29 Hofmann 2009 zeigt die Verbreitung an der wachsenden Erwähnung von Simulationen in der Projektdatenbank GEPRIS der DFG und von EU-Rahmenprogrammen.

30 Frigg und Reiss 2009, S. 14.

31 Vgl. beispielsweise Zacharias, Peter und Bingert 2011.

der Fokus in anderen Forschungsbereichen stärker in der Gestaltung, Konstruktion und insbesondere Kontextualisierung der Simulation. In Klaus G. Troitzsch 2019 Beschreibung zur Konstruktion von Simulationen wäre die Optimierung wohl am ehesten in dem Bereich des Prozesses der Implementierung und den *Parameter Settings* anzusiedeln, wohingegen das Design und die Kontextualisierung auch den Aspekten der *Formalization* und *Instantiation* zuzuordnen sind. In geisteswissenschaftlichen Kontexten wäre hier noch ein weiterer Zugang denkbar: Die explorative Annäherung. Mentale Modelle stehen hier nicht zwingend in einem Zusammenhang zu einem realen Zielsystem und wären so kein empirischer Ansatz. Die Simulationsexperimente variieren in den Anforderungen und wurden nicht für Effizienz oder einen geringen Ressourcenverbrauch optimiert. Für die Berechnungen wurden zwei Systeme verwendet: Ein Desktopsystem (Intel i7-6700K, 16 Gigabyte RAM, Linux (Ubuntu 19.04), SWI-Prolog 8.2.X, Python 3.X) und ein Hochperformance-System (32 Kerne, 64 Gigabyte RAM, Netzspeicher, Ubuntu 18.04). Auch wurden einige Experimente erfolgreich unter Microsoft Windows 10 und Apple macOS 10.13 getestet. Um eine bessere Lesbarkeit und Modularität der Beispiele zu gewährleisten, wurde wo möglich auf stärkere Optimierungen verzichtet. Entsprechend sind viele Module wohlwissend in Hinblick auf den Ressourceneinsatz ineffizient.

Zusammenfassend lassen sich die vorgestellten Experimente folgend einordnen: Die Experimente lassen sich als wissenschaftliche Simulationen von selbstorganisierten Systemen verstehen. Analog zu den Simulationsmethoden, wie sie etwa in Hofmann 2009 zu den Untersuchungen zur Dynamik sozialer Praktiken aufgezeigt und eingesetzt werden, dienen die hier entwickelten Simulationen der Untersuchung des Forschungsprozesses. Das Kernkonzept dieser Methode besteht darin, Forschungsprozesse für Simulationsexperimente zu modellieren, diese in Simulationsreihen mit jeweils veränderten Parametern zu durchlaufen und die Ergebnisse zu untersuchen. Dabei können die gesetzten Parameter der Experimente sowohl konstruiert als auch an der Realität angelehnt sein. Ein weiterer Bestandteil, neben der Beschreibung, Durchführung und Untersuchung der Experimente, stellt die Modellierung dar. So lässt sich etwa die Verbreitung bestimmter Fakten oder die Entwicklung gewählter Agenten über die Zeit im Zusammenhang zu bestimmten Konfigurationsänderungen beobachten.

1.4 Die Theoriefindung in simulierten Umgebungen

Die Agenten der Simulation führen Handlungen durch. *Lernen, Entdecken, Beschreiben, Erklären, Ableiten* sind alles Handlungen in der Forschung und auch zentrale Begriffe in der Entwicklung von *Künstlichen Intelligenzen*³².

In der Forschung beschäftigen sich Bereiche der Informatik, Mathematik, Philosophie und dem Querschnittsfeld der Neurowissenschaften damit, dem Computer menschenähnliche kognitive Fähigkeiten zu verleihen. Es werden Methoden entwickelt, Computern Fähigkeiten zum selbstständigen Handeln durch erlerntes Wissen und somit der Fähigkeit zum Erfassen und Handeln von und in neuen Situationen zu verleihen. Die Forschung an solchen künstlichen Intelligenzen hat gerade in den letzten Jahren eine erhebliche Entwicklung erfahren, wenn sie auch dem Ziel von vollständig autonom denkenden und handelnden Programmen noch weit entfernt zu sein scheint.

Der Begriff *Künstliche Intelligenz* beschreibt einen ganzen Kanon an Methoden, Fragestellungen und Paradigmen. Er umfasst unter anderem die Themenschwerpunkte des maschinellen Lernens, autonomen Handelns oder die Mustererkennung.

In der Literatur gibt es viele unterschiedliche Definitionen und Verwendungen zu zentralen Begriffen, wie etwa der Intelligenz, des Lernens oder der Kreativität. Gerne wird darauf verwiesen, dass es im Kern der Begrifflichkeiten darum geht, den Computer zu befähigen, seine Umwelt zu erfahren, eigenständig daraus zu lernen und darin zu agieren. Somit ergibt sich die verbindende Vorstellung eines autonom agierenden Systems. Paul Thagard beschreibt dies knapp und pragmatisch mit den Worten: „Artificial intelligence (AI) is the branch of computer science concerned with getting computers to perform intelligent tasks“³³ und hatte im gleichen Kontext bereits 1988 beschrieben wie die KI-Forschung und die Wissenschaftstheorie von gemeinsamen Perspektiven profitieren können.³⁴ Diese Beschreibung mag für diesen Kontext genügen, da es nicht Ziel der Simulationen ist, in diesem Sinne *Intelligenz* zu schaffen, sondern die Modellierung von Forschungsprozessen zu zeigen.

Die aktuelle Forschung befasst sich insbesondere mit der Optimierung und Anwendung von künstlichen neuronalen Netzen. Zwischen einer Menge von Ausgangszuständen und potenziellen Zielzuständen wird ein Netz von künstlichen Neuronen erzeugt, welche logische Operatoren abbilden. Je nach Anwendungsfall kann das Netz auf bestimmte Aufgaben trainiert werden und so die Gewichtung von Übergängen zwischen den Neuronen verändern. Die Beschaffenheit der Neuronen und

32 Wenn auch ich diesen Begriff für nicht sonderlich gut gewählt halte, ist es ein fester Term und wird als solcher benannt.

33 Thagard 1988, S. 2.

34 Vgl. die entsprechenden Thesen in ebd., S. 4ff. Eine präzise Antwort auf die Frage wie sich Intelligenz genau definiert, bleibt er jedoch auch in letzter Instanz schuldig.

Netze ist vom gewählten Modell abhängig. Ein einfaches Beispiel ist ein Neuron n_i , welches eine Menge an Eingängen e_n , ein Bias b und einen Ausgang besitzt. Die Summe der Eingangssignale und des Bias wird dann einer Aktivierungsfunktion übergeben und so entschieden, ob das Neuron selbst feuert. Ein einfaches Beispiel für eine solche Aktivierungsfunktion ist die Sigmoid Funktion: $\frac{e^x}{e^x+1}$. Diese bildet die Summe der Eingangswerte $\Sigma(e_n * w_n)$ auf ein Intervall $[0, 1] = \{x \in \mathbb{R} | 0 \leq x \leq 1\}$ ab. Durch ein Training werden die Gewichtungen w der Eingangssignale und des Bias bestimmt. Der Ausgangswert a des Neurons n_i wird somit durch $\frac{1}{1+e^{-\Sigma(e_n * w_n)}}$ bestimmt. Ausgangszustände können etwa Bildpunkte eines Kamerabildes oder Wörter aus Texten sein und die Netze dafür verwendet werden, Personen auf Bildern oder Erzählstilen in Romanen zu identifizieren. Um diese Tätigkeiten auszuführen, müssen die Netze zunächst anhand von größeren Datenmengen trainiert werden. Das trainierte Netz kann danach verwendet werden, um etwa Segmentierungen durchzuführen und beispielsweise spezifische Objekte in Bildern erkennen. In der Praxis werden diese Netzwerke durch Vektoren beschrieben. Die Eingangszustände der Inputvektoren werden als *features* bezeichnet. Dies sind die beobachtbaren Entitäten und werden als Informationen im Netzwerk verarbeitet.

Evolutionäre Algorithmen bieten einen alternativen Ansatz. Das Konzept ist an die Bestenauslese angelehnt. Aus einer Menge von Variationen eines Algorithmus wird der beste gewählt. Auf dessen Grundlage werden weitere Varianten erzeugt, wieder angepasst und anschließend evaluiert. Dieser Prozess ähnelt dem zeitlichen Ablauf einer Theorieentwicklung. Hypothesen werden aufgestellt und über die Zeit verfeinert. Einzelne Hypothesen sind zwar ebenfalls spezifisch, doch erlauben sie, anders als neurale Netze, eine klare Abgrenzung.

Diese aktuell verbreiteten Verfahren Simulationsmodelle ausfolgenden Gründen ungeeignet: Trainierte Netzwerke sind problemspezifisch: Sie werden auf eine definierte Menge an Daten trainiert und können so spezifische Zustände erkennen und in den Parametern arbeiten. Dieses Vorgehen bietet sich etwa bei der Klassifikation von Daten an. Somit sind Vorhersagen modellspezifisch. Insbesondere wird hier der Begriff der *Intelligenz* sehr eng gefasst und insbesondere in der Anwendung von neuronalen Netzen stark auf das Erlernen von Handlungsmustern eingeschränkt. Darüber hinaus sind diese Netze der Betrachtung verschlossen. Statt konkreter Konzepte bilden die Netze abstrakte Muster aus. Entscheidungen sind von außen schwer nachvollziehbar und somit nicht für eine Erklärung im Sinne des *reasoning* nutzbar.

Thagard 1988 beschreibt eine andere Herangehensweise: Phänomene werden beobachtet und anhand von Ableitungen aus bestehendem Wissen erklärt. Dazu wird eine Menge von Konzepten und Regeln vorgegeben und eine Frage definiert, welche es durch das Wissen zu beantworten gilt. Über Ableitungen wird dann versucht die Fragestellung anhand bekannter Konzepte und Regeln zu beantworten. Thagard nennt hierzu ein Beispiel, in dem Schallwellen anhand des Konzepts der Wellenaus-

breitung beschrieben werden.³⁵ Für die Definition der Ableitung stützt sich Thagard auf die von Charles Sanders Peirce geprägte Definition der Abduktion.³⁶ Eine Deduktion würde nur zeigen, dass das beobachtete Phänomen direkt durch eine Hypothese der Wissensbasis erklärt werden könne. Dazu müsse das Phänomen aber bereits bekannt sein. Beispielsweise würde ein Phänomen q beobachtet und es ist bekannt, dass $p \rightarrow q$. So würde hypothetisch p als Ursache für q gelten, wenn genau diese Beziehung definiert ist. Eine Hypothese über neue Phänomene als Ergänzung der Wissensbasis würde so allerdings nicht entstehen. Durch die Abduktion wäre aber genau dies möglich. Wenn etwa eine Erklärung für das Phänomen $G(a)$ gesucht wird und bekannt ist, dass allgemein gilt $F(x) \rightarrow G(x)$, so könnte etwa $F(a)$ eine Erklärung und somit eine mögliche Hypothese sein.³⁷

Insbesondere unter der Annahme von konsistentem Wissen, also zueinander widerspruchsfreien Sätzen in der Wissensbasis, lässt sich diese Idee vergleichsweise einfach in Prolog realisieren. Die Herausforderungen liegen bei der Beobachtung und Definition der Problembeschreibung oder wissenschaftlichen Fragestellung,³⁸ bei dem Umgang mit komplexen und möglicherweise nicht mehr konsistenten Wissensbasen, der Approximation von Daten und insbesondere in der Aufstellung neuer Hypothesen. Die Ordnung, welche durch die Verwendung der starken Taxonomie vorgegeben wird, ist problematisch. Denn so wird eine strenge Ordnung und Konsistenz verlangt, welche sich kaum in einer Welt mit unterschiedlichen Perspektiven, Grenzen der Wahrnehmung und unvollständigen Sprachen realisieren lässt.

Die hier angewandte Idee in der Simulation schließt dort an: Agenten nehmen eine Welt voller Ereignisse mit unterschiedlichen Typen wahr. Ihr Ziel ist es, Muster zu finden und die Ereignisse in ihren zeitlichen und räumlichen Kontext zu setzen. Konzepte können als Beziehungen der Phänomene im Raum zueinander verstanden werden, in der Simulation als *shape* bezeichnet. So können Theorien aufgebaut werden, welche auch Vorhersagen ermöglichen. Die so identifizierten Muster bedürfen keines Training eines Netzwerks, sind allerdings in ihrer Anwendbarkeit deutlich eingeschränkter.

1.5 Prolog als Simulationsumgebung

Für soziale Simulationen werden häufig spezialisierte Entwicklungsumgebungen für Simulationen, wie etwa NetLogo³⁹ oder MASON⁴⁰ verwendet. Auch werden Spra-

35 Ein Simulationsdurchlauf dafür wird in ebd., S. 209-224 angegeben.

36 Vgl. ebd., S. 52.

37 Vgl. ebd., S. 55.

38 Dies beschreibt Thagard selbst in ebd., S. 175f.

39 Siehe <https://ccl.northwestern.edu/netlogo/> (zuletzt aufgerufen am 25.07.2020).

40 Siehe <https://cs.gmu.edu/~eclab/projects/mason/> (zuletzt aufgerufen am 25.07.2020).

chen wie Python⁴¹, JavaScript, Java⁴², Processing⁴³ oder Umgebungen wie AnyLogic⁴⁴ für die Gestaltung von Simulationen verwendet. Gerade NetLogo hat eine große Nutzerschaft im Kontext der sozialen Simulationen und bietet durch die reduzierte Syntax, integrierten graphischen Werkzeugen und die vorgegebene Struktur (Patches, Turtles, Ticks) einen sehr praxisorientierten Zugang. Jedoch kann genau diese feste Struktur eine starke Anpassung und Abstraktion der intendierten Simulation notwendig machen. Die hier vorgestellten Konzepte sind in Prolog implementiert. Theorien werden mittels (Daten-)Strukturen und Modellen repräsentiert und ausgewertet. Modelle und Datenstrukturen lassen sich mit wenig Aufwand als Prädikate, Listen und Ausdrücken in Prolog beschreiben. Prolog bietet durch das Backtracking eine einfache Möglichkeit Ableitungen zu verwenden. Dies ist etwas für *reasoning* und viele weitere Prozesse vorteilhaft und in anderen Umgebungen nur mit größerem Aufwand möglich. Es lassen sich formal beschriebene Theorien im Idealfall direkt über die Definition eines entsprechend strukturierten Prädikats übertragen. Ein Beispiel dafür findet sich in Abschnitt 2.2.1. Entsprechend bietet sich Prolog für diese Art von Simulationen an, auch wenn es sich um eine relativ exotische Programmiersprache handeln mag.⁴⁵ Ein erwähnenswerter Nachteil zeigt sich bei der Visualisierung und Datenauswertung. SWI-Prolog bietet zwar eine Vielzahl von Schnittstellen (etwa die Datenverwaltung mittels ODBC, RDF-Unterstützung, Grafikausgabe über XPCE, Schnittstellen mittels Webservern über pengine und swish, Module für die R Integration, C Integration) und die Möglichkeit Auswertungen und Berechnungen in Prolog durchzuführen, doch bedürfen all diese Möglichkeiten weitere Arbeitsschritte und die Anbindung an externe Bibliotheken. Hierzu wurde für die Beispiele Python gewählt, da hier eine Vielzahl an Werkzeugen für die Datenauswertung und Visualisierung vorhanden sind.

41 Siehe <https://www.python.org/> (zuletzt aufgerufen am 25.07.2020).

42 Siehe <https://openjdk.java.net/> (zuletzt aufgerufen am 25.07.2020).

43 Siehe <https://processing.org/> (zuletzt aufgerufen am 25.07.2020).

44 Siehe <https://www.anylogic.de/> (zuletzt aufgerufen am 25.07.2020).

45 Vgl. Balzer und Manhart 2011.

2 Die Entwicklung und Darstellung wissenschaftlicher Theorien

Wissenschaftliche Theorien sind ein zentraler Bestandteil der Forschung und entsprechend auch ein wichtiger Aspekt der Simulation. Die Wissenschaftstheorie widmet sich der Struktur der Wissenschaft und speziell auch den Theorien.¹ Der Theoriebegriff ist jedoch kein über alle Disziplinen gesetzter Term, welcher gleichbedeutend in allen Bereichen der Forschung verwendet wird. Ebenso wie sich die vielen Disziplinen und Teildisziplinen der Wissenschaft in Gegenstandsbereichen und den jeweiligen Methoden unterscheiden, variiert auch die Art, Gestalt und Bedeutung von wissenschaftlichen Theorien. Und auch aus der Metabetrachtung der Wissenschaftstheorie und Wissenschaftsforschung gibt es eine Vielfalt von Positionen, Begriffen und dazugehörigen Begriffsbedeutungen.

Entsprechend ist zu klären, was genau im Folgenden und speziell in der Simulation unter dem Term *Theorie* verstanden und abgebildet wird. Nach Balzer, Moulines und Sneed 1987 lässt sich eine Theorie *T* im Kern aus einer Menge von realen Systemen, auf denen die Theorie ihre Anwendung findet und einer Menge von Modellen, welche diese Systeme abbilden, beschreiben.

Erkenntnisse entspringen in der Forschungspraxis beispielsweise aus empirischen Beobachtungen von Phänomenen in diesen Systemen. Die Beobachtungen werden nach wiederholten Mustern, Korrelationen, Ähnlichkeiten, oder etwas allgemeiner gefasst, Sinnzusammenhängen untersucht und in den Kontext bisherigen Wissens gesetzt. Erklärungen für die beobachteten Phänomene werden als Hypothese in einer normierten Fachsprache formuliert und somit die Struktur für Modelle beschrieben. Sofern sich diese über die Zeit durch kritische Prüfungen oder Anwendungen bewährt und keine Einwände gegen eine Allgemeingültigkeit entstehen, wird eine solche Hypothese schließlich als geltende Theorie gesehen. Kuhn spricht in diesem Bild von einem Paradigma.²

Die Datenerhebung ist in der empirischen Forschung ein zentraler Punkt: Daten werden in dem Prozess der Beobachtung anhand normierter und beschriebener Methoden erhoben, welche aber auch selbst Gegenstand anderweitiger Theorien sind. Die Erhebung der Daten erfolgt etwa in Experimenten, Simulationen, Metaanalysen, Studien oder direkten Beobachtungen am Objekt. Neben der Normierung der Daten bedürfen auch die Methoden einer Anerkennung der jeweiligen Gemeinschaft, um die daraus entsprungene Daten nicht durch vermeidbare methodische Fehler zu belasten. Da die Beschreibung von Phänomenen und des Beobachtungs-

1 Vgl. Balzer und Brendel 2018, S. 17f.

2 Vgl. Kuhn 1962.

raums ebenfalls auf einem theoretischen Unterbau aufbaut, ordnet sich jede ergänzende Theorie selbst in ein Netz von Theorien und Begrifflichkeiten ein. Die Vernetzung zu anderen Theorien und theorieunabhängigen Begriffen³ ist dabei eine weitere Ebene, welche über den jeweiligen Einzeltheorien liegt. Die Abgrenzung zwischen T-theoretischen Begriffen, welche als Teil der Theorie formuliert werden und solchen, die von der Theorie unabhängig sind, birgt weitere Implikationen. Dieses Muster gilt insbesondere für Bereiche der angewandten Naturwissenschaften, in denen empirische Beobachtungen auf Grundlage bestehender Theorien aufbauen oder dort weiter ansetzen.

Doch daneben gibt es noch viele weitere Forschungsparadigmen, wie etwa in theoriegetriebenen⁴ Disziplinen oder welchen, bei denen der Begriff der Empirie anders zu deuten ist. In der Mathematik unterscheidet sich der Theoriebegriff naturgemäß von empirischen Disziplinen.⁵

2.1 Perspektiven auf Methoden und Theorien

Formale Methoden finden überall dort Anwendung, wo diese Form der Präzision im Ausdruck oder die Methoden zur klaren Ableitung notwendig sind. In der Chemie, Biologie, Soziologie, Psychologie, Medizin und vielen weiteren Disziplinen finden etwa die Statistik oder die Modelltheorie ihre Anwendung. Die Physik nutzt verschiedenste mathematische Werkzeuge zur Beschreibung physikalischer Vorgänge, zur präzisen Beschreibung zum Aufbau von Modellen und für Vorhersagen. Eine Ausgangsbasis kann dabei eine empirische Beobachtung sein, welche dann normiert wird. Rückschlüsse von theoretischen Vorhersagen auf reale Systeme erfolgen in der Experimentalphysik. Beispielsweise hat das *ATLAS Experiment* am *Large Hadron Collider* 2012 einen experimentellen Nachweis des 1964 von Peter Higgs postulierten Higgs-Teilchens erbracht und somit das in der Physik als *Standardmodell* bezeichnete Theoriekonstrukt in einem Fundament gestärkt.⁶ Es wurde ein experimenteller Nachweis über die Existenz des Teilchens erbracht, welches bis dahin nur theoretisch postuliert wurde. Erst durch eine Vielzahl technischer Entwicklungen war es möglich, einen entsprechenden Teilchenbeschleuniger mit Messinstrumenten zu bauen und so die empirische Beobachtung zu ermöglichen. Die von Albert Einstein im Jahr 1916 postulierten Gravitationswellen wurden 2016⁷ in einer Beobachtung nachgewiesen.

3 Vgl. Balzer, Moulines und Sneed 1987, S. 57f.

4 Hier im Sinne von Gegenpol zum empirischen Forschen.

5 Die Ideenlehre Platons ist ein entsprechender klassischer Diskurs der Philosophie.

6 Vgl. Collaboration 2012.

7 Vgl. Abbott u. a. 2016.

Beide Beispiele zeigen zum einen das Zusammenspiel von Vorhersagen und der empirischen Forschung. In diesem Zusammenspiel werden mathematische Methoden und Ausdrücke genutzt, welche nicht selbst empirischen Beobachtungen entspringen. Daraus ergibt sich ein Zusammenspiel zwischen Kausalbeschreibungen der experimentellen Physik und mathematischen Modellen.⁸

Während etwa die Nullhypothese oder Alternativhypothese in empirischen Theorien zum Nachweis ihrer Gültigkeit oder eben statistischen Unabhängigkeit genutzt wird, verlangen mathematische Theorien nach klaren Beweisen in definierten formalen Systemen und Regeln. Geisteswissenschaftliche Forschung ist dagegen selten formal und unterscheidet sich entsprechend methodisch stark von Disziplinen, welche mithilfe von formalen Systemen arbeiten und argumentieren. Anstelle solcher Systeme, empirisch beobachtbarer oder ableitbarer Phänomene und Theorien, treten Diskurse und Interpretationen. In der Vielzahl geisteswissenschaftlicher Disziplinen existiert ein breiter Methodenbaum. Dieser unterscheidet sich etwa im Stil der Argumentationen und der Art des geführten Diskurses. Um ein paar Beispiele zu nennen: Die Geschichtswissenschaft führt einen Diskurs, aufbauend auf Datengrundlagen oder im Falle der neuesten Geschichte auch auf Beobachtungen zur Geschichtsschreibung.⁹ Dabei unterscheidet sich naturgemäß jedoch der Wahrheitsanspruch und die Deutung der Hypothesen von denen in formalen Disziplinen.¹⁰ Im methodischen Zentrum vieler geisteswissenschaftlicher Disziplinen steht die Hermeneutik. Forschungsgegenstände werden im sozialen und historischen Zusammenhang kontextualisiert und diskutiert.¹¹ In den Musikwissenschaften dagegen finden sich etwa sowohl formale Systeme als auch historische und kulturelle Untersuchungen mit entsprechenden Methoden. Klänge werden physikalisch analysiert, Harmoniesysteme aufgestellt und klassifiziert, aber auch die historische Entwicklung thematisiert. Literaturwissenschaftler*innen sprechen in Bezug zu Theorien, etwa von *der Theorie des Romans im 17. Jahrhundert* oder einer spezifischen Literaturgattung. Damit wird eine Menge von Eigenschaften, Beispielwerken oder spezifischen Autoren zur Abgrenzung eines Korpus gegenüber anderen Publikationen definiert. So lassen sich durch die sprachlichen Beschreibungen zwar ebenfalls Modellmengen definieren, doch enthalten die Hypothesen subjektive Beschreibungen.

Um die Wissenschaft als System in einer Simulation abzubilden, wäre es natürlich wünschenswert, ein möglichst breites Spektrum der vielen Nuancen aufzugreifen. Doch ist dies bei der unüberschaubaren Zahl der Forschungsbereiche und entsprechenden Methoden leider nicht möglich. In den Simulationsexperimenten wird die

8 Vgl. Lange 2013 S. 486ff.

9 Genau diese empirischen *Fragmente* sind allerdings etwa für Hempel 1942 Belege für den empirischen Charakter der Geschichtsforschung.

10 Vgl. Rösen 2013, S. 55f.

11 Vgl. Doublet 2003, S. 62ff.

empirische Beobachtung von Ereignissen und darauf aufbauenden Ableitung als Beispiel betrachtet, aber entsprechend in einem sehr engen Rahmen. Die Ereignisse sind Teil eines statischen zweidimensionalen endlichen Ereignisraums $\langle x, y \rangle$ mit diskreter Zeit $t_i \in N$ und Ereignistypen als weitere Dimensionen. Die beobachtbaren Ereignisse sind jeweils genau einem Typ zugeordnet und finden zu genau einem Zeitpunkt statt. Die Menge der Typen ist endlich und konstant, die Zeitpunkte diskret und ebenfalls durch die Simulationslaufzeit begrenzt. Die Methoden, mit denen die Agenten die Ereignisse untersuchen, sind ebenso eng definiert.

2.1.1 Wissenschaftstheoretische Beschreibung von Theorien

In der Wissenschaftstheorie haben sich unterschiedliche Perspektiven auf Theorien und den Forschungsprozess entwickelt. Moulines 2008 zeichnet hier eine Entwicklung größerer wissenschaftstheoretischer Perspektiven in fünf Phasen, beginnend um 1885. Im Kontext der Simulation sind insbesondere zwei Perspektiven relevant: Der in Kuhn 1962 formulierte Forschungsprozess, welcher eine Kontextualisierung von Theorie zu sozialen Handlungen herstellt und der wissenschaftstheoretische Strukturalismus, welcher eine umfassende Repräsentation von Theorien und Netzen erlaubt. Kuhn beschreibt ein Konzept zur Entwicklung von Paradigmen und der Strukturalismus ergänzt die formale Darstellung und bietet eine begriffliche Präzisierung.

Die Wissenschaft durchläuft einen Kreislauf, in dem Paradigmen so lange die *normal science* bestimmen, bis diese aufgrund von Anomalien nicht mehr zu halten sind und diese entweder angepasst werden oder durch eine Revolution Alternativen an ihre Stelle treten. Dies alles geschieht im Zusammenwirken von Wissenschaftsgemeinschaften. Kuhns Beschreibung erlaubt es, die Forschung somit als Prozess zu verstehen. Forschende interagieren miteinander und formen Theorien. Der Stellenwert dieser Theorien ändert sich über die Zeit. Aus einer Vielzahl von Erklärungsmodellen für ein Phänomen bildet sich ein Paradigma heraus. Weitere Forschung und neuere Erkenntnisse lassen sich möglicherweise mithilfe des geltenden Paradigmas erklären und verlangen nach einer Hilfhypothese oder einem gänzlich neuen Paradigma. Jedoch bleibt Kuhn bei den Details im Vagen. Sowohl der in Kuhns Arbeit zentrale Begriff des Paradigmas wird in unterschiedlicher Weise verwendet als auch viele der weiteren Begriffe.¹² Einen genaueren Einblick in die eigentliche Struktur der Theorien erlaubt der wissenschaftstheoretische Strukturalismus: Die Theorie steht im Fokus und die soziale Struktur lässt sich durch die Verbindung von Theorien, intendierten Anwendungen und den jeweiligen Forschergruppen in ihrer Entwicklung darstellen. Sie bezeichnen größere richtungsweisende Anschauungen und bestehen aus einem Geflecht von einzelnen Theorieelementen.

12 Vgl. Masterman 1974.

2.2 Theorien und Modelle

Ein Theorieelement besteht in seiner Grundform aus einer Menge von Modellen M und intendierten Systemen der realen Welt I . Hinzu kommt eine Datensammlung als Teilmenge der intendierten Systeme: $D \subset I$. Eine Theorie hat somit die Gestalt $T = \langle M, I, D \rangle$.¹³ Die Datensammlung besteht aus Fakten, welche etwa der Beobachtung der intendierten Systeme entspringen. Erfüllen diese Daten die Axiome der Theorie T , bilden diese ein Modell für die Theorie $T : x \in T$. Anders formuliert ist x ein Modell der Theorie T , wenn es sich aus Grundmengen, Hilfsbasismengen und Relationen zusammensetzt: $x = \langle G_1, \dots, G_p, A_1, \dots, A_q, R_1, \dots, R_r \rangle$. Diese Struktur bildet dann, so die Anforderung, eine normierte Beschreibung eines realen Systems ab. Die Grundmengen beheimaten die Objekte und Gegenstandsbereiche der Theorie. Die mathematischen Konstrukte werden durch die Hilfsbasismengen beschrieben und die Menge der Relationen beschreibt die Beziehungen der Objekte. Wie in Kapitel 5 beschrieben wird, werden die Relationen über die dort genauer beschriebenen *shapes* repräsentiert. Die Grundmengen beheimaten in der Simulation *events*. Theorien lassen sich in dieser Form leicht in Prolog übersetzen. Das folgende Beispiel führt in die hier verwendete Prolog-Notation ein und gibt zudem die Beschreibung einer strukturalistisch rekonstruierten Theorie.

2.2.1 Strukturalistische Theorien in Prolog

Computerprogramme bestehen aus einer Reihe von Anweisungen, welche durch den Computer sequenziell oder in nebenläufigen Listen von Sequenzen abgearbeitet werden. Hochsprachen verfügen über komplexere Abstraktionen und erleichtern es so, umfangreichere Anwendungen und Modelle zu beschreiben. Während der Kompilierung des Quelltextes in ausführbare Maschinenbefehle werden diese in Instruktionen für die physikalische Hardware übersetzt. Diese können sich je nach Computerarchitektur und Betriebssystem unterscheiden. In der Interpretation und Kompilierung der Anwendung werden also über mehrere Schritte hinweg komplexe Anweisungen in sehr rudimentäre Instruktionen übersetzt und schlussendlich durch die Hardware ausgeführt.

Prolog stellt hier als deklarative Programmiersprache eine Besonderheit dar: In Prolog werden Programme nicht nur durch einfache Anweisungen (imperativ) beschrieben, sondern bestehen aus einer Verschachtelung von einfachen atomaren Ausdrücken, Prädikaten (diese werden in Prolog auch Fakten genannt) und komplexen prädikatenlogischen Ausdrücken (auch als Regeln bezeichnet). Anstelle einer abstrakten Ablaufdefinition wird eine Wissensdatenbank mit Regeln und Fakten geschaffen und das Problem beschrieben. Der Lösungsweg zu den beschriebenen Problemen

¹³ Vgl. Balzer und Brendel 2018, S. 20ff.

wird von Prolog erarbeitet und dadurch das Programm erzeugt. Die Ausführung besteht somit in dem Versuch des Prolog-Interpreters die Anfrage (query) an die Wissensdatenbank zu beantworten. Dazu werden die Programme zur Laufzeit in Instruktionen übersetzt, da diese sich jederzeit in ihrer Struktur ändern können. Konkret geschieht dies wie folgt: Fakten und Regeln bzw. Relationen werden als Horn-Klauseln der Form $head \leftarrow tail_1 \wedge \dots \wedge tail_n$ bzw. in Prolog `head:-tail_1,...,tail_n` beschrieben. Die Ausführung beginnt durch eine Behauptung, welche Prolog dann versucht zu beweisen. Ein Interpreter erzeugt aus den einzelnen Ableitungsschritten zur Laufzeit dann jeweils die Maschinenbefehle. Dies geschieht folgenderweise: Der Prolog Interpreter wird versuchen, die Abfrage anhand eines Beweismechanismus (Backtracking, baumartige Abarbeitung) auf Grundlage der Wissensbasis des Programms zu beantworten. Durch Rekursion lassen sich Programmflüsse steuern. Die in vielen Programmiersprachen üblichen Schleifenkonstrukte lassen sich zwar in Prolog nachstellen, sind aber eher unüblich, da der Vorteil von Prolog gerade in der deklarativen Natur liegt und die Programmflüsse nicht statisch, wie etwa in imperativen Programmiersprachen, vorgegeben werden. Eine Simulation besteht also aus einer Beschreibung von Daten in Form von einfachen Prolog-Fakten und Regeln. Simulationsläufe werden so durch Anfragen an Prolog gestartet. Dabei wird Prolog die Aufgabe gestellt, zu beschreiben, wie der Zustand der Simulation in einer vorgegebenen Zahl von Durchläufen ist. Die Simulationsschritte ergeben sich dann in den vielen Schritten, welche Prolog durchläuft, um die Anfrage zu beantworten. Die einzelnen Beweisschritte bilden die Handlungen der Agenten oder andere Prozesse in der Simulation.

Da Prolog mit Listen anstelle von Mengen operiert, müssen notwendige Mengenoperationen auf Listen abgebildet werden. Dabei ist zu beachten, dass sich Listen in Prolog, neben den offensichtlichen Limitationen durch die technische Begrenzung auf abzählbare und endliche Elemente, in zwei Aspekten von entsprechenden mengentheoretischen Mengen unterscheiden: Listen in Prolog können mehrfach das selbige Element enthalten und die Liste definiert sich auch über die Position der Elemente innerhalb einer Liste. Beides lässt sich jedoch umgehen und so ein Verhalten entsprechender Mengen simulieren. Dazu bietet Prolog `is_set/1` und `list_to_set` \leftrightarrow /2. Die Objektmengen einer entsprechend des Strukturalismus definierten Theorie lassen sich dadurch als Liste darstellen. Die deklarative Natur und die dynamische Typisierung von Prolog erlauben es, Programme als prädikatenlogische Ausdrücke zu lesen und so Theorien, welche entsprechend formuliert sind, mit wenig Aufwand in Prolog zu adaptieren. Ein mengentheoretisches Prädikat, die Grundlage der Theoriebeschreibung, welches die Beziehung von Grundmengen und Relationen $\langle G_1, \dots, G_m, R_1, \dots, R_n \rangle$ beschreibt, lässt sich so bereits direkt übertragen. Ein Bei-

spiel Suppes zeigt dies anhand der Quasiordnung auf einer zweistelligen Relation.¹⁴ Die Menge A und die zweistellige Relation R bilden die Quasiordnung $\mathfrak{A} = \langle A, R \rangle$, wenn die Objekte aus A in R liegen und R reflexiv und transitiv ist. Dies lässt sich folgenderweise in Prolog übersetzen:¹⁵

```

1 qo(A,R):-
2   product(A,A,P),
3   forall( (member([X,Z],P)), transR(R,X,Z)),
4   forall( (member([X2,X2],P)), reflR(R,X2,X2)).
5 product(A,B,C):-
6   findall([X,Y],(member(X,A),member(Y,B)),C).
7 transR(R,X,Z):-
8   Rel1 =.. [R,X,Y],
9   Rel2 =.. [R,Y,Z],
10  call(Rel1), call(Rel2).
11 reflR(R,X,X).

```

Ein konkretes Beispiel wäre etwa $\mathfrak{A} = \langle a, b, r \rangle$, also die zweistellige Relation $r(X, Y)$ mit den Daten $[a, b]$. In Prolog wäre ein solches Beispiel etwa der Aufruf `qo([a,b,c ↦],r)` und mit Faktenbasis:

```

1 r(a,a).
2 r(a,b).
3 r(b,c).
4 r(a,c).
5 r(c,a).
6 r(c,b).

```

Etwas komplexer, aber dennoch anschaulich, ist ein Beispiel aus der strukturalistischen Rekonstruktion der Newtonschen Mechanik nach Balzer, Moulines und Sneed 1987. Ein Element dieser Theorie ist die klassische Stoßmechanik: Im Kern des Modells steht die Aussage $\sum_{p \in P} m(p) * v(p, t_1) = \sum_{p \in P} m(p) * v(p, t_2)$. Der Gesamtimpuls $m(p) * v(p)$ zum Zeitpunkt t_1 aller Partikel $p \in P$ im System entspricht dem Gesamtimpuls zu t_2 . Dieses Beispiel lässt sich folgendermaßen in Prolog formulieren, indem das Prädikat für potentielle Modelle beschrieben wird. Dafür wird `potential_model/2` erzeugt, welches die Elemente der Struktur definiert:

```

1 :-use_module(funktionen).
2 :-use_module(daten).
3
4 potential_model(ksm,K):-
5   K=[P,T,V,M],
6   not(P = []),

```

¹⁴ Vgl. Suppes 1957, S.250 oder auch als Beispiel in Andreas und Zenker 2014.

¹⁵ Das Beispiel dient der Anschaulichkeit und stellt nicht die kürzeste Form dar.

```

7  member(T, [t1, t2]),
8  member(Pp, P),
9  v(Pp, T, V),
10 m(Pp, M),
11 true,
12 M > 0.

```

Als Argument muss eine Liste $[P, T, V, M]$ übergeben werden, welche im Prädikat weiter definiert wird. Die Masse $m(Pp, M)$ und der Impuls $v(Pp, T, V)$ müssen als Daten zu den Partikeln gegeben sein. Die Prädikate können für eine generischere Definition noch weiter ausdefiniert werden.¹⁶ Das eigentliche Modell, welches die Hauptaussage der Theorie trägt, wird dann wie folgend beschrieben:

```

1  model(ksm, X) :-
2    X = [P, T, V, M],
3    potential_model(ksm, X),
4    impuls(P, t1, Impuls_t1),
5    gesamt_impuls(Impuls_t1, Ges_Impuls),
6    impuls(P, t2, Impuls_t2),
7    gesamt_impuls(Impuls_t2, Ges_Impuls).

```

Die Berechnung des Gesamtimpulses $gesamt_impuls/2$ ist dabei gesondert definiert. Ein Modell für das Theorieelement besteht aus einer Menge von Partikeln und Impulsen.

```

1  model(ksm, [P, T, V, M]).

```

2.2.2 Theorien in der Simulation

Die Agenten bauen Theorien in folgender Form: $theory(TID, PotModID, ModID) \leftrightarrow .$ Eine Theorie stellt hier ein potentiell Modell der Form $pot_model(PotModID \leftrightarrow , BaseType, Types)$ in Relationen zu einem Modell $model(ModID, PotModID, \leftrightarrow Relations)$. Das potentielle Modell beschreibt die Struktur der Theorie durch die darin vorkommenden Ereignistypen, während das eigentliche Modell die konkreten Relationen zwischen den Ereignissen beschreibt. Diese Relationen werden als Vektoren in einem Ereignisraum beschrieben. Konkret werden die Positionen zum Vorkommen spezifischer Ereignistypen im Raum in Relation zu einer Basis dargestellt: $[BaseType, BaseLocation, [Type1, Relative_Location1], [Type2, \leftrightarrow Relative_Location2], \dots]$. Diese Relationen können auch zeitliche Verläufe

¹⁶ Etwa kann die Zeit als Menge mit zwei Elementen, das Prädikat jedoch dann für mehr als zwei Partikel definiert werden, etc. Die Zahlenräume (für die Abbildung auf \mathbb{R}, \mathbb{R}^3) sind durch Prolog bzw. dem verarbeitenden System begrenzt. Prädikate können maximal in Prolog 64-bit im Falle von *integers* oder entsprechend *double precision floats* als Informationen tragen.

bei Hinzunahme der Zeit als Dimension beschreiben. Theorien können in dieser Form Netze bilden, indem etwa verschiedene Theorien die gleichen potentiellen Modelle teilen oder es Schnittmengen zwischen den Relationen der Modelle gibt. Ebenso kann sich eine Theorie in einem Netz entwickeln, indem das Modell durch ein passenderes ersetzt wird und dadurch ggf. eine passendere Theorie entsteht. Das Problem der Theoriebeladenheit wird dadurch umgangen, als dass es nur eine allgemein verfügbare umfassende Beobachtungssprache gibt, welche theorieunabhängig und objektiv ist. Diese starke Idealisierung wurde bewusst so festgelegt, da andernfalls der Fokus der Experimente auf sprachtheoretische Fragestellungen verschoben würde.

2.3 Theoriefindung in der Simulation

Die Theorien, welche die Agenten in den Simulationsexperimenten beispielhaft aufbauen, sind auf die wesentlichen Komponenten reduziert. In den Experimenten nehmen die Agenten, mit den im Folgenden beschriebenen Einschränkungen, die Ereignisse in einer simulierten Welt wahr. In realen Systemen können Ereignisse wahrgenommen und sprachlich beschrieben werden.¹⁷ Die Wahrnehmung von Ereignissen in realen Systemen ist subjektiv und standpunktabhängig. Ereignisse können Teil größerer Ereignisse sein und selbst in Teilereignisse zerlegt werden. Dabei gilt für die subjektive Wahrnehmung der Ereignisse: Für alle Ereignisse $e_n \in E$: Ist Ereignis $e_1 \subseteq e_2$ und $e_2 \subseteq e_3$, so ist auch $e_1 \subseteq e_3$. Führt etwa eine politische Debatte e_1 zu einer Massenzusammenkunft e_2 und diese zu einer erhöhten Zahl von Virusübertragungen e_3 , so ist das Ereignis der politische Debatte e_1 Teil der konkreten Virusübertragung e_3 . Wenn zwei Ereignisse e_1 und e_2 gegenseitig ineinander liegen, so sind diese identisch: $(e_1 \subseteq e_2) \wedge (e_2 \subseteq e_1) \rightarrow e_1 = e_2$. Jedes Ereignis ist Teil seiner selbst und somit reflexiv. Somit stehen Ereignisse in einer Halbordnung. Diese Ordnung lässt sich über die Sprache formulieren. Konkrete Ereignisse können etwa aus Handlungen oder komplexen Konstrukten bestehen. Die Vorführung eines Theaterstücks kann beispielsweise aus Szenen aufgebaut sein. Die jeweiligen Szenen selbst bestehen wieder aus Einzelhandlungen, Gesten und gesprochenen Sätzen der Schauspieler*innen. Diese ergeben eine Dramaturgie, welche wieder eine Obermenge der Szenen ist. Diese lässt sich etwa linguistisch, historisch oder theaterwissenschaftlich beschreiben, zerlegen oder neu kontextualisieren und somit wieder in den Kontext zu anderen Ereignissen setzen. Physikalische Ereignisse, wie etwa der Zerfall eines Atoms, können etwa anhand chemischer Reaktionen beobachtet werden und ein Teil eines Ereignisses in der Farbveränderung einer Flüssigkeit bestehen. Der Zerfall kann aber auch durch ein anderes Ereignis, wie etwa der Be-

¹⁷ Vgl. Balzer und Brendel 2018, S. 18ff.

obachtung der Spuren des Zerfalls in einer gesättigten Ethanol-Gasmischung einer Nebelkammer, beobachtet werden. Dies wäre ebenfalls Teil dieses Ereignisses. Ereignisse in der Simulation sind atomar und die Welt ist beschränkt in ihrer Ausdehnung und Dimensionalität. Die möglichen Typen von Ereignissen sind konstant und ebenfalls in der Zahl beschränkt. Andernfalls würde die Komplexität der Simulation zu stark steigen und potentiell nicht mehr berechenbar sein. Je höher die Zahl der Ereignisse, Raumdimensionen, Ausdehnung und Typen gewählt wird, um so höher muss die zeitliche Auflösung der Simulation gewählt werden, um den Agenten Gelegenheit zu bieten, die Welt erfassen zu können. Da das Ziel der Simulationsexperimente aber nicht in einer möglichst komplexen Simulationswelt liegt, sondern die Verbindung von Prozessen und Ebenen aufgezeigt werden soll, wurden diese Parameter entsprechend stark beschränkt. In der Definition der Welten wurde hingegen eine große Dynamik zugelassen:

Diese können für jeden Simulationslauf neu erzeugt werden. Agenten führen ihre Beobachtungen in den Welten durch und versuchen diese dann zu beschreiben (siehe Kapitel 5.2.2). Ein Modell der Struktur $x = \langle G_1, \dots, G_p, A_1, \dots, A_q, R_1 \dots R_r \rangle$ enthält als Grundmengen und Hilfsmengen die Liste der Ereignisse mit entsprechenden Beobachtungszeitpunkten und deren Positionen. Als Relation wird eine als *Form* bzw. *shape* bezeichnet Liste von Eventtypen in Relation zu einer Basis beschrieben. Die Form Shape = [BaseLocation, [EventType1, Relative_Location1], [EventType2 \leftrightarrow , Relative_Location2], ...] beschreibt die spezifische Konstellation von Objekttypen und ist sowohl Grundlage von Ausgangs- und Zielzuständen der Gesetze als auch möglicher Inhalt von Hypothesen.

2.3.1 Simulierte Welten auf Grundlage von Ereignissen und Gesetzen

Peter Gärdenfors beschreibt Kausalzusammenhänge durch die Abbildung von Ereignissen mittels Vektoren in Aktions- und Zustandsräumen.¹⁸ In Gärdenfors Konzept wird die menschliche Wahrnehmung auf Grundlage dieser Ereignisse beschrieben. Diese unterscheidet sich von anderen Lebewesen dadurch, dass Menschen die treibende Kraft von Ereignissen erkennen und mit diesen in Verbindung bringen. Dies führt Gärdenfors auf mentale Repräsentationen zurück. Ereignisse lassen sich demnach durch Vektoren in den beiden Zustandsräumen beschreiben. Aktionen bedingen Ereignisse über eine Kraft. Diese Ereignisse sind dann die eigentlichen Zustandsveränderungen in einem Vorgang und werden als Vektoren im Ereignisraum beschrieben. So wird der Start- und Zielzustand markiert. Durch Muster in wiederhol-

18 Vgl. Gärdenfors 2020.

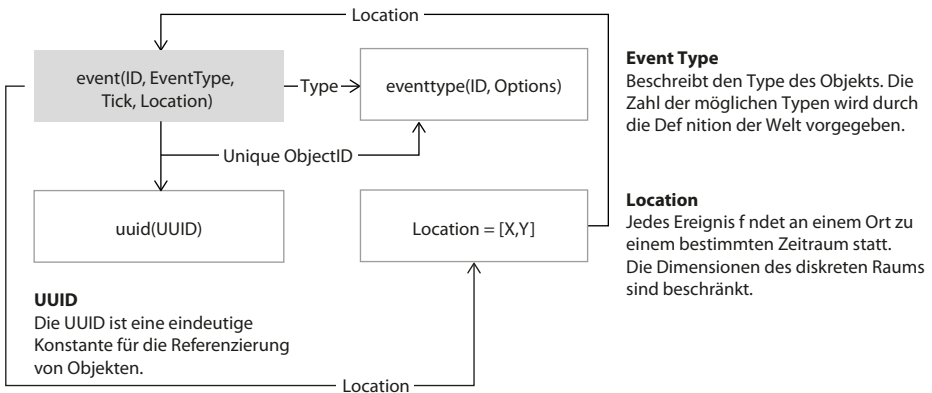


Abbildung 2.1: Definition der Ereignisse

ten Beobachtungen solcher Zustände können Kausalzusammenhänge erkannt und komplexere Zusammenhänge in der Ursache und Wirkung wahrgenommen werden.

Wie zuvor beschrieben, beruht die Beschreibung der simulierten Welt W in den Simulationsexperimenten ebenfalls auf der Darstellung von Ereignissen. Diese unterscheiden sich jedoch von denen Gärdenfors'. Ereignisse sind Teil einer Welt, welche auch aus Typen, einer diskreten Zeit, einem Raum und Gesetzen besteht. Jedes Ereignis $Event$ ist von genau einem Typ $Type_i$ aus der Menge der Typen und findet zu einem Zeitpunkt $t \in T$ statt. Entsprechend bezeichnet ein Ereignis hier nicht einen Zustandsübergang, sondern den konkreten Zustand zu einem Zeitpunkt an einem spezifischen Ort $\langle x, y \rangle \rightarrow \mathbb{N}$ im Raum statt: $event(ID, Type, Tick, Location)$. Der Raum ist diskret und konstant. Prinzipiell kann dieser auch höherdimensional sein, wird den Rechenaufwand mit n^x Rechenschritten allerdings deutlich erhöhen. Die Menge der Ereignistypen wird bei der Erzeugung einer Welt definiert und ist konstant. Eine spezifische Konstellation von zeitgleich eintretenden Events definierten Typs, zu einem Zeitpunkt t_i wird *shape* genannt. Die *shapes* werden über relative Positionen von einer Menge von Events und Typen zueinander beschrieben. Diese Nachbarschaftsrelation wird in Kapitel 5.2.2 beschrieben. Der Zustandsübergang der Welt W von t_i zu $t_i + 1$ wird durch Gesetze beschrieben. Zu jedem Zeitpunkt bzw. Tick t_n finden diese genau einmal ihre Anwendung. Alle Ereignisse innerhalb eines Ticks sind somit zeitgleich. Die Gesetze sind über die gesamte simulierte Welt und Zeit konstant und erzeugen neue Ereignisse für den nächsten Zeitpunkt t_{n+1} folgenderweise: Wenn eine *shape*, also eine Konstellation von Ereignissen spezifischen Typs, zu t_n zueinander als hinreichende Bedingung vorliegt, so wird in $t_n + 1$ ein Zielzustand mit einer neuen definierten *shape* erzeugt und das Gesetz als Konsequenz umgesetzt: $law_i^{t_n} : shape_p^{t_n} \mapsto shape_q^{t_n+1}$. Die Anfangsbedingung wird als *law precondition*: $law_pre(ID, Precondition)$ und die Konsequenz

als *law target* $law_target(LawID, TargetState)$ beschrieben. Das lässt sich analog zu Gärdenfors natürlich auch als Vektor in einem Zustandsraum verstehen. Durch die Anwendung der Gesetze zu jedem $t_n \in T$ verändert sich die Welt entsprechend der Ausgangs- und Zielzustände über die Zeit. Die Abbildung 2.2 zeigt diesen Aufbau eines Gesetzes. Das Gesetz *base_law/3* ergibt sich aus der Verbindung der beiden Prädikate *law_pre/2* und *law_target/2*, welche eindeutige Bezeichner besitzen, und jeweils einer Liste der Ausgangs- und Zielzustände. Die vom Gesetz betroffenen Objekttypen und die Komplexität ergibt sich aus der Obermenge.

Gärdenfors beschreibt Aktionen, welche ein Ereignis hervorruft, als ein Muster von wirkenden Kräften.¹⁹ Die von Gärdenfors genannten Aktionen und Muster lassen sich in dem hier beschriebenen Ansatz über die Ereignisse und *shapes* darstellen, jedoch mit dem Vorteil, dass eine Konstellation von Ereignissen und Ereignistypen nicht an die Gesetze gebunden ist. Agenten, welche die Welt wahrnehmen, können so aus der Beobachtung von Orten oder einzelnen Ereignistypen, komplexere Zusammenhänge bilden. Gärdenfors' Idee, die hinreichende Bedingung eines Ereignisses von der Konsequenz zu trennen, bleibt damit erhalten. Einzelne Ereignisse können jedoch Teil mehrerer Gesetze sein und so auf unterschiedliche Weise miteinander wirken.

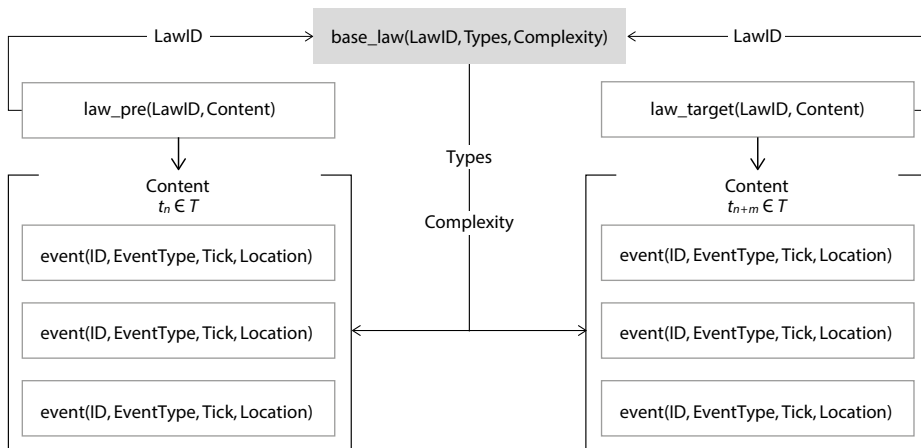


Abbildung 2.2: Gesetze bestehend aus einem Ausgangszustand und einem Zielzustand

Dabei wird bei der Erzeugung der Welt zwischen zwei Arten von Gesetzen unterschieden: Die Basisgesetze bilden den Ausgangspunkt, auf welchem Folgegesetze aufbauen. Das bedeutet konkret, dass Folgegesetze als Ausgangszustand eine Teilmenge der Zielzustände der Basisgesetze oder andere Folgegesetze besitzen. Dadurch er-

¹⁹ Vgl. Gärdenfors 2020, S. 4f.

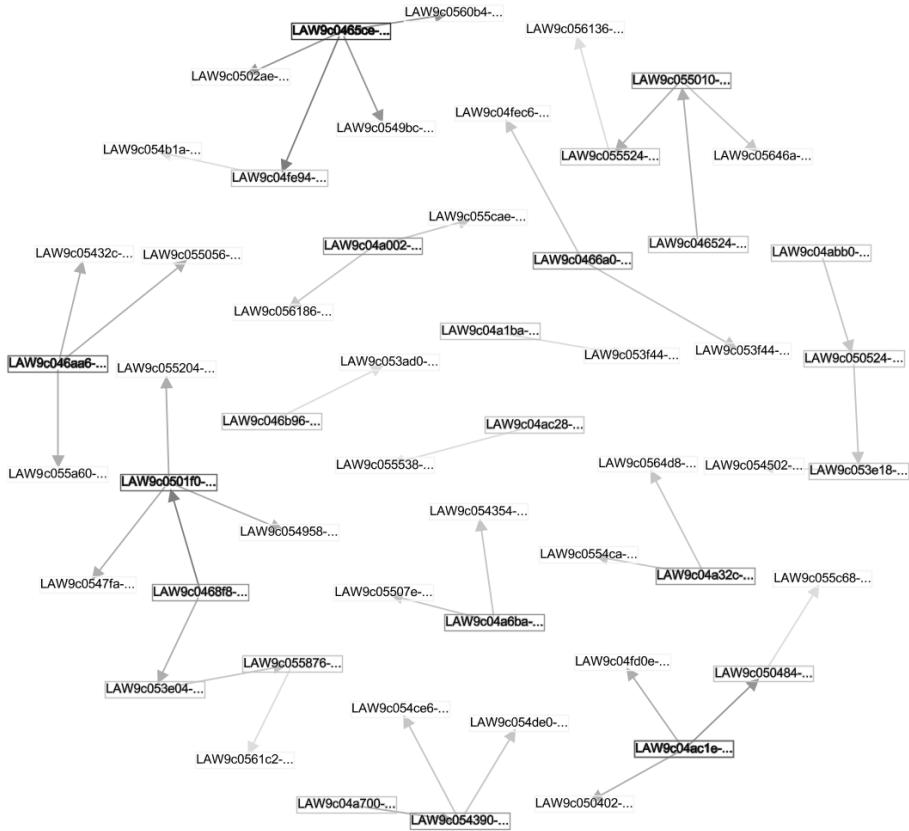


Abbildung 2.3: Struktur der Gesetze einer simulierten Welt

geben sich Kausalketten. Diese können in komplexeren Welten auch zirkulär sein. Auch können sich die *shapes* in den Anfangsbedingungen und Zielzuständen unterschiedlicher Bäume ähneln oder gleichen. Die Abbildung 2.3 zeigt, wie sich dies konkret gestaltet. Es werden die Basisgesetze und die Folgegesetze in einer simulierten Welt anhand ihrer eindeutigen Bezeichnung dargestellt. Die hervorgehobenen Gesetze sind jeweils die Basisgesetze, welche dann ihre Zielzustände als Ausgangspunkt vererben. So entstehen viele kleine Baumstrukturen. In der Grafik werden allerdings keine Überschneidungen oder zirkulären Strukturen dargestellt. Diese ergeben sich durch den Inhalt der Gesetze, den *shapes*.

Die Agenten können so nicht nur dediziert die Auswirkung eines spezifischen Gesetzes untersuchen, sondern auch das Zusammenspiel verschiedener Gesetze. Tatsächlich nehmen die Agenten die Gesetze auch nur über das Aufkommen von Ereignis-

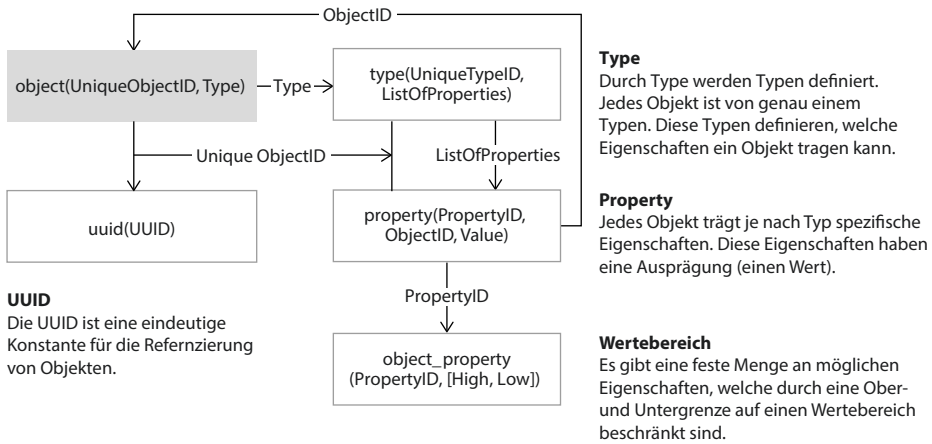


Abbildung 2.4: Definition der Objekte, Typen und Eigenschaften

nissen und Ereignistypen zu einem Zeitpunkt an einem Ort wahr. Darauf aufbauend stellen diese dann die Hypothesen auf. Was genau durch die Ereignisse repräsentiert wird, ist nicht festgelegt. Dies können etwa chemische Prozesse, soziale Interaktion oder auch vollkommen andere Ereignisse sein.

Alternativ kann die Welt, wie in Abbildung 2.4 gezeigt, aus dedizierten Objekten mit einer Menge von Eigenschaften aufgebaut werden. Diese Eigenschaften können Ausprägungen tragen, beispielsweise als Repräsentation von Orten, Farben, Kräften, Wortformen oder geometrischen Strukturen. Typen ergeben sich aus der spezifischen Menge der Eigenschaften der Objekte. Gesetze können sich auf die Eigenschaften oder Ausprägungen auswirken. Diese Variante ist jedoch für komplexere Simulationen weniger praktikabel, da die Welt nicht dynamisch gestaltet werden kann und auch die Theoriefindung in dieser Form prinzipienbedingten Einschränkungen unterliegt.

2.3.2 Aufbau von Hypothesen in der Simulation

Die in Kapitel 5.3 beschriebenen Handlungen erlauben es den Agenten Ereignisse wahrzunehmen und anhand von Beobachtungen Hypothesen zu formulieren. Gärdenfors unterscheidet drei Ebenen der Beobachtung als Grundlage der Hypothesenbildung²⁰ und sieht eine Kombination dieser Ebenen als geeigneten Ansatz, Problemstellungen mittels Induktion zu lösen. Die linguistische Ebene beschreibt die Wahrnehmung durch Beobachtungssätze, unterliegt jedoch klassischen Induktionsproblemen. Um diesen zu entkommen, führt Gärdenfors *conceptual spaces* auf einer kon-

20 Vgl. Gärdenfors 2005, S.125f.

zeptionellen Ebene ein. Beobachtungen werden durch die Abbildung auf Konzepte innerhalb eines Konzeptraums beschrieben. Jeder Punkt in diesem Raum bezeichnet dabei ein konkretes Objekt der Beobachtung. Regionen in diesem Raum bilden die Konzepte über die Ähnlichkeiten bzw. geteilte Eigenschaften der Objekte. So unterteilt sich der Raum auch in Kategorien, über die sich Konzepte voneinander abgrenzen.

Die Agenten der Simulation nehmen Ereignisse und deren Typen in Relation zueinander, zu einem Raum und in der Zeit durch Beobachtungen wahr. Dadurch entstehen *shapes*, welche in ihrer Struktur als Konzept verstanden werden können. Dabei ist jedoch die Beobachtungssprache, in welcher Ereignisse durch die Agenten beschrieben werden, nicht von der Welt getrennt und in diesem Punkt stark idealisiert. Diese Trennung ist prinzipiell in der Simulation möglich, findet allerdings aufgrund der damit verbundenen Komplexität durch linguistische und logische Implikationen nicht statt.

Je nach Typ, Fähigkeit, Ressourcen und Handlungsoptionen haben die Agenten die Möglichkeit, Beobachtungen oder ganze Reihen von Beobachtungen in der Welt durchzuführen. Sollten die Agenten bisher noch keine Beobachtungen durchgeführt haben, wählen sie einen ihnen bereits bekannten Ereignistypen, den sie untersuchen wollen, oder beobachten einen Ort im Raum, auf der Suche nach ihnen noch nicht bekannten Ereignistypen.

3 Soziale Prozesse in der Wissenschaft

Zwischen den Akteuren der Wissenschaft entstehen soziale Interaktionen durch gemeinsame Handlungen. Es werden etwa Vorträge gehalten, Daten ausgetauscht, Projekte bewertet oder Diskurse geführt. Solche Handlungen entstehen in sozialen Räumen, in denen die maßgeblichen Interaktionen der Wissenschaft geschehen. Das macht die Wissenschaft auch zu einem Gegenstand der Sozialforschung. Über den Stellenwert sozialer Gefüge in der Wissenschaft gibt es diverse Diskurse und darin auch Extrempositionen, welche den sozialen Anteil im Wissenschaftsgeschehen entweder als gering oder maßgeblich betrachten. Ein Beispiel ist hier der soziologische Konstruktivismus, welcher die Wissenschaft mit allen Handlungen und Erkenntnissen auf ein pures soziologisches Konstrukt reduziert und somit etwa dem kritischen Rationalismus entgegensetzt. Dazu wurde insbesondere in den 1990ern eine teils mediale Debatte über Wissenschaftskulturen und Perspektiven auf die Wissenschaft unter dem Schlagwort *science wars* geführt. Latour und Woolgar 2013 und andere vertreten die Meinung, dass Wissenschaft und Fakten Teile von Gruppenhandlungen sind und so die Wissenschaft selbst ein pures soziales Konstrukt sei. Diese Art von Debatten ist für die Simulation nicht relevant und wird entsprechend nicht weiter betrachtet. Vielmehr ist die Betrachtung der sozialen Interaktionen, wie sie in den Laborstudien durchgeführt wird, von besonderem Interesse. Diese dienen als Vorbild für die Aktionen der Agenten und werden somit auch in einem gewissen Umfang Grundlage der Simulationsmodelle. Genau hier bietet die Wissenschaftssoziologie durch die Betrachtung von Interaktion von Forschenden und dem Wirken von einzelnen Akteuren oder Akteurenkonstellationen die notwendigen Modelle.¹ Der scheinbare Widerspruch zwischen Latour, Knorr-Cetina und den methodisch- oder theoriegeleiteten Perspektiven löst sich durch eine Trennung der Beschreibungsebenen auf. Hierzu hat etwa Kuhn durch die Verbindung der Rolle von Wissenschaftsgemeinschaften mit den Prozessen der Theorieentwicklung bereits eine Brücke benannt. Diese wurde im Strukturalismus auch klar in die formale Theoriebeschreibung mit aufgenommen. Nun ist zu klären, wie diese Wissenschaftsgemeinschaften agieren. Forschende, als Element solcher Gemeinschaften, führen Handlungen in sozialen Räumen durch, etwa bei der Teilnahme an Konferenzen, bei der Arbeit in der Bibliothek oder im Labor. Knorr-Cetina untersucht diese Räume und beschreibt ausführlich soziale Interaktion der Akteure in diesen Räumen.² Die zugrunde liegenden Handlungen orientieren sich nach Knorr-Cetina an den Wahlmöglichkeiten zur gegebenen Zeit am gegebenen Ort.³ Forschende nutzen die zu dem Moment gegebenen Möglichkeiten, um

1 Vgl. Gläser 2006, S. 67ff.

2 Vgl. Knorr-Cetina und Harré 2012.

3 Vgl. ebd., S. 63-72.

mittels normierter Methoden die erreichbaren und zielführenden Ergebnisse zu erzielen. Zur Beschreibung solcher sozialen Räume werden gerne auch Analogien zu anderen sozialen Räumen herangezogen: Insofern viele Handlungen der Forschenden denen in der Produktion ähneln, werden Labore und Denkräume als Produktionsstätten beschrieben, in denen mittels sozialer Handlungen Wissen fabriziert wird. Wissen ist in dieser Analogie das Produkt einer Produktionskette, in der Forschende interagieren und Erkenntnisse bereitstellen. Dies sind etwa auch wissenschaftliche Artikel, welche das Ergebnis streng normierter Forschungsprozesse sind. Im Kontext der zunehmenden Kommerzialisierung der Wissenschaft wird diese Analogie weitergedacht: Die Wissenschaft wird als ökonomisches System verstanden, in dem Wissen produziert und dann als Ware gehandelt wird.⁴ Mit wissenschaftlichen Verlagen, kommerziellen Laboren und spezialisierten Betrieben entstehen Produktionsketten, welche der Marktwirtschaft ähneln. Wissenschaftler agieren in diesem Konstrukt als Wissensproduzenten, welche kollektiv durch Forschungs- und Wissenschaftshandlungen Wissen generieren und auch zum Teil selbst wieder konsumieren.⁵

3.1 Forschungs- und Wissenschaftshandlungen

Handlungen der Wissenschaft lassen sich in zwei Kategorien gliedern: Forschungshandlungen und Wissenschaftshandlungen. Während die einen im direkten Zusammenhang mit dem Erforschen stehen, ermöglichen die anderen die Forschung durch die Schaffung der notwendigen Rahmenbedingungen.⁶ Zu den Forschungshandlungen zählen intellektuell schöpferische Handlungen ebenso wie das strenge Anwenden von Methoden, die genaue Beobachtung und strukturierte Dokumentation und Beschreibung. Handlungen zur Planung und Verwaltung zählen hingegen zu den Wissenschaftshandlungen. Darüber hinaus weisen Balzer und Manhart 2011 auf die Unterscheidung zwischen genuinen sozialen Handlungen der Wissenschaft und anderen Handlungen hin, welche in der Wissenschaft ihre Anwendung finden.⁷

Ein konkretes Beispiel einer Wissenschaftshandlung ist die Antragsgestaltung zur Akquirierung von finanziellen Mitteln, mit der notwendigen Verwaltung zur Verteilung der Mittel. Der Antrag kann zwar wissenschaftliche Annahmen enthalten, welche selbst Grundlage einer Forschungshandlung sind, doch wird die eigentliche Forschung mit dem Projekt beantragt und die erwarteten Ergebnisse und anzuwendenden Methoden im Antrag beschrieben, mit dem Ziel die erforderlichen Mittel für die

4 Vgl. Balzer 2003.

5 Vgl. Gläser 2006, S. 11f.

6 Vgl. Krohn und Küppers 1990, S. 307f.

7 Etwa ist eine Erhebung im Rahmen einer quantitativen Umfragereihe eine genuine wissenschaftliche Handlung, während Unterhaltungen auch Alltagshandlungen sind, die natürlich auch im Alltag der Forschenden und im Kontext der Forschung stattfindet.

Durchführung der Forschung zu erhalten. Eine Forschungshandlung hingegen wäre beispielsweise eine bestimmte Handlung zur Untersuchung von spezifischen Eigenschaften einer Proteinstruktur oder der Akt der Interpretation eines Romans in Bezug zu seinem historischen und sozialen Entstehungsumfeld. Hierbei wird eine Methode auf einen Forschungsgegenstand angewendet. Die Ergebnisse werden dann in einem intellektuellen Prozess interpretiert. Das Ziel dieser Handlung ist es, Wissen zu generieren. Forschungshandlungen bedürfen Zeit, um diese auszuführen, und Ressourcen wie Instrumente oder Mitarbeiter*innen zur Durchführung von Messreihen oder Befragungen. Beides kann zu einem gewissen Maße über Geld abstrahiert werden. In dem Verhältnis zwischen Forschungs- und Wissenschaftshandlungen entsteht eine gegenseitige Abhängigkeit: Ohne die notwendigen Rahmenbedingungen kann keine Forschung existieren. Die Forschung benötigt die zuvor genannten Ressourcen, doch um diese Ressourcen zu erlangen und den Aufwand gegenüber Förderern und der Gesellschaft zu rechtfertigen, werden hinreichend vorzeigbare Forschungsergebnisse und Perspektiven benötigt. Dadurch werden wieder weitere Wissenschaftshandlungen ermöglicht, die wieder die notwendigen Rahmenbedingungen verlangen. So ergibt sich ein Kreislauf. Die Attraktivität von Institutionen wird durch eine gute Reputation gesteigert. Diese Reputation können die Institutionen auch über herausragende Forschung erlangen. Dies kann auch Einfluss auf das Budget der Institutionen haben. Ein Beispiel hierzu ist die Exzellenzstrategie des Bundes und der Länder.⁸

Seit 2005 wurde in verschiedenen Finanzierungsrunden eine Reihe von Universitäten ausgezeichnet und diese haben so einen zeitweisen Zugang zu zusätzlichen Finanzmitteln erhalten.⁹ 2018 wurde dieser Wettbewerb über die Clusterinitiative und einen langfristigen Status als Exzellenzuniversität noch einmal erweitert und verschärft. Im Verlauf des Verfahrens wurden Cluster, eine thematische Sammlung von Forschungsschwerpunkten und Projekten, als auch die Universitäten durch Standortkonzepte zur Forschung und Lehre jeweils begutachtet¹⁰ und gefördert. Wie die Beispiele zur Mitteleinwerbung zeigen, gibt es hier ein Wechselspiel: Es gilt für Forschende mit ihren begrenzten Zeitressourcen eine ideale Balance zwischen Forschungs- und Wissenschaftshandlungen zu finden und für Institutionen darum, eine ideale Umgebung für Forschende zu schaffen. Ein hohes Lehrdeputat, ausgelöst durch ein schlechtes Lehrenden-Studierenden-Verhältnis, kann diese Balance stören.

⁸ Das Verfahren wird durch die DFG getragen. Beschrieben wird es auf folgender Seite: <http://www.dfg.de/foerderung/exzellenzstrategie/index.html> (zuletzt aufgerufen am 01.06.2020).

⁹ Siehe <http://www.dfg.de/foerderung/programme/exzellenzinitiative/index.html> (zuletzt aufgerufen am 01.06.2020).

¹⁰ Wobei hier neben der wissenschaftlichen Begutachtung sicherlich auch eine politische Dimension zu beachten ist.

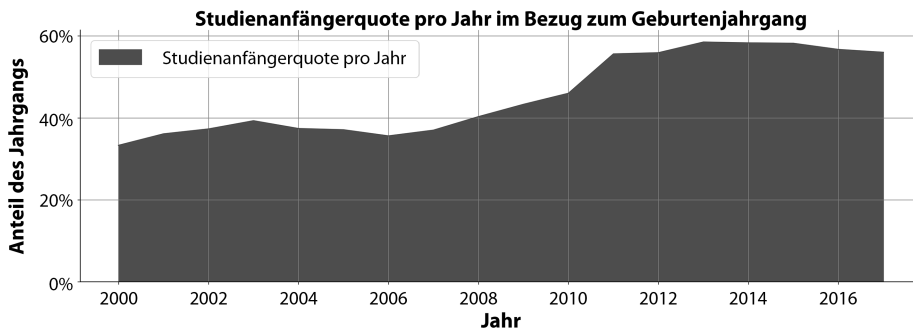


Abbildung 3.1: Studierende pro Jahrgang.

3.1.1 Forschung und Lehre

Der Einfluss von Entscheidungen der Wissenschaftspolitik auf die Entwicklung der Forschung lässt sich in der aktuellen Entwicklung der deutschen öffentlichen Forschungslandschaft beispielsweise in zentralen Veränderungen der letzten Jahre erkennen: Das Verhältnis zwischen der Anzahl von Studierenden zu wissenschaftlich Beschäftigten an Universitäten hat sich signifikant zu Ungunsten des Betreuungsverhältnisses der Studierenden verändert.¹¹ Der Anteil der Studierenden eines Geburtenjahrgangs ist in den letzten 17 Jahren zunehmend gestiegen, wie die Abbildung 3.1 verdeutlicht.¹² Im gleichen Zeitraum ist die absolute Zahl der Studierenden von 1.799.338¹³ im Wintersemester 2000/2001 um 58,1% auf 2.844.978 im Wintersemester 2017/2018 gestiegen.¹⁴ Insbesondere in den Jahren 2000 bis 2017 haben sich größere Veränderungen ergeben: Die Zahl der Studierenden an deutschen Hochschulen ist von 2.002.729 auf 2.844.978 (entspricht 42,0%) angestiegen. Im gleichen Zeitraum ist die Zahl der Beschäftigten an den Universitäten jedoch nur von 812.104 auf 1.099.485, also um 35,4% gestiegen. Die Abbildung 3.2 zeigt die genaue Aufschlüsselung über die Zeit. Die wachsende Zahl der Studierenden wurde nicht im gleichen Maße durch eine Aufstockung des Personals an den Hochschulen abgedeckt. Dies könnte eine be-

11 Dohmen und Wrobel 2018, S. 11 beschreibt die Veränderungen in dem Zeitraum 1995 bis 2015.

12 Die Daten wurden dem Bericht Statistisches Bundesamt (Destatis) 2017, S. 11 entnommen und zeigen den Trend zum steigenden Verhältnis der Studierenden pro Jahr im Zusammenhang zu dem entsprechenden Geburtenjahrgang an. Entsprechend Statistisches Bundesamt (Destatis). 2016, S. 2 lagen die Geburten in den relevanten Jahrgängen 1980 bis 2000 zwischen 865.789 (1980) und 766.999 (2000).

13 Das statistische Bundesamt gibt auf der Website eine leicht vom Bericht abweichende Zahl von 1.798.863 Studierenden an deutschen Hochschulen an: <https://www.destatis.de/DE/ZahlenFakten/Indikatoren/LangeReihen/Bildung/lrbilo1.html> (zuletzt aufgerufen am 27.10.2018).

14 Der Bericht Statistisches Bundesamt (Destatis) 2018, S. 13 gibt eine genaue Aufschlüsselung zu Studierenden an deutschen Hochschulen. In diesem Zeitraum sind bei den Extremwerten jedoch auch zwei Ereignisse zu berücksichtigen: Zum einen wurden seit 2007 über die Zeit in einigen Bundesländern die Zahl der Schuljahre bis zum Erwerb der allgemeinen Hochschulreife von 13 auf 12 Jahre reduziert. Im Jahr 2011 wurde die Wehrpflicht nach mehreren Schritten der Reduktion endgültig ausgesetzt.

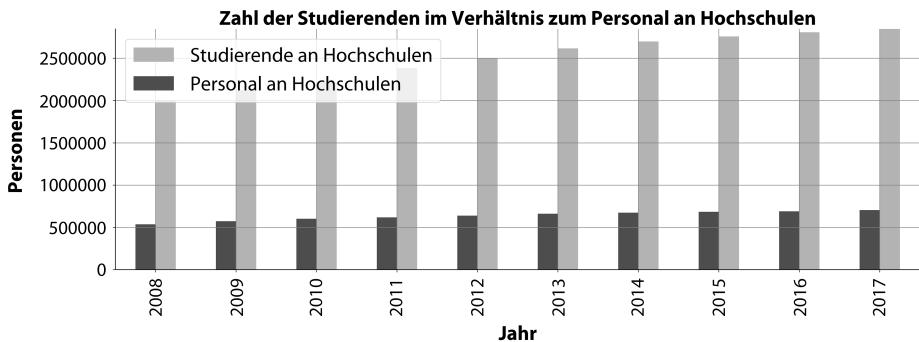


Abbildung 3.2: Verhältnis zwischen Personal und Studierenden

wusste Entscheidung sein, da der Zuwachs der Studierenden durch Änderungen der Bildungspolitik stark schwankte und zwischen 2000 und 2010 die Geburtenrate rückläufig war. Dennoch steigen die absoluten Zahlen der Studierenden.

Ebenso steigt mit der Zahl der Studierenden auch die Belastung durch Lehre und hat das Verhältnis zwischen Forschungs- und Wissenschaftshandlungen zugunsten der Wissenschaftshandlungen verschoben.¹⁵ Dies hat zur Folge, dass Forschende mehr Zeit aufbringen müssen, um die Lehre durchzuführen, oder dass Studierende durch den schlechteren Betreuungsschlüssel ein insgesamt schlechteres Betreuungsverhältnis im Studium erfahren. Während Professor*innen 2008 im Mittel 51,9 Studierende betreuen, sind es 2017 bereits 59,8 Studierende.

Forschende als Arbeitskraft

Der Anteil der wissenschaftlichen und künstlerischen Beschäftigten¹⁶ hat sich von 51,1% auf 56,0% gegenüber Beschäftigten in der Verwaltung zugunsten der wissenschaftlichen und künstlerischen Beschäftigten verschoben. Bei genauerer Betrachtung fällt hier jedoch auf, dass die Zahl der Professorinnen und Professoren zwar absolut leicht steigt, doch im Anteil rückläufig ist (von 7,18% zu 6,75%). Die absoluten Zahlen der Dozent*innen und Assistent*innen sind insgesamt rückläufig. Zuwachs verzeichnen die zumeist befristeten Stellen in der Lehre und Qualifikation, etwa bei wissenschaftlichen Mitarbeiterinnen oder Lehrbeauftragten. Diese Beobachtung lässt sich auch dadurch erklären, dass die Art der Forschung sich von Expertentätigkeiten zu Teamleistungen verändert.

¹⁵ Dies wird auch in dem Bereich der internationalen Expertenkommission zur Evaluation der Exzellenzinitiative Internationale Expertenkommission zur Evaluation der Exzellenzinitiative 2016 auf S. 13f. herausgearbeitet.

¹⁶ Dies sind etwa Projektmitarbeiter*innen oder Lehrkräfte.

Milojević, Radicchi und Walsh 2018 beschreiben wie sich in den USA die Art der Beiträge zur Forschung und gleichzeitig die Karriere Forschender über die Zeit von 1960 bis 2010 veränderte. Dazu untersuchten sie die drei Forschungsdisziplinen Astronomie, Ökologie und Robotik. In diesen nahm die Zahl von Forschenden, welche über ihre Anstellung an Instituten einzig als Co-Autoren und nicht als Hauptautoren in Publikationen vertreten sind, im Verhältnis zur Gesamtzahl der Publikationen deutlich zu. Gleichzeitig hat sich die Verweildauer von Personen an Forschungsinstitutionen von 35 Jahren (1960) auf 5 Jahre (2010) deutlich verkürzt. Die Zahl von befristet Beschäftigten stieg über den Beobachtungszeitraum, ebenso die Zahl von Forschenden, die keine Gelegenheit erhielten, ein eigenes Profil auszubauen. Solche Forschende waren in ihrer gesamten Karriere als Unterstützung größerer Vorhaben tätig. Diese Entwicklung steht im Kontrast zur stärkeren Selektion in der Vergangenheit, in der Forschende entweder die Auslese in den Karrierepfad überstehen oder entsprechend nicht weiter in der Forschung tätig sind.

Durch den relativen Abbau der Verwaltung liegt ein Anstieg des Verwaltungsaufwands auch bei den wissenschaftlichen Beschäftigten nahe. Wie in Kapitel 4.1 beschrieben wird, steigt der Anteil der Drittmittel an Universitäten und somit auch der Anteil von entsprechenden befristeten Stellen. Sowohl der Mehraufwand kurzfristiger Verträge, als auch die Verwaltung dedizierter zweckgebundener Mittel bedarf seitens der Universitäten administrativer Bearbeitung. Der erhöhte Verwaltungs- und Betreuungsaufwand geht allerdings mit gleichzeitiger Reduzierung des entsprechenden Verwaltungspersonals einher.¹⁷

3.1.2 Drittmittelförderung: Forschende im Wettbewerb

Zum anderen wurde der Wettbewerb zwischen Forschenden um Fördermittel durch den Ausbau der konkurrenzbetonten Drittmittelförderung verstärkt, mit dem Ziel die Qualität und Effizienz der Forschung durch einen offenen Wettbewerb um Mittel zu steigern. Auch hierdurch wurde das Verhältnis zwischen Forschungs- und Wissenschaftshandlungen aufgrund des damit verbundenen steigenden Verwaltungsaufwands verändert. Ebenso wird hier möglicherweise auch durch Programmförderungen ein Einfluss auf die Freiheit der Forschung genommen. Denn Forschungsmittel werden auch themengebunden ausgeschrieben (siehe Kapitel 4.1).

Fraglich ist, ob durch die Drittmittelförderung wirklich eine Qualitätssteigerung erfolgt. Schließlich müssen sich nun Forschende vermehrt und frühzeitig um Förder-

¹⁷ Dies widerspricht interessanterweise der medialen Debatte über den Zuwachs an Verwaltungsstellen an Universitäten. Diese Wahrnehmung mag vermutlich auch darauf beruhen, dass Forschende durch die steigende Zahl an Projektförderungen stärker mit Verwaltung beschäftigt und der Verwaltungsaufwand insgesamt gestiegen ist. So entsteht ein engerer Kontakt zu Verwaltungsprozessen. Auch werden gesetzliche Regelungen, etwa bei Einstellungsverfahren oder Ausschreibungen zunehmend verschärft.

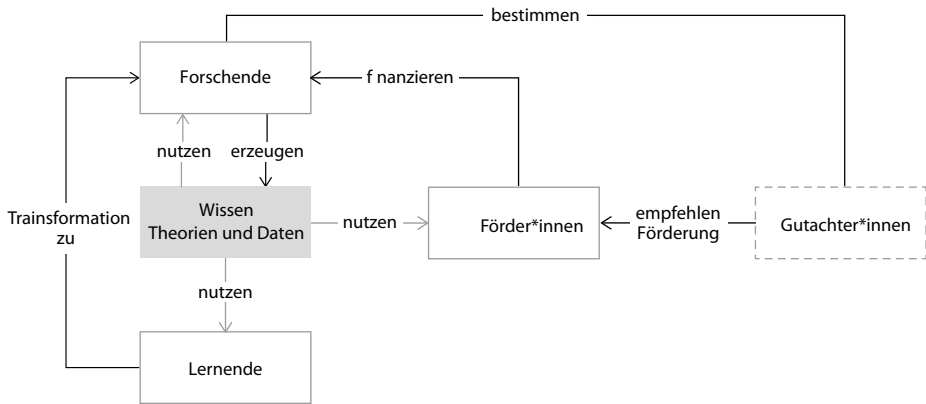


Abbildung 3.3: Rollen in der Wissenschaft

mittel bemühen und dabei mit anderen konkurrieren. Da die Mittel aber begrenzt sind, werden Forschungsprojekte nicht unbedingt an der Dauer der während des Forschungsprozesses aufkommenden Bedarfe ausgerichtet, sondern anhand des anfänglich beantragten Volumens oder der in der Ausschreibung vorgesehenen maximalen Förderdauer. Existenzielle Überlegungen der Antragssteller können zudem auch inhaltliche Motivationen überschatten.

Trotz der politischen Orientierung der Forschungsfinanzierung an marktwirtschaftlichen Strukturen unterscheidet sich der Wettbewerb in der Forschung von anderen ähnlichen Konkurrenzsituationen, wie solchen in der Marktwirtschaft, signifikant. Zum einen sind Forschende, trotz der Konkurrenzsituation, durch die aufeinander aufbauende Forschung aufeinander angewiesen. Zum anderen sind sie auch durch die gegenseitigen Begutachtungsprozesse vom gegenseitigen Wohlwollen abhängig.¹⁸ Diese Faktoren verzerren einen freien Wettbewerb.

3.2 Akteure und Gemeinschaften in der Wissenschaft

Auf Grundlage der beschriebenen Beispiele zur Struktur der universitären Forschung werden in den Simulationen drei Rollen in der Wissenschaft herausgestellt: 1. die Forschenden, 2. die Fördernden und 3. die Studierenden. Diese drei Rollen stehen in einer engen Wechselbeziehung durch zentrale Handlungen zur Generierung von Wissen und weiteren Artefakten der Forschung, wie etwa Theorien, Daten und Methoden.

¹⁸ Vgl. Winterhager 2015, S. 234.

1. Die Forschenden vollziehen die intellektuellen und handwerklichen Handlungen, um neues Wissen zu generieren und zu verbreiten. Dies geschieht beispielsweise durch Handlungen zum Planen und Durchführen von Experimenten oder deren Auswertung und Publikation. Aber auch die Lehre und das Verfassen von entsprechender Literatur zählt hierzu. Das Wissen wird jedoch primär durch die Gruppe der Forschenden generiert. Sie benötigt neben einer finanziellen Ausstattung auch Zugang zu Forschungsgeräten, notwendiger Literatur und Daten, sowie Zeit, um ihre Forschung durchzuführen. Die Forschung hat Daten, Hypothesen und die daraus potenziell entstehenden Theorien als Ergebnis. Forschende sind zumeist an Universitäten, angrenzenden Institutionen (in Deutschland etwa Helmholtz-Zentren, Akademien, Max-Planck-Institute, Leibniz-Zentren), privaten Einrichtungen (etwa Stiftungen, Nichtregierungsorganisationen) oder privatwirtschaftlichen Forschungseinrichtungen angegliedert.
2. Die Fördernden folgen unterschiedlichen Motivationen. Ihr Handeln wird sowohl aus gesellschaftlichen und damit verbundenen politischen Beweggründen motiviert, als auch aus wissenschaftlichen oder moralischen Überzeugungen. Fördernde verteilen die von der Gesellschaft oder privaten Mittelgeber*innen erbrachten Mittel an die Forschenden (beispielsweise über Projekte oder Institutsstiftungen) und Studierenden (etwa über Stipendien). Forschende können über potenzielle Rollen als Gutachtende Empfehlungen für die Verteilung der Mittel geben oder sich direkt als Expert*innen in Gremien an der Wissenschaftspolitik beteiligen. Beispiele sind hier etwa der *Ausschuss für den Europäischen Raum für Forschung und Innovation* (ERAC) der Europäischen Union oder der *Wissenschaftsrat* (WR) auf bundesdeutscher Ebene.
3. In der dritten Gruppe, den Studierenden, werden diejenigen repräsentiert, welche das erarbeitete Wissen erlernen. Diese nehmen das Wissen auf, bilden Überzeugungen aus und entwickeln sich potenziell zu Forschenden. Während der wissenschaftlichen Ausbildung werden erste Forschungsleistungen erbracht, etwa bei der Anfertigung von Studien- und Abschlussarbeiten, der Promotion oder Beschäftigungen am Institut. Die Ausbildung kann einen starken Einfluss auf die Verbreitung von Theorien und die Weiterverbreitung sowie den Fortbestand von Überzeugungen haben.

In der Simulation werden die Agenten als Typ einer dieser drei beschriebenen Gruppen zugeordnet: `Type = [forschende, foerderer, studierende]`. Die Gruppe vermag zu wechseln, doch sind die Agenten zu jedem Zeitpunkt jeweils genau einer Gruppe zugeordnet. Ebenso tragen sie eine eindeutige Kennzeichnung und eine Menge an zusätzlichen Eigenschaften: `agent (Type, AgentID, Properties)`. Jeder Typ ermöglicht den Agenten spezifische Aktionen und Attribute, welche in den Experimenten definiert werden.

3.2.1 Wissenschaftsgemeinschaften

Eine andere Ordnungsebene ergibt sich durch die Gruppierung anhand von Forschungsschwerpunkten und den Überzeugungen zu Theorien. Durch die Übereinstimmung der Überzeugungen zu diesen Aspekten lassen sich Wissenschaftsgemeinschaften bzw. *Scientific Communities* beschreiben. Forschende teilen darin Überzeugungen zu Theorien und arbeiten gemeinsam an deren Ausarbeitung und Verbreitung mittels Publikationen. Der Begriff der *Wissenschaftsgemeinschaften* findet eine breite Verwendung in der Wissenschaftsforschung und wird als Sammelbegriff für Gruppen Forschender mit übereinstimmenden Überzeugungen oder Zielen verwendet. Gläser kritisiert die unscharfe Verwendung des Begriffs¹⁹ und definiert diesen selbst als „kollektive Produzenten [...], deren soziale Ordnung durch den gemeinsamen Bezug aller Produzenten auf das existierende Wissen hergestellt wird.“²⁰

Ein wichtiger Aspekt der parallel stattfindenden Aktionen in Kooperation und Konkurrenz findet sich in Thagards Definition: „Science today is performed by large communities of scientists working sometimes in collaboration, sometimes in conflict. Problem solving, hypothesis formation, and experimentation in a given field may be distributed across individuals. Different groups may be involved in the advocacy and exploration of conflicting hypotheses.“²¹

Eine Beschreibung von Wissenschaftsgemeinschaften findet sich im Strukturalismus, welcher in Anlehnung an Kuhns Begrifflichkeiten *Communities* in den Kontext zu Theorien und insbesondere der Theoriehistorie setzt. Wissenschaftsgemeinschaften werden als die Teilmengen der Forschenden²² beschrieben, welche Überzeugungen zu einer Theorie teilen.²³ Jedoch folgt in ebd. auch direkt der Verweis, dass es einer „further sociohistorical clarification“²⁴ des Begriffs bedarf und dass die einzelnen verwendeten Komponenten „[...] have been used by many authors in a more or less implicit way. We shall consider each of them successively and try to make it as clear and explicit as we can“²⁵. Moulines 1979 beschreibt die Wissenschaftsgemeinschaften als eine Gruppe von Forschenden, welche in einer wissenschaftlichen Sprache kommunizieren und einen gemeinsamen Methodenansatz verfolgen. Die Gruppe existiert üblicherweise über mehrere Forschendengenerationen und bildet so eine

19 Vgl. Gläser 2006, S. 44f.

20 Ebd., S. 261.

21 Thagard 1988, S. 182.

22 Hierbei wird bereits betont, dass diese Definition nicht nur auf Menschen zutrifft, sondern etwa auch für KI-Systeme gelten kann. Diese Sichtweise erlangt gerade durch die Fortschritte im Bereich der Forschung an künstlichen Intelligenzen wieder an Aktualität.

23 Vgl. Balzer, Moulines und Sneed 1987, S. 213f.

24 Ebd., S. 213f.

25 Ebd., S. 211.

genidentische Einheit.²⁶ Die Forschenden besitzen entsprechend Balzer und Brendel 2018 als zentrale Eigenschaften Namen, welche sie identifizieren, Zeiträume, in denen sie aktiv sind, und Überzeugungen.²⁷ Wissenschaftsgemeinschaften bestehen aus einer Menge von Forschenden über üblicherweise mehrere Generationen hinweg. Die Mitglieder der Forschungsgemeinschaft stehen im Austausch und teilen Arbeitsweisen, Ansichten und Methoden. Modelle werden um die Beschreibung von Personen und historischen Perioden ergänzt, doch wird insbesondere auch die Beschreibung der Überzeugungen und Intentionen betont. Überzeugungen beziehen sich dort mitunter auf Fakten, welche einen gewissen approximativen Grad zwischen den Modellmengen und einer Theorie bilden können. Ebenso werden Theorien und Personen in einen zeitlichen Bezug gesetzt. Hierbei entsteht eine Entwicklungsgeschichte zum jeweiligen Theorienetz. Die Netzgeschichte $\text{hist}(z, T, D, P)$ beschreibt das Verhältnis einer Theorie T zum Zeitpunkt $z \in Z$, den Daten D zum Zeitpunkt z und den Personen, welche diese Fakten als Überzeugung P tragen. Das bedeutet, dass dann entsprechend $\forall p \in P : \text{bel}(p, P)$ gilt.²⁸ Die Simulation kann dies in begrenzten Umfang abbilden: Theorien T_n bilden Netze über gemeinsame potentielle Modelle m_p . Auch können Theoriefolgen entstehen, indem Ketten von *shapes* durch einzelne Theorien beschrieben werden. Die Autoren dieser Theorien können retrospektiv ebenso als Gemeinschaften identifiziert werden.

3.3 Wissenschaftliche Handlungen auf Grundlage von Wissen, Überzeugungen und Intentionen

Wissen ist ein traditionelles Thema in der Philosophie und wird auch etwa in der Psychologie, Neurologie oder der Informationswissenschaft untersucht. Eine Simulation von Prozessen in der Wissenschaft benötigt eine anschauliche Definition, welche eine breite Spannbreite dieser Perspektiven abdeckt, dabei aber weder zu sehr ausgrenzt, noch zu allgemein wird. Das Hauptkriterium ist dabei aber auch die Praktikabilität und Anschaulichkeit: Die gewählte Definition muss implementierbar sein und darf nicht durch prinzipienbedingte Lücken zu ungewollten rekursiven Schleifen führen. Agenten bauen ihr Wissen auf Grundlage ihrer simulierten Wahrnehmung auf. Das Wissen soll verarbeitet werden und als Grundlage weiterer Aktionen dienen. Hier bieten sich zum Belief Revision Systeme an, in welchen ein Wissensoperator $K_i(\phi)$ und ein Axiomensystem Wissen und dessen Gültigkeit und Ableitung beschreiben. Pragmatischer und prozessorientierter geht Langley vor: Langley beschreibt den Prozess des Forschens und der wissenschaftlichen Entdeckung mit dem

26 Vgl. Moulines 1979, S. 419.

27 Vgl. ebd., S. 241f.

28 Vgl. ebd., S. 236f.

Ziel der Ausweitung des Wissens über die Außenwelt: „The scientific enterprise is dedicated to the extension of knowledge about the external world.”²⁹ Primär stellen Langeley et. al. vier große Arten von Aktivitäten heraus: (1) Die Suche nach Daten, (2) eine auf ein Minimum an externen Annahmen beschränkte Beschreibung der Daten, (3) die Formulierung von beschreibenden Theorien und (4) das Testen von Theorien. Zieht man an dieser Stelle die Unterscheidung zwischen den bereits erwähnten Forschungs- und Wissenschaftshandlungen ein, so sind alle von Langley genannten Arten den Forschungshandlungen zuzuordnen. Die dafür notwendigen Wissenschaftshandlungen und somit Handlungen zur Darstellung des Wissenschafts-systems sowie der Wechselbeziehung zu umliegenden Systemen, wie der Wirtschaft, Gesellschaft oder Politik, werden vollständig ausgelassen. Auch werden geisteswissenschaftliche, theoretische oder mathematische Forschungshandlungen nicht oder nur teilweise durch diese Kategorien abgedeckt. Konkret sind Handlungen die Verwirklichung von geplanten Vorgängen und Intentionen. Innerhalb des Zustandsraums D aller möglichen Zustände wird die Teilmenge des Ist-Zustand D_i durch eine Handlung über eine Zeitperiode $p = (t_1, \dots, t_n) \in T$ in einen neuen Zustand überführt.

In der Simulation orientiert sich die Implementierung an dem bereits einleitend erwähnten *Belief-Desire-Intention-Model*³⁰. Dies beschreibt Handlungen in einer Wechselbeziehung zu den individuellen Überzeugungen und Intentionen. Mithilfe von Überzeugungen und Annahmen sowie Absichten werden Handlungen geplant und ausgeführt. Dazu besitzt jeder Agent eine Überzeugungsbasis, eine Menge aller Überzeugungen die der Agent zum Zeitpunkt t_i vertritt. Überzeugungen lassen sich als Prädikate beschreiben. $bel(a, t, \phi, \delta)$ drückt aus, dass ein Agent a zum Zeitpunkt t eine Überzeugung ϕ mit einer Ausprägung δ besitzt. Konkret hat eine solche Überzeugung in der Simulation folgende Form `belief(AgentID, Tick, \leftrightarrow ContentType, Content, Domain, Strength)`. Wie im Beispiel ersichtlich, kann die Überzeugung darüber einen spezifischen Typ besitzen und einem Bereich zugeordnet werden. Diese beiden Aspekte erleichtern die Filterung der Überzeugungen in der Simulation, sind jedoch nicht notwendige Bedingung. Überzeugungen können sich graduell unterscheiden und so kann auch eine negative Überzeugung in Bezug auf den Inhalt entstehen. Der Grad der Überzeugung wird in der Simulation von der Quelle bestimmt. Führen Agenten Beobachtungen durch, so sind die Agenten von ihrer Beobachtung stark überzeugt, wohingegen Daten aus gelesenen Publikationen zunächst neutral aufgenommen werden. Diese bedürfen zunächst einer Validierung oder einer vertrauenswürdigen Quelle.

²⁹ Langley u. a. 1987, S. 18. Er bezieht sich hierbei primär auf die positivistische Perspektive empirischer Forschungsdisziplinen.

³⁰ Wooldridge 2000, S. 21.

Die Intention ergibt sich aus einem Handlungsbaum. Die Handlungen ergeben sich auch den Überzeugungen zu Daten, Theorien und den verfügbaren Ressourcen. Fernziel ist es in der thematisch festgelegten Simulation neue Erkenntnisse zu erlangen und diese dann weiterzugeben. Darüber hinaus besitzen die Agenten Interessen, welche subjektive Ziele bestimmen. So sind die Agenten an der Erforschung bestimmter Eventtypen oder Punkten im Ereignisraum der simulierten Welt interessiert und planen ihre Aktionen, um genau diese Aspekte zu untersuchen. Daraus ergeben sich dann die jeweiligen Intentionen und ergründen dann die konkreten Aktionen bzw. Handlungen. In Kapitel 5.3 werden die genauen Handlungen und Handlungs bäume konkreter beschrieben und im Kontext der Simulation aufgeführt.

3.4 Kommunikation in der Wissenschaft

Kommunikation erfüllt in der Wissenschaft mehrere Funktionen. Neben sozialen Aspekten zählt dazu auch der Austausch von Wissen und Argumentationen, welche in der Simulation betrachtet werden. Entsprechend Hagenhoff u. a. 2007 lassen sich zwei Arten von Zielgruppen definieren: Zum einem über die nach innen gerichtete Kommunikation, mit der Zielgruppe der Wissenschaftlerinnen und Wissenschaftler sowie der Studierenden. Zum anderen geschieht dies über die nach außen gerichtete Kommunikation, bei welcher die Öffentlichkeit und Geldgeber als Zielgruppe benannt werden.³¹

Die wissenschaftliche und nach innen gerichtete Kommunikation erfüllt dabei vier Funktionen: (1) Der Zeitpunkt des Erkenntnisgewinns soll damit festgehalten oder dokumentiert werden. (2) Anerkennung und Begutachtung sollen zu einer Legitimation führen. Und es soll (3) eine Verbreitung und (4) Archivierung des Wissens erfolgen.³² Die Kommunikation findet neben dem direkten verbalen Austausch auch auf vielfältigen weiteren Wegen statt, welche sich über viele Jahre in der Wissenschaft herausgebildet haben. Dazu zählen beispielsweise Publikationen von Aufsätzen in Fachzeitschriften oder Vorträge auf Konferenzen. Durch den gegenseitigen Austausch in solchen Medien oder Veranstaltungen werden Diskurse geführt. Die Kommunikation etwa durch das Publizieren und dem Rezipieren von Artikeln zwischen Autoren und Rezipienten findet unidirektional statt oder im direkten Dialog bidirektional. Publikationen wissenschaftlicher Artikel in Zeitschriften richten sich an eine weitestgehend unbestimmte Leserschaft. Ähnlich verhält es sich etwa bei einem Vortrag auf einer Konferenz. Dort können die Vortragenden ein Publikum erreichen. Eine darauffolgende Diskussion kann einen bidirektionalen Austausch ermöglichen.

31 Vgl. ebd., S. 6.

32 Vgl. N. Taubert 2017. Taubert beschreibt diese Funktionen in archivarischem Terminus als *Registrierung*, *Zertifizierung*, *Verbreitung* und *Archivierung*.

Dialoge mit anderen Systemen, etwa der Politik oder Öffentlichkeit, werden jedoch wieder auf andere Weise geführt. Dazu zählen etwa populärwissenschaftliche Darstellungen oder Wissensformate in Form von Dokumentationen oder öffentlichen Vorträgen. Ein wesentlicher und nachhaltiger Teil der Kommunikation in der Wissenschaft erfolgt über Publikationen, welche in der Simulation besonders betrachtet werden.

3.4.1 Publikationen

In Publikationen werden sowohl Diskurse geführt als auch Wissen vermittelt. Dies erfolgt beispielsweise in Form von Monografien oder Aufsätzen in Konferenzbänden, Sammelbänden, Fachzeitschriften, wissenschaftlichen Blogs und Internetbeiträgen. Einmal publiziert, stehen die publizierten Informationen bereit, um von anderen gelesen und bewertet zu werden.

Insbesondere haben Lehrbücher oder allgemeiner Lehrmaterialien einen hohen Einfluss auf die Verbreitung von Überzeugungen, da diese sowohl zur direkten Vermittlung von Wissen als auch als direkte Referenz auf die aktuelle Forschung in der Lehre verwendet werden.³³ Der Beginn des Wandels der direkten Kommunikation über Briefe zwischen Gelehrten hin zu einer strukturierten und breiteren Form des Austauschs über Publikationen lässt sich auf das Erscheinen des ersten wissenschaftlichen Journals *Philosophical Transactions* der britischen Royal Society im Jahr 1666 datieren.³⁴ Bauer beschreibt ebenfalls vergleichbare Entwicklungen in der Antike, dem Gelehrtentum des Islam oder der mittelalterlichen Scholastik. Wie Bauer im Weiteren beschreibt, hat sich nicht nur die Art der Wissenschaftskommunikation systematisiert, sondern auch die Profession als solche herausgebildet. Damit ging laut Bauer auch ebenfalls eine Professionalisierung der Wissenschaftskommunikation einher.³⁵ Die schriftliche wissenschaftliche Diskussion hat sich darauffolgend über die Jahre von einem direkten Austausch und Briefwechsel zunehmend in eine breite Kommunikation über Fachzeitschriften ausgeweitet. Es werden Artikel mit Theesen, Sichtweisen, Belegen oder Daten verfasst und diese dann in Fachzeitschriften diskutiert. Das Wachstum der Wissenschaft und die damit einhergehende Ausdifferenzierung von Fachdisziplinen, Methoden, Theoriennetzen und Ansichten führten über die Jahre zu einer großen Vielfalt an Fachzeitschriften.

Die Rolle der Publikation hat sich durch das Hinzukommen eines weiteren Aspekts noch erweitert. Wissenschaftliche Publikationen in qualitativ hochwertigen Fachzeitschriften tragen zur Reputation der Forschenden bei. Eine Publikation in etablierten Journals gilt beispielsweise als prestigeträchtig und als ein Qualitätsmerkmal für die

33 Vgl. Giere 1997 S. 62ff.

34 Vgl. Kronick 1962 und Hagenhoff u. a. 2007, S. 1.

35 Vgl. Bauer 2017, S. 21ff.

dort veröffentlichte Arbeit. In den Naturwissenschaften wird etwa eine Publikation in *Nature* und angegliederten Zeitschriften als Ritterschlag verstanden. Für die Besetzung von Stellen in der Wissenschaft oder das Einwerben von Forschungsmitteln ist die Reputation von Bedeutung und die Bewertung der Publikationen somit näher zu betrachten. Es gibt mehrere Ansätze diese Effekte genauer zu betrachten und Publikationen in Bezug zu ihrer Wirkung zu metrisieren. So setzt etwa der *h-Index* als bibliometrisches Maß die Anzahl der Publikationen und Zitationen ins Verhältnis oder gibt der *Journal Impact Factor* das Zitationsverhältnis zu Zeitschriften an. Diese Ansätze sagen jedoch nichts über die Qualität oder eine weitere Vergleichbarkeit von Arbeiten aus. Tatsächlich ist eine Vergleichbarkeit über ein generisches Maß problematisch: Forschungsdisziplinen allgemein und insbesondere Teildisziplinen haben eigene Kulturen und Traditionen und somit auch Spezifika in der Publikation und Zitation. So verliert ein generisches Maß an Aussagekraft und birgt die Gefahr einer Missinterpretation.

In diesem Kontext ergeben sich interessante Fragestellungen in Bezug zwischen Publikationen und der Forschung. So untersucht etwa Hagenhoff u. a. die Änderungen der Wissenschaftskommunikation und sagt insbesondere dazu: „[...] Publikationen [sind] seit dem 20. Jahrhundert von essenzieller Bedeutung für den Karrierefortschritt eines einzelnen Wissenschaftlers. Zum anderen wuchs auch das weltweite Budget für die Forschung in beträchtlichem Maß. Beides führte zu einem noch stärkeren Wachstum der Menge der wissenschaftlichen Publikationen. Im Verhältnis hierzu wuchs das Budget für die Literaturversorgung der Wissenschaftler in geringerem Maß, sodass eine flächendeckende, vollständige Literaturversorgung nicht mehr gewährleistet werden konnte.“³⁶ Alternative Publikationsmodelle scheinen hier attraktiv. Die Experimente im Simulationsbeispiel 1 in Kapitel 6.2.1 beschreiben dazu einzelne Aspekte unterschiedlicher Publikationsprozesse und Publikationsmodelle.

Entwicklung der Publikationen, Budgets und Forschenden

Oberflächlich lässt sich zunächst anhand der aktuellen Budgets und dem Zuwachs der Forschenden ein Trend simulieren. Die Abbildung 3.4 zeigt eine Projektion der Entwicklung im Zuwachs von Forschenden und Publikationen, den Budgets der Bibliotheken und Preisen. Als Datengrundlage diente dazu eine Betrachtung der Entwicklung der letzten 10 Jahre in diesen Bereichen. Aufbauend auf der Analyse der durchschnittlichen Budgets von Bibliotheken aus Daten des statistischen Bundesamts wurde ein möglicher Trend für die kommenden 30 Jahre fortgeschrieben. Betrachtet wurden dabei die gemittelten Publikationspreise pro Werk, der anteiligen Zahl der Publikationen und der Zahl der Forschenden. Zu sehen ist ein Anstieg der For-

36 Hagenhoff u. a. 2007, S. 1.

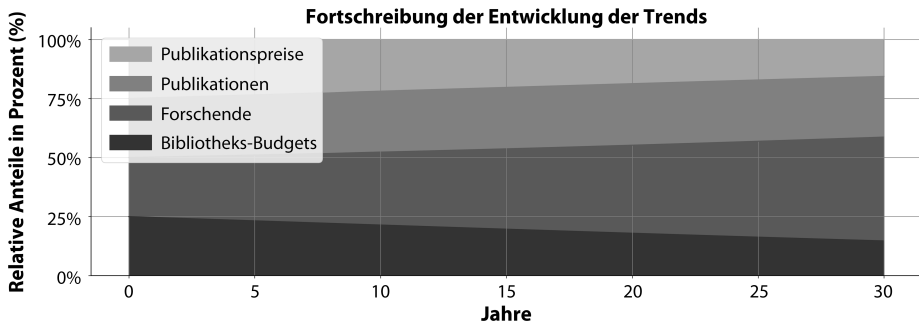


Abbildung 3.4: Relative Entwicklung wachsender Kosten von Publikationen in Bezug zur Zahl von Forschenden und Budgets für Bibliotheken

schenden, Publikationen und Budgets. Es ist ersichtlich, dass in diesem Trend die Budgets der Bibliotheken weder dem starken Anstieg der Forschenden noch der entstehenden Publikationen entspricht. Ebenso ist zu erkennen, dass das Budget der Bibliotheken bereits fast durch Preissteigerungen in der Literaturversorgung ausgeschöpft wird. Die Effekte dieser potenziellen Entwicklung lassen sich in einer Agentensimulation untersuchen.

3.4.2 Publikationsprozesse

Das Feld der wissenschaftlichen Publikationen befindet sich aktuell im Umbruch. Während sich über etwa die letzten 400 Jahre³⁷ eine Struktur aus wissenschaftlichen Fachzeitschriften aufgebaut hat, entwickelt sich durch die Digitalisierung eine Parallelstruktur über digitale Zeitschriften, Repositorien und Fachportale. Diese orientieren sich zum Teil an bewährten Peer-Review Verfahren, in dem Gutachter vor der Veröffentlichung Artikel einer Qualitätskontrolle unterziehen.

Doch werden auch alternative Verfahren eingesetzt. Etwa gibt es verschiedenste Konzepte für Open-Peer-Review Verfahren, welche die bisherigen Verfahren mit dem Ziel einer besseren Transparenz ergänzen oder ersetzen. Auch wird zunehmend zusätzlich zur Publikation die Veröffentlichung der vollständigen zugehörigen Daten und eine klare methodische Beschreibung zur Datenerhebung und Datenauswertung gefordert, wodurch die Publikation in allen Aspekten überprüfbar werden soll. Diese Forderung bestand zwar zu einem gewissen Grad immer, ließ sich aber gerade bei größeren Ergebnisdaten nicht in Druckform realisieren. Die Deutsche Forschungsgemeinschaft fordert entsprechend für die Förderung von Forschungsvorhaben die Einhaltung der *guten wissenschaftlichen Praxis*, in welcher die Vorhaltung aller not-

³⁷ Spier 2002, S. 357.

wendigen Daten zur Rekonstruktion der Forschungsergebnisse von mindestens zehn Jahren verlangt wird.³⁸

Die starke Zunahme an wissenschaftlichen Publikationen und der Weiterentwicklung von Publikationsprozessen bringt auch problematische Entwicklungen mit sich: In *fraud journals* werden Artikel unter der Legitimation eines Reviews publiziert. Jedoch erfolgt ein solches nicht und es werden letztendlich alle Einreichungen publiziert.³⁹ Die Zeitschriftentitel und Gestaltungen orientieren sich dabei an renommierten Fachzeitschriften, um den Eindruck von Seriosität zu erzeugen. Diese Entwicklung ermöglicht es beispielsweise in der medizinischen Forschung methodisch schwache Studien zu publizieren und so eine vermeintliche Wirksamkeit von Produkten nachzuweisen. Dahinter stehen dann etwa wirtschaftliche Interessen, wenn die Wirksamkeit des Produkts als *wissenschaftlich belegt* beworben werden kann oder für eine Zulassung entsprechende Studien nachgewiesen werden können. Aber auch der wachsende Publikationsdruck in der Forschung mag Forschende dazu bewegen, einen schnellen und einfachen Weg zu wählen, um die Zahl der Publikationen zu erhöhen.

3.4.3 Gutachter in Reviewprozessen

Gutachterverfahren sind ein gängiges Mittel zur Auswahl von Publikationen zur Veröffentlichung, Projekten zur Förderung oder anderen Entscheidungsprozessen und dienen etwa der Qualitätssicherung oder Mittelvergabe. Hierbei werden wissenschaftliche Arbeiten durch Gutachter beurteilt und entschieden, ob diese qualitativen Normen erfüllen. Im Wissenschaftsbetrieb werden diese Verfahren bei der Auswahl von Publikation von Aufsätzen, bei der Auswahl von Konferenzbeiträgen oder bei der Förderung von Projektanträgen in den unterschiedlichsten Arten eingesetzt. Um wissenschaftlichen Ansprüchen zu genügen, sind zwei zentrale Komponenten dabei die Transparenz des Verfahrens und die Anonymität, um Beeinflussungen möglichst zu vermeiden. Beide Aspekte müssen jedoch gegeneinander abgewogen werden und können sich zum Teil ausschließen.

Weingart nennt bei der Funktion von Peer-Review-Verfahren drei Kontexte: 1. Die Beurteilung wissenschaftlicher Arbeiten vor der Veröffentlichung, wie es etwa bei Journal Publikationen durchgeführt wird; 2. bei der Entscheidung über die Finanzierung von Forschungsprojekten, etwa durch Drittmittelförderung durch die Deutsche Forschungsgesellschaft; 3. bei der Beurteilung von Forschungsergebnissen als Grundlage für politische Entscheidungen. Dabei unterteilt Weingart diese Kontexte noch in die beiden Wirkungskategorien *nach innen gerichtet* und *nach außen gerichtet*, in

³⁸ Vgl. Forschungsgemeinschaft 2019.

³⁹ Vgl. Beall 2018 und Beall 2017, S. 274.

dem Peer-Reviews jeweils andere Funktionen erfüllen, wie etwa das Vertrauen in einen neutralen Prozess zu stärken.⁴⁰ In einem *double-blind peer review* Verfahren sind beispielsweise dem Autor eines Aufsatzes die Gutachter nicht bekannt. Ebenso sollten die zu begutachtenden Arbeiten möglichst keine Rückschlüsse auf die Autoren erlauben, um auch hier eine Anonymität zu gewährleisten. Letzteres lässt sich jedoch nur schwer sicherstellen, da insbesondere in kleineren Forschungsbereichen bereits aus den Literaturangaben oder dem Themenschwerpunkt der Autor oder die Forschergruppe erkenntlich werden kann. Der Verlag Elsevier rät etwa den Autoren, hier neben der Entfernung von Namen und Förderkennzeichen auch die Literaturangaben für das Review zu anonymisieren.⁴¹

Die Brisanz der Thematik wird in Bezug zur Konkurrenz um Fördergelder oder Deutungshoheiten und den damit verbunden Reputationsgewinn deutlich. Anonymisierte Gutachterverfahren sollen die Konkurrenz und das durch die Begutachtung bestehende Machtgefüge unterbinden. Jedoch gibt es hier die genannten systembedingten Grenzen. Da Gutachterverfahren nur bedingt über Disziplinengrenzen hinweg möglich sind, lässt sich die Problematik prinzipienbedingt nicht durchbrechen. Bei größeren Fachbereichen mag sich dieses Problem nicht stellen, da hier genügend Forschende aktiv sind, als dass direkte Rückschlüsse nicht immer möglich sind.

Balzer und Will beschreiben ein formales Modell für Gutachtersysteme.⁴² Ein Gutachtersystem, bestehend aus einem gewählten Gremium, dem Gutachterkreis, bewertet ein Projekt. Die Bewertung des Projekts setzt sich aus den Einzelgutachten der Mitglieder des Gremiums zusammen. Der Prozess des Gutachtens wird darin als zeitliche Abfolge von Handlungen beschrieben, in welchem Gutachter die zu beurteilenden Projekte zu definierten Zeitpunkten auf einer gesetzten Skala bewerten. Eine Bewertungsrelation $bew(t, g, P^*, x) = n$ stellt die Verbindung zwischen einem Gutachter g des Projekts x und den in diesem Projekt tätigen Mitarbeiter*innen P^* mit einer Note n her. Auf dem formalen Gerüst werden sieben Hypothesen und weitere Spezialisierungen vorgestellt. In der Simulation wird eine vereinfachte Version für diverse Gutachterprozesse eingesetzt. Hier entsteht ebenfalls ein Gutachten, bei dem auch eine Summe von Einzelbewertungen der Gutachter zu einer Gesamtbewertung zusammengezogen werden. Die Abbildung 3,5 verdeutlicht diesen Prozess.

Die Gutachter und Projektteams sind jeweils Mengen von Agenten, wobei in der Simulation beide jeweils Teilmengen der Forschenden sind. Die von Balzer und Will beschriebenen Zeitpunkte werden durch Ticks in der Simulation beschrieben. Projekte sind bereits als solche in der Simulation als abstrakte Größe definiert. Diese

40 Vgl. N. Taubert und Weingart 2016, S. 25.

41 Siehe <https://www.journals.elsevier.com/social-science-and-medicine/policies/double-blind-peer-review-guidelines> (zuletzt aufgerufen am 01.06.2020).

42 Vgl. Balzer und Will 2019 (Manuskript) <http://www.balzerprof-unimunich.de/gutacht.pdf> (zuletzt aufgerufen am 01.06.2020).

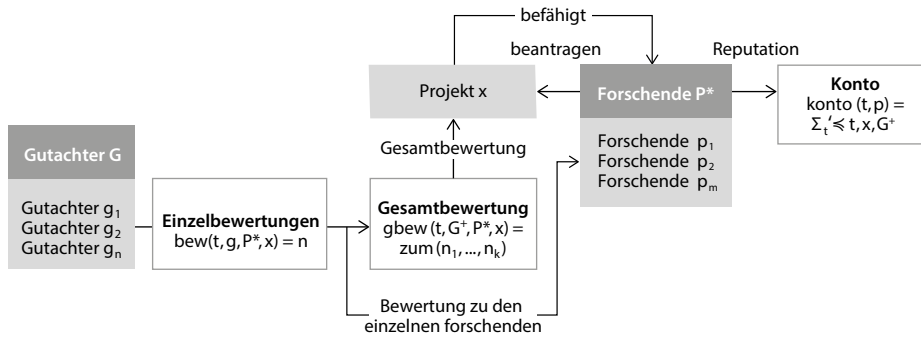


Abbildung 3.5: Bewertungen und Gutachten

werden in der Simulation ebenfalls von Forschenden geschrieben und dann von Gutachtern bewertet. Die Note ist $n \in \mathbb{N}$, welche durch eine Ober- und Untergrenze $n_{\min} \leq n \leq n_{\max}$ beschränkt wird. Die eigentliche Bewertung bzw. Note, wie diese durch Balzer und Will bezeichnet wird, erfolgt durch die Regel $\text{bewertung}(\text{Tick} \leftrightarrow, \text{Gutachter}, \text{Projektteam}, \text{Projekt}, \text{Note})$. Diese Regel bestimmt eine Note, welche von einem Gutachter für ein Projekt und entsprechende Mitarbeiter vergeben wird.

3.5 Empirische Datengrundlage für Publikationsnetzwerke

Solla Price hat vorgeschlagen ein qualitatives und messbares Maß für die soziale Struktur der Wissenschaft zu benennen und so eine Brücke zwischen Methoden der qualitativen und quantitativen Wissenschaftsforschung zu schlagen.⁴³ Price benannte dabei die Netzwerke zwischen Autoren von wissenschaftlichen Artikeln als mögliche Quelle für solche Untersuchungen. Entsprechende Kennzahlen und Größen sind ebenso notwendig, um realitätsnahe Simulationsmodelle erzeugen zu können. Metadaten über Publikationen, also etwa Informationen über die Autorenschaft oder das Publikationsdatum, bieten eine gute Quelle, um Netzwerke und Wissenschaftsgemeinschaften genauer zu betrachten und darauf aufbauend Modelle zu erstellen.

Unter der Annahme, dass Forschende zumeist mit Vertretern gleicher Theorien und Ansichten publizieren, lassen sich diese Metadaten verwenden, um Publikationsnetzwerke aufzuzeigen und zu untersuchen.

Als Datenquelle dient hierfür das von der Cornell-Universität seit 1992 betriebene Online-Archiv für Forschungspublikationen *arXiv*⁴⁴. Das Archiv enthält der-

43 Vgl. Price 1971.

44 Siehe <https://arxiv.org/> (zuletzt aufgerufen am 25.07.2020)

zeit⁴⁵ 1.377.332 Publikationen zu primär naturwissenschaftlichen Publikationen. Dazu zählen etwa Fachbereiche aus der Physik, Mathematik, Biologie, Statistik und Informatik. Insbesondere in der Physik ist arXiv eine offene verbreitete Publikationsplattform.⁴⁶ Die Abbildung 3.6 zeigt die Anzahl von neu publizierten Artikeln. Es ist zu erkennen, dass es einen starken Anstieg der Publikationen pro Jahr auf arXiv gibt. Dieser Anstieg ist im Zuge der Digitalisierung der Kommunikation in der Forschung auch so zu erwarten.

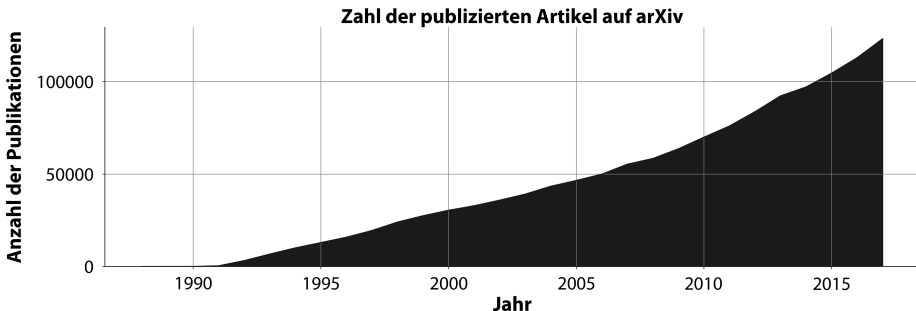


Abbildung 3.6: Anstieg der Publikationen pro Jahr

Die Plattform ist dadurch insbesondere als Datengrundlage geeignet, da sie in den genannten Fachbereichen weite Verbreitung findet und für eine Auswertung mit ihren diversen Zugangsformen eine Vielzahl an Möglichkeiten zur Datenerhebung bereitstellt. Es bietet sich also an, die Publikationsdaten von arXiv genauer zu betrachten. Dies alleine fasst natürlich nicht das von Price geforderte Maß. Diese Datenerhebung ist so nur an Price' Konzept angelehnt und erfüllt den Zweck, eine Idee für realistische Daten und Kennzahlen in den Simulationsbeispielen zu gewinnen.

3.5.1 Datenerhebung und Analyse von arXiv

Forschende stehen mit ihren Publikationen in Relation zu anderen Forschenden und stehen dadurch in einem Netz: Entweder durch die Co-Autorschaft oder Zitation. Publizieren sie etwa gemeinsam, entsteht so eine Beziehung. Diese kann transitiv weitergeführt werden und so kann ein Geflecht an Beziehungen ersichtlich werden. Gemeinsam mit Metadaten zu den Fachbereichen lassen sich Gruppen identifizieren. Für die Untersuchung dieser Publikationsnetzwerke ist also von Interesse zu analysieren, wer mit wem und wann gemeinsam eine Publikation in spezifischen Themen-

⁴⁵ Stand 13.04.2018.

⁴⁶ Hagenhoff u. a. 2007, S. 29-35 beschreibt arXiv und dessen Historie. Die dort beschriebenen Prozesse haben sich allerdings zum Teil in den letzten Jahren geändert. Dies betrifft beispielsweise die Publikations-Policy.

bereichen veröffentlicht hat.⁴⁷ Dazu stehen die in Tabelle 3.1 beschriebenen Metadaten in arXiv über die Publikationen für eine Auswertung zur Verfügung.

Bezeichnung	Beschreibung
title	Titel der Publikation
creator	Autoren der Publikation
subject	Zuordnung zum Fachbereich
description	Abstract zum Inhalt der Publikation
date	Publikationsdatum
type	Art der Publikation
identifier	Eine eindeutige ID zur Identifikation der Publikation

Tabelle 3.1: Verfügbare Metadaten

für die Publikation. So können falsche Zuordnungen bei gleichen Titeln verhindert werden. Der Bereich *astro-ph.GA* beschreibt beispielsweise den Unterbereich *Astrophysics of Galaxies* in der Kategorie *Astrophysics*.

Bevor die Netzwerke genauer betrachtet werden, sollen durch diese Daten zunächst grundlegende statistische Daten zu Publikationen von Forschenden erhoben werden, etwa wie viele Publikationen durchschnittlich von Autoren veröffentlicht werden. Dazu lassen sich die Publikationen pro Autor anhand der Autorennamen und Identifier der Publikationen zählen. Das Ergebnis zeigt sich in der Verteilung in Tabelle 3.2.

Anzahl der Publikationen	Anzahl der Autoren
1–5	636815
5–10	98929
10–15	38086
15–20	71720
mehr als 20	89100

Tabelle 3.2: Publikationen pro Autor

abweichungen bei Abkürzungen von Vornamen. Für die quantitative Analyse sind beide Effekte aber zu vernachlässigen. Ein Tal ergibt sich bei 10 - 15 Publikationen. Da-

Die Qualität der Metadaten ist zwar schwankend, da Namen etwa teilweise ausgeschrieben oder abgekürzt werden und es keine eindeutigen Kennungen für Personen gibt. Für die später aufzubauenen Simulationsbeispiele ist jedoch nur ein Trend notwendig und dieser ist trotz der genannten Einschränkungen klar erkennbar. Die Zuordnung zum Fachbereich im Feld *subject* bezieht sich auf die arXiv Fachkategorien. Das Feld *identifier* bietet eine eindeutige Bezeichnung

Die meisten Autoren waren an bis zu 5 Publikationen über arXiv beteiligt. Jedoch ist insbesondere hier die Ungenauigkeit im Ausgangsmaterial zu bedenken: Zur Auswertung der Daten wurden die in den Publikationen angegebenen Namen verwendet. Dabei kann es zum einen zu Überschneidungen in den Namen⁴⁸ kommen und zum anderen gibt es auch Ab-

⁴⁷ Um diese Metadaten zu erfassen, wurden diese von der Plattform per OAI-PMH zu jeder Publikation ausgelesen und die Verbindungen zwischen den Autoren und Publikationen ausgewertet.

⁴⁸ Um solche Falschzuordnungen zu vermeiden, wird zunehmend die Verwendung von ORCID Angaben von Verlagen und Archiven empfohlen. Ein ORCID ist ein eindeutiger Identifier, welcher von Forschenden beantragt werden kann. So wird eine eindeutige Identifikation zwischen Forschenden und Publikationen hergestellt.

nach steigt wieder die Zahl der Autoren an, die an mehr Publikationen beteiligt waren. Die Spitze bilden einige wenige Autoren, etwa an erster Stelle Saharon Shelah mit 715 Publikationen, dicht gefolgt von A.C. Fabian (707 Publikationen).

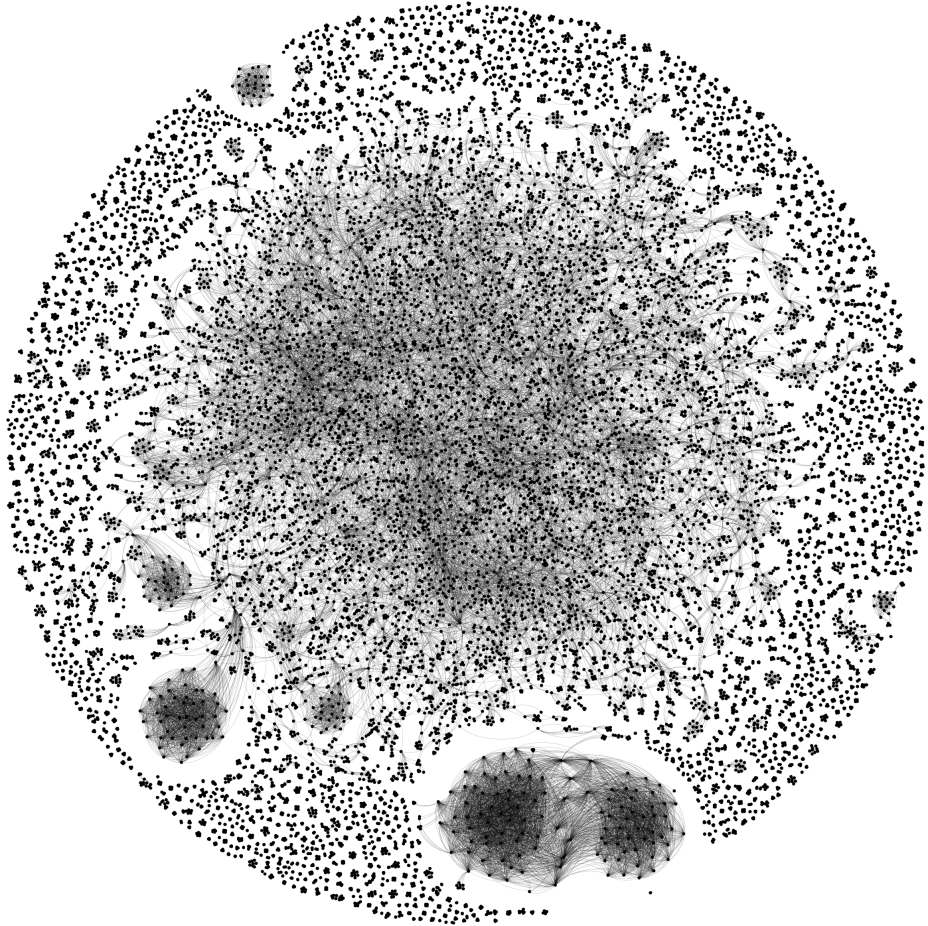


Abbildung 3.7: Netzwerk der Co-Autoren

Autoren auf der Plattform haben im Median zwei Publikationen dort publiziert. Im Durchschnitt sind $\frac{1}{n_i} \sum_{n=1}^{Pub} Aut_i = 3,45$ Autoren an einer Publikation beteiligt, mit einer Standardabweichung von 2,03 und der Median \tilde{x} liegt bei 3 Autoren pro Publikation. Auf Grundlage der Daten können nun auch die Beziehungen zwischen Autoren mittels einer Netzwerkanalyse analysiert werden. Hierzu wurde für eine genauere Analyse und Übersichtlichkeit der Teilbereich *Computer Vision and Pattern Recognition* über einen beschränkten Zeitraum von zehn Jahren analysiert. Dieser Be-

reich und Zeitraum wurden zur Veranschaulichung der Abbildung gewählt, da größere Netzwerke schnell zu groß für eine anschauliche Abbildung werden. Die Abbildung 3.7 zeigt die Vernetzung von Autor*innen in diesem Teilbereich.

Autoren werden in der Abbildung durch Punkte repräsentiert und mit Co-Autoren über eine gemeinsame Publikation durch Linien verbunden. Neben vielen kleinen Clustern, in denen Autoren gemeinsam publiziert haben, sind mehrere größere Häufungen zu erkennen. Hier gibt es eine Vielzahl von Beziehungen zwischen Autoren. Price' Vermutung, dass sich Wissenschaftsgemeinschaften in Publikationsnetzwerken erkennen lassen, lässt sich in diesen Daten nachvollziehen. Ein weiterer Hinweis ergibt sich dadurch, dass die Cluster zum Teil Inseln bilden, die nur mit sehr wenigen Verbindungen zur umgebenden Masse verbunden sind.

Aber dieses Bild lässt auch weitere Vermutungen zu: In dem dargestellten Fachbereich könnten etwa mehrere Themenbereiche subsumiert sein, die inhaltlich wenig Verbindungen haben und auf diese Weise etwa die Autoren voneinander separieren. Neben den kleinen und größeren Inseln, sind auch eine Vielzahl von einzelnen Autoren (bzw. Publikationen) zu sehen, welche in keiner Beziehung zu den anderen Autoren stehen.

4 Wissenschaft im Kontext der Gesellschaft

Beginnend mit der Phase des Frühkapitalismus um 1600 und der Aufklärung hat sich die Wissenschaft als System zu ihrer heutigen Form entwickelt.¹ Sie ist insbesondere seit der Industrialisierung, mit ihrem Einfluss auf die gesellschaftliche Entwicklung, ein zentraler gesellschaftlicher Bestandteil und dabei Triebmotor der kapitalistischen Gesellschaft: Erkenntnisse der Forschung fließen in die industrielle Verwertung, ermöglichen Produktentwicklungen und prägen mit den daraus entstehenden Produkten die Märkte und damit auch den Alltag der industriellen Gesellschaften.

Ein aktuelles Beispiel ist hier die *digitale Transformation*, welche bestehende Prozesse in der Alltagskommunikation und Interaktion (z.B. über soziale Medien, Emails, Foren, virtuelle Konferenzen oder digitale Publikationen) maßgeblich ändert. In der Industrie und Wirtschaft wird die Planung und Durchführung von Produktionsketten oder die Verwaltung durch digitale Verarbeitungsschritte ergänzt oder abgelöst. Die Entwicklung dorthin lässt sich in folgende Phasen unterteilen: Aus der theoretischen Grundlagenarbeit in der Mathematik, Physik und Mechanik entwickelten sich Verfahren und technische Grundbedingungen für komplexe Rechenmaschinen. Blaise Pascal hatte hier möglicherweise 1642 mit seiner Rechenmaschine Pascaline die erste ihrer Art geschaffen. Dies geschah nicht unbedingt zielgerichtet und um Computer entsprechend der heutigen Vorstellung zu schaffen. Jedoch erlaubten diese Arbeiten in der nächsten Phase der praktischen Anwendung dieser Konzepte etwa Charles Babbage im 19. Jahrhundert eine programmierbare Rechenmaschine zu entwickeln, welche sich in ihrer Komplexität und Funktionalität von den vorhergehenden Ansätzen abgrenzte und generische Rechenoperationen durchführen konnte. Etwa ein Jahrhundert später wurde von Konrad Zuse der erste digitale und elektronische generische Rechner Z1 geschaffen.² Die notwendigen Grundlagen in der Mathematik und Informationstheorie wurden etwa durch Alan Turing und Claude Shannon bereit. In der zweiten Hälfte des 20. Jahrhunderts wurden Techniken zur Vernetzung von Computersystemen entwickelt, welche auf der Grundlagenforschung einer Vielzahl von Forschenden aufbaut. Durch die Entwicklung von massentauglichen Fertigungsverfahren und der Entwicklung von kompakten Schaltkreisen stieg die Verbreitung von Computersystemen sprunghaft an und eine neue Industrie entstand. Die aktuelle Entwicklung, Prozesse, Arbeitsabläufe und Güter in den digitalen Raum zu verlagern, baut auf diesen Entwicklungen auf. Unterschiedliche Forschungsstränge haben die Grundlage für einen gesellschaftlichen Wandel geschaffen, in der die Kommunikation global verläuft und sich kulturelle Grenzen auflösen. Die Entwick-

1 Vgl. Zilsel 1976, S. 49f.

2 Zu dieser Zeit existierten mehrere parallele Entwicklungen zu digitalen Rechenmaschinen.

lung mathematischer Verfahren, der Mechanik, Elektrotechnik, Informatik bis hin zur Digitalisierung zeigt beispielhaft, wie verschiedene Forschungsstränge die Gesellschaft grundlegend ändern. Solche Entwicklungen haben einen Einfluss auf das soziale Zusammenleben, die Ethik, Kultur, Politik und Gesetzgebung eines Landes. Bildung und Informationen im Allgemeinen werden insbesondere durch die Vernetzung und Digitalisierung für eine immer breitere Menge an Menschen zugänglich. Gleichzeitig wandeln sich Berufsbilder und Tätigkeitsfelder. Kapitalistische Gesellschaften sind auf einen Geldfluss angewiesen und benötigen dazu Produkte durch Innovationen. So zeigt sich, dass die Forschung wesentliche Aspekte des sozialen Zusammenlebens bedient.

Die Bedeutung der Forschung für die Gesellschaft spiegelt sich auch in den öffentlichen Finanzierungen wider. Die finanziellen Mittel für Forschung und Lehre sind in den Jahren 2001 bis 2013 etwa in den OECD-Ländern um 35%, in Russland um 59% und um 591% in China gestiegen.³

In Deutschland wird die universitäre Forschung zu einem Großteil öffentlich durch direkte Zuwendungen über die Länder oder den Bund finanziert. Neben der direkten öffentlichen Finanzierung existieren auch weitere Einnahmen durch Drittmittel (welche aus öffentlichen Ausschreibungen oder privaten Stiftungen stammen können), wirtschaftlichen Ausgründungen oder Kooperationen. Die medizinische Forschung ist hier ein besonders markantes Beispiel. Universitäre Krankenhäuser agieren auch als wirtschaftliche Unternehmen, gliedern sich aber auch in die Forschung und Lehre ein. Daneben existieren auch private Universitäten oder Fachhochschulen, welche sich primär durch Beiträge und Trägerschaften finanzieren.

Die Gesellschaft stellt Ansprüche an öffentlich finanzierte Forschung und steuert diese etwa über Drittmittelprogramme zur Forschungsförderung. Aber auch die kommerzielle Forschung unterliegt über den Handel von Produkten und die Käuferresonanz Einflussfaktoren. Die kommerzielle Forschung unterliegt dem Druck, die finanziellen Aufwände für die Forschung wieder zu erwirtschaften und Gewinn zu erzielen. Dieser Faktor ist so nicht in der öffentlichen Forschung gegeben. An dessen Stelle treten jedoch andere Konkurrenzfaktoren, wie etwa die Reputation und der Publikationsdruck.

Daneben existieren auch intrinsische gesellschaftliche Forschungsprozesse. Dazu zählen Gutachterverfahren bei der Forschungsförderung oder der Veröffentlichung von Artikeln und Beiträgen in Publikationen. Auch die Lehre, welche der Wissenschaft Ressourcen in Form von ausgebildeten Forschenden zur Verfügung stellt, ist

³ Siehe entsprechende Übersicht im OECD-Bericht Organisation for Economic Co-Operation and Development 2016, S. 153. Der Bericht wurde hier nicht genauer hinterfragt und die zugrundeliegenden Analysen und Kriterien nicht genauer beschrieben. Hier bleibt sicherlich auch zu hinterfragen, wie sich diese Zahlen zusammensetzen und zueinander verhalten. Insbesondere bei dem Beispiel Chinas mag etwa die stärkere Verbindung zwischen Staat und Wirtschaft die Vergleichbarkeit erschweren.

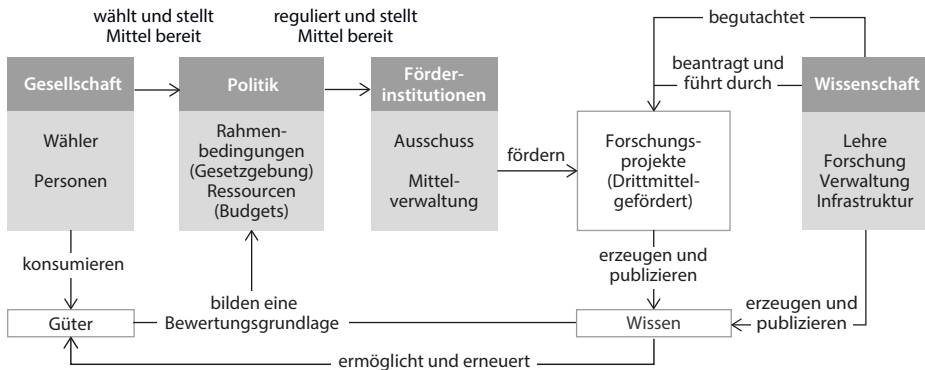


Abbildung 4.1: Forschung und Gesellschaft

ein relevanter Prozess. Durch die Fokussierung auf Schwerpunkte in der Lehre werden auch Grundsteine für die spätere Forschung gelegt. Hierbei spielen ökonomische Prozesse genauso wie politische Prozesse eine wichtige Rolle. Die hier skizzierten verschränkten Prozesse bedürfen einer genaueren Betrachtung, damit diese beispielhaft modelliert werden können.

4.1 Die deutsche und europäische Forschungslandschaft

Die deutsche Forschungslandschaft ist durch ihre Vielzahl von Institutionen und Einrichtungen in Bezug auf die Strukturierung und Organisation heterogen und dezentral organisiert. Es existieren diverse Gremien und Organisationen, welche die öffentliche und private Forschung in Deutschland organisieren. Eine zentrale Rolle darin trägt der Wissenschaftsrat. Dieser führt etwa die Akkreditierung nichtstaatlicher Hochschulen durch, berät bei wissenschaftspolitischen Fragen, unterstützt bei Evaluationen und Forschungsprogrammen. Daneben agieren weitere Vereinigungen und Organisationen. Politisch gesteuert wird die Wissenschaftspolitik in den jeweiligen Landesministerien und dem Bundesministerium für Bildung und Forschung. Wie bereits zuvor beschrieben, lässt sich die Forschung zwischen der öffentlichen, gestifteten und privatwirtschaftlichen Forschung unterscheiden, wobei der Fokus hier auf der ersteren liegt. Die universitäre Forschung bezieht ihr Budget aus zwei Quellen: Zum einen der institutionellen Grundfinanzierung und zum anderen über Drittmittel.⁴ Drittmittel sind Zuwendungen, die neben den Grundfinanzierungen der Universitäten und auf Grundlage weiterer Verträge, etwa durch Forschungsprogramme,

4 Winterhager 2015, S. 34.

an diese vergeben werden.⁵ Diese Art der Forschungsfinanzierung hat über die Jahre an Bedeutung gewonnen.⁶ Der Anteil der Drittmittel an der Forschungsfinanzierung ist damit über die Jahre gestiegen und somit auch die Konkurrenz in der Projektförderung und damit verbundenen Antragsstellung.⁷

Auch wenn die Freiheit der Forschung etwa im Hochschulrahmengesetz⁸ und im Grundgesetz durch den Art. 5 Abs. 3 – dort heißt es: „Kunst und Wissenschaft, Forschung und Lehre sind frei. Die Freiheit der Lehre entbindet nicht von der Treue zur Verfassung.“⁹ – verankert ist, so besteht doch eine starke Abhängigkeit zu den Finanzierungsquellen und den dort getroffenen Vorgaben. Bei der öffentlichen Forschung sind das etwa Vorgaben der Projektträger. Unterschiedliche Förderlinien definieren die Art der Forschungsprojekte und Programme, welche Vorhaben zu einem Themenschwerpunkt bündeln. Die Themenschwerpunkte werden durch die Förderer bestimmt. Da die programmgesteuerte Drittmittelforschung in einem immer stärkeren Ausmaß das Forschungsbild prägt,¹⁰ ist der Arm der drittmittelfinanzierten Forschung den Programmen und den damit verbundenen thematischen, inhaltlichen und strukturellen Vorgaben verpflichtet. Noch drastischer beschreibt es Winterhager: „Das Dilemma ist eindeutig: Massive Haushaltskürzungen in Hochschulen und Kliniken beschränken Forschung und Lehre. Drittmittel tragen dazu bei, diese Defizite in den öffentlichen Kassen wieder auszugleichen und einzigartige Forschung zu ermöglichen. Die Einwerbung von Drittmitteln wurde zudem durch die neuen Universitäts- und Besoldungsgesetze zur Dienstpflicht der Hochschullehrerinnen und Hochschullehrer erhoben.“¹¹

Neben einer Vielzahl von Stiftungen finanziert der Bund über die beiden großen Förderarme der Deutschen Forschungsgemeinschaft (DFG) und der Förderung über das Bundesministerium für Bildung und Forschung (BmBF) eine Vielzahl öffentlicher Forschungsprojekte. Hierbei gibt es unterschiedliche Verfahren, wie diese Projekte eine Finanzierung erhalten. Üblicherweise wird durch einen Drittmittelantrag ein Forschungsvorhaben skizziert. Nach einer Begutachtung wird dem Antrag in Teilen oder vollständig stattgegeben oder die Finanzierung wird abgelehnt.¹²

5 Diese Art von Finanzierung existiert auch für andere Institutionen und wirtschaftliche Betriebe.

6 Tröger 2004, S. 8ff.

7 Winterhager 2015, S. 36f.

8 Hochschulrahmengesetz (HRG), § 4 Freiheit von Kunst und Wissenschaft, Forschung, Lehre und Studium, https://www.gesetze-im-internet.de/hrg/_4.html (zuletzt aufgerufen am 25.07.2020).

9 Siehe https://www.gesetze-im-internet.de/gg/art_5.html (zuletzt aufgerufen am 25.07.2020).

10 Vgl. ebd., S. 224.

11 ebd., Vorwort, S. V.

12 Eine direkte und dauerhafte Förderung des Bundes ist aufgrund des im Artikel 30 des deutschen Grundgesetzes geregelten Kooperationsverbots nur begrenzt im Rahmen der von Artikel 91b genannten Kooperationen möglich.

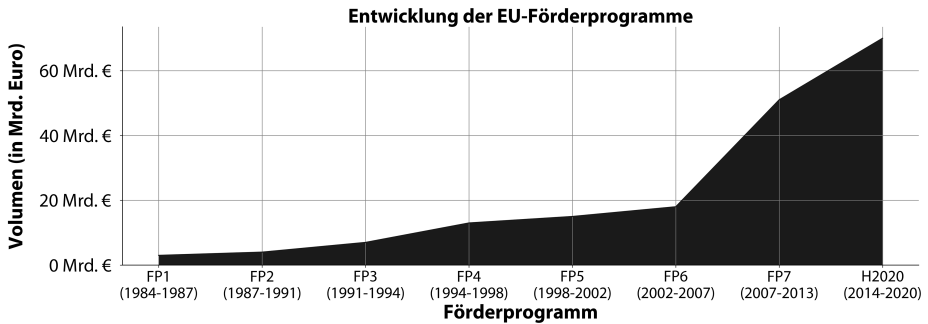


Abbildung 4.2: Fördervolumen der EU-Forschungsprogramme

Auch die Europäische Union fördert Forschungsprogramme und hat seit 1984 über die Jahre acht Frameworkprogramme gestartet. Diese sind kontinuierlich im Budget gestiegen. Die institutionelle Forschung spaltet sich in private und öffentliche Forschung auf, wobei es bei dieser Unterscheidung gerade bei detaillierter Betrachtung eine Vielzahl von Überschneidungen und Graubereiche sichtbar werden.¹³ Durch Kooperationen entstehen ebenfalls Verbindungen zwischen gewerblichen und akademischen Einrichtungen. Hierbei werden etwa Forschungs- und Entwicklungsaktivitäten in Netzwerken und Clustern gebündelt. Darüber hinaus ist die Kooperation auf europäischer und internationaler Ebene zu einer entscheidenden Komponente der Natur- und Geisteswissenschaften in Deutschland geworden. Die private Forschung wird zumeist durch wirtschaftliche Trägerschaften finanziert. Dem gegenüber steht die öffentliche Forschung, welche etwa an Universitäten beheimatet ist und primär durch den Staat getragen wird.

4.1.1 Drittmittelgeförderte Forschung als Prozess am Beispiel der DFG-Förderung

Der Prozess zur Förderung eines Projektes über die DFG ist in mehrere Phasen unterteilt. In diesem Beispiel wird die Entstehung eines Projekts unabhängig einer konkreten Ausschreibung oder eines gezielten Förderprogramms in einem Schwerpunktprogramm beschrieben.

Zu Beginn eines Forschungsantrags wird ein Problem oder ein Ziel skizziert. Die Fragestellung und der aktuelle Forschungsstand werden beschrieben, sowie eine Lösungsstrategie und ein Finanzierungsplan. Dieser Antrag wird an die DFG übermittelt. Die DFG bedient sich zur Bewertung des Forschungsantrags eines Netzwerks von Experten, welche den Antrag inhaltlich begutachten. Auf Grundlage der Gutach-

¹³ Beispiele sind etwa Ausgründungen aus Forschungsprojekten oder Förderlinien, welche eine wirtschaftliche Kooperation oder Nachnutzung fördern.

ten wird über eine Förderung des Projekts entschieden. Unterschiedliche Förderprogramme stellen Rahmenbedingungen an das Projekt, die einreichenden Forschenden und Institutionen. Dies können etwa Forderungen an die Zusammensetzung der beantragten Stellen sein (etwa richtet sich eine Förderung nur an Forschende in bestimmten Karrierestufen) oder aber auch der potentielle Rahmen für Förderdauer und maximaler Förderhöhe werden vorgegeben. Die Bewilligungsquote der Vorhaben lag in den Jahren 2011 bis 2014 je nach Disziplin zwischen 28,3 und 37,2 Prozent.¹⁴

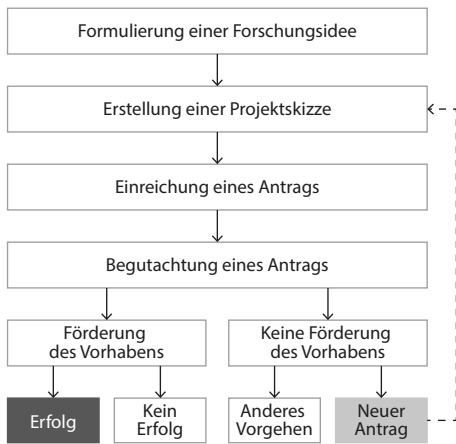


Abbildung 4.3: DFG-Antragsverfahren als Prozess

Für die Formulierung eines Vorhabens bedarf es eines Konzepts, welches den Antrag mit der zugrunde liegenden Fragestellung motiviert und auch die angedachte Methode hinreichend belegt. Auf Grundlage dessen erfolgt die wissenschaftliche Prüfung der Relevanz und Durchführbarkeit. Neben der inhaltlichen Bewertung werden ebenfalls die angedachten Projektpartner auf ihre Eignung (etwa durch Publikationen und institutionelle Anbindungen) begutachtet. Die DFG fordert für die Beantragung von Forschungsgeldern darüber hinaus auch Aussagen zur nachhaltigen Forschung und der Einhaltung der „guten Wissenschaftlichen Praxis“¹⁵.

Dies betrifft insbesondere datenintensive Forschung, welche etwa nachweisen muss, wie und auf welche Weise die erhobenen Daten gespeichert werden. In größeren Verbänden (etwa Sonderforschungsbereichen) werden diese Aspekte in einem eigens dafür geschaffenen INF-Teilprojekt neben den wissenschaftlichen Fachbereichen eigenständig begutachtet. Nach dem Zusammentragen der notwendigen Angaben und der Klärung weiterer Formalitäten zur Projektdurchführung an der Institution, kann eine Projektskizze erstellt werden. Diese wird bei der DFG einem Fachreferat zugeordnet und eingereicht. Nach der Einreichung bestimmt die DFG Gutachter und bittet diese um eine Beurteilung des Forschungsvorhabens. Dies kann bei größeren Vorhaben auch in mehreren Schritten erfolgen und auch durch örtliche Begehungen ergänzt werden, in welchen sich die Forschergruppen mit ihren Arbeiten und Konzepten vorstellen. Als weiteren Aspekt enthält der Forschungs-

¹⁴ Die Daten wurden den Angaben der DFG entnommen. Siehe https://www.dfg.de/dfg_profil/zahlen_fakten/statistik/bearbeitungsdauer/index.html (zuletzt aufgerufen am 25.07.2020).

¹⁵ Deutsche Forschungsgemeinschaft 2013.

antrag einen Projektplan zum Zeitablauf und den erhofften Ergebnissen sowie einen Finanzplan, welcher die Verwendung der beantragten Mittel und der Infrastruktur beschreibt. Dies alles fließt in die Gutachten ein. Auf Grundlage der Gutachten wird dann in mehreren Stufen über eine Förderung entschieden. Hierbei werden die genauen Schritte auch vom Volumen der beantragten Mittel beeinflusst. Die DFG entscheidet ebenfalls darüber, ob das gesamte Vorhaben, entsprechend der Skizze, gefördert wird, ob nur Teilbereiche gefördert werden, oder ob Personalmittel oder sonstige Ausstattung, beispielsweise in Form von Reise- oder Publikationskosten, gekürzt, gestrichen oder gefördert werden.

Im Fall einer positiven Entscheidung beginnt die Projektphase, in der das Forschungsprojekt seine Arbeit aufnimmt. Mit der Durchführung des Projekts werden entweder die gewünschten Ergebnisse graduell oder vollständig erzielt oder das Projekt scheitert in Bezug auf das Erreichen der Projektziele.

Wenn das Projekt nicht gefördert, also abgelehnt wird, obliegt es den Forschenden den Antrag anzupassen und sich an einer weiteren Förderung zu versuchen, alternative Förderwege über andere Instrumente zu betrachten oder die Idee zu verwerfen. Die Abbildung 4.3 verdeutlicht die wesentlichen Stationen dieses Prozesses. Jede der in der Abbildung 4.3 gezeigten Schritte hat eine Wahrscheinlichkeit zu scheitern. Etwa kann ein Antrag nicht gestellt werden, weil vorher Mittel wegfallen oder die Entscheidung der Gutachter fällt negativ aus. Der Pfad zu einer erfolgreichen Förderung besitzt dabei einige Unwägbarkeiten, insofern jeder Schritt zum Scheitern des Gesamtziels führen kann: Mit jedem Schritt ergeben sich die beiden Möglichkeiten $\Omega = \{\text{erfolgreich, nicht erfolgreich}\}$. In dem einfachsten ungewichteten Fall liegt dabei die Wahrscheinlichkeit P zum Eintritt eines spezifischen Ereignisses A bei $P(A) = \frac{1}{2}$. Die Schritte sind dabei im einfachsten Fall stochastisch unabhängig. Dies kann jedoch bei mehrstufigen Antragsverfahren anders sein. In diesen Verfahren werden zunächst Voranträge gestellt. Aus den eingereichten Voranträgen wird aufgrund einer Begutachtung eine Teilmenge selektiert und eine Auswahl an Projekten darum gebeten einen Vollertrag zu stellen. Dabei verändert sich die Basismenge und somit auch die Wahrscheinlichkeit.

Begutachtung eines Forschungsprojekts in der Simulation

Ein Projekt besteht in der Simulation aus einer Liste von Antragsstellern, Gutachtern, der Vorhabensbeschreibung bzw. des Antrags und einer Bewertungsfunktion. Dabei gibt es zwei Strategien zur Begutachtung des Vorhabens: Zum einen können der Antrag und die mit dem Antrag assoziierten Antragssteller durch die Gutachter mittels der Bewertungsfunktion bewertet werden oder die Bewertung der Antragssteller wird alternativ in einem blinden Verfahren nicht berücksichtigt. Konkret ist das Simulationsmodell folgenderweise aufgebaut:

1. Es existiert einen Antrag, welcher eine Fragestellung beschreibt. In der Simulation besteht die Fragestellung aus einer Liste von Typen oder Eventkonstellationen.
2. Es existiert entweder eine Menge von Agenten des Typs `reviewer`, welche sich auf das Fachgebiet des Antrags spezialisiert haben, oder es werden zufällige Gutachter gewählt. Die Spezialisierung wird über das Interesse und Wissen der Gutachter über die im Antrag gelisteten Fragestellungen und Typen abgebildet.
3. Als Bewertungsgrundlage werden die Einzelbewertung der Gutachtenden mittels `reviewer_bias(Rev, Subject, Fitting)` bestimmt. Die Originalität des Antrags wird dazu über `originality(AgentID, Subject, OriginalityScore)` bewertet. Dazu wird dann die Vertrautheit der Antragsstellenden über `familiarity ↔ (AgentID, Subject, FamiliarityScore)` ermittelt.
4. Die Reviewer geben in ihrer Einzelbewertung $bew(t, g, P^*, x)$ (siehe Abschnitt 5.3) entweder eine Förderempfehlung oder lehnen die Förderung ab. Um dies zu bestimmen, wird geprüft, ob die im Antrag beschriebenen Fragestellungen im Interesse des Gutachters liegen und somit `subset(ProposalSubject, Subject)` gilt. Ist dies der Fall, dann wird eine Förderung empfohlen. Wenn dies nicht der Fall ist, wird eine Förderung nicht empfohlen.
5. Durch die Bewertungsfunktion wird die Gesamtbewertung

$gbew(t, G^+, P^*, x) = \frac{1}{G^+} \sum_{i=n}^{G^+} g_i \in G^+$ als Mittelwert über die Summe der Einzelbewertungen `sum_list(Results, SumResults)` gebildet und mit der Bewertung zur Originalität und Vertrautheit verrechnet: `Result is ((SumRes/LenRes*Quality ↔)+OriginalityScore)+FamiliarityScore`. Zuletzt wird bestimmt, ob das Ergebnis der Bewertungsfunktion $gbew(t, G^+, P^*, x)$ über dem Schwellwert δ liegt:

$$\text{Result} > \text{Threshold} \begin{cases} 0 \text{ bzw. nicht gefördert, wenn } \delta \leq g_n \\ 1 \text{ bzw. gefördert, wenn } \delta > g_n \end{cases}$$

4.2 Die Dynamik von Forschung und Lehre

Die Erzeugung und Verbreitung von Wissen innerhalb des Wissenschaftsprozesses beinhaltet zum einen die Erforschung von Gegenstandsbereichen und zum anderen die Vermittlung der Forschung, deren Ergebnissen und den Forschungsmethoden. Die Erforschung kann sowohl den Erkenntnisgewinn als auch die Weiterentwicklung zum Ziel haben, während die Vermittlung eine Notwendigkeit ist, um den Fortlauf des Prozesses zu gewährleisten. Während die Erforschung die Wissensbasis vergrößert,¹⁶ wird Wissen durch die Lehre vermittelt. Diese Prozesse können auf

¹⁶ Natürlich sagt dies noch nichts über Wahrheit, Beschaffenheit oder dem allgemeinen Verhältnis zu einer möglichen Außenwelt oder Wahrnehmung aus. Dieser Art gestaltete erkenntnistheoretische Fragestellungen werden in dem Kontext leider nicht behandelt werden können.

verschiedene Art in Hinblick auf unterschiedliche Aspekte untersucht und beschrieben werden. Die Soziologie bietet dabei Beschreibungsmodelle, welche die sozialen Strukturen und Prozesse beschreiben. Die Ökonomie bietet Beschreibungsmodelle für Prozesse der Wertschöpfungskette in der Gesellschaft. Die Vermittlung von Wissen und das Erlernen von Informationen wird von der Psychologie, Pädagogik und auf abstrakter Art durch die Philosophie untersucht.

Forschungshandlungen stehen in einem direkten Zusammenhang zur Untersuchung des Forschungsgegenstands. Dies umfasst die Anwendung oder Entwicklung geeigneter Forschungsmethodik auf eine Fragestellung oder einen Gegenstandsbe- reich. Dies ist beispielsweise die Untersuchung einer Zellkultur auf bestimmte oder unbestimmte Merkmale, der mathematische Beweis einer Vermutung, die Kontex- tualisierung eines historischen Zeitungsberichts oder ein Erklärungsmodell zu einer Rezeption eines Kunstwerks aufgrund von wahrnehmungstheoretischen Annahmen.

Die Forschungsgegenstände liegen im einfachsten Fall innerhalb der direkten Wahrnehmung. Etwa könnte dies die Beobachtung eines Wetterphänomens sein. Doch vielfach sind diese Beobachtungen nur mit weiteren Annahmen über die Bedingun- gen und mithilfe von Messapparaturen, über deren Funktionsweise ebenfalls Annah- men bestehen, möglich. Salopp gesagt, wenn sich Objekte oder Phänomene durch ihre Winzigkeit, enorme Größe, Ferne oder weiteren Eigenschaften unseren Sinnen entziehen, bedarf es Instrumente wie Mikroskope, Detektoren, Teleskope oder an- dersartiger Messapparaturen. Eine Annahme enthält eine empirische Beobachtung, an welcher sich die Annahme messen lässt. Die Beobachtung induziert eine Annah- me, aus welcher sich etwa eine Hypothese formulieren lässt. So lassen sich vereinfacht die Grundpfeiler der empirischen Methoden beschreiben. Dieser lässt sich re- lativ anschaulich auf wenige Grundbegriffe und Schritte reduzieren, etwa mit Hypo- thesen, Vorhersagen, Beobachtungen oder Messungen, Artefakten oder Forschungs- gegenständen.

Diese Begrifflichkeiten sind jedoch in den Literaturwissenschaften, der Philoso- phie oder der Kunstgeschichte nicht so einfach zuzuordnen. Die Methoden sind hier deutlich abstrakter. Der Rezipient wird zumeist nicht nur durch seine Sinne, son- dern auch zu einem hohen Maß durch seine Erfahrung selbst zum Messinstrument. Beobachtungen und Hypothesen sind im hohem Maße abhängig vom jeweiligen Be- obachter. Auch sind diese in ihrer Form zumeist komplexer formuliert und bedürfen zum Teil selbst einer Interpretation. Dadurch entsteht ein andersartiger Forschungs- zyklus, der an einigen Stellen sicherlich Analogien bietet und auch formal strukturiert sein kann, es aber nicht sein muss. Viele unterschiedliche Strömungen und Traditionen haben hier unterschiedlichste Ansätze hervorgebracht.

Doch stehen die beiden Traditionen hier noch allzu stark im Kontrast zueinander. Es gibt viele Graubereiche und gerade durch die immer stärkeren interdisziplinären Methoden findet hier ein Wandel statt. Klassische geisteswissenschaftliche Bereiche

werden durch computerbasierte Verfahren stärker formalisiert. Andere Bereiche der Naturwissenschaft werden durch das Voranschreiten der Technik immer stärker auch mit ethischen Fragestellungen konfrontiert. Und beides wird durch eine Gesellschaft getragen, welche die Forschung trägt, doch auch Rechenschaft abverlangt.

4.3 Die gesellschaftliche Wahrnehmung der Wissenschaft

Die Wissenschaft steht in zahlreichen Wechselbeziehungen zur Ökonomie, Politik und der weiteren Gesellschaft.¹⁷ In der Gesellschaft tritt die Wissenschaft dabei in zwei zentralen Punkten in den Vordergrund: Zum einen geschieht dies durch die Produktentwicklung. Wie eingangs skizziert, werden wissenschaftliche Erkenntnisse genutzt, um neue Waren zu erzeugen. Und die Wirtschaft betreibt zu diesem Zweck auch selbst Forschung in den jeweiligen Entwicklungsgebieten. Dies können sowohl Konsumgüter sein als auch Dienstleistungen, insbesondere auch medizinische. Der zweite Punkt betrifft die Politik, welche mit Reglementierungen, Forschungs- und Bildungspolitik, Finanzierungen sowie Förderprogrammen dediziert Einfluss auf die Entwicklung des Wissenschaftssystems hat. Die Politik bildet in einer Demokratie auch das Wertesystem der Gesellschaft ab. So mag eine konservative Politik die Forschung an Stammzellen ablehnen und eine ökologisch orientierte Regierung die Forschung für fossile Technologien. Darüber hinaus hat die Wissenschaft in der modernen Gesellschaft längst auch einen neuen Stellenwert als Unterhaltungsware erlangt. So schreibt Fuhrin etwa: „Die mediale Vermittlung von Wissenschaft bekommt dabei eine eigenständige Funktion innerhalb der Massenmedien. Es geht dabei allerdings nicht mehr um Aufklärung und Teilhabe, sondern die Medien instrumentalisieren Wissenschaft als einen Bereich unter anderen, um die Aufmerksamkeit einer beliebigen Öffentlichkeit zu bekommen.“¹⁸ Die Wissenschaft bekommt so eine Rolle in der Medienindustrie. Weiter heißt es: „Wissenschaft wird zur Nachricht und wird an ihrem Nachrichtenwert gemessen. Dabei zählen vor allem Sensationen und potenzielle Katastrophen.“¹⁹ Dies hat auch Auswirkungen auf die Wissenschaft: Zu der Reputation über Wissenschafts- und Forschungshandlungen kommen nun noch Aspekte der öffentlichen Wahrnehmung hinzu. Ein Beispiel zeigte die Medienpräsenz des Virologen Prof. Dr. Christian Drosten während der Corona-Pandemie. Dieser stand neben seiner Forschungstätigkeit den Medien als Experte zur Verfügung. Dies führt dazu, dass seine wissenschaftlichen Tätigkeiten ebenfalls Teil des öffentlichen Diskurses wurden und Prof. Drosten sich dazu ebenfalls in der Öffentlichkeit erklärt.

17 Vgl. Krause und Möller 2008, S. 11.

18 Vgl. Fuhrin 2013, S. 17.

19 Vgl. ebd., S. 17.

Fuhrin sieht allerdings hier eine Entfremdung zwischen Wissenschaft und Gesellschaft: „Die institutionelle Wissenschaft mit all ihren Ausdifferenzierungen entfernt sich weiter von der Öffentlichkeit und erscheint dieser als fremd, weil sie kommunikativ von der Gesellschaft separiert ist.“²⁰

²⁰ Vgl. ebd., S. 17.

Teil II

Simulation

5 Die Simulationsumgebung

Dieses Kapitel beschreibt die Simulationsumgebung und die darin abgebildeten Prozesse. Dieses in Prolog gebaute Grundgerüst erlaubt es, die Experimente auf generische Weise zu implementieren, durchzuführen und zu vergleichen. Die Ausgabe erfolgt dabei als Werteliste, welche dann mit weiteren Werkzeugen zur Datenanalyse weiterverarbeitet werden kann. Die Simulation stellt aber in den Experimenten auch einige Informationsfilter bereit, um die Daten bereits in den Simulationsläufen aufzubereiten, um einzelne Aspekte herauszustellen.

5.1 Aufbau der Simulationsumgebung

Das Simulationsprogramm ist modular aufgebaut und jedes Modul lässt sich für die Beschreibung der einzelnen Experimente konfigurieren und erweitern. Neben einer Zahl von generischen Parametern stehen für die spezifischen Funktionalitäten jeweils einzelne Konfigurationsmöglichkeiten bereit. In den jeweiligen Modulen können dann die Agenten, Handlungen und die simulierte Welt jeweils mit eigenen Regeln definiert und angepasst werden. Die Tabelle 7.1 gibt eine Übersicht über die in den Experimenten verwendeten Module des Kernsystems.

Der Quelltext der Simulation ist unter der Adresse <https://gitlab.gwdg.de/dkurzaw1/eine-simulation-von-wissenschaftsprozessen-in-prolog> verfügbar. Die freie *General Public Licence V3*¹ erlaubt eine Nachnutzung, Anpassung und Weiterentwicklung der Umgebung und der Experimente unter den darin genannten Bedingungen. Um die Simulation auszuführen, ist eine Installation von SWI-Prolog in Version 8.3.3 und den in `simulation.pl` genannten Zusatzmodulen notwendig. Die Zusatzmodule werden von Prolog üblicherweise selbstständig nachgeladen. Für die Ausführung einer Simulation müssen sich alle Module im selben Verzeichnis befinden und die Konfigurationen gesetzt sein. Unter GNU/Linux oder macOS lässt sich die Simulation mittels `swipl -f simulations.pl` laden und dann mittels des Aufrufs `start.` starten. Die Abbildung 5.1 zeigt den Start der Simulation.

Nach jedem Simulationslauf wird eine Zusammenfassung mit einer Übersicht der wichtigsten Parameter ausgegeben. Bei komplexeren Simulationen kann längere Zeit keine Rückmeldung erfolgen und es bedarf entsprechend etwas Geduld. Die Simulation bietet eine Anbindung an Grafana², wodurch der aktuelle Status der Simulation graphisch überwacht werden kann. Sofern nicht anders konfiguriert, nutzt das Simulationsprogramm an einigen Stellen parallele Verarbeitungsschritte und verwendet al-

¹ Siehe <https://www.gnu.org/licenses/gpl-3.0.de.html> (zuletzt aufgerufen am 25.07.2020).

² Siehe <https://grafana.com/> (zuletzt aufgerufen am 25.07.2020).

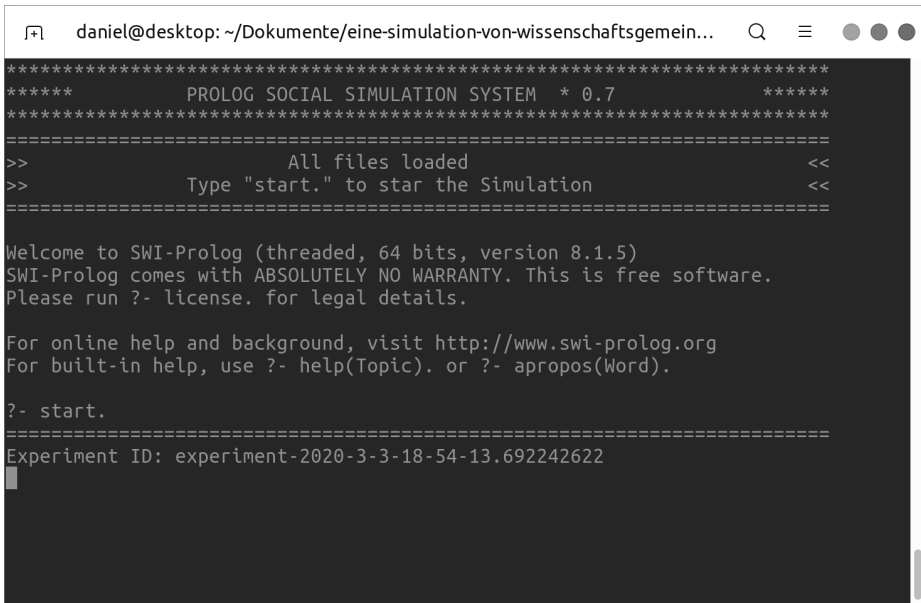


Abbildung 5.1: Start der Simulation

le verfügbaren Prozessorkerne. Die Speicherbegrenzung ist relativ hoch gewählt und es werden viele Daten in dem Speicher abgelegt. Entsprechend können komplexere Simulationen auf moderaten Systemen zu Ressourcenproblemen führen und bedürfen ggf. Optimierungen.

5.1.1 Simulationskern

Die Struktur der Simulation ist dabei folgende: Der Ablauf aller Vorgänge wird durch zwei zentrale ineinander geschachtelte Schleifen gesteuert. Die äußere Schleife bereitet alle Simulationsläufe eines Experiments, wie in Abbildung 5.2 gezeigt, vor.

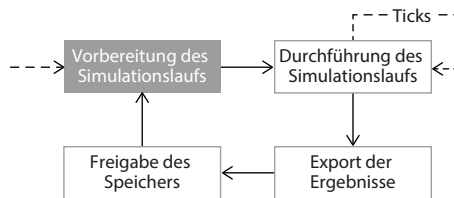


Abbildung 5.2: Schema der äußeren Schleife

Nachdem alle Parameter für den Lauf gesetzt worden sind und die Konfiguration eingelesen wurde, wird ein `pre_run` durchgeführt, es werden initial Agenten erzeugt und der Lauf gestartet. Danach werden die Ergebnisse ausgegeben und der Speicher geleert.

```

1 run(Seed):-
2   % set_random(seed(Seed)),
3   initialisation_run,
  
```

```

4   number_of_ticks_for_prerun(NumberOfTicksPreRun),
5   number_of_ticks(NumberOfTicks),
6   update_number_of_ticks(NumberOfTicks, NumberOfTicksPreRun,
   ↪ NewNumberOfTicks),
7   prerun(NumberOfTicksPreRun),
8   StartActualRun is NumberOfTicksPreRun + 0,
9   initialisation_agents,
10  run(StartActualRun, NewNumberOfTicks),
11  write_results,
12  cleanup.

```

Für den Lauf kann `run(Seed)` entweder einen gewählten *seed*³ übernehmen oder mittels des Zufallsgenerators bestimmt werden.

Die innere Schleife

Die innere Schleife steuert alle Prozesse, welche parallel zu einem simulierten Zeitschritt t_n geschehen. Die Abbildung 5.4 zeigt diese Prozesse. Die Schleife wiederholt ihren Durchlauf über alle simulierten Zeitpunkte $T = \{t_{start}, \dots, t_{end}\}$ entsprechend der Konfiguration in `number_of_ticks(Ticks)` und dem Vorlauf der Simulation: `number_of_ticks_for_prerun(Ticks)`. Der Vorlauf wird benötigt, um die Effekte einer sich im Start befindenden Welt zu filtern.

Die innere Schleife `run(Start, End)` ruft den eigentlichen `tick` auf und aktualisiert nach erfolgreichem Durchlauf die Simulationszeit t_n auf t_{n+1} .

```

1   run(Counter, End):-
2     Counter<End,
3     not(abort_run(true)),
4     Counter1 is Counter + 1,
5     tick,
6     update_sim_time(Counter1),
7     run(Counter1, End).

```

Zum Ende des Laufs werden die Gesetze exportiert, damit diese ebenfalls ausgewertet oder für weitere Simulationsläufe verwendet werden können.

```

1   run(End, End):-
2     not(abort_run(true)),
3     run_id(RunID),
4     export_laws,
5     assertz(finish_state(RunID, completed)).

```

3 Der *seed* steuert den Zufallsgenerator und kann so die Verteilung der Pseudozufallszahlen bestimmen.

Konsistenzprüfung

Wurde in dem vorhergehenden Tick t_{n-1} in der Konsistenzprüfung eine Inkonsistenz festgestellt, bricht die Simulation ab und ein neuer Run wird gestartet. Die bisher durchlaufenen Ticks werden noch gespeichert, allerdings wird der Run als fehlerhaft markiert und an dieser Stelle abgebrochen.

```

1 run(_X, _Y):-
2     abort_run(true),
3     run_id(RunID),
4     assertz(finish_state(RunID, failed)).

```

Die Abbruchbedingungen sind notwendig, um die Berechnungszeit der Simulation zu reduzieren und auch um ungewollte Artefakte von instabilen Welten zu reduzieren. Um dem entgegenzuwirken, werden durch `check_world_consistency(Tick)` folgende Tests durchgeführt:

1. Es konnten im Simulationslauf keine Gesetze angewendet werden. Dies deutet darauf hin, dass die Welt in einem Zustand ist, bei dem kein Gesetz mehr greift – etwa, wenn keine *shape* gefunden werden kann. Die Agenten könnten in dieser Welt nicht agieren.
2. Die Welt enthält keine Events. Dieser Fall tritt meist in Kombination zu 1. ein. Die Welt ist in einem Zustand, in dem die Gesetze keine Ereignisse mehr erzeugen können. Aus diesem Zustand kann sich die Welt nicht mehr erholen, da ohne Vorbedingungen zur Laufzeit keine neuen Ereignisse erzeugt werden.
3. Findet sich ein Muster in der Anzahl der aktiven Gesetze über die letzten Ticks, deutet dies sowohl auf eine sehr kurze Schleife als auch auf eine Welt hin, in der relativ wenig Veränderungen geschehen. Der Lauf wird entsprechend abgebrochen.
4. Ist die Zahl der aktiven Gesetze über mehrere Ticks konstant, so deutet dies darauf hin, dass alle anwendbaren Gesetze immer getriggert werden. Dies ist nicht wünschenswert, da die Agenten hier keine Muster erkennen können. Wenn es zu allen $t \in T$ keine Veränderung gibt, da zu jederzeit alle Gesetze aktiv sind, gibt es für die Agenten keine wahrnehmbare Veränderung. Die einzige Erkenntnis der Agenten ist dann hierbei, dass zu jedem Zeitpunkt alles und immer geschieht.

Zum Ende des Simulationsexperiments wird dann eine Zusammenfassung der einzelnen Läufe mit dem Status (erfolgreich beendet, nicht erfolgreich beendet) und der Zeit für die Berechnung angezeigt.

Wenn, wie in der Abbildung 5.3 gezeigt, eine Vielzahl von Simulationsläufen nicht erfolgreich ist, ist dies nicht als Fehlschlag des Experiments zu verstehen. Der Abbruch aufgrund der oben genannten Bedingungen ist keine Aussage über die Qualität der Simulationsergebnisse. Viel mehr bedeutet dies, dass eine Reihe von Läufen

```

cloud@simulation (10.254.1.7) - byobu
experiment-2020-7-4-0-15-20.797185182
run-2020-7-4-1-33-24.375825166
% 2,588,271 inferences, 500.073 CPU in 525.804 seconds (95% CPU, 5176 Lips)
=====
EXPERIMENT OVERVIEW
RUN-ID ..... RESULT
>> run-2020-7-4-0-15-21.077580928 ..... failed <<
>> run-2020-7-4-0-15-58.871549844 ..... failed <<
>> run-2020-7-4-0-16-53.597726106 ..... failed <<
>> run-2020-7-4-0-17-12.408818721 ..... completed <<
>> run-2020-7-4-0-22-16.031227111 ..... failed <<
>> run-2020-7-4-0-22-42.102777004000004 ..... failed <<
>> run-2020-7-4-0-23-8.723162650999999 ..... failed <<
>> run-2020-7-4-0-29-53.316860437 ..... failed <<
>> run-2020-7-4-0-31-9.095500707 ..... failed <<
>> run-2020-7-4-0-32-33.69333744 ..... failed <<
>> run-2020-7-4-0-33-55.442413568 ..... failed <<
>> run-2020-7-4-0-35-16.720185995 ..... failed <<
>> run-2020-7-4-0-36-42.532774925 ..... failed <<
>> run-2020-7-4-0-38-0.860872983 ..... failed <<
>> run-2020-7-4-0-39-0.989969015 ..... failed <<
>> run-2020-7-4-0-40-3.640888452 ..... failed <<
>> run-2020-7-4-0-41-43.08836627 ..... failed <<
>> run-2020-7-4-0-43-9.297573804 ..... failed <<
>> run-2020-7-4-0-57-4.161784887 ..... completed <<
>> run-2020-7-4-1-3-53.754263639 ..... failed <<
>> run-2020-7-4-1-4-29.960193872 ..... failed <<
>> run-2020-7-4-1-9-27.915302753 ..... failed <<
>> run-2020-7-4-1-10-5.543021917 ..... failed <<
>> run-2020-7-4-1-30-31.746700525 ..... failed <<
>> run-2020-7-4-1-32-56.33928132 ..... failed <<
>> run-2020-7-4-1-33-24.375825166 ..... failed <<
? -

```

Abbildung 5.3: Zusammenfassung der Simulationsergebnisse

in der gewählten Konfiguration eine der beschriebenen Abbruchbedingungen triggern. Der Grund hierfür kann in den Modellen liegen, jedoch auch in der Grundkonfiguration. So können etwa zu viele oder zu wenige Gesetze oder Events vorhanden sein. Nun kann entweder die Gesamtzahl der Läufe innerhalb eines Experiments erhöht oder die Grundkonfiguration angepasst werden. Sollte dies nicht wünschenswert oder erfolgreich sein, so ist vermutlich ein Simulationsmodell ungünstig gewählt.

Prozesse innerhalb eines Simulationsschritts

Innerhalb der Schleife werden dann zu jedem Tick die in Abbildung 5.4 gezeigten Schritte durchlaufen.

```

1  tick:-
2      sim_time(Tick),
3      update_world,
4      % update_range,
5      update_agents,
6      perform_actions,
7      check_world_consistency(Tick).

```

Die Welt wird, wie in Kapitel 5.2.2 beschrieben, aktualisiert. Gesetze werden dabei angewendet und Ereignisse entsprechend erzeugt. Danach werden die Agenten aktualisiert.

```

1  update_agents:-
2      agent_activity,
3      create_new_agents,
4      update_list_of_active_agents.

```

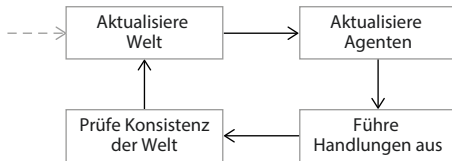


Abbildung 5.4: Schema innerhalb eines Ticks

Es wird geprüft, ob die Agenten noch aktiv sind. Dazu gibt es in der Simulation eine Wahrscheinlichkeitsverteilung. Die Idee für die Implementierung fußt auf den Ergebnissen der in Kapitel 3 besprochenen Verweildauer in der Forschung und den Daten aus der Analyse der Publikationen. Die Werte wurden allerdings

nicht direkt übernommen, da in der Simulation aus diversen Gründen⁴ nur eine kleine Zahl von Agenten aktiv ist. Die hier gezeigten Werte wurden von der ursprünglichen Analyse soweit abgeändert, dass sich in der Simulation ein stabiles Verhältnis zwischen neuen und ausscheidenden Agenten bildet, um ein zu starkes Ansteigen der Zahl der Agenten zu vermeiden. Sind die Agenten weniger als 6 Ticks aktiv, liegt die Wahrscheinlichkeit, dass diese die Wissenschaft verlassen bei 20% pro Tick. Die 6 Ticks wurden in Anlehnung an das Wissenschaftszeitvertragsgesetz gewählt. Demnach können wissenschaftliches und künstlerisches Personal sachgrundlos bis zu sechs Jahren beschäftigt werden. Der Wahrscheinlichkeitswert wurde gewählt, um die Zahl der Agenten pro Simulationslauf konstant zu halten. Bis zum Erreichen des Alters 45 ist die Wahrscheinlichkeit gering, steigt jedoch mit zuneh-

⁴ Dies sind insbesondere die Laufzeit der Simulation, welche sich mit jedem Agenten verlängert und auch die Anschaulichkeit, welche mit größeren Mengen von Agenten verloren geht.

mendem Alter. Ab 45 Ticks steigt die Wahrscheinlichkeit, die Wissenschaft zu verlassen, mit jedem Jahr – $\text{Probability is } (0.05 * (\text{Age} - 45)^2 + 55)$ – relativ schnell an. Die Abbildung 6.1 zeigt die Verteilung.

```

1 check_agent_active(Agent, Age):-
2   Age < 6 *->
3   (
4     random(0,100,Random),
5     Random < 20,
6     set_agent_inactive(Agent)
7   );
8   Age >= 45 *->
9   (
10    Probability is (0.05*(Age-45)^2+55),
11    random(0,100,Random),
12    Random > Probability,
13    set_agent_inactive(Agent)
14   );
15   (
16    Probability is (0.01*Age),
17    random(0,100,Random),
18    Random < Probability,
19    set_agent_inactive(Agent)
20   ).

```

Als nächstes werden neue Agenten erzeugt. Abhängig vom Experiment geschieht dies entsprechend spezifischer Regeln oder zufallsgesteuert:

```

1 construct_agent:-
2   select_agent_type(AgentType),
3   id("AGENT", AgentID),
4   random_simple_name(SimpleName),
5   sim_time(Tick),
6   default_resources(Res),
7   select_random_domain(DomainID),
8   determine_interests(DomainID, Interests),
9   random(0,10,RndAbility), Ability is (RndAbility/10),
10  assertz(agent(AgentType, AgentID, properties{created:Tick, resources:
11    ↪ Res, reputation:1.0, ability:Ability, action_queue:[],
12    ↪ research_domain:[DomainID], list_of_interests: Interests,
13    ↪ simplename:SimpleName})).
14
15 select_agent_type(AgentType):-
16   list_of_types(List),
17   random_member(AgentType, List).

```



```

16 determine_interests(DomainID, Interests):-
17     research_domain(DomainID, EventTypes),
18     length(EventTypes, NumberOfTypes),
19     maxInterest(MaxInterest),
20     NumberInterests is round((NumberOfTypes/100)*MaxInterest)+1,
21     random_sublist(NumberInterests, EventTypes, Interests).

```

5.1.2 Simulationszeit

Je nach Simulationsziel und Experiment können Ticks andere Zeitdimensionen repräsentieren. Dieses Verhältnis zwischen Simulationszeit und den Zeiteinheiten der repräsentierten Systeme wird als *Auflösung* bezeichnet. Wird etwa in der Simulation mit jedem Tick ein Tag repräsentiert, hat diese Simulation eine zeitlich höhere Auflösung, als wenn etwa ein Tick ein Jahr repräsentiert. Bei Experimenten, in denen komplexe Prozesse zwischen Agenten abgebildet werden, ist es sinnvoller eine feinere Auflösung zu wählen, als etwa bei Experimenten, die den Fokus auf gesamtgesellschaftliche Entwicklungen legen.

Parameter	Beschreibung
number_of_agents_at_start/1	Definiert die Zahl der Agenten, welche zu Beginn eines Simulationslaufs unabhängig erzeugt werden.
number_of_facts_at_start/1	Beschreibt die Menge an Fakten, welche zu Beginn eines Simulationslaufs erzeugt werden. Dies ist nur notwendig, wenn die Simulation mit dem Konzept von Fakten arbeitet.
number_of_beliefs_at_start/1	Gibt die Menge von <i>beliefs</i> der Agenten an, die zum Start erzeugt werden.
number_of_domains/1	Definiert die Zahl der Fachbereiche, die vorgegeben werden, etwa Kernphysik, Chemie oder neuere Geschichte.
maxWorkingTime/1	Hier wird die maximale Anzahl der Ticks festgelegt, welche ein Agent aktiv sein kann.
maxFounding/1	Gibt die Höchstsumme einer Förderung an, welche vergeben werden kann.
maxInvestment/1	Definiert die Höchstgrenze an Ausgaben, welche Agenten für eine Aktion tätigen können.
maxInterest/1	Gibt den Maximalwert für Interessen innerhalb des Forschungsgebiets an.
default_resources/1	Beschreibt die Ressourcen, welche Agenten bei ihrer Erzeugung erhalten.
action_planning/1	Bestimmt die Strategie, mit welcher Handlungen gewählt werden. Diese kann entweder entsprechend einer vordefinierten <i>strategy</i> oder <i>random</i> erfolgen.

Tabelle 5.1: Agentenkonfiguration

Die Simulation ist aus technischen Gesichtspunkten zunächst nicht sonderlich komplex. Die Iteration über die Schleifen und die Interaktion zwischen den Agenten bietet keine größeren Herausforderungen. Jeder Agent bekommt entsprechend der Parameter die Möglichkeit Aktionen zu bestimmten Zeitpunkten auszuführen. Simulationsläufe werden über Parameter in `configuration.pl` gesetzt. Die Tabelle 5.1 gibt einen Überblick über die Parameter und eine kurze Beschreibung.

Die Konfiguration der Simulationsläufe erfolgt über die in Tabelle 5.2 beschriebenen Prädikate.

Parameter	Beschreibung
<code>number_of_ticks/1</code>	Hier wird die Zahl der Ticks definiert, welche ein Simulationslauf durchläuft.
<code>number_of_runs/1</code>	Legt die Zahl der Simulationsläufe pro Aufruf fest.
<code>output_filename/1</code>	Definiert den Namen für eine Ausgabedatei.
<code>output_subdirectory/1</code>	Bestimmt das Verzeichnis, in dem die Ergebnisse gespeichert werden.

Tabelle 5.2: Simulationskonfiguration

5.1.3 Die Planung von Aktionen

Für die Planung der Aktionen der Agenten gibt es zwei Optionen, welche über `action_planning(random)` zufällig gewählt bzw. über die Einstellung `action_planning` \leftrightarrow (`strategy`) einer Strategie folgen. Werden die Aktionen zufällig gewählt, wird bestimmt, welche Aktionen der jeweilige Agent auf Grundlage seiner Eigenschaften ausführen kann. Danach wird aus dieser Liste eine zufällige Aktion gewählt und dem Agenten dann in die Queue eingetragen. Im nächsten Tick wird der Agent dann diese Aktion ausführen.

```

1 plan_random_action_for_agent(AgentID):-
2   find_valid_actions_for_agent(AgentID, List_Of_Actions),
3   random_member(Action, List_Of_Actions),
4   sim_time(Tick),
5   ActionContent = action(Action, AgentID, Tick),
6   agent(Type, AgentID, Attributes),
7   ActionQueue = Attributes.get(action_queue),
8   append([ActionContent], ActionQueue, NewActionQueue),
9   retract( agent(Type, AgentID, Attributes)),
10  assertz( agent(Type, AgentID, Attributes.put(action_queue,
    ↪ NewActionQueue))).

```

Wenn die Aktionen entsprechend einer Strategie geplant werden, ist der Ablauf komplexer. Entsprechend der gesetzten Strategie planen die Agenten ihre nächsten Handlungen, um ihr Ziel zu erreichen. Die Abbildung 5.5 verdeutlicht den entsprechenden Entscheidungsbaum, welcher die Handlungen für forschende Agenten bestimmt. Jede Aktion verlangt nach entsprechenden Ausgangsbedingungen, wie Ressourcen oder

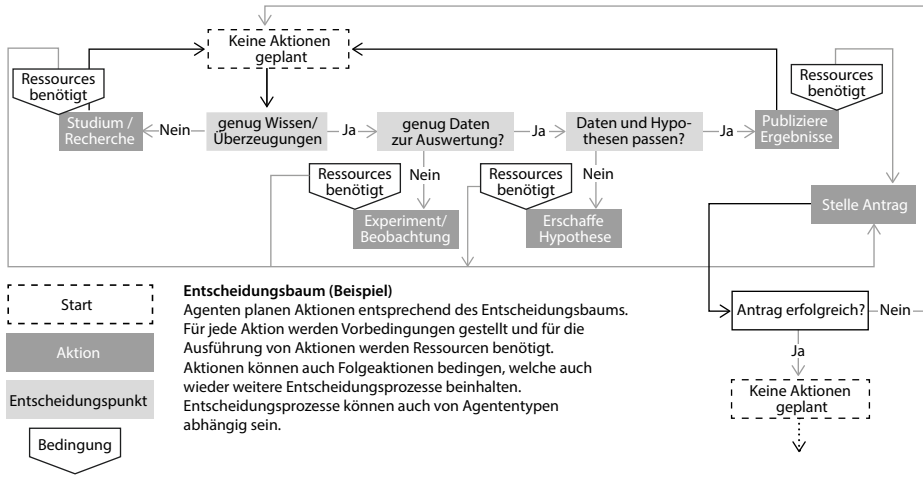


Abbildung 5.5: Aktionen und Ressourcen

ausreichendem Wissen. Sind keine Aktionen geplant, stehen die Agenten vor der Wahl sich weiteres Wissen anzueignen oder bei hinreichendem Wissen Daten zu erheben, diese auszuwerten, Hypothesen aufzustellen oder diese zu belegen und zu publizieren (siehe Kapitel 3.4.1). Diese Aktionen benötigen Ressourcen, welche eingeworben werden müssen (siehe Kapitel 4.1).

5.1.4 Eindeutige Namen durch UUIDs

Die Eindeutigkeit von Namen ist wichtig, um etwa Agenten oder Ereignisse untereinander unterscheiden zu können und Überschneidungen und falsche Zuordnungen ausschließen zu können. Um in der Simulation eine Eindeutigkeit von Namen zu gewährleisten, werden Universally Unique Identifier (UUID)⁵ verwendet. Eine Wahrscheinlichkeit einer Überschneidung von UUIDs ist vernachlässigbar gering. An allen Stellen, an denen das System automatisch Objekte (Agenten, Prädikate für Objekte oder Eigenschaften, Ausgabedateien, etc.) erzeugt, werden UUIDs als Bezeichner verwendet. Ein UUID ist ein 16-bit hexadezimal-Wert. Prolog liefert das Prädikat `uuid(UUID)` zur Erzeugung einer UUID in der Bibliothek `library(uuid)`⁶. Zur Erzeugung von Zufallszahlen wird die Standardimplementierung `uuid(random(Number))` von SWI-Prolog verwendet. Diese basiert auf dem Mersenne-Twister-Verfahren und bietet eine hinreichende Qualität.⁷ In einigen Beispielen werden die *seeds* zur Ge-

5 Die genaue Spezifikation ist hier zu finden: <https://tools.ietf.org/html/rfc4122> (zuletzt aufgerufen am 01.06.2020).

6 Siehe <http://www.swi-prolog.org/pldoc/man?section=uuid> (zuletzt aufgerufen am 01.06.2020).

7 [http://www.swi-prolog.org/pldoc/doc_for?object=f\(random/1\)](http://www.swi-prolog.org/pldoc/doc_for?object=f(random/1)) (zuletzt aufgerufen am 01.06.2020).

Parameter	Beschreibung
<code>number_of_events/1</code>	Bestimmt die Zahl der Ereignisse zum Simulationsstart.
<code>number_of_properties/1</code>	Beschreibt die Zahl der Eigenschaften, welche Objekte tragen können (für spezifische Experimente).
<code>number_of_objects/1</code>	Legt die Zahl von Objekten im Universum fest (für spezifische Experimente).
<code>number_of_laws/1</code>	Definiert die Zahl von Gesetzen, welche bei jedem Simulationslauf generiert werden.
<code>number_of_followup_laws/1</code>	Beschreibt die Zahl von Gesetzen, welche auf den Zielzustand anderer Gesetze aufbauen.
<code>number_of_types/1</code>	Dieser Wert bestimmt die Zahl der möglichen Ereignistypen.
<code>universe_bounders_2d/2</code>	Die Größe des Universums lässt sich durch die maximale Ausdehnung $\langle x, y \rangle$ angeben.

Tabelle 5.3: Weltengenerierung mit Ereignissen

nerierung der Zufallszahlen mittels `set_random(seed(Seed))` fixiert. Das bedeutet, dass der Zahlenpool, aus dem die Zufallszahlen generiert werden, vorgegeben wird und sich so bei erneuten Durchläufen nicht verändert. Dadurch kann eine Vergleichbarkeit zwischen unterschiedlichen Simulationsläufen mit unterschiedlichen Parametern hergestellt werden. Dies ist in den entsprechenden Beispielen angegeben.

5.2 Generierung der simulierten Umgebung

Die Agenten agieren innerhalb einer dynamischen und zur Laufzeit generierten Umgebung. Es wird eine Welt erzeugt, welche entweder aus abstrakten Ereignissen mit Typen und Relationen oder aus konkreten Objekten mit Eigenschaften und Relationen zueinander besteht. Gesetzmäßigkeiten haben Einfluss auf die Welt und verändern diese. Für die Erzeugung der simulierten Welten und Gesetze stehen folgende Parameter zur Verfügung:

Für die alternative Definition des Universums über Objekte, Eigenschaften und Relationen, stehen dazu noch folgende Parameter bereit:

Parameter	Beschreibung
<code>relation_arity/1</code>	Definiert Zahl der Argumente für Relationen (für spezifische Experimente).
<code>number_of_proto_objects/1</code>	Definiert die Zahl von Objektprototypen beim Simulationsstart (nur für spezifische Experimente).
<code>number_of_object_types/1</code>	Bestimmt die Nummer von Objekttypen (für spezifische Experimente).
<code>number_of_relations/1</code>	Bestimmt die Anzahl von Relationen zum Start eines Simulationslaufs (für spezifische Experimente).
<code>law_complexity/1</code>	Definiert die Komplexität von Gesetzen. Dies kann etwa die Zahl der erhaltenen Objekte und Relationen sein (für spezifische Experimente).
<code>maximal_properties_per_type/1</code>	Definiert die maximale Anzahl von Eigenschaften pro Objekttyp (für spezifische Experimente).

Tabelle 5.4: Weltengenerierung mit Objekten

Eine Beschreibung zum Aufbau der simulierten Welt findet sich in Kapitel 2.3.1. In der Simulation sind folgende Varianten implementiert:

5.2.1 Variante 1a: Ereignisse und Typen

Die Welt besteht aus Ereignissen, die jeweils zu genau einem Zeitpunkt und an einem Ort stattfinden. Jedes Ereignis ist eines gewissen Typs, welche mit der Welt gegeben sind. Erzeugt wird die Welt folgenderweise: Zunächst wird die initiale Konfiguration geladen, welche die Typen definiert, den Aufbau der Gesetze bestimmen und im Allgemeinen definiert, wie sich die Welt zusammensetzt. Dann werden die entsprechenden Typen generiert und eine Zahl von Events für den ersten Start erzeugt. Darauf folgen die Gesetze der simulierten Welt und ein Export von Daten zum Startzustand zur Auswertung der Simulation.

```

1 create_world:-
2     init_config,
3     create_event_types,
4     create_initial_events,
5     create_laws,
6     create_lawtree,
7     export_law_visualisation.
```

Zu jedem Tick wird die Welt aktualisiert. Gesetze finden ihre Anwendung und neue Ereignisse entstehen. Dazu wird der Zustand der Welt zu dem spezifischen Zeitpunkt bestimmt und die Regeln auf diesen Zustand angewandt.

```

1 update_world:-
2     state_of_the_universe(_State),
3     types_in_universe(_Types),
```

```

4   check_laws,
5   remove_old_universe.

```

Nach einem Simulationslauf wird die Welt mit allen Bestandteilen aus dem Speicher entfernt.

```

1  remove_world:-
2      remove_config,
3      remove_event_types,
4      remove_events,
5      remove_listings,
6      remove_laws.

```

Für die Anwendung der Gesetze wird der Zustand der Welt bestimmt und zunächst geprüft, welche Gesetze überhaupt ihre Anwendung finden. Sind die notwendigen Bedingungen für die Anwendung eines Gesetzes gegeben, was bedeutet, dass entsprechende Ereignisse Teil der aktuellen Welt sind, wird das Gesetz ausgeführt und neue Ereignisse für den Zeitpunkt $t + 1$ erzeugt.

```

1
2  check_law(PreLaw):-
3      PreLaw = [LawID, PreCon],
4      sim_time(Tick),
5      types_in_universe(TypesInUniverse, Tick),
6      ( subset(PreCon, TypesInUniverse) ->
7          apply_law(LawID);
8          true).

```

Wenn die Vorbedingungen erfüllt sind, kann das Gesetz zur Ausführung kommen. Dazu wird mittels `apply_law(LawID)` der Zeitpunkt bestimmt, zu dem das Gesetz angewendet wird.

```

1  apply_law(LawID):-
2      sim_time(Tick),
3      law_pre(LawID, Precondition),
4      ( apply_once(LawID, Precondition, Tick), fail; true).

```

Danach werden dann mittels `apply_once(LawID, Precondition, Tick)` die Events im Folgezeitpunkt entsprechend der Beschreibung des Zielzustands erzeugt. Auch wird die Zustandsbeschreibung der aktuellen in der Simulation vorkommenden Typen aktualisiert.

```

1  apply_once(LawID, Precondition, Tick):-
2      types_in_universe(TypesInUniverse, Tick),
3      remove_list(TypesInUniverse, Precondition, NewTypes),
4      retractall( types_in_universe(_,_) ),
5      asserta( types_in_universe(NewTypes, Tick) ),

```

```

6   NewTick is Tick + 1,
7   law_target(LawID, TargetState),
8   foreach( member(Target,TargetState), create_specific_event(Target,
    ↪ NewTick) ).

```

5.2.2 Variante 1b: Ereignisraum

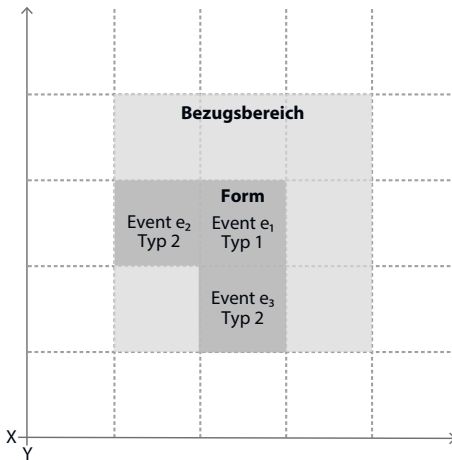


Abbildung 5.6: Abbildung von Ereignissen im Raum

Aufbauend auf dem beschriebenen Aufbau der Welt aus Ereignissen und Typen wird ein mehrdimensionaler Raum geschaffen. Jedes Ereignis erhält neben dem Typen auch eine Position im Raum. In den Simulationsbeispielen ist der Raum zweidimensional $D_n = \langle X, Y \rangle$. Damit findet ein Ereignis in den Dimensionen $\langle x, y \rangle, event_type, tick_n$ statt. Dies ermöglicht mehrere Prozesse: Zum einen können Agenten nun konkrete Beobachtungen im Raum durchführen und zum anderen kann die Örtlichkeit in den Gesetzen der jeweiligen Welt berücksichtigt werden. Die statische Definition der beiden Bezugssysteme *Ort* und *Zeit* widerspricht an dieser Stelle einer relativistischen Interpretation des Raums.

So kann etwa ein Gesetz als Bedingung haben, dass eine örtliche Relation zwischen zwei Ereignissen gewissen Typs zu einem festen Zeitpunkt besteht und als Folge dessen eine andere örtliche Konstellation von Ereignissen im Raum erzeugt. Diese Konstellationen von Ereignissen im Raum werden in der Simulation als *Form* bzw. *shape* bezeichnet. Abbildung 5.6 verdeutlicht den Bezugsbereich der Gesetze und der Form.

Diese Formen sind in der Simulation jedoch auf ein Raster von 9 Feldern begrenzt, wie in der Abbildung dargestellt. Zum einen wird die Komplexität $x * y * |law|$ bei der Berechnung für jeden Tick durchgeführt und somit dann relativ schnell sehr rechenintensiv und bringt hingegen wenig Mehrwerte. Hinzu kommt auch, dass bei der steigenden Komplexität das Universum mit mehr Ereignissen geflutet werden muss, um die Wahrscheinlichkeit für günstige Anfangsbedingungen zu erhöhen. Das steigert ebenfalls die Komplexität und Berechenbarkeit der Simulation, ohne wesentlich an Aussagekraft zu gewinnen. Ein Gesetz beschreibt dann, wenn mehrere Ereignisse bestimmten Typs zu einem Zeitpunkt t_n vorliegen, einen Zielzustand an gleichem Ort zu t_{n+1} .

```

1 create_law:-
2     base_law(LawID, Types, Complexity),
3     random(0, Complexity, ComplexityPre),
4     random(0, Complexity, ComplexityTarget),
5     findall(ElementPre, (between(0, ComplexityPre, C), build_element(
6         ↪ ElementPre)), ElementsPre),
7     assertz(law_pre(LawID, ElementsPre)),
8     findall(ElementTarget, (between(0, ComplexityTarget, C),
9         ↪ build_element(ElementTarget)), ElementsTarget),
10    assertz(law_target(LawID, ElementsTarget)).
11
12 % Returns an element: A location and event type
13 build_element(Element):-
14     list_of_event_types(Types),
15     random_member(Type, Types),
16     random_relative_location(Location),
17     Element = [Type, Location].

```

Da auch gezielt Gesetze erzeugt werden, welche auf dem Zielzustand anderer Gesetze aufbauen, entsteht eine Kette, welche auch zirkulär sein kann. Die Abbildung 2.3 zeigt eine solche Struktur. Die jeweiligen Knoten zeigen einzelne Gesetze und die Kanten die davon jeweils abgeleiteten Gesetze. Es entstehen Kausalketten, welche die Agenten untersuchen können. Das Zusammenspiel der Gesetze in diesen Ketten bedingt die Struktur der *shapes*. Erzeugt wird dies auf folgende Weise:

```

1 % Builds laws, which build upon existing laws
2 lawtree:-
3     base_law(LawID, Types, Complexity),
4     findall([TID, TState], law_target(TID, TState), ChainDomain),
5     random_member(ChainTuple, ChainDomain),
6     ChainTuple = [SourceID, ChainPrecondition],
7     assertz(lawtree(SourceID, LawID)),
8     assertz(law_pre(LawID, ChainPrecondition)),
9     random(0, Complexity, ComplexityTarget),
10    findall(ElementTarget, (between(0, ComplexityTarget, C),
11        ↪ build_element(ElementTarget)), ElementsTarget),
12    assertz(law_target(LawID, ElementsTarget)).

```

Bei jedem Update der Welt wird ein Raster über die gesamten Koordinaten geschoben und durch `check_shapes([C1,C2])` geprüft, an welcher Stelle Gesetze anzuwenden sind. Die Koordinate in der Liste `[C1,C2]` zeigt dabei auf das jeweilige Zentrum eines Rasters. Diese Funktion lässt sich potentiell über Vektoren effektiver berechnen, ist aber durch die verschachtelte Schleife anschaulich und erlaubt es an jeder Stelle einzugreifen und weitere Prozesse einzubinden.


```

1 update_universe:-
2   universe_bounders_2d(BounderX, BounderY),
3   between(1, BounderX, C1),
4   ( between(1, BounderY, C2),
5     ( thread_create_in_pool(threadpool,
6       check_shapes([C1,C2]), _, []) )
7   ),fail;true.

```

5.2.3 Variante 2: Objekte, Relationen und Eigenschaften

Objekte, Eigenschaften und Relationen können transformiert werden. Dies geschieht in dem Modul `create_world.pl`. Bei der Erzeugung der Welt werden Eigenschaften mittels `create_property(Property)` erzeugt, welche Objekte tragen können. Die Objekttypen werden durch jeweils spezifische Mengen an Eigenschaften durch das Prädikat `create_object_type(Type)` definiert. Mithilfe der Objekttypen werden durch `create_object_prototype(ObjectPrototype)` Prototypen erzeugt. Prototypen lassen sich als Modell verstehen. Sie beschreiben, wie das Objekt konstruiert ist und geben den Bauplan für konkrete Objekte, welche mittels `create_object(PrototypID, ↪ Object)` erzeugt werden. Es entsteht ein konkretes Objekt mit folgender Form: `object(ObjectID, Properties)`. Dabei ist `ObjectID` ein eindeutiger Bezeichner und `Properties` eine Liste von konkreten Eigenschaften, welche von den Agenten etwa über Beobachtungen erfahren werden können.

5.3 Aktionen der Agenten im Detail

In den Simulationsexperimenten stehen den Agenten die in diesem Abschnitt näher beschriebenen Aktionen zur Verfügung. Diese werden entsprechend der in den Abschnitten 5.1.3 und 5.3 skizzierten Abläufe geplant, um die jeweiligen Ziele und Absichten zu erfüllen. Die Aktionen sind unabhängig und möglichst generisch definiert. Somit können diese auch bei dynamischen Strategien in unterschiedlichen Konstellationen stattfinden. Hierarchien ergeben sich aber durch die jeweiligen notwendigen Bedingungen. Publikationen können etwa erst gelesen werden, wenn auch solche geschrieben wurden. Theorien lassen sich erst formulieren, wenn Daten analysiert sind, wozu Daten vorhanden sein müssen. Diese Steuerung wird allerdings nicht in den Aktionen vorgenommen, sondern in der Aktionsplanung. Diese beschreibt die Wege, wie die Agenten entsprechend des BDI-Konzepts die Ziele über die jeweiligen Einzelaktionen erfüllen können. Die Aktionen bzw. Handlungen bauen dafür auf den in Kapitel 3 und Kapitel 4 beschriebenen Prozessen auf, sind allerdings dabei stark idealisiert und insbesondere für die in Kapitel 6 beschriebenen Beispiele angepasst. Durch diese Aktionen werden die in Tabelle 5.5 beschriebenen Prädikate erzeugt, welche dann in Kapitel 6 zur Auswertung betrachtet werden.

Prädikat	Beschreibung
<code>data(DataID, AgentID, Tick,</code> \hookrightarrow <code>DataType, BaseType, BaseLocation</code> \hookrightarrow <code>, DataContent)</code>	Agenten speichern in <code>data/7</code> ihre Beobachtungen. Diese stehen keinen anderen Agenten zur Verfügung und repräsentieren ein Laborbuch oder die persönlichen Notizen. Daten können nach einiger Zeit auch archiviert werden.
<code>theory(CreatorID, TheoryType,</code> \hookrightarrow <code>TheoryID, Link, PotMod, Mod)</code>	Hier wird die Verbindung zwischen <code>pot_model/3</code> \leftrightarrow und <code>model/3</code> hergestellt. Daten können instanziiert werden und sich zu Modellen der Theorie qualifizieren.
<code>pot_model(PotModID, BaseType, Types)</code>	Das potentielle Modell M_p beschreibt, welche Typen in Relation stehen.
<code>model(ModID, PotModID, Relations)</code>	Die eigentlichen Relationen der Theorie werden im Modell beschrieben. Die Relationen sind durch <i>shapes</i> beschrieben, spezifische Konstellationen von Ereignissen, Typen zu bestimmten Orten in zeitlicher Abfolge.
<code>citation(Author, CitedAuthor,</code> \hookrightarrow <code>CitedPublication)</code>	Die Angabe einer Zitation einer Publikation als Referenz zur Analyse.
<code>belief(AgentID, Tick, Subject,</code> \hookrightarrow <code>Content, Domain, Str)</code>	Die Überzeugungsbasis der Agenten wird durch <code>belief/6</code> ausgedrückt. Ein Agent <code>AgentID</code> erlangt zum Zeitpunkt <code>Tick</code> die Überzeugung <code>Content</code> . Diese Überzeugung kann einem <code>Domain</code> und spezifischen <code>Subject</code> zugeordnet werden, muss aber nicht. Die Stärke der Überzeugung wird durch die Quelle determiniert. Selbst beobachtete Ereignisse führen zu einer starken Überzeugung, Messergebnisse oder Beobachtungen aus Publikationen von Agenten mit niedriger Reputation führen hingegen zu schwächeren Überzeugungen.
<code>publication(PublicationID, Journal,</code> \hookrightarrow <code>ListOfAuthors, Tick, Type, Content)</code>	Die Publikation ist ein zentrales Kommunikationsmittel zwischen den Agenten. Diese werden von Agenten verfasst und enthalten etwa Theorien oder Beobachtungen. Andere Agenten können diese Publikationen lesen und bewerten. <code>Type</code> beschreibt, ob die Publikation eine Theorie oder Daten enthält und <code>Content</code> ist eine Liste der Inhalte, etwa die IDs von Theorien.

Tabelle 5.5: Durch Aktionen erzeugte Prädikate

Aktion: Planung und Durchführung von Handlungen

Die Planung und Durchführung von Handlungen ist ein zusammenhängendes Konstrukt von einzelnen Aktionen:

Definition 5.1. Handlungen der Agenten

1. Eine Handlung beschreibt die Interaktion der Agenten mit der simulierten Umgebung und anderen Agenten. Den Agenten stehen dazu die diesem Kapitel beschriebenen Handlungen zur Verfügung. Diese bestehen zum Teil aus einer Kombination mehrerer Einzelaktionen. Beispielsweise besteht ein Gutachten zur Förderung oder Publikation aus der Anfertigung des zu begutachtenden Antrags oder

der Publikation, der Auswahl der Gutachter, der Sammlung und Auswertung der einzelnen Entscheidungen.

2. Jeder aktive Agent $\forall x \in \text{list_of_active_agents}$ führt zu jedem Simulationszeitpunkt $t \in T$ im Hauptlauf eine Handlung durch. Dazu wird zu jedem Tick für jeden Agenten `perform_action(AgentID, Queue)` aufgerufen.
3. Die Handlungen werden durch `plan(Agent, Action)` geplant und so in eine `action_queue` eingeliebert.
4. Die `action_queue` wird zu jedem Simulationszeitpunkt aufgerufen und jeweils die erste Handlung durchgeführt. Dadurch wird die Queue sequenziell abgearbeitet.
5. Wenn `action_queue = \emptyset` ist, dann werden entsprechend der *beliefs* und den Eigenschaften des Agenten neue Handlungen durch `plan(Agent, Action)` geplant.
6. Handlungen können aus einer Verkettung mehrerer Einzelhandlungen bestehen. Ein Beispiel ist etwa eine Versuchsreihe, welche aus einer Reihe von Einzelversuchen über mehrere Ticks stattfindet.
7. Je nach Definition können Handlungen Ressourcen verbrauchen. Diese werden dem Agenten bei der Durchführung der Handlung durch `spend_resources(Agent \leftrightarrow ID, Price)` von dem Kontigent durch `resources` abgezogen.
8. Hat der Agent nicht die notwendigen Ressourcen zur Durchführung der Handlung, versucht der Agent neue Ressourcen durch die Aktion `execute_action(\leftrightarrow funding, AgentID, Options)` einzuwerben.
9. Es gibt generische Handlungen, welche alle Typen von Agenten durchführen können und spezifische Handlungen, die auf eine Teilmenge der Typen beschränkt ist. Die Steuerung erfolgt über `action_type(action, [agent_type, (...)])`.

Die Planung weiterer Handlungen auf Grundlage der persönlichen Eignung und insbesondere der *beliefs* ist eine zentrale Handlung, welche allen Agententypen zur Verfügung steht. Je nach Strategie, wie in Kapitel 5.1.3 beschrieben, verfahren die Agenten jedoch unterschiedlich in der Planung.

Definition 5.2. Planung von Handlungen anhand einer Strategie (BDI)

1. Zu jedem Typ von Agenten existiert ein spezifisches Ziel, welches der Agent versucht durch einzelne Aktionen (Intentionen) zu erfüllen. In den Simulationsbeispielen sind dies Studierende, Forschende und Fördernden. Die Planung erfolgt über `strategic_action(AgentID, AgentType)`. Konkret werden dann die einzelnen Aktionen über `plan(Action, AgentID)` in die Queue eingeführt.
2. Ziel der Studierenden ist es *beliefs* aufzubauen, welche in einer potentiellen weiteren Tätigkeit als Forschende notwendige Grundlage sind.

3. Die Forschenden haben zum Ziel neue Theorien zu publizieren und dabei die Reputation zu erhöhen. Dazu forschen sie, lesen relevante Publikationen und werben Fördermittel ein.
4. Ziel der Förderer ist es, vielversprechende Forschung mit Ressourcen auszustatten. Als Bewertungsgrundlage lesen Förderer aktuelle Publikationen und erweitern so ihre *beliefs* zum aktuellen Stand der Forschung. Je nach Experiment können hier weitere Vorgaben (etwa durch ein Bias zu spezifischen Gebieten) hinzukommen.
5. Grundlage zur Wahl der einzelnen Intentionen sind *beliefs*. Diese enthalten Referenzen zu Daten und Theorien. Die Entscheidungsgrundlagen werden in `plan(↪ Action, AgentID)` definiert.

Konkret ermöglicht das Prädikat `strategic_action(AgentID, AgentType)` jedem Agententypen eine eigene Strategie zu verfolgen. Alle Agenten müssen Ressourcen einsetzen, um die eigenen Ziele zu erfüllen. Diese Ressourcen können durch Förderungen (etwa Stipendien, Nebentätigkeiten oder Drittmittel) eingeworben werden (siehe Beispiel in Abbildung 5.5). Wenn definierte Bedingungen für eine Aktion in `plan(Action, AgentID)` getriggert werden, wird diese Aktion in die Action Queue des Agenten geschrieben. Diese Liste enthält alle geplanten Aktionen und wird sequenziell abgearbeitet. Diese Liste kann auch verworfen werden, etwa wenn die Ressourcen zur Neige gehen oder der Agent seinen Fokus ändert. Ist die Liste leer, so wird der Agent aufgefordert neue Aktionen zu planen. Ein Beispiel ist etwa folgende Strategie, in der Agenten vom Typ `researcher` als Ziel haben, eine Theorie zu publizieren. Dazu führen sie zunächst Experimente durch, lesen Literatur im Kontext und versuchen dann auf dieser Grundlage eine Theorie zu formulieren. Wenn dies gelingt, wird diese publiziert:

```

1 strategic_action(AgentID, researcher):-
2     sufficient_bel_for_publish(AgentID)*->
3     plan(publish, AgentID);
4     ( plan(make_theory, AgentID),
5       plan(read_related_publication, AgentID),
6         plan(experiment_series, AgentID)).

```

Die jeweiligen Aktionen können selbst noch weitere Aktionen mit sich führen. So besteht eine `experiment_series` aus einer Reihe von Einzelbeobachtungen. Die Ergebnisse der Einzelbeobachtungen können selbst aber etwa auch als Daten publiziert werden.

Aktion: Zwei Varianten der Theoriebildung

Die Aktion `execute_action(make_theory ,AgentID, _Options)` ermöglicht es dem Agenten eine Theorie im Sinne von der in Kapitel 2 beschriebenen Form zu formulieren. Diese Theorien sind wie folgend aufgebaut:

Definition 5.3. Simulierte Theorien

1. Eine Theorie T hat in der Simulation die Form: `theory(CreatorID, TheoryType, ↪ TheoryID, Link, PotMod, Mod)`.
2. Ein Modell für eine solche Theorie, muss zum einen das `pot_model(PotModID, ↪ BaseType, Types)` erfüllen und zum anderen die konkreten Anforderungen aus `model(ModID, PotModID, Relations)`.
3. `Link` beschreibt die Beziehung zu anderen Theorien und kann entsprechend eine Liste von Verweisen (IDs) zu anderen Theorien enthalten.
4. `pot_model(PotModID, BaseType, Types)` definiert die Typen, welche Teil der Theorie sind.
 - a) `BaseType` beschreibt einen Objekttyp, zu welchem die Theorie eine Aussage trifft.
 - b) `Types` ist eine Liste aller weiteren in der Theorie vorkommenden Eventtypen, welche in einer Beziehung stehen.
5. `model(ModID, PotModID, Relations)` gibt die konkreten Relationen an, welche zwischen dem `BaseType` und `Types` besteht. Diese Beziehung wird über `shapes` ausgedrückt.
6. Jede Theorie T ist genau einem durch `TheoryType` bezeichnetem Typ zugeordnet.
7. Durch `CreatorID` wird der Agent referenziert, welcher die Theorie verfasst hat.
8. `TheoryID` ist ein eindeutiger Bezeichner der Theorie und dient als Referenz innerhalb der Simulation.

In der Simulation gibt es zwei Typen von Theorien: Theorien vom ersten Typ beschreiben eine Relation zwischen zeitgleich stattfindenden Ereignistypen.

Definition 5.4. Theoriebildung vom Typ 1

1. Ziel ist es, eine Relation der folgenden Form in den Beobachtungen zu identifizieren: Für alle Paare von Ereignissen x_1, x_2 mit Ereignistypen p, q gibt es eine Relation R , so dass gilt: $\forall x_1 \forall x_2 \exists R : Event_{x_1}^p \wedge Event_{x_2}^q : R(x_1, x_2)$
2. Es wird ein `BaseType` bestimmt, welcher in der Liste der untersuchten Eventtypen des Agenten liegen: `BaseType ⊆ Interests`
3. Zu dem `BaseType` gibt es eine nicht leere Liste von `DataObjects`, welche der Agent durch Beobachtungen oder Literatur erlangt und beschreibt, welche Typen von Ereignissen im Zusammenhang mit dem `BaseType` beobachtet wurden.
4. Die Beobachtungsdaten enthalten eine Liste von zeitgleichen Nachbarereignissen zu dem gewählten `BaseType`. Es werden die relative Position und der Ereignistyp erfasst.
5. Es wird nach zwei Arten von Kandidaten von Relationen gesucht:

- a) Ereignistypen, welche immer in der Nachbarschaft zum `BaseType` auftreten. Diese liegen in $\text{relatives} = \text{observation}_1 \cap \text{observation}_2 \cap (\dots) \cap \text{observation}_n$
- b) `potential_relatives` liegen in vielen, jedoch nicht im Schnitt aller Beobachtungen und sind somit vermutlich über die Verbindung mit weiteren Typen und Relation mit dem `BaseType` verbunden: Dazu wird der beste Kandidat aus der Häufigkeitsverteilung der Typen in $\bigcup(\text{observation}_n)$ ausgewählt.

Theorien von Typ 2 ermöglichen es eine Beziehung im Sinne der Ursache und Wirkung herzustellen. Diese Theorien beschreiben die Zustandsveränderung über die Zeit.

Definition 5.5. Theoriebildung vom Typ 2

1. Eine Theorie von Typ 2 beschreibt eine Kausalrelation R durch den Übergang von shape_1 in t_1 zu shape_2 in t_2 : $\forall x \in \text{shape}_1^{t_1} \forall y \in \text{shape}_2^{t_1} : R(x, y)$. Dabei bezeichnen x, y eine Liste von Objekttypen mit einer relativen Position zu einem `BaseType`.
2. Es wird ein `BaseType` bestimmt, welcher in der Liste der untersuchten Eventtypen des Agenten liegen: $\text{BaseType} \subseteq \text{Interests}$
3. Zu dem `BaseType` gibt es eine nicht leere Liste von `DataObjects` zu t_n und t_{n+1} .
4. Es wird in den Beobachtungen zu t_n und $t_n + 1$ nach wiederkehrenden charakteristischen shapes gesucht, die in jedem Beobachtungspaar $\text{observation}_p^{t_n}, \text{observation}_q^{t_n+1}$ liegen. Wird zu jeder $\text{observation}_p^{t_n}$ ein `shape_r` und in der darauf folgenden $\text{observation}_q^{t_n+1}$ ein `shape_s` gefunden, so kann auf $\text{shape}_r^{t_n+1} \rightarrow \text{shape}_s^{t_n+1}$ geschlossen werden.

Konkret wird durch `get_interest(AgentID, BaseType)` zuerst geprüft, welche Typen im Fokus des Agenten liegen. Dieser Eventtype wird als `BaseType` bezeichnet. Es werden dann alle Beobachtungen auf das Vorkommen dieses Typen gefiltert:

```

1 get_observed_events(AgentID, BaseType, DataObjects) :-
2   findall(
3     [BaseType, BaseLocation, DataContent],
4     actions:data(DataID, AgentID, _Tick, _DataType, BaseType, BaseLocation,
5       ↪ DataContent),
6     DataObjects).
```

Ein `DataObject` bündelt die Beobachtungen aus `data/7` zu einer Liste, bestehend aus `[BaseType, BaseLocation, DataContent]`. Den eigentlichen Inhalt enthält `DataContent`, eine Liste von beobachteten Objekten zu einem spezifischen Zeitpunkt und an einem Ort: `fitting_data(AgentID, BaseType, DataObjects)`

Aktion: Die Publikation von Theorien und Daten

Publikationen erlauben es den Agenten Wissen untereinander auszutauschen. Diese haben Theorien oder Beobachtungsdaten zum Inhalt.

Definition 5.6. Publikation

1. Eine Publikation hat die Struktur:
`publication(PublicationID, Journal, ListOfAuthors, Tick, Type, Content)`
2. Sie ist von einem der drei möglichen Typen:
 $Type \in \{data, theory, observed_types\}$.
3. Jede Publikation hat mindestens einen Autor: $ListOfAuthors \neq \emptyset$.
4. `Tick` bezeichnet den Zeitpunkt, zu dem die Publikation publiziert wurde.
5. `Journal` bezeichnet eine Publikationsplattform, etwa eine Fachzeitschrift, Verlagsreihe oder ein Repositorium für Daten.
6. Der eigentliche Inhalt der Publikation ist eine Liste `Content`. Deren Inhalt kann sein:
 - a) eine nicht leere Liste von Ereignissen (anhand der IDs)
 - b) eine Referenz zu einer Theorie (durch eine Theorie ID)
7. Je nach Simulationsexperiment kann auch eine Publikationsplattform (dies repräsentiert etwa eine Zeitschrift oder einen Verlag) als weiteres Argument angehängt werden.

Um eine Publikation zu erzeugen, wird ein Publikationsprozess durchlaufen: Es wird zunächst eine Menge von `beliefs` mit hinreichender Überzeugung ausgesucht. Dies kann entsprechend einer Strategie erfolgen oder zufällig, wie in diesem Beispiel:

```

1 select_bel_to_publish(AgentID, Theory):-
2     findall(
3         actions:belief(AgentID, Tick, theory, Theory, Domain, Str),
4         (actions:belief(AgentID, Tick, theory, Theory, Domain, Str), writeln
5             ↪ (Str)),
6         BellList),
7     random_member(actions:belief(_, _, theory, Theory, _, _), BellList).
```

Je nach Typ (Datenpublikation, Beobachtung oder Publikation einer Theorie), werden die entsprechenden Daten ausgewählt.

```

1 execute_action(publish, AgentID, _Options):-
2     select_bel_to_publish(AgentID, Theory),
3     get_observed_events(AgentID, BaseType, DataObjects),
4     (...)
```

Wenn eine Theorie publiziert wird, werden in jedem Fall die zugrunde liegenden Daten auch publiziert.

```

1  publish([AgentID], theory, Theory),
2  publish([AgentID], data, DataObjects).

```

Werden Beobachtungen zum Vorkommen von Ereignistypen publiziert, so werden diese direkt aus den Beobachtungsdaten entnommen und gesondert publiziert:

```

1  execute_action(publish_types, AgentID, _Options):-
2  findall(Observation, actions:data(DataID, AgentID, Tick, ContentType,
3  _, _, Observation), ObservedTypes),
4  publish([AgentID], observed_types, ObservedTypes).

```

Für die Publikation kann auch ein Review-Verfahren notwendig sein:

Definition 5.7. Peer-Review

1. Ein Review-Verfahren besteht aus einer Menge von Gutachtenden, den Autoren, den zugeordneten Überzeugungen und Daten sowie einer potentiellen Publikation (dem Entwurf bzw. Draft).
2. Die Gruppe der Reviewer besteht aus Forschenden, welche eine Überschneidung in den Forschungsgebieten zu den Autoren aufweisen: $\text{intersection}(\text{RevDomain}, \hookrightarrow \text{AuthorDomain}, \text{Intersection})$
3. Die Reviewer beurteilen die Publikation anhand der *soundness*. Dieser Wert ergibt sich aus einer Überschneidung von in der Publikation vorkommenden *BaseTypes* \hookrightarrow und den entsprechenden Vorkommen in den *beliefs* der Gutachter. Je mehr Überzeugungen sie zu diesem Typen haben, desto besser bewerten sie das Vorhaben. Die Bewertung unterteilt sich in zwei Varianten:
 - a) In einem Blindverfahren sind den Gutachtenden die Autoren nicht bekannt. Entsprechend werden *reputation* und *abiliy* nicht berücksichtigt, sondern nur *soundness*.
 - b) Bei einem Non-Blind-Verfahren, wird die Eignung der Autoren *ability* und Reputation *reputation* berücksichtigt und zusammen mit *soundness* zu einer Gesamtbewertung verrechnet:
$$\text{review} = \text{reputation} + \text{abiliy} + \text{soundness}$$
4. Ein Schwellwert *threshold* definiert die Hürde zur Akzeptanz der Publikation. Liegt $\text{review} > \text{threshold}$ wird das Draft angenommen, andernfalls abgelehnt.

Aktion: Publikationen lesen

Zum Lesen von Publikationen werden Publikationen gesucht, welche im Interessensgebiet des Akteurs liegen. Aus der Menge wird dann eine Publikation ausgewählt. Dies kann entweder zufällig geschehen oder anhand von Merkmalen, wie etwa nach Reputation des Autors. Ist die Publikation ausgewählt, werden die darin beschriebenen Inhalte als *belief* aufgenommen. Dieser *belief* enthält zunächst eine neutra-

le Bewertung (0.5) und kann etwa durch die Validierung noch gestärkt oder abgeschwächt werden. Sollte es keine Publikation zu dem gewählten Thema geben, wird als *belief* hinterlegt, dass dieses Thema noch nicht erforscht ist. Dieser Prozess wird folgendermaßen gesteuert:

```

1 execute_action(read_publication, AgentID, _Options):-
2   get_interest(AgentID, Interest),
3   find_related_publications(Interest, PublicationsAuthors),
4   PublicationsAuthors \= [] ->
5   (select_publication(PublicationsAuthors, Publication),
6    update_belief_with_publication(publication, AgentID, Publication));
7   no_publication(AgentID, Interest).

```

Aktion: Antragsverfahren für Drittmittel

Die Beantragung von Drittmitteln ist in der Simulation eine der komplexeren Aktionen. Alle Agenten benötigen Ressourcen für die meisten Aktionen. Sind diese verbraucht, müssen neue eingeworben werden. Kapitel 4.1 skizziert die Funktion von Drittmitteln in der Forschung und Kapitel 3.4.3 beschreibt die Gutachterverfahren, welche in der Vergabe von Fördermitteln angewendet werden. Der Effekt von Drittmitteln auf die Forschung soll in der Simulation über das Zusammenspiel der Ressourcen, Aktionen und Förderung dargestellt werden. Für eine Förderung wird durch die Agenten ein Antrag geschrieben, durch Gutachter geprüft und auf dieser Grundlage dann über die Förderung entschieden. Dies hat nicht nur Auswirkungen auf die Ressourcen, sondern kann auch einen Effekt auf die Reputation haben. Ein Antrag als Grundlage einer Förderung hat folgende Form:

Definition 5.8. Antrag

1. Ein Antrag ist eine Liste der Form: $\text{Proposal} = [\text{Subject}, \text{Domain}, \text{Quality}]$.
2. *Subject* ergibt sich aus der Liste der Interessen des Agenten. Konkret ist dies eine Liste von Ereignistypen, mit dem sich der Agent beschäftigt.
3. *Domain* kann als Fachbereich verstanden werden. Dies ist eine Menge von Ereignistypen, wobei gilt: $\text{Subject} \subset \text{Domain}$.
4. Die Qualität des Antrags ergibt sich in der Simulation aus der individuellen Befähigung *Ability* und der Reputation des Agenten: $\text{Quality is } (\text{Ability} + \leftrightarrow \text{Reputation}) / \text{WeightingFactor}$. Die Reputation ergibt sich auch aus simulierten wissenschaftlichen Leistungen und ist somit auch als Faktor berücksichtigt.

Die Aktion ist dann folgenderweise aufgebaut:

```

1 execute_action(funding, AgentID, _Options):-
2   write_proposal(AgentID, Proposal),

```

```

3   review(AgentID, Proposal, Result),
4   Result = "funding_recommended" ->
5   execute_funding(AgentID);
6   reject_funding(AgentID).

```

Das Gutachtendenverfahren ähnelt dem der Publikation: Eine Gruppe von Gutachtern bewerten den Antrag anhand von Kriterien. Dabei können auch Eigenschaften des Antragsstellenden berücksichtigt werden. Hinzu kommen zwei zentrale Faktoren für die Bewertung:

1. `originality(AgentID, Subject, OriginalityScore)`: Die Originalität beschreibt, wie viele Publikationen zu den in `Subject` genannten Typen im Verhältnis zu anderen Publikationen existieren:

$$Originality = WeightingFactor - \frac{NumberRelatedPublications}{NumberOfAllPublications} \times 100.$$
 Ein niedriger Wert ist für die Beantragung entsprechend vorteilhaft.

2. `familiarity(AgentID, Subject, FamiliarityScore)`: Hat der Agent selbst bereits in dem Bereich veröffentlicht, in dem der Antrag geplant ist, so ist dies ebenfalls vorteilhaft, da er Erfahrungen auf dem Gebiet vorweisen kann. Der Wert ergibt sich aus der Menge der relevanten Publikationen und wird durch einen Faktor gewichtet, um in Relation zu den anderen Kriterien zu stehen:

$$FamiliarityScore = \frac{RelatedPublications}{WeightingFactor}$$

Wenn es Gutachter gibt, welche sich mit den in `Subject` genannten Themen beschäftigen haben, werden diese über den Antrag entscheiden. Die Gutachter vergleichen die Liste der in `Subject` genannten Typen mit den eigenen. Wenn sich eine Schnittmenge ergibt, stimmen diese einer Förderung zu.

```

1 review(AgentID, Proposal, Result):-
2   Proposal = [Subject, Domain, Quality],
3   select_reviewer(Domain, ListOfReviewerIDs)->
4   conduct_review(blind, AgentID, Proposal, ListOfReviewerIDs, Result
5   ↪ );
6   new_subject(blind, AgentID, Proposal, ListOfReviewerIDs, Result).

```

Sollte es keine Gutachter zu dem beantragten *subject* geben, wird das Projekt als neuartig betrachtet und die Begutachtung erfolgt nur aufgrund der beschriebenen Parameter des Antrags. Das Ergebnis der Bewertung des Antrags und der Empfehlungen der Gutachter wird zu einer Gesamtwertung verrechnet. Liegt diese über einen gesetzten Schwellenwert, erfolgt eine Förderung. Andernfalls wird das Vorhaben abgelehnt:

```

1   Result is ((SumResults/LenResults*Quality)+OriginalityScore)+
2   ↪ FamiliarityScore,
3   Result>Threshold ->
4   Result = "funding_recommended" ;

```

```
4 Result = "funding_rejected".
```

Daraus ergibt sich folgendes Gutachterverfahren für die Beantragung von Drittmittel mittels eines Antrags:

Definition 5.9. Gutachterverfahren

1. Ein Review-Verfahren besteht aus einer Menge von Gutachtenden, den Antragstellern, den jeweiligen Überzeugungen und Daten und dem eigentlichen Antrag und der Qualität des Antrags.
2. Die Gruppe der Reviewer besteht aus Gutachtenden, welche eine Überschneidung in Interessensgebieten und Fachbereich zu den Autoren aufweisen.
3. Die Qualität des Antrags hängt von der Befähigung und dem bisherigen Erfolg des Antrags ab: $Quality = \frac{(Ability+Reputation)}{WeightingFactor}$
4. Die Reviewer beurteilen den Antrag anhand des OriginalityScores, der familiarity, und je nach Verfahren, weiterer Faktoren:
 - a) Analog zu den Gutachterverfahren der Publikation sind den Gutachtenden in einem Blindverfahren die Autoren nicht bekannt. Entsprechend werden reputation und ability nicht berücksichtigt.
 - b) Bei einem Non-Blind-Verfahren werden zusätzlich zu den Einzelbewertungen der Gutachter, die Eignung der Autoren ability und Reputation reputation berücksichtigt und zu einer Gesamtbewertung verrechnet:

$$Review = (\sum(Einzelbewertung) * \frac{Einzelbewertung}{|Einzelbewertungen|}) * Quality + OriginalityScores + Familiarity$$
5. Ein Schwellwert threshold definiert die Hürde zur Akzeptanz der Publikation. Liegt $review > threshold$ wird das Draft angenommen, andernfalls abgelehnt.

Aktion: Beobachtung von Ereignistypen

In der Simulation haben die Agenten die Möglichkeit zwei verschiedene Arten von Beobachtungen durchzuführen: Zum einen können die Agenten Ereignistypen und deren Umgebung zu einem spezifischen Moment beobachten und zum anderen können sie Beobachtungsreihen über die Zeit durchführen. Im ersten Fall wählt der Agent einen Ort aus und beobachtet, welche Ereignisse an diesem Ort zu diesem Moment stattfinden und welche Ereignisse in der Nähe geschehen. Dies kann der Agent dann als Einzelbeobachtung publizieren und in einer Analyse etwa Relationen zwischen Ereignistypen feststellen.

```
1 execute_action(observation_types, AgentID, _Options):-
2   agent(Type, AgentID, Dict),
3   sim_time(Tick),
4   events_in_tick(Tick) *->
```

```

5   (   observe_location(Tick, PossibleObservation),
6       extract_types(PossibleObservation, Types),
7       analyse_observation(Dict, Types, Observation),
8       publish_observation_types(AgentID, Tick, Observation));
9   (...)
```

Die eigentliche Beobachtung stellt eine Liste aller Nachbarn zu einem Event des untersuchten Typs aus:

```

1   observe_location(Tick, PossibleObservation):-
2       findall(EID, create_world_module:event(EID, _,Tick,_), EIDs)*->
3       (   random_member(EventID,EIDs),
4           neighborevents(EventID,PossibleObservation) );
5       (...)
```

Aktion: Beobachtung von Ereignissen

Eine Beobachtungsreihe findet zu mindestens zwei aufeinander folgenden Zeitpunkten statt. Dabei werden die Veränderungen an einem Ort und seiner Umgebung untersucht. Bei wiederholten Beobachtungen von spezifischen Konstellationen von Ereignissen können die Agenten so auf Muster schließen und darauf aufbauend Hypothesen bilden. Da diese Aktion über mehrere Zeitschritte erfolgt, wird zunächst geprüft, ob es sich um eine neue Beobachtung handelt oder diese ein Teil einer Reihe ist:

```

1   execute_action(observation_experiment, AgentID, _Options):-
2       agent(Type, AgentID, Attributes),
3       sim_time(Tick),
4       (ongoing_experiment(AgentID, Tick)*->
5           continue_observation(AgentID, ListOfBaseEvents);
6           make_new_observation(AgentID, ListOfBaseEvents)),
7       observation(AgentID, Tick, ListOfBaseEvents).
```

Handelt es sich um eine Fortsetzung sammelt das Prädikat `continue_observation` \leftrightarrow (AgentID, ListOfEventIDs) die bisherigen Beobachtungen aus t_{n-1} und nutzt diese dann als Referenz, um wieder anzuknüpfen.

Andernfalls wird mittels `make_new_observation(AgentID, Observation)` bestimmt, welcher Ereignistyp im Interesse des Agenten liegt und eine Liste der aktuell entsprechenden beobachtbaren Ereignisse aufgebaut. Auf dieser Grundlage findet dann die eigentliche Beobachtung statt. Abhängig von der Befähigung `Ability` des Agenten können mehr oder weniger Objekte wahrgenommen werden:

```

1   observation(AgentID, Tick, ListOfBaseEvents):-
2       agent(_, AgentID, Attributes ),
3       Ability is Attributes.get(ability),
```

```

4     length(ListOfBaseEvents, ListOfBaseID ),
5     Scope is round(ListOfBaseID*Ability),
6     random_sublist(Scope, ListOfBaseEvents, Observation),
7     (...)
8     observe_event(AgentID, Event, Tick)
9     (...)

```

Handelt es sich um eine neue Beobachtung, so wird zunächst ein zu untersuchender Eventtype aus `interests` gewählt und dann alle potentiellen beobachtbaren Ereignisse zu dem aktuellen Tick t gesammelt:

```

1 make_new_observation(AgentID, Observation):-
2     sim_time(Tick),
3     get_interest(AgentID, EventType),
4     findall(ID, create_world_module:event(ID, EventType, Tick, Location),
5             ↪ Observation).

```

Die konkrete Beobachtung ist dann eine Teilmenge der beobachtbaren Ereignisse und abhängig von der Befähigung des Agenten:

```

1 observation(AgentID, Tick, ListOfBaseEvents):-
2     agent(_, AgentID, Attributes ),
3     Ability is Attributes.get(ability),
4     length(ListOfBaseEvents, ListOfBaseID ),
5     Scope is round(ListOfBaseID*Ability),
6     %( RndScope > 400 *-> Scope is 400; Scope is RndScope ),
7     random_sublist(Scope, ListOfBaseEvents, Observation),
8     cores(NumberOfCores),
9     findall(Goal, observation_concurrent(Goal, Observation, AgentID, Event,
10     ↪ Tick), Goals),
10    concurrent(NumberOfCores, Goals, []).

```

Nachdem die Liste der zu beobachtbaren Events festgelegt ist, werden diese mitsamt der Nachbarelemente beobachtet und die Beobachtung als Daten gespeichert. Zu jedem Datensatz wird ein `belief` erzeugt. Da diese Daten aus der eigenen Beobachtung stammen, ist der Agent sehr überzeugt von der Richtigkeit der Beobachtung und bewertet diese mit 1.0.

```

1 observe_event(AgentID, Event, Tick):-
2     neighborevents(Event, Observation),
3     create_world_module:event(Event, BaseType, Tick, Location),
4     ContentType = observation_data,
5     uuid(DataID),
6     assertz(actions:belief(AgentID, Tick, ContentType, DataID, Domain, 1.0)
7     ↪ ),
7     assertz(actions:data(DataID, AgentID, Tick, ContentType, BaseType,
8     ↪ Location, Observation)),

```

8 (...)

Wird eine Beobachtungsreihe fortgesetzt, so wird wieder die gleiche Stelle beobachtet wie zu t_{n-1} :

```

1 continue_observation(AgentID, ListOfEventIDs):-
2   msg_with_simple_name(AgentID, 'action: continue_observation'),
3   sim_time(Tick), PTick is Tick -1,
4   actions:data(DataID, AgentID, PTick, _ContentType, BaseType, Location,
5     ↪ _Observation),
6   findall(ID, create_world_module:event(ID, EventType, PTick, Location),
7     ↪ ListOfEventIDs),
8   addlog(AgentID, subaction, continue_observation,_).

```

Aktion: Wechsel des Interessengebiets

Die Agenten haben die Möglichkeit ihr Interessensgebiet zu verlagern. Dies ist etwa notwendig, wenn sich keine Ereignisse des Interessensgebiets mehr beobachten lassen und es auch keine Literatur dazu gibt. Finden die Agenten also keine Daten mehr zu ihrem Interessensgebiet, so wechseln sie dieses mit der Zeit. Der Grund dafür ist die dynamische Welt: So kann es über die Laufzeit geschehen, dass Regeln zur Erzeugung bestimmter Ereignistypen nicht mehr getriggert werden und so diese Typen nicht mehr zur Beobachtung zur Verfügung stehen. Die Agenten suchen sich dann einen anderen Schwerpunkt, da sie zumindest nicht mehr empirisch zu dem bisherigen Fokus forschen können. Hier ließe sich natürlich auch eine historische Forschung implementieren, welche dann nur auf Basis von Publikationen forscht. Das ist aber in den Beispielen nicht implementiert.

Der neue Fokus wird aus den Publikationen entnommen. So ist die Wahrscheinlichkeit gegeben, dass der neu gewählte Ereignistyp auch wirklich aktuell in der Welt vorkommt.

```

1 execute_action(select_new_interest, AgentID, _):-
2   msg_with_simple_name(AgentID, 'action: select new type (interest)'),
3   findall(ObservedTypes, (actions:belief(AgentID, _Tick, observed_types,
4     ↪ ObservedTypes, Domain, Str), Str > 0.5),
5     ↪ ListOfInterestingObservations),
6   select_observation(ListOfInterestingObservations, Observation),
7   select_new_type(Observation, Type),
8   switch_interest(AgentID, Type).

```

Aktion: Validierung von Theorien des Typs 1

Für die Validierung von Theorien des Typs 1 prüfen die Agenten, ob sich Daten aus der eigenen Erfahrung in die Theorie einsetzen lassen. Lässt sich die Theorie

mit der Einsetzung der eigens beobachteten Daten bestätigen, steigt der *belief* zur Theorie und die Reputation des entsprechenden Autors. Wenn die Theorie sich nicht bestätigen lässt, sinkt die Überzeugung zur Theorie und auch die Reputation des Autors der Theorie.

```

1 execute_action(validation_type1, AgentID, _Options):-
2     select_t1_theory(AgentID, TheoryID)*->
3     theory_t1_validation(AgentID, Theories);
4     (...).

```

Für die Validierung wird zu einer gegebenen TheoryID geprüft, ob sich eine Relation zwischen dem BaseType und weiteren Events entsprechend der Theorie in den eigenen Daten nachvollziehen lässt.

```

1 theory_t1_validation(AgentID, TheoryID):-
2     TheoryType = type1,
3     actionstheory(CreatorID, TheoryType, TheoryID, Link, PotMod, Mod),
4     unfold_theory(TheoryID, [BaseType, T1, T2]),
5     actions:data(DataID_T1, AgentID, PTick_T1, ContentType_T1, BaseType,
6         ↪ Location_T1, Observation_T1)*->
7     ( T2 = [Count, Dist, [Elety, ElePos]],
8       ( member(Elety, Observation_T1)*->
9         theory_t1_validation(AgentID, TheoryID, validated);
10        theory_t1_validation(AgentID, TheoryID, not_validated)));
11     (...).

```

5.3.1 Aktionen zu Täuschungen in der Wissenschaft

Nicht zuletzt durch das Aufkommen diverser Plagiatsaffären in der Politik ist das Thema der Täuschung in der Forschung in den öffentlichen Diskurs gerückt. Bei den zuletzt breit diskutierten und prominenten Fällen über Plagiatsfunde in Dissertationen von bekannten Politikern mag der Effekt auf die Theoriebildung gering gewesen sein, doch gibt es auch Beispiele, bei denen die Täuschung stärker ins Gewicht gefallen ist. Ein bekanntes Beispiel ist hier der Genetiker Woo Suk Hwang. Durch die Manipulation von Daten hatte Hwang vorgegeben, menschliche Stammzellen in hoher Qualität klonen zu können, was ihm allerdings nicht gelang.⁸ In der Pharmaindustrie ist dieses Problem ebenfalls ausgeprägt, um etwa die Wirksamkeit von Präparaten positiver darzustellen.⁹ Doch wie hoch sind die Einflüsse von Täuschung, Fälschung und gezielter Informationsfilterung im Forschungsprozess? Wie sehr prägt beispielsweise ein Plagiat die Theorieentwicklung? Nicht nur die vorsätzliche Täuschung, sondern auch unbemerkte Fehler können problematisch sein. In dieser Simulation sollen

⁸ Vgl. Cyranoski 2009.

⁹ Vgl. Smith 2006.

genau diese Aspekte betrachtet werden. In einem geschlossenen System mit Wechselbeziehungen zwischen den simulierten Forschenden werden die Auswirkungen von Plagiaten und Fehlern untersucht. Zuvor müssen jedoch erst die Begriffe klar getrennt und definiert werden:

Plagiat

Ein Plagiat ist eine Übernahme von Forschungsergebnissen ohne dies kenntlich zu machen. Eine präzise Definition kann stark vom Sachbereich abhängen und variiert. Insbesondere im deutschen Wissenschaftsraum ist die Diskussion zu diesem Begriff aufgrund einiger Fälle von Plagiaten in Dissertation ein bekannter Politiker großflächig in den Medien diskutiert wurden.

Fälschung

Die Fälschung ist bereits ein komplexerer Fall. Bei einer Fälschung werden etwa Fakten produziert, die ideal zu einem Bezugssystem von intendierten Anwendungen passt, aber so nicht nachgewiesen wurde. Es ist eine Hypothese, die als Fakt weitergegeben wird. Dies können etwa in der Praxis Messdaten sein, die den gewünschten Resultaten entsprechend angepasst wurden, oder auch Daten, die nicht verifizierbar sind.

Fehler

Der Fehler ist noch komplexer, da dieser Fall in graduellen Abstufungen in der Empirie ein fester Bestandteil des Experiments ist. Fehler können an unterschiedlichen Stellen auftreten und sowohl eine bessere als auch eine schlechtere Passung zur Folge haben und sowohl in der Methodik als auch in der Durchführung liegen. Ein Experiment kann falsch aufgebaut sein, ein Buchdruck kann falsch datiert oder eine Zahl von einem Messgerät falsch übertragen werden.

In der Simulation würden sich Plagiate über `citation` abbilden lassen, doch da die Inhalte der Publikationen einfach gehalten sind und zumeist nur aus einer Referenz auf eine Theorie oder direkt auf spezifische Daten bestehen, würde dies nicht sonderlich anschaulich sein. Ein Plagiat wird stattdessen durch die Übernahme einer Publikation unter eigenem Namen dargestellt:

Definition 5.10. Plagiat

Ein $Agent_p$ plagiiert $Agent_q$, wenn $Agent_p$ die Elemente [PubID, Type, Content] einer Publikation `publication(PubID, Journal, [AgentID], Tick, Type, Content ↪)` übernimmt und unter dem eigenen Namen wieder veröffentlicht. Dies geschieht folgenderweise:


```

plagiarize(AgentID):-
  (...)
  findall(Publications, actions:publication(PublicationID,_,_,_,_),
    ↪ ListOfPubs),
  select(PublicationID, ListOfPubs),
  publication(SelectedPublication, _Journal, Authors, PublicationTick,
    ↪ Type, _Content), ListOfPubs),
  asserta(actions:publication(SelectedPublication, _Journal, [AgentID],
    ↪ ActualTick, Type, Content)).

```

Um die Plagiate verfolgen zu können, wird alle zehn Ticks ein Agent aufgefordert ein Plagiat zu begehen. Bei der Rezeption von Publikationen prüfen die Agenten, ob die gerade gelesene Publikation bereits schon einmal publiziert wurde. Ist dies der Fall, wird der Autor der später erschienen Publikation eines Plagiats beschuldigt und mittels Reputationsverlusts abgestraft. Dies hat Auswirkungen auf seine weiteren Möglichkeiten: Non-Blind-Reviews bei der Beantragung von Mitteln oder der Publikation werden schlechter ausfallen.

Fälschungen und Fehler sind ähnlicher Struktur. Entsprechend kann beides auf gleiche Weise dargestellt werden: Daten und Theorien werden publiziert, welche nicht durch andere Agenten bestätigt werden können. Dies können bewusste Fälschungen oder auch Fehler sein. In die Beobachtung *observation_experiment* eines Agenten *Agent_p* werden Events bzw. Eventtypen ergänzt, welche nicht in der Beobachtung vorgekommen sind. Dazu wird eine Theorie des Typs *T1* erstellt, welche sich auf diese Daten beruft. Durch die in Kapitel 5.3 *theory_t1_validation* beschriebene Validierung und die Einsetzung der Beobachtungsdaten, kann geprüft werden, ob die Theorie anhand anderer Daten nachvollziehbar ist. Ist dies nicht der Fall erfährt *Agent_p* eine Reduktion der Reputation.

6 Analyse von Simulationsläufen

In diesem Kapitel werden Beispiele und Simulationsexperimente vorgestellt. Die Hintergründe der Simulationsbeispiele wurden in den Kapiteln 2 bis 4 erläutert und die zentralen Komponenten dazu in Kapitel 5 vorgestellt. Die nun hier gezeigten Simulationsexperimente gliedern sich in Schwerpunktbereiche, die jeweils einen spezifischen Aspekt betonen.

6.1 Basis Simulation

Der erste Schwerpunktbereich zeigt die Verteilung einzelner Aspekte der Simulation wie die Entwicklung der Agenten über die Zeit oder die Zahl der Theorien und Publikationen. Die darin erhobenen Basisdaten dienen zum einen der Verifikation der gewünschten Verteilungen, zum anderen als Referenz für die weiteren Experimente. Dies ermöglicht die Auswirkungen der Parameter zu dem Referenzmodell herauszuarbeiten. Die hier angegebene Basiskonfiguration wurde auch für die weiteren Simulationen verwendet, sofern dies nicht gesondert erwähnt ist. Für die in dem Abschnitt genannten Ergebnisse der Basissimulation wurden fünf vollständige Simulationsläufe ausgewählt.

6.1.1 Basiskonfiguration

Parameter	Beschreibung
number_of_events	1000
number_of_laws	20
number_of_followup_laws	40
number_of_types	15
universe_bounders_2d	< 35, 35 >
law_complexity	3

Tabelle 6.1: Standardkonfiguration der simulierten Welt

Die Simulationsexperimente in der Basiskonfiguration durchlaufen 100 Ticks als Vorlauf zur Generierung der Welt und 300 eigentliche Ticks, in denen die Agenten Aktionen durchführen. Die Ticks werden in den Experimenten als Jahre verstanden. Der Seed für die Generierung von Zufallszahlen wird nicht vorgegeben. Jedes Experiment durchläuft 25 Einzelläufe mit gleicher Konfiguration. Dabei ist jedoch zu beachten, dass viele

Läufe aufgrund gesetzter Parameter nicht bis zum Ende durchlaufen. Grund dafür sind instabile Welten, die entweder zu stark wachsen und somit nicht mehr berechenbar werden oder zusammenfallen. Letzteres geschieht, wenn die Konstellation von Regeln keine Ergebnisse mehr erzeugen kann.

Gesucht wird also nach einer stabilen Welt, in der die Gesamtzahl der Ergebnisse pro Tick relativ konstant bleibt. Als Modell für die Weltenerzeugung wird die Variante 1b entsprechend Kapitel 5.2.2 verwendet. Die entsprechenden Parameter

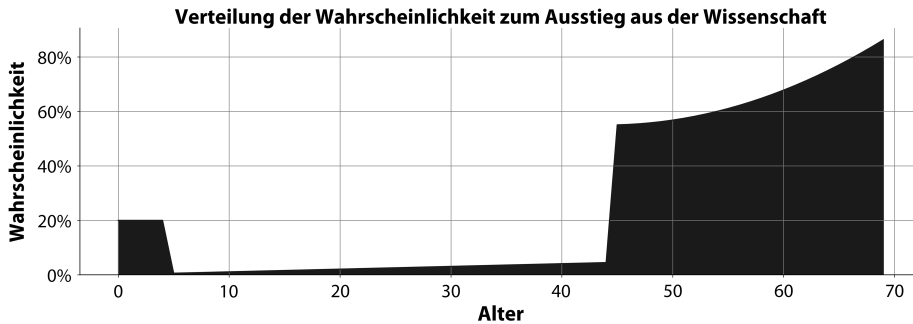


Abbildung 6.1: Wahrscheinlichkeit zum Ausstieg nach Agentenalter in %

sind in der Tabelle 6.1 aufgeführt. Zu Beginn eines Simulationslaufs werden zwanzig Agenten zufälligen Typs erstellt. Zu jedem Tick werden mit einer Wahrscheinlichkeit von 70% bis zu zehn Agenten erstellt. Die Wahrscheinlichkeit für den Ausstieg variiert mit dem Agentenalter. Die Abbildung 6.1 zeigt die dafür zugrunde liegende Wahrscheinlichkeitsverteilung, welche für die Simulationsexperimente verwendet wurde. Die Verteilung wurde so gewählt, um die in Kapitel 3.1 beschriebenen Kennzahlen zur Verweildauer in der Forschung aufzugreifen. Da die Datenlage hier allerdings nur exemplarisch ist, wird hier keinesfalls ein Anspruch eines statistisch validen Modells erhoben. Die Kennzahlen bieten vielmehr eine Orientierung und wurden mit dem Ziel zwei Effekte abzubilden idealisiert: Es soll gezeigt werden, dass der Einstieg in eine Wissenschaftskarriere mit einer hohen Hürde verbunden ist,

Parameter	Beschreibung
<code>maxFounding</code>	20
<code>maxInvestment(6)</code>	6
<code>default_resources(10)</code>	10

Tabelle 6.2: Ressourcen der Agenten

dann allerdings relativ stabil bleibt. Nach etwa 45 Berufsjahren soll die Wahrscheinlichkeit zum Ausstieg wieder stärker ansteigen. Die Aktionsplanung erfolgt nach den im Modul `actions` \hookrightarrow definierten Strategien (und nicht in der alternativen Konfiguration zufällig).

6.1.2 Entwicklung der Agenten

Wie in Kapitel 3.1 beschrieben, ist die Verweildauer von Personen in der Forschung über die Jahre auf etwa fünf Jahre gesunken. Der Mittelwert in den Simulationsexperimenten liegt bei 13,36 und im Median bei drei Ticks. Dabei ist zu beachten, dass hier neben der Gruppe der Forscher ebenfalls die beiden Gruppen Studenten und Förderer einbezogen wurden. Die Tabelle 6.3 lässt erkennen, dass die Verteilung über die Läufe stabil ist. Dies deutet darauf hin, dass die dynamischen Prozesse zur Erzeugung und dem Ausscheiden der Agenten tatsächlich die gewünschte Verteilung ab-

	Alle	Run 1	Run 2	Run 3	Run 4	Run 5
Median	3	4	3	3	3	3
Mittelwert	13, 36	13, 82	13, 61	13, 15	13, 21	13, 00

Tabelle 6.3: Verteilung des Alters

Tick	Res.	Rep.	Que	Action	Data	Bel.	Pub.	Int.
253	11	1.20	1	observation_experiment	1	1	0	1
254	9	1.20	0	make_theory	1	1	0	1
255	8	1.20	1	observation_types	1	1	0	1
256	6	1.20	1	publish_types	2	2	0	1
257	5	1.20	0	select_new_interest	2	2	1	1
257	5	1.20	0	new interest	2	2	1	1
259	4	1.20	4	observation_experiment	2	2	1	2
260	3	1.20	3	observation_experiment	13	13	1	2
261	1	1.20	2	observation_experiment	13	13	1	2
263	1	1.20	0	funding	26	26	1	2
263	1	1.44	0	Funding approved	26	26	1	2
265	16	1.44	5	observation_experiment	26	26	1	2

Res. = Ressources, Rep. = Reputation, Bel. = Belief, Pub = Publication, Int. = Interests

Tabelle 6.4: Auszug aus den Aktionen eines Agenten

bilden. Die Abbildung 6.2 zeigt dabei die Spannweite der Altersverteilung über die fünf betrachteten Simulationsläufe.

6.1.3 Aktionen in den Simulationsbeispielen

Die Aktionen der Forschenden werden entsprechend des Handlungsbaums in Abbildung 5.5 und der Beschreibungen in Kapitel 5.1.3 und 5.3 geplant und durchgeführt. Die Studierenden und Reviewer haben `read_publication` als Handlung. Die Reviewer werden anlassbezogen, etwa bei der Aktion `funding` in Begutachtungen einbezogen und lesen ansonsten aktuelle Publikationen, um die Wissensbasis aktuell zu halten. Die Studierenden lesen ebenfalls und werden nach etwa 5 Ticks (entsprechend einer Zufallsverteilung) in den Status eines Forschenden gehoben und können entsprechend weitere Handlungen durchführen.

Für die Analyse der Aktionen schreibt jeder Agent ein `diary`, welches wichtige Schritte und die jeweiligen Auswirkungen für die spätere Analyse der Simulationsergebnisse aufzeichnet. Das Beispiel 6.4 zeigt einen Auszug, beschränkt auf die Aktionen und der ausgewählten Parameter über fünf Ticks, in denen das Zusammenspiel von Parametern gut ersichtlich ist. Zu sehen ist, dass ein Agent Beobachtungen durchführt und sich dann an einer Theorie auf Grundlage nur eines Datensets versucht (Tick 253 - 254). Es glückt dem Agenten nicht, die Theorie zu formulieren, und

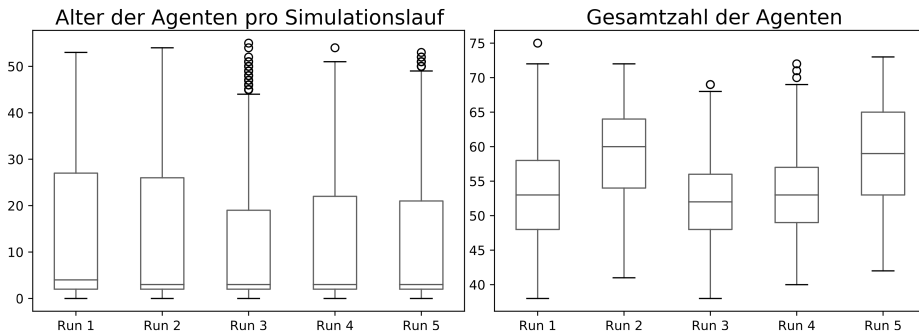


Abbildung 6.2: Häufigkeitsverteilung und Gesamtzahl der Agenten pro Lauf

es werden weitere Beobachtungen durchgeführt (Tick 255). Die Beobachtungen werden publiziert. Anstelle eines neuen Versuchs zur Formulierung einer Theorie sucht sich der Agent nun ein weiteres Fokusgebiet und führt erneut Beobachtungen durch. Mit zwei Beobachtungen (Tick 259 - 260) ist der Agent deutlich erfolgreicher als in Tick 253, jedoch scheitert die letzte Beobachtung in Tick 261. Der Agent hat noch weitere Handlungen geplant (Que = 2 in Tick 261), jedoch müssen diese verworfen werden, da neue Ressourcen benötigt werden (Tick 263). Der Antrag auf Förderung ist erfolgreich und zehn Ressourceneinheiten werden bereitgestellt. Dadurch steigt auch die Reputation des Agenten von 1.2 auf 1.44 und es können wieder weitere Aktionen durchgeführt werden (ab Tick 265). Im weiteren Verlauf wird der Agent jedoch kein Glück haben und keine Theorie aufstellen können. Der Agent publiziert noch diverse Beobachtungsdaten und schafft es insgesamt 89 Beobachtungen in 21 Messreihen durchzuführen. Diese wird der Agent in seinen 48 aktiven Ticks in sieben Publikationen beschreiben. Die Befähigung *ability* ist mit 0.1 relativ gering. Entsprechend war es dem Agenten möglicherweise nicht möglich, hinreichend viele Datenmengen zu sammeln oder die Ereignistypen im Interesse des Agenten waren nur selten aufzufinden. Die Abbildung 6.3 zeigt die Verteilung der gewählten Aktionen eines Agenten in der Gruppe der Forschenden. In der Verteilung nehmen die Beobachtungen den größten Teil der Aktionszeit ein. Messreihen können über mehrere Ticks stattfinden und stehen am Anfang des Handlungsbaums, während die Theoriefindung und das Publizieren nur stattfinden, wenn auch genug Daten vorhanden sind. Dies lässt sich in der Verteilung entsprechend erkennen.

6.2 Simulationsbeispiele zu Publikationsformen

In Kapitel 3.3 werden Publikationen als zentrales Mittel der wissenschaftlichen Kommunikation beschrieben. Publikationen finden in der Wissenschaft über verschiedene Medien statt. Wissenschaftliche Zeitschriften sind dabei eine übliche Form einen

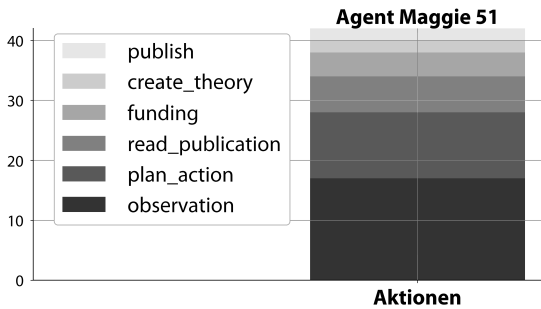


Abbildung 6.3: Verteilung der Aktionen der Forschenden *Maggie51*

wissenschaftlichen ungerichteten Diskurs zu führen. Ungerichtet ist dieser, da er von einer Gruppe oder einem einzelnen Autor an eine unbestimmte Leserschaft gerichtet ist. Antworten folgen nach möglichen Diskussionen auf Konferenzen, in Folgebeiträgen der Autoren oder Kommentaren in weiteren Artikeln anderer. Das Publikationswesen, mit seiner fast 400 Jahre alten Tradition, befindet sich allerdings in einem Umbruch. Digitalisierung und Open Access sind die zwei entscheidenden Faktoren, welche die traditionellen Strukturen von Verlagen durchbrechen und zu einer Pluralisierung des Wissens führen. Bei steigenden Kosten für Literatur und einer steigenden Nutzerschaft (siehe Abschnitt 3.3.1) werden alternative Modelle für Forschende und Institutionen zunehmend attraktiver. Doch was für Folgen hat dies auf die Theorieentwicklung? In diesem Experiment wird genau das dargestellt. Aufbauend auf dem Basisexperiment forschen die Agenten und publizieren ihr Wissen. Doch werden in diesem Experiment einige Rahmenbedingungen geändert: Zum einen wird der Theoriebegriff und insbesondere die Theoriefindung simplifiziert, um vorhersagbar mehr Publikationen provozieren zu können. Agenten führen zwar noch Beobachtungen im Ereignisraum durch, die darauf aufbauenden Theorien werden allerdings nicht maschinell gesucht, sondern als Hüllen vorgegeben. Publikationen erfolgen dafür in entsprechenden Zeitschriften. Die so publizierten Artikel werden dadurch anderen Agenten zugänglich. Es wird in dem Experiment zwischen zwei Arten von Journals unterschieden: Peer-Reviewed und offen (siehe Kapitel 3.4.2). Im ersten Fall entscheidet ein Gutachten über die Annahme, im zweiten Fall entfällt das Gutachten und der Schwellenwert. Doch fällt in der Simulation der Reputationsgewinn dann auch entsprechend geringer aus. Die Wirkung von Open-Access-Modellen, in denen Artikel, je nach Variante, frei zugänglich oder auch frei veröffentlicht werden können, wird ebenfalls aufgegriffen.

Definition 6.1. Reviewed Journal

1. Ein Journal J hat die Struktur

Journal = [UniqueID, Threshold, ReputationFactor, Type].

2. Reputationsfaktor R , welcher den Reputationsgewinn durch eine Publikation in dem entsprechenden Journal beeinflusst: $Gain = Reputation * R$
3. Jede Publikation ist in diesem Beispiel genau einem Journal zugeordnet.
4. Journals können entweder als `open_access` zur Publikation und im Zugriff frei sein oder `closed` sein und Ressourcen für die Veröffentlichung und den Zugriff verlangen.

Im Falle eines Peer-Reviewed Journals kommt noch der Prozess der Begutachtung hinzu:

5. Die Entscheidung über die Annahme der Publikation treffen Gutachter, analog des in den Abschnitten 3.4.3 und 5.9 beschriebenen Prozessen. Dabei werden die Einzelbewertungen der Begutachtenden zu einer Gesamtbewertung $score = \sum(review_n)$ zusammengerechnet.
6. Ein für jedes Journal spezifischer Schwellenwert θ entscheidet auf Grundlage der Bewertungen über die Annahme: $score \mapsto \begin{cases} accepted : score \geq \theta \\ rejected : score < \theta \end{cases}$

6.2.1 Experiment 1: Reviews und Open Access

In diesem Experiment werden die Auswirkungen von drei Peer-Reviewed Journals mit unterschiedlich starken Schwellenwerten θ^1 im Vergleich zu einem System ohne Schwellenwert² und deren Auswirkung auf die Theorieentwicklung betrachtet. Dazu werden zwei dedizierte Simulationsläufe verglichen. Die Journals mit Schwellenwert sind definiert als $J = [[low_impact, 5, 1.01, closed], [mid_impact, 15, 1.05, closed], [high_impact, 50, 1.1, closed]]$ und die Schwellenwerte sind so gewählt, dass eine sichtbare Abstufung zwischen den Journals entsteht. In dem zweiten Lauf liegt der Schwellwert bei 0. Als Kriterium für das Review werden von den Autoren die Zahl der passenden Beobachtungsdaten als Beleg und die Befähigung `ability` des Autors berücksichtigt. Hinzu kommt noch die Schnittmenge der Interessen zwischen den Autoren und den Reviewern. Die Gruppe der Reviewer setzt sich in dem Experiment aus drei Agenten zusammen, die eine Überschneidung im Forschungsschwerpunkt `research_domain` vorweisen. Wenn keine passenden Agenten vorhanden sind, werden zufällige Agenten gewählt. Wie Abbildung 6.4 zeigt, lässt sich die Verteilung in der Akzeptanzrate der Journals anhand der Publikationszahlen und Ablehnungen wiedererkennen. Für jede erfolgreiche Publikation erhalten die Autoren einen leichten Gewinn an Reputation ($\mathcal{R} = < impact_{Journal} : 1.01, midimpact_{Journal} : 1.05, highimpact_{Journal} : 1.1 >$), welcher den Erfolg bei Reviewverfahren und

¹ Simulationslauf: run-2020-7-23-15-32-57.963709592

² Simulationslauf: run-2020-7-23-17-19-47.93621087

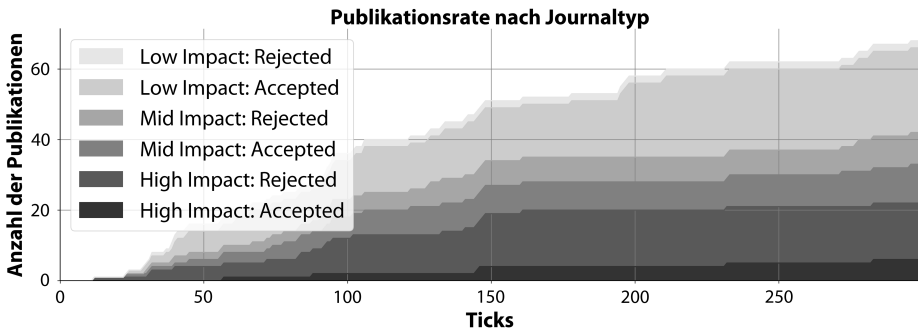


Abbildung 6.4: Absolute Anzahl von Publikationen und Ablehnungen nach Journal

die Sichtbarkeit der eigenen Forschung erhöht. Haben die Agenten bei der Erforschung wenig Erfolg, so zeigt sich das auch in der Zahl und Verteilung der Publikationen. Der Simulationslauf run-2020-7-23-15-32-57.963709592 zeigt genau diesen Fall: In den Daten ist ersichtlich, dass es dort insgesamt nur 14 Datensets mit erfolgreichen Beobachtungen gab. Bei Vergleichsläufen sind dies oft mehrere hundert Beobachtungen. Die Agenten konnten so nur wenige Theorien aufstellen und diese dann auch nur in einer niedrigen Qualität. Mit wenigen Beobachtungen konnten diese Theorien auch nicht validiert werden und entsprechend erfolgten auch weniger Publikationen. Bei einem Vergleich von Theorien und Publikationen wird ersichtlich, dass sich dieses Verhältnis mit einem Peer-Review-Verfahren umkehrt.³ Die Zahl der Publikationen wird nur noch durch die Ressourcen und Zeit begrenzt. Bei einer weiteren Anhebung der Schwellenwerte sinken die Publikationszahlen und die Verteilung der Publikationen nach Journalklassen verändert sich wie erwartet.

Die Abbildung 6.5 zeigt die absolute Verteilung der Erfolgsrate bzw. Ablehnung der Publikation in den jeweiligen Journaltypen im mit den höheren Schwellenwerten low impact: 5, mid impact: 15, high impact: 50.⁴ In diesem Experiment schaffen es keine Artikel in einem high impact Journal publiziert zu werden. Auch ist es nur ein vergleichsweise geringer Anteil an der Gesamtheit der Artikel, welcher in einem mid impact Journal publiziert wird. Dies gelingt auch erst nach mehreren Generationen und nach über 280 Ticks Laufzeit. Selbst die low impact Journals haben hier eine hohe Ablehnungsrate. Über die Laufzeit ist die Verteilung der Artikel auf die Journalklassen dabei über mehrere Agenten-Generationen relativ konstant. Der späte Erfolg in den mid impact Journals lässt auch über die Zeit die steigende Erfolgsrate in

³ Ein direkter Schluss auf den Anteil der publizierten Theorien ist jedoch so nicht möglich, da nicht ausgeschlossen ist, dass Theorien mehrfach publiziert werden. Da sich die Qualität jedoch nicht ändert, wird die Erfolgsrate nur minimal durch eine mögliche steigende Reputation des Autors wachsen. Von daher kann dieser Wert als Indikator gesehen werden.

⁴ Simulationslauf run-2020-7-23-15-36-58.321325302

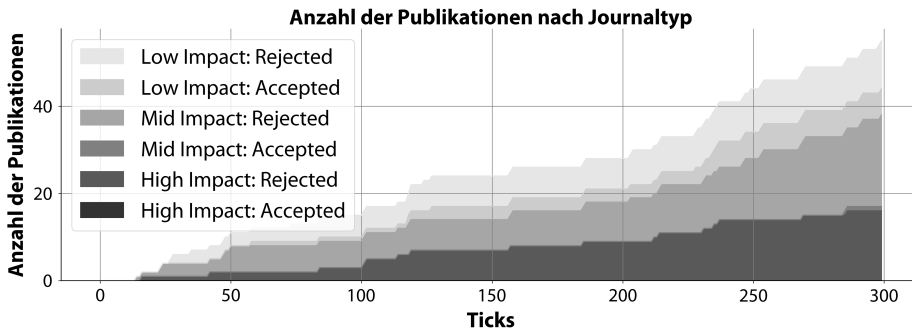


Abbildung 6.5: Die Verteilung der Erfolgsrate der Publikation nach Journaltypen

der Theoriebildung erklären. Darauf deutet auch der zum Ende der Laufzeit sinkende Anteil an low impact Publikationen hin.

6.2.2 Experiment 2: Publikationen und Zitationen

Nun stellt sich die Frage, wie der Effekt unterschiedlicher Publikationsstrategien auf die Verbreitung des Wissens und die Entwicklung im System hat. Dazu wurde ein Simulationslauf erstellt, in dem zu jedem Journaltypen und jeder Publikation eine Zitation gemessen wurde.

Definition 6.2. Zitation

1. Eine Zitation hat in der Simulation die Form $\text{citation}(\text{Author}, \text{ReferenceAuthor} \leftrightarrow, \text{ReferenceArticle})$ und wird als Teil der Aktion `read_publication_rnd` ausgelöst.
2. Der `basetype` der in der Publikation beschriebenen Theorie muss im Interesse des Agenten liegen: $\text{base_type} \in \text{list_of_interests}$
3. Eine Schnittmenge zwischen den Interessen der beteiligten Agenten A_i : $\bigcup(\text{list_of_interests}_i) \neq \emptyset$ und den Forschungsbereichen $\bigcup(\text{research_area}_i) \neq \emptyset$ muss bestehen. Diese Bedingung ersetzt die Passung durch die Aktion `validation`, führt aber zu einer ähnlichen Verteilung.⁵
4. Sind diese Bedingungen erfüllt, kann der Agent selbst eine Publikation durchführen. Dabei wird eine Zitation als Referenz erzeugt.

⁵ Wünschenswert wäre es an dieser Stelle, die Überzeugungen als Grundlage für die Bewertung einer Publikation zu wählen und die Passung über die Aktionen zur Validierung zu bestimmen, damit die Zitationen sich tatsächlich stärker am Inhalt der Publikation orientieren. Dies kann zwar in der Simulation abgebildet werden, übersteigt allerdings die Ressourcen in einem Simulationslauf. Entsprechend wurde hier ein Wert bestimmt, der die wesentlichen Faktoren zwischen Autor und Rezipient abgleicht und so auch ein ähnliches Bewertungsverhältnis herbeiführt.

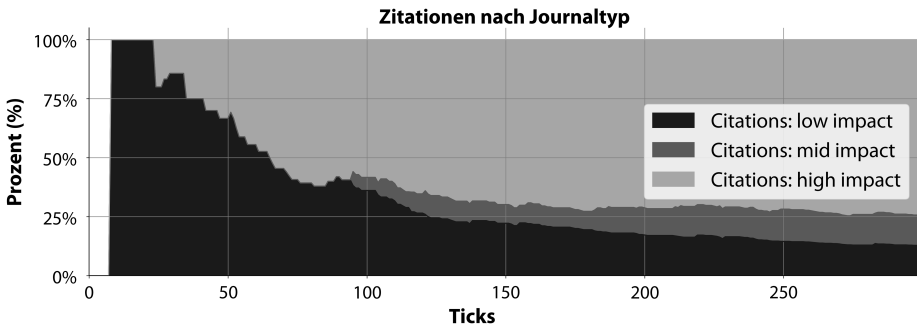


Abbildung 6.6: Die Anteil der Zitationen nach Journaltyp

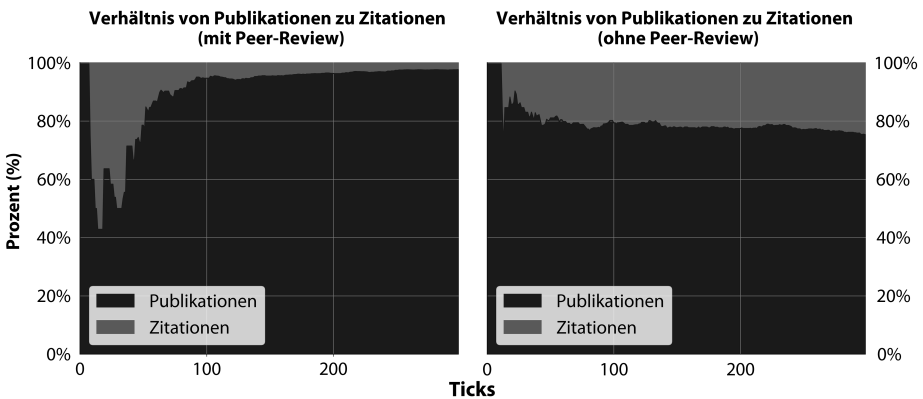


Abbildung 6.7: Publikationen im Verhältnis zu Zitationen (mit und ohne Peer-Review)

Publikationen fördern die Theorieentwicklung durch Daten, doch die Zeit der Agenten ist begrenzt und ein Überangebot von Publikationen muss nicht der Gesamtentwicklung zuträglich sein. Jedoch kosten Reviewprozesse möglicherweise Zeit, da Artikel überarbeitet werden müssen. Dies sichert aber wieder die Qualität. In Abbildung 6.6 ist zu sehen, dass Letzteres auch ein wesentlicher Faktor ist. Die Agenten lesen und zitieren Artikel bevorzugt aus höher bewerteten Journals. Dazu suchen sie zunächst nach einem Artikel, welcher in der `list_of_interests` liegt und beginnen bei der Suche in den *high impact* Journals. Werden sie dort nicht fündig, suchen sie in weiteren Journals. Abbildung 6.7 stellt nun das Verhältnis von Publikationen und Zitationen zwischen einem System mit offenen Publikationen und einem mit Peer-Review gegenüber. Das Ergebnis wirkt zunächst überraschend: Peer-Reviewed Journals erhalten weniger Zitationen als die offenen Journals. Der Grund hierfür liegt darin, dass in einem offenen System die von einer Publikation abgeleitete Beiträge selbst auch kein Review durchlaufen. Dadurch kann nicht nur insgesamt mehr publiziert werden, sondern es können auch mehr Folgepublikationen entstehen.

6.3 Simulationsbeispiel zu Ressourcen als Grundlage von Handlungen

Die Ressourcen der Simulation sind eine virtuelle Einheit. Diese repräsentieren einen Faktor der investiert werden muss, um Forschungshandlungen durchzuführen. Dies können Geldmittel sein, aber auch etwa Zeiteinheiten an einem Großgerät oder in einem Labor. Ebenfalls kann dieser Wert auch als Faktor für die Freiheit von anderen Verpflichtungen, wie etwa der Lehre, eingesetzt werden. In diesem Beispiel werden die Ressourcen als Geldmittel verstanden und im Kontext der Forschungsförderung betrachtet. Die Abbildung 6.8 zeigt zwei prägnante Zustände einer Simulationsreihe, in welcher die maximalen Ressourcen, welche von den Agenten über einen Antrag eingeworben werden können, zwischen fünf und zwanzig Einheiten liegt. Dabei weichen die Regeln dieser Simulation, wie folgt von der Basiskonfiguration ab:

1. Die Simulation läuft insgesamt 200 Ticks.
2. Alle Agenten haben eine harte Altersgrenze von 45 Ticks.
3. Sie verlassen auch das System, sobald sie keine Mittel mehr zur Verfügung haben (aufgebraucht und kein Erfolg bei der Einwerbung).

6.3.1 Experiment 3: Forschungsbudgets und Wissen

Die Simulationsläufe wurden über 200 Ticks durchgeführt und die Agenten hatten eine feste Altersobergrenze von 45 Ticks. Es wird gezeigt, wie viele Überzeugungen pro Tick im System vorhanden sind (innerhalb der aktiven Agenten). Es ist zu sehen, dass die Peaks in der linken Grafik (max. 20 Einheiten) deutlich höher sind. Die Agenten können mehr Wissen erlangen. Mit dem Ausscheiden der ersten Generation fällt dies jedoch ab. Die Überzeugungen scheinen überproportional zu der Anzahl der Agenten zu steigen. In der zweiten Variante lassen sich viele kleine Peaks erkennen. Doch bleiben alle weit hinter denen der ersten Variante zurück. Ebenfalls ist zu erkennen, dass die Zahl der Agenten ebenso stark fluktuiert. Dies liegt daran, dass die Agenten in vielen Fällen frühzeitig das System aufgrund von Ressourcenmangel verlassen. Während bei einer geringen Förderung und hoher Konkurrenz die Beliefs im System teils stark zurückfallen, lässt sich bei einer höheren Förderung eine stabile Basis an Überzeugungen auch zwischen den Peaks im System erkennen. Betrachtet man die Aktionen eines einzelnen Agenten mit maximal sechs Ressourceneinheiten pro Antrag in Abbildung 6.9, so wird im Vergleich einem Agenten in Abbildung 6.3 ersichtlich, dass dieser kaum die Möglichkeit hat, Forschung zu betreiben. Dies lässt sich auch auf der Theorieebene erkennen: Die Gesamtzahl der Theorien liegt in dem unterfinanzierten System bei 48 Theorien und im Referenzlauf bei 71. Das gleiche Bild zeigt sich bei den erhobenen Datensets mit 255 gegenüber 29820. In dem unterfinanzierten System ist es den Agenten nicht gelungen zu publi-

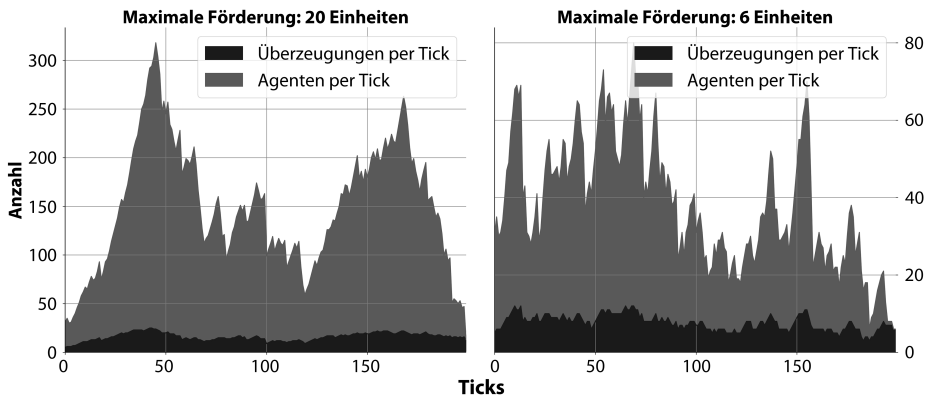


Abbildung 6.8: Vergleich zweier Fundingstrategien: Obergrenze bei der Bewilligung von 20 Ressourceneinheiten vs. 6 Einheiten

zieren. Die Gesamtzahl der Publikationen im Referenzsystem liegt dagegen bei 511 Publikationen mit 313 Zitationen.

6.4 Einordnung

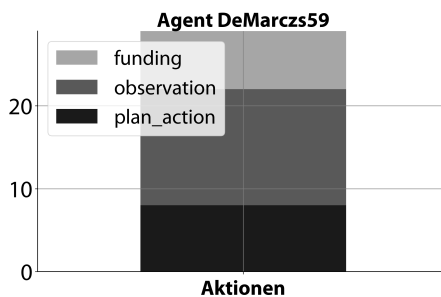


Abbildung 6.9: Aktionen des unterfinanzierten Agenten *DeMarcz59*

Die Wissenschaft zeigt sich als komplexes selbstorganisierendes System in vielen verschachtelten Facetten. Ziel dieser Arbeit war es, einige dieser Facetten und ihre Verbindungen in einer Simulation darzustellen. Die hier gezeigten Beispiele sollten genau dies als Anwendungsmöglichkeit der Simulation demonstrieren. In den Kapiteln 2 bis 4 wurden grundlegende Rahmenbedingungen angesprochen, welche zu der in Kapitel 5 gezeigten Simulation führen. Die Simulationsbeispiele zu den Gutachterprozessen, der Publikation und Drittmittelbeantragung, der Auswirkung von Forschungsfinanzierungen auf die Wissensbasis oder die Dynamik der Agenten in einem System sollten die Verbindung der drei Ebenen – Theorie-, Sozial-, und Gesellschaftsebene – demonstrieren. Die Beschreibung der Theorie wurde an den wissenschaftstheoretischen Strukturalismus angelehnt, diverse soziale Aspekte wurden aufgenommen. Gutachterprozesse wurden auf deren Grundlage durchlaufen. Die Agenten konnten in simulierten und dynamischen Umgebungen eigenständig Theorien innerhalb des gesetzten Rahmens aufbauen. In einem Ereignisraum wurden Konzepte als Formen erkannt und in den Theorien formuliert. Auf

Die Wissenschaft zeigt sich als komplexes selbstorganisierendes System in vielen verschachtelten Facetten. Ziel dieser Arbeit war es, einige dieser Facetten und ihre Verbindungen in einer Simulation darzustellen. Die hier gezeigten Beispiele sollten genau dies als Anwendungsmöglichkeit der Simulation demonstrieren. In den Kapiteln 2 bis 4 wurden grundlegende Rahmenbedingungen angesprochen, welche zu der in Kapitel 5 gezeigten Simulation führen. Die Simulationsbeispiele zu den Gutachterprozessen, der Publikation und Drittmittelbeantragung, der Auswirkung von Forschungsfinanzierungen auf die Wissensbasis oder die Dynamik der Agenten in einem System sollten die Verbindung der drei Ebenen – Theorie-, Sozial-, und Gesellschaftsebene – demonstrieren. Die Beschreibung der Theorie wurde an den wissenschaftstheoretischen Strukturalismus angelehnt, diverse soziale Aspekte wurden aufgenommen. Gutachterprozesse wurden auf deren Grundlage durchlaufen. Die Agenten konnten in simulierten und dynamischen Umgebungen eigenständig Theorien innerhalb des gesetzten Rahmens aufbauen. In einem Ereignisraum wurden Konzepte als Formen erkannt und in den Theorien formuliert. Auf

deren Grundlage entstanden Überzeugungen, welche zu Publikationen führten. Alle diese Aspekte können jeweils weiter untersucht werden und bieten spannende Fragestellungen, auf welche die Simulation spezialisiert werden kann. Die Simulationsumgebung bietet deutlich mehr Funktionen, als in den Experimenten dargestellt wurden. Die Experimente wurden entsprechend bewusst auch als *Simulationsbeispiele* benannt, da diese nur einen Ausschnitt der möglichen Anwendungen der Simulation zeigen. Hier bietet sich auch eine Perspektive: Alle angesprochenen Beispiele eignen sich für eine weitere Vertiefung. Die in den Kapiteln 2 bis 4 erwähnten Prozesse könnten in der vorgestellten Simulationsarchitektur sicherlich noch tiefer beschrieben werden und so weitere Experimente ermöglichen. Wenn auch die Simulation zu den gezeigten Ergebnissen geführt hat, so haben die Experimente oft die Möglichkeiten der Sprache bzw. des Interpreters überschritten und wurden so durch technische Begrenzungen eingeschränkt. Komplexere Läufe sind immer am Rande der Ausführbarkeit durchlaufen, selbst auf einem System mit 32 Rechenkernen und 64 GB RAM. Eine ressourcenbewusstere Implementierung mit einer Auslagerung von Prozessen der Theoriefindung und Weltengenerierung in eine in dieser Hinsicht effizienteren Umgebung, kann entsprechend noch deutlich umfassendere Simulationsläufe erlauben und insbesondere besser gesellschaftliche Prozesse auf Grundlage der einzelnen Agenten abbilden.

Teil III

Implementierung

7 Die kommentierte Simulationsumgebung

Dieses Kapitel beschreibt die kommentierte Simulationsumgebung. Der vollständige Programmcode kann online über ein eingesehen und heruntergeladen werden.¹ In dem Repository sind der Quelltext in der Basiskonfiguration der Simulation und die in dieser Arbeit besprochenen Datensätze zu den vorgestellten Simulationsläufen zu finden. Das Programm steht unter der GPL-Lizenz² und darf entsprechend der dort beschriebenen Lizenzbestimmungen frei verwendet und erweitert werden. In den jeweiligen Simulationsexperimenten wurden spezifische Konfigurationen verwendet, welche in den entsprechenden Beschreibungen ersichtlich sind. Diese beinhalten teils spezifische Prozesse, welche nicht in der Basiskonfiguration enthalten oder auskommentiert sind. In der Basiskonfiguration sind entsprechend nicht alle hier gezeigten Prozesse aktiv. Die in diesem Kapitel beschriebene Konfiguration bietet eine Orientierung für den Aufbau der Simulationsexperimente und der Definitionen der Prädikate. In der Tabelle 7.1 werden die Module und ihre Funktionen kurz erläutert. Eine genauere Beschreibung der Funktionen ist in den Kapiteln 5 und 6 zu finden. Die Module für die Textausgabe und Speicherung von Simulationsdaten werden aus Platzgründen nicht abgedruckt und können online eingesehen werden.

Modul	Beschreibung
<code>actions.pl</code>	Dieses Modul enthält die Strategien zur Planung und Durchführung von Aktionen.
<code>actions_definition.pl</code>	Die Definition der einzelnen Aktionen befindet sich in diesem Modul.
<code>agents.pl</code>	Die Agenten werden über die hier implementierten Funktionen erzeugt.
<code>configuration.pl</code>	Die Konfiguration der Simulationsläufe erfolgt in diesem Modul. Die Parameter und deren Auswirkungen werden in der Tabelle 5.3 beschrieben.
<code>create_world.pl</code>	Dieses Modul bietet die Möglichkeit, simulierte Welten für jeden Simulationslauf spezifisch zu erzeugen.
<code>file_output.pl</code>	Die Funktionalitäten zum Export der Ergebnisse in unterschiedliche Formate finden sich in diesem Modul.
<code>publications.pl</code>	In diesem Modul werden alle Funktionen bezüglich zu Publikationen definiert.
<code>screen_output.pl</code>	Funktionen zur Ausgabe auf den Bildschirm werden hier gesammelt.
<code>simulation.pl</code>	Dies ist das Hauptmodul und enthält die Simulationslogik. Hier werden alle Kernfunktionen und die zentralen Schleifen definiert. Diese Datei muss zum Starten der Simulation geladen werden.
<code>utils.pl</code>	Alle Hilfsprädikate werden zugunsten der Übersichtlichkeit in diesem Modul ausgelagert. Es stehen eine Vielzahl von Funktionen zur Simulation und Auswertung bereit.

Tabelle 7.1: Übersicht über die Module der Simulation

¹ <https://gitlab.gwdg.de/dkurzaw1/eine-simulation-von-wissenschaftsprozessen-in-prolog>

² Die GNU GPL Version 3 findet sich unter <https://www.gnu.org/licenses/gpl-3.0.html>

7.1 Modul: simulation.pl

```

1  /** <module> PROLOG SOCIAL SIMULATION SYSTEM main module
2  This is the main simulation module and includes the main logic for
    ↪ simulation runs and ticks.
3  @author Daniel Kurzawe
4  @version ALPHA
5  @see http://www.daniel-kurzawe.de
6  @license GNU/GPL V.3 http://www.gnu.org/copyleft/gpl.html
7  */

```

Warnungen bezüglich ungenutzter Variablen werden mittels `style_check(-singleton ↪)` unterdrückt und `callgraph` für eine Visualisierung der Programmstruktur eingebunden.

```

8  :- style_check(-singleton).
9  %%%
10 %%% For debugging %%%
11 %%%
12 %%%
13 %%% Visualisation of program structure %%%
14 :- use_module(library(callgraph)).
15 %%% run command: module_dotpdf(user, [method(unflatten([f1(4),c(4)]))]).
16 %%%

```

Im folgenden Abschnitt werden alle notwendigen Module geladen. Eine Übersicht über die Module und deren Funktionen findet sich in Tabelle 7.1 und in Abschnitt 5.1.

```

17 %%%
18 %%% Moduls %%%
19 %%%
20 %%%
21 %:- module(simulation).
22 %:- module(kernel,[sim_time/1]).
23 %%% CSV file output library %%%
24 :- use_module(library(csv)).
25 %%% Action definitions and logic %%%
26 :- use_module('utils.pl').
27 %%% Action definitions and logic %%%
28 :- use_module('configuration.pl').
29 %%% Action definitions and logic %%%
30 :- use_module('actions.pl').
31 %%% Agents definitions and logic %%%
32 :- use_module('agents.pl').
33 %%%

```

```

34  %%% Create laws %%%
35  :- use_module('create_world_event.pl').
36  %%% World facts database %%%
37  % :- use_module('world_facts.pl').
38  %%% Screen output %%%
39  :- use_module('screen_output.pl').
40  %%% File output %%%
41  :- use_module('file_output.pl').
42  %%% Publication addition %%%
43  :- use_module('publication.pl').
44  %%% For calling influxdb and grafana (monitoring) over RESTful interface
45  :- use_module(library(http/http_client)).

```

Die im Folgenden definierten dynamischen Prädikate dienen der Ablaufsteuerung und Konfiguration. Auch wird die Zahl der verfügbaren CPU-Kerne für die nebenläufige Berechnung definiert und ein *Thread-Pool* geschaffen. Standardmäßig werden alle verfügbaren Kerne verwendet.

```

46  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
47  %%% Initialisation of variables and Constants %%%
48  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
49  % Dynamic definition for predicates, acting as "global variables"
50  :- dynamic([sim_time/1, run_id/1, experiment_id/1, experiment_dir/1,
51             ↪ diary_dir/1, abort_run/1, abort/2]).
52  version('0.7').
53  :- cls.
54  :- current_prolog_flag(cpu_count,NumberOfCores),
55     thread_pool_create(threadpool, NumberOfCores, []).

```

Zum Start der Simulation über `start/0` werden alle Parameter initialisiert und die äußere Hauptschleife des Simulationskerns beginnt. Diese Schleife iteriert über die Zahl der definierten Simulationsläufe in einem Experiment.

```

55  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
56  %%% Simulation start %%%
57  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
58  :- message(welcome,_).
59
60  start:-
61     initialisation_experiment,
62     message(experiment,_),
63     number_of_runs(Runs),
64     foreach( between(0,Runs,Seed),
65             time(run(Seed)) ),
66     message(border,_),
67     message(runstate,_).

```

Alternativ lässt sich auch ein Menü definieren. Dadurch kann ein Simulationslauf einzeln gesteuert werden. In der Standardkonfiguration läuft die Simulation jedoch nach dem Start selbstständig ab und arbeitet die gegebenen Konfigurationen ab.

```

68 start_menue:-
69     menu('Simulation Options',
70         [ start : 'Start a simulation run'
71           , check_configuration : 'Check configuration'
72           , halt    : 'Quit'
73           ], Choice), call(Choice).

```

Die Hauptschleife `run/1` steuert alle Prozesse zu einem Simulationslauf (`run`): Es werden alle Datenstrukturen initialisiert, die Zahl der zu durchlaufenden Zeitpunkte (`ticks`) festgelegt und dann ein Vorlauf (`prerun`) gestartet. Danach werden die Agenten erzeugt und der eigentliche Simulationslauf (`run`) gestartet. Nach dem Durchlauf werden die Ergebnisse in die entsprechenden Dateien geschrieben und die Ergebnisse ausgegeben. Zum Abschluss werden dann alle Daten aus dem Speicher gelöscht und somit alles für den nächsten Lauf wieder freigegeben.

```

75 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
76 %%% Simulation: Main loop %%%
77 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
78
79 run(Seed):-
80     initialisation_run,
81     number_of_ticks_for_prerun(NumberOfTicksPreRun),
82     number_of_ticks(NumberOfTicks),
83     update_number_of_ticks(NumberOfTicks,NumberOfTicksPreRun,
84         ↪ NewNumberOfTicks),
85     prerun(NumberOfTicksPreRun),
86     StartActualRun is NumberOfTicksPreRun + 0,
87     initialisation_agents,
88     writeln('start run'),
89     run(StartActualRun,NewNumberOfTicks),
90     write_results,
91     cleanup.

```

Ein Vorlauf ermöglicht es in der simulierten Welt einen stabilen Zustand zu erreichen, bevor die Simulationsexperimente starten und vermeidet den Effekt von noch nicht eingependelten Prozessen und Wechselwirkungen einer frischen Welt auf die Experimente.

```

91 prerun(NumberOfTicksPrerun):-
92     writeln('... performing prerun'),
93     ( between(0, NumberOfTicksPrerun, C),
94       update_world,

```

```

95     update_sim_time(C),
96     fail;true ).

```

Der Simulationslauf erfolgt in `run(Counter,End)`. In diesem werden die Simulationszeitpunkte über `tick/0` aufgerufen. Hier kann auch ein Live-Monitoring³ erfolgen. Über `run(End,End)` können gezielt Bedingungen für den letzten Tick in einem Durchlauf festgelegt werden.

```

97 run(End,End):-
98     not(abort_run(true)),
99     message(finish,_),
100    run_id(RunID),
101    export_laws,
102    assertz(finish_state(RunID, completed)).
103
104 run(Counter,End):-
105     Counter<End,
106     not(abort_run(true)),
107     format('~t<< Tick: ~w >>~t~72|~n',[Counter]),
108     Counter1 is Counter + 1,
109     tick,
110     update_sim_time(Counter1),
111     run(Counter1,End).
112
113 run(_X,_Y):-
114     abort_run(true),
115     writeln(' >>> ABORT RUN <<<'),
116     message(finish,_),
117     run_id(RunID),
118     assertz(finish_state(RunID, failed)).

```

Die Aktivitäten der Agenten werden über `update_agents/0` gesteuert. Hier werden zu jedem Tick die Aktivität, Erzeugung und Prüfung angestoßen.

```

119 update_agents:-
120     agent_activity,
121     create_new_agents,
122     update_list_of_active_agents.

```

Die Prozesse innerhalb eines Zeitabschnitts werden dann über `tick/0` gesteuert. Hier wird zu Beginn die Welt über `update_world/0` aktualisiert. Simulationsläufe können

3 Beispielsweise über <https://grafana.com/> oder ähnliche Werkzeuge. In der Simulation kann etwa Grafana direkt über die REST-API angesprochen werden und so etwa die Zahl der aktiven Agenten oder des gerade laufenden Ticks direkt im Ablauf beobachtet werden. Dies bietet gerade bei zeitintensiven Läufen den Vorteil, dass wichtige Parameter direkt beobachtet werden können.

über `update_range/0` stark optimiert und die Laufzeit erheblich verkürzt werden. Doch steigt der Speicherbedarf dabei massiv.⁴ Danach werden die Agenten aktualisiert und können ihre Aktionen ausführen. Zum Schluss eines Ticks wird die Welt auf ihre Konsistenz geprüft. Wenn die Welt zusammenfällt oder zu stark expandiert, wird der Simulationslauf gestoppt und ein nächster gestartet.

```

123 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
124 %%% Simulation: Tick %%%
125 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
126
127 tick:-
128     sim_time(Tick),
129     write('Update World | '),
130     update_world,
131
132     % Precalculation of the reach of each event (high ram consumption)
133     % update_range,
134
135     write('Update Agents | '),
136     update_agents,
137
138     write('Perform Actions | '),
139     perform_actions,
140
141     write('Check World \n\nl'),
142     check_world_consistency(Tick),
143
144     number_of_events_in_tick(Tick, NumberEvents),
145     format('Events: ~w ~t~72|~n', [NumberEvents]),
146     findall(log(_,_,_,_),
147             log(_RunID, Tick, _, law, apply_law, _),Laws),
148     length(Laws, LenLaws),
149     format('Active Law: ~w ~t~72|~n', [LenLaws]).

```

Die Stabilität der Welt wird über einige Tests durch `check_world_consistency/1` geprüft. Dabei wird nach Endlosketten in den Regeln, einem übermäßigen Wachstum der Ereignisse oder dem Zusammenbruch der simulierten Welt geprüft.

```

150 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
151 %%% Consistency test %%%
152 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
153

```

⁴ In den Simulationsexperimenten haben oft die verfügbaren 64GB des Simulationservers nicht ausgereicht, weswegen diese Option nur für kleinere Experimente zu empfehlen ist.

```

154 check_world_consistency(Tick):-
155     active_laws_in_tick(Tick),
156     check_events(Tick),
157     check_short_pattern(Tick),
158     check_laws(Tick).
159
160 check_events(Tick):-
161     number_of_events_in_tick(Tick, NumberEvents),
162     format('Number Events: ~w ~t~72|~n',[NumberEvents]),
163     NumberEvents=0 *-> (abort_run, writeln( ">>> ABORT - NO EVENTS <<"
        ↪ ));true.

```

Es kann in der Simulation zu unerwünschten Rückkopplungen zwischen Gesetzen kommen. Um dies auszuschließen, werden die jeweilig drei letzten Ticks darauf überprüft:

```

164 check_short_pattern(Tick):-
165     law_pattern(Tick)->(abort_run,writeln( ">>> ABORT - SHORT PATTERN
        ↪ <<"));true.
166
167 law_pattern(Tick):-
168     BTick is Tick -1,
169     BBTick is Tick -2,
170     BBBTick is Tick -3,
171
172     abort(Tick,LenActiveLaws),
173     abort(BTick,LenActiveLawsB),
174     abort(BBTick,LenActiveLaws),
175     abort(BBBTick,LenActiveLawsB).
176
177 check_laws(Tick):-
178     laws_repeate(Tick)->(abort_run,writeln( ">>> ABORT - ALL LAWS
        ↪ ACITVE<<"));true.
179
180 active_laws_in_tick(Tick):-
181     findall(log(_,_,_,_),log(_,Tick, _,law, apply_law, _),Laws),
182     length(Laws, LenActiveLaws),
183     asserta(abort(Tick, LenActiveLaws)).
184
185 laws_repeate(Tick):-
186     active_laws_in_timeframe(Tick,LenActiveLaws),
187     abort(Tick,LenActiveLaws).
188
189 active_laws_in_timeframe(Tick,Laws):-
190     BTick is Tick -1,
191     BBTick is Tick -2,

```

```

192     abort(Tick, LenActiveLaws),
193     abort(BTick, LenActiveLaws),
194     abort(BBTick, LenActiveLaws).

```

Eine zentrale Bedingung für einen Abbruch ist die wiederholte Ausführung der gleichen Gesetze, da dies auf eine ungewollte Schleife hindeutet. Darauf testet `compare_active_laws/0`

```

1 compare_active_laws(Tick):-
2     OT is Tick -1,
3     OTT is Tick -2,
4     abort(Tick, LenLaws),
5     abort(OT, LenLaws),
6     abort(OTT, LenLaws),
7     writeln('>>>> RUN ABORTED - UNINTENDING LOOP / ALL LAWS FIRE AT
    ↪ SAME TIME <<<<<').

```

Sollte ein Simulationslauf abgebrochen werden müssen, geschieht dies über `abort_run`
`↪ /0`

```

1 abort_run:-
2     retractall(abort_run(_)),
3     asserta(abort_run(true)).

```

Zu Beginn eines jeden Simulationslaufs finden diverse Initialisierungen statt. Es werden Agenten erzeugt, Überzeugungen zugewiesen, Fakten generiert oder etwa die Welt mit ihren Gesetzen geschaffen.

```

4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 %%% Initialisation %%%
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 create_agents(Number_Of_Agents):-
8     foreach(between(1, Number_Of_Agents, _X), construct_agent).
9
10 assign_beliefs(Number_Of_Beliefs_Per_Agent):-
11     foreach(between(1, Number_Of_Beliefs_Per_Agent, _),
    ↪ random_belief_for_each_agent).

```

Die Initialisierung der Simulation wird mittels `initialisation_run/0` angestoßen. Dabei wird die Simulationszeit gesetzt, Verzeichnisse und Konstanten erzeugt und die Welt geschaffen.

```

1 initialisation_run:-
2     asserta(sim_time(0)),
3     asserta(abort_run(false)),
4     run_id,
5     create_dir(run),
6     create_dir(diary),

```

```

7     create_world,
8     create_domains,
9     check_configuration,
10    init_results(between_ticks).

```

Für die Initialisierung der Agenten wird `initialisation_agents/0` verwendet.

```

1  initialisation_agents:-
2      number_of_agents_at_start(Number_Of_Agents),
3      create_agents(Number_Of_Agents),
4      number_of_facts_at_start(Number_Of_Facts),
5      forall(agent(Type,ID,Attributes),
6             agent(Type,ID,Attributes),List_Of_Agents),
7             update_list_of_active_agents,
8             assertz(actions:publication(dummy, dummy,dummy,dummy,dummy,dummy))
9             ↪ ,
10             create_first_run(List_Of_Agents).
11
12 create_first_run(List):-
13     member(Agent,List),
14     asserta(time(0,0,Agent)),
15     fail;true.
16
17 initialisation_experiment:-
18     experiment_id,
19     prepare_folder,
20     read_name_file.

```

Um die Experimente und Läufe eindeutig zu identifizieren, werden entsprechende Bezeichner über `experiment_id/0` und `run_id/0` erzeugt und vergeben.

```

21 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22 %%% IDs %%%
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 experiment_id:-
25     get_time(TimeStamp),
26     stamp_date_time(TimeStamp,Time,'UTC'),
27     Time=..[_ ,Year,Month,Day,Hour,Min,Sec|P],
28     atomic_list_concat([experiment,Year,Month,Day,Hour,Min,Sec],"-",
29                        ↪ Experiment_id),
30     retractall(experiment_id(_)),
31     asserta(experiment_id(Experiment_id)).
32
33 run_id:-
34     get_time(TimeStamp),
35     stamp_date_time(TimeStamp,Time,'UTC'),
36     Time=..[_ ,Year,Month,Day,Hour,Min,Sec|P],

```



```

35     atomic_list_concat([run,Year,Month,Day,Hour,Min,Sec],"-",Run_ID),
36     retractall(run_id(_)),
37     asserta(run_id(Run_ID)).

```

Nach dem Beenden eines jeden Laufs, werden alle Prädikate in den entsprechenden Namensräumen aus dem Speicher gelöscht. Zu wird sichergestellt, dass es zu keinen Überschneidungen kommt.

```

38     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39     %%% Cleanup %%%
40     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Nach jedem Lauf wird der Speicher geleert. Dies wird mittels `cleanup/0` angestoßen.

```

1 cleanup:-
2     retractall(abort_run(_)),
3     remove_world,
4     retractall(abort(_,_)),
5     retractall(agents:agent(_,_,_)),
6     retractall(actions:belief(_,_,-,-,-)),
7     retractall(fact(_,_,_)),
8     retractall(fact(_)),
9     retractall(actions:model(_,_,_)),
10    retractall(actions:pot_mod(_,_,_)),
11    retractall(actions:theory(_,-,-,-,-)),
12    retractall(actions:data(_,-,-,-,-,-)),
13    retractall(actions:log(_,-,-,-,-,-)),
14    retractall(events_in_range(_,_)),
15    retractall(actions:publication(_,-,-,-,-,-)),
16    retractall(research_domain(_,_)),
17    retractall(sim_time(_)).

```

Durch `write_results/0` werden ausgewählte Ergebnisse eines Simulationslaufs in die jeweiligen Dateien ausgegeben. Es kann auch ein umfangreiches Protokoll mittels `write_results(log)` erzeugt werden, welches sämtliche Vorgänge des Laufs im Detail aufzeichnet.

```

18     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19     %%% Write results %%%
20     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die Ergebnisse eines Laufs werden mit Hilfe von `write_results/0` in dedizierte Dateien geschrieben.

```

1 write_results:-
2     write_results(alter_csv),
3     write_results(agent_statistic_age_csv),
4     write_results(theories),

```

```

5     % write_results(log),
6     write_results(eventsper tick),
7     write_results(overview).

```

7.2 Modul: configuration.pl

```

1  /** <module> Configuration module
2
3  This module contains all the configuration parameter for a simulation
4     ↪ experiment.
5
6  @author Daniel Kurzawe
7  @version ALPHA
8  @see http://www.daniel-kurzawe.de
9  @license GNU/GPL V.3 http://www.gnu.org/copyleft/gpl.html
10 */
11 :-module(configuration,
12     [cores/1,
13     number_of_agents_at_start/1,
14     number_of_beliefs_at_start/1,
15     number_of_facts_at_start/1,
16     number_of_ticks/1,
17     number_of_domains/1,
18     output_filename/1,
19     maxWorkingTime/1,
20     maxFounding/1,
21     maxInvestment/1,
22     maxInterest/1,
23     number_of_runs/1,
24     default_resources/1,
25     action_planing/1,
26     relation_arity/1,
27     number_of_events/1,
28     number_of_proto_objects/1,
29     number_of_properties/1,
30     number_of_objects/1,
31     number_of_object_types/1,
32     number_of_laws/1,
33     number_of_followup_laws/1,
34     number_of_relations/1,
35     number_of_types/1,
36     number_of_ticks_for_prerun/1,

```

```

37     law_complexity/1,
38     maximal_properties_per_type/1,
39     output_subdirectory/1,
40     universe_bounders_2d/2,
41     world_generation/2]).
42
43 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
44 %% System Configuration          %%
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46 :- dynamic(number_of_ticks/1).

```

Für eine Reproduzierbarkeit der Simulationsläufe wird ein *seed* für die Zuweisung der vom Zufallsgenerator bereitgestellten Zufallszahlen verwendet.

```

47 % The seed for the random number generator
48 %:- set_random(seed(111)).

```

Um Simulationen genügend Speicher zur Verfügung zu stellen, wird Prolog erlaubt, einen größeren Speicherbereich zu reservieren. Ebenso werden alle verfügbaren CPU-Kerne für die Simulation reserviert. Dies kann auch hier begrenzt werden. Einige Module erlauben parallele Verarbeitung. Dadurch kann ein Simulationslauf massiv beschleunigt werden, geht aber zu Lasten der Hauptspeichers.

```

49 % Stack definition for old Prolog versions
50 %:- set_prolog_stack(global, limit(1073741824)).
51 %:- set_prolog_stack(local, limit(1073741824)).
52
53 % Stack definition for modern Prolog versions
54 :- set_prolog_flag(stack_limit, 10 000 000 000).
55 :- current_prolog_flag(cpu_count, NumberOfCores),
56     assertz(cores(NumberOfCores)).
57 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
58 %% Agent Configuration          %%
59 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die Zahl der Agenten zum Simulationsstart wird vordefiniert und diese dann zum Beginn eines Simulationslaufs erzeugt. Sollten in *agents.pl* bereits Agenten vorgegeben sein, addiert sich diese Zahl.

```

60 number_of_agents_at_start(20).

```

Gleiches gilt für die Zahl der Überzeugungen der Agenten zum Start. Hier muss zwischen den zwei Simulationsarten mit der Weltengenerierung mit Events oder Objekten unterschieden werden (siehe Kapitel 5.2): Wird eine Welt über Events generiert, werden Überzeugungen nicht über diese Option gesteuert, da diese vollständig dynamisch generiert werden und keine Startbedingung benötigen.

```

61 number_of_beliefs_at_start(0).

```


Die Zahl der vorgenerierten Fakten kann durch `number_of_facts_at_start` festgelegt werden. Diese Fakten addieren sich zu möglichen manuell vordefinierten Fakten.

```
81 number_of_facts_at_start(20).
```

Die Zahl der Ticks vor dem Start der eigentlichen Simulation kann durch `number_of_ticks_for_prerun` angegeben werden. Hierdurch wird sichergestellt. Je nach Simulationsziel macht es Sinn, diese Zahl hinreichend groß zu wählen (über eine Forschergeneration) oder sehr klein, wenn gerade Anfangsbedingungen einer entstehenden Gesellschaft gezeigt werden sollen.

```
82 number_of_ticks_for_prerun(100).
```

Die Zahl der Ticks für einen Simulationslauf wird durch `number_of_ticks` definiert. Diese Zahl addiert sich zu dem Vorlauf der Simulation.

```
83 number_of_ticks(300).
```

`number_of_runs` legt die Zahl der Simulationsläufe für ein Experiment fest.

```
84 number_of_runs(25).
```

Die Welt kann auf zwei Weisen erzeugt werden: Zum einem kann diese vordefiniert sein oder mit Hilfe der folgenden Parameter erzeugt werden. Diese Option kann über `world_generation(generator,no_file)` gesetzt werden.

```
85 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
86 %%% Create world (main configuration) %%%
87 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88 world_generation(generator,no_file).
89 %world_generation(predefined,'world.py').
```

Sofern die Simulation Objekte und Relationen beschreibt, kann die Stufe der Relationen vordefiniert oder durch einen vorgegebenen Wert begrenzt werden.

```
90 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
91 %%% Create world (Object based world) %%%
92 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
93 % highest_arity_for_relations(2).
```

Die maximale Stelligkeit von Relationen zwischen Objekten wird durch `relation_arity/3` beschrieben.

```
94 relation_arity(3).
```

Gleiches gilt für die maximale Zahl von Eigenschaften, welche ein Objekt tragen kann.

```
95 number_of_properties(5).
```


Durch `number_of_events` kann die Zahl der Events definiert werden, die bei der Erzeugung der Welt zu Tick 0 bereitgestellt werden.

```
112 number_of_events(1000).
```

Die maximale Zahl der Typen von Ereignissen kann beschränkt werden. Dieser Wert hat einen unmittelbaren Einfluss auf die Performance der Simulation.

```
113 number_of_types(15).
```

Die Dimensionalität der Welt kann global definiert werden.

```
114 universe_bounders_2d(35,35).
```

```
115
```

```
116 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
117 %%% Output Configuration %%%
```

```
118 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Der Name für die Datei zur Ergebnisübersicht kann manuell festgelegt werden. Weitere Parameter können in `results.pl` definiert werden.

```
119 output_filename('results.csv').
```

Ebenso kann das Verzeichnis, in welchem die Ergebnisdaten gespeichert werden, festgelegt werden.

```
120 output_subdirectory('results').
```

7.3 Modul: `utils.pl`

In diesem Modul werden globale Prädikate definiert, welche an vielen Stellen der Simulation ihre Anwendung finden und nicht einem spezifischen Modul zuzuordnen sind. Die Benennung und Kommentare im Quelltext beschreiben dabei die Funktionalität der Prädikate.

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%% Utilitis %%%
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5 :- module(utils,[
6   agent_age/2,
7   agent_resources/2,
8   agent_reputation/2,
9   find_all_beliefs_for_AgentID/2,
10  check_configuration/0,
11  checking_results_dir/0,
12  cls/0,
13  create_dir/1,
```

```
14  distribution_of_elements/2,
15  export_law_visualisation/0,
16  find_all_actions/1,
17  find_all_agents/1,
18  find_all_facts/1,
19  id/2,
20  in_range/2,
21  in_extended_range/2,
22  length_of_queue/2,
23  list_of_active_agents/1,
24  list_of_inactive_agents/1,
25  log/6,
26  addlog/4,
27  monitor_status/0,
28  multi_intersection/2,
29  multi_append/2,
30  neighborevents/2,
31  number_of_events_in_tick/2,
32  nextto/2,
33  prepare_folder/0,
34  prop_call/2,
35  random_location/2,
36  random_relative_location/1,
37  random_simple_name/1,
38  random_sublist/3,
39  read_name_file/0,
40  start_monitoring/0,
41  update_range/0,
42  update_sim_time/1,
43  relative_to_absolut_location/3,
44  absolut_to_relativ_location/3,
45  remove_list/3,
46  simple_name_to_id/2,
47  shape/2,
48  spend_resources/2,
49  startup_time/6,
50  sublist/2,
51  switch_dir/1,
52  update_attributes/3,
53  update_list_of_active_agents/0,
54  update_number_of_ticks/3,
55  write_tick_results/0)].
56
57 :-dynamic([log/6, list_of_active_agents/1, events_in_range/3]).
58
```



```

59 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
60 %%% Math operations
61 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
62
63 mean_value(List,MeanValue):-
64     sum_list(List,Sum),
65     length(List,Length),
66     MeanValue is div(Sum,Length).
67
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
69 %%% List operations
70 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die folgenden Listenoperationen erleichtern die Auswahl von Elementen oder die Bildung von Ober- und Teilmengen.

Eine zufällige Teilmenge $RandM \subset M$ wird über `random_sublist/2` generiert.

```

71 % Returns a random subset of given list
72 % random_sublist(+Number_of_random_elements, +list, -Random_sublist).
73 random_sublist(0, _, []).
74 random_sublist(Count, List, [X|SelectedFromRemaining]) :-
75     random_member(X, List),
76     select(X, List, RemainingList),
77     C1 is Count - 1,
78     random_sublist(C1, RemainingList, SelectedFromRemaining).

```

Die Schnittmenge $\bigcap_{i=1}^n List_i$ über eine unbestimmte Anzahl von Listen wird über `multi_intersection/2` gebildet.

```

79 multi_intersection([L1|R],Result):-
80     multi_intersection(L1,R,Result).
81
82 multi_intersection(Result,[],Result):-!.
83
84 multi_intersection(L1,[L2|R],Result):-
85     intersection(L1,L2, NResult),
86     multi_intersection(NResult,R, Result).

```

Die Vereinigung $\bigcup_{i=1}^n List_i$ über eine unbestimmte Anzahl von Listen wird entsprechend analog über `multi_union/2` gebildet.

```

87 multi_append([L1|R],Result):-
88     multi_append(L1,R,Result).
89
90 multi_append(Result,[],Result):-!.
91
92 multi_append(L1,[L2|R],Result):-

```

```

93  append(L1,L2, NResult),
94  multi_append(NResult,R, Result).

```

Über `distribution_of_elements/2` lässt sich die Verteilung der Elemente einer Liste zurückgeben.

```

95  distribution_of_elements(List, Distribution):-
96      findall([Number, Element], countall(List,Element,Number), OCList),
97      length(List, LenList),
98      findall([OccurrenceCount, Percentage, BaseElement],
99          ( member([OccurrenceCount, BaseElement],OCList),
100            Percentage is 100/LenList*OccurrenceCount ), Distribution).
101
102  count([],X,0).
103  count([X|T],X,Y):- count(T,X,Z), Y is 1+Z.
104  count([X1|T],X,Z):- X1\=X,count(T,X,Z).
105
106  countall(List,X,C) :-
107      sort(List,List1),
108      member(X,List1),
109      count(List,X,C).

```

Mittels `sublist/2` lässt sich bestimmen, ob eine Liste L_1 Teilliste einer weiteren Liste L_2 ist.

```

110  sublist( [], _ ).
111  sublist( [X|XS], [X|XSS] ) :- sublist( XS, XSS ).
112  sublist( [X|XS], [_|XSS] ) :- sublist( [X|XS], XSS ).

```

Eine gegebene Liste von Elementen L_1 kann mittels `remove_list/3` aus einer Liste L_2 entfernt werden.

```

113  remove_list(L, [], L):- !.
114  remove_list([X|Tail], [X|Rest], Result):- !, remove_list(Tail, Rest,
115      ↪ Result).
116  remove_list([X|Tail], L2, [X|Result]):- remove_list(Tail, L2, Result).
117  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
118  %%% Dynamic listings
119  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die folgenden Prädikate bilden Listen, welche in der Simulation an mehreren Stellen Verwendung finden. Je nach Häufigkeit der Verwendung, werden diese dann auch im Speicher abgelegt, um die Zahl der Operationen zu minimieren.

```

120  find_all_actions(List_of_available_actions):-
121      findall(Actions,action(Action,Agent,Tick),
122      ↪ List_of_available_actions).

```

```

122
123 find_all_agents(List_of_available_agents):-
124     findall(ID,agent(Type,ID,List_of_Attributes),
125             ↪ List_of_available_agents).
126
127 find_all_beliefs_for_AgentID(BelList,AgentID):-
128     sim_time(Tick),
129     findall( belief(Agent, Tick, Type, BeliefContent, Domain, Degree) ,
130             belief(Agent, Tick, Type, BeliefContent, Domain, Degree), BelList).
131
132 find_all_facts(List_of_facts_in_World):-
133     findall(ID,fact(ID),List_of_facts_in_World).
134
135 update_list_of_active_agents:-
136     findall(AgentID,(agent(Type,AgentID,Prop),not(IA=Prop.get(inactive))),
137             ↪ ListOfActiveAgents),
138     retractall(list_of_active_agents(_)),
139     asserta(list_of_active_agents(ListOfActiveAgents)).
140
141 list_of_inactive_agents(ListOfInactiveAgents):-
142     findall(AgentID,(agent(Type,AgentID,Prop),(IA=Prop.get(inactive))),
143             ↪ ListOfInactiveAgents).
144
145 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
146 %%% Length of specific lists
147 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die Länge der Liste geplanter Aktionen wird mittels `length_of_queue/2` bestimmt.

```

148 length_of_queue(AgentID,LOQ):-
149     agent(Type,AgentID, Attributes),
150     Que = Attributes.get(action_queue),
151     length(Queue LOQ).
152
153 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
154 %%% Update agents
155 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Einige spezifische Operationen für Agenten finden an mehreren Stellen der Simulation ihre Anwendung und sind entsprechend global definiert.

Durch `spend_resources/2` werden die Ressourcen eines gegebenen Agenten um einen gegebenen Umfang reduziert.

```

156 spend_resources(AgentID,ResourcesToSpend):-
157     agent(Type,AgentID, Attributes),
158     Resources = Attributes.get(resources),
159     NewResources is Resources - ResourcesToSpend,

```

```
157     update_attributes(AgentID,resources,NewResources) .
```

Weitere Eigenschaften können über `update_attributes/3` aktualisiert werden. Hierfür wird die ID des gewünschten Agenten, der zu aktualisierenden Eigenschaft und des neuen Werts übergeben.

```
158 update_attributes(AgentID,Selection,NewValue):-
159     agent(Type,AgentID, Attributes),
160     retract( agents:agent(Type, AgentID, Attributes) ),
161     assertz( agents:agent(Type, AgentID, Attributes.put(Selection,
162         ↪ NewValue)) ).
163
164 %%% Agent statistics
165 %%%
```

Durch `agent_age/2` kann das Alter eines Agenten bestimmt werden. Das Alter beschreibt die Zahl der vergangenen Ticks seit der Erzeugung des Agenten.

```
166 agent_age(AgentID,Age):-
167     sim_time(Tick),
168     agent(Type,AgentID,Properties),
169     Created = Properties.get(created),
170     Age is Tick - Created.
```

Mithilfe von `agent_resources/2` lassen sich die verfügbaren Ressourcen eines gewählten Agenten einsehen.

```
171 agent_resources(Agent,Resources):-
172     agent(Type,Agent,Properties),
173     Resources = Properties.get(resources) .
```

Die Reputation ist über `agent_reputation/2` zu erfragen.

```
174 agent_reputation(Agent, Reputation):-
175     agent(Type,Agent,Properties),
176     Reputation = Properties.get(reputation) .
177
178 %%% Screen update
179 %%%
```

An einigen Stellen wird der Bildschirm der Kommandozeile neu gezeichnet. Dies geschieht über `cls/0`.

```
181 % cls = clear screen for clean output
182 cls :- write('\e[H\e[2J') .
183
184 %%%
```

```

185 %%% Date and time
186 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Das Dateimanagement für die Ereignisse der Simulationsläufe, markiert diese über Datum und Uhrzeit zum Start. Diese werden über `startup_time/6` bestimmt.

```

187 startup_time(Year,Month,Day,Hour,Minute,Second):-
188     get_time(X),
189     stamp_date_time(X,TimeString,localc),
190     TimeString = . . [_ ,Year,Month,Day,Hour,Minute,Second,_,_,_].

```

Über `update_number_of_ticks/3` werden die Zahl der Ticks mit denen des Simulationsvorlaufs verrechnet.

```

191 update_number_of_ticks(NumberOfTicks,NumberOfTicksPrerun,NewNumberOfTicks
    ↪ ):-
192     retractall(configuration:number_of_ticks(NumberOfTicks)),
193     NewNumberOfTicks is NumberOfTicks + NumberOfTicksPrerun,
194     asserta(configuration:number_of_ticks(NumberOfTicks)).

```

Die Simulationszeit, also der aktuell vorherrschende Tick, wird über `update_sim_time/1` aktualisiert.

```

195 update_sim_time(NewTime):-
196     retractall(sim_time(_)),
197     assert(sim_time(NewTime)).
198
199 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
200 %%% Simulation - File Output %%%
201 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Für die Speicherung der Simulationsdaten werden Ordner angelegt und Dateien geschrieben. Die dafür notwendigen Operationen werden hier definiert.

```

202 %%% write tick results to file %%%
203 write_tick_results:-
204     output_filename(Filename),
205     open(Filename,write,Stream),
206     write(Stream, "Test"),
207     close(Stream).
208
209 % Check if results dir exists
210 prepare_folder:-
211     working_directory(SimDir,SimDir),
212     asserta(sim_dir(SimDir)),
213     checking_results_dir,
214     create_dir(experiment).
215

```

```

216 % Create folder for result files
217 checking_results_dir:-
218     exists_directory('./results');
219     make_directory('./results').
220
221 % Create folder for running experiment
222 create_dir(experiment):-
223     experiment_id(Experiment_id),
224     concat('./results/',Experiment_id,ExDir),
225     asserta(experiment_dir(ExDir)),
226     make_directory(ExDir).
227
228 % Create folder for single runs
229 create_dir(run):-
230     experiment_id(ExperimentID),
231     run_id(RunID),
232     format(atom(RunDir),'./results/~w/~w',[ExperimentID, RunID]),
233     asserta(run_dir(RunDir)),
234     make_directory(RunDir).
235
236 create_dir(diary):-
237     experiment_id(ExperimentID),
238     run_id(RunID),
239     format(atom(Diary),'./results/~w/~w/~w',[ExperimentID, RunID,'diary']),
240     asserta(diary_dir(Diary)),
241     make_directory(Diary).
242
243 switch_dir(diary):-
244     diary_dir(Diary),
245     simulation:working_directory(_, Diary).
246
247 switch_dir(run):-
248     run_dir(RunDir),
249     simulation:working_directory(_, RunDir).
250
251 switch_dir(simulation):-
252     sim_dir(SimDir),
253     simulation:working_directory(_,SimDir).
254
255 switch_dir(experiment):-
256     experiment_dir(ExDir),
257     simulation:working_directory(_,ExDir).
258
259 addlog(AgentID, Type, Content, Value):-
260     sim_time(Tick),

```

```

261     run_id(RunID),
262     assertz(log(RunID, Tick, AgentID, Type, Content, Value)).
263
264
265     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
266     %%% Monitoring with influxDB and Grafana %%%
267     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die Simulation ermöglicht es influxDB und Grafana für die Visualisierung von Simulationsläufen zu verwenden. Die notwendigen API-Aufrufe sind hier beschrieben.

```

268 % Optional: Enables live monitoring with influxDB and Grafana
269 monitor_status:-
270 %     thread_pool_property(main,backlog(B)),
271 %     writeln(B),
272     find_all_agents(Agents),
273     length(Agents, Number_Of_Agents),
274     atom_concat('simulation,run=ID,tick=1 agents=',Number_Of_Agents,
275               ↪ Atom),
276     http_post( [protocol(http),
277               host('192.168.0.11'),
278               port(8086),
279               path('/write?db=mydb')],
280             atom(Atom),
281             _R, []).
282
283 start_monitoring:-
284     thread_create_in_pool(main, monitor_status, _ID, [stack_limit(2
285               ↪ 000 000 000)] ).
286
287     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
288     %%% Simulation - Basic failure tests %%%
289     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die hier beschriebenen Tests prüfen, ob einige Kernbestandteile richtig konfiguriert sind und notwendige Verzeichnisse angelegt sind.

```

288 check_configuration:-
289     check_file_names,
290     check_action_planing.
291
292 check_file_names:- true.
293 /*
294     ( output_filename('results.csv'),
295       output_subdirectory('results') ) ;
296     writeln('*** RESULT FILE NOT SET ***').
297 */

```

```

298
299 check_action_planing:- true.
300 /*
301     action_planing(random),
302     action_planing(strategy),
303     retract_all(action_planing(strategy)),
304     writeln('*** CONFLICT IN STRATEGY - WILL USE RANDOM DEFINITION
           ↪ ***').
305 */
306
307 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
308 %%% Events - Shapes, locations, ect. %%%
309 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die simulierte Welt kann auch im Terminal als ASCII gezeichnet werden. Diese Funktion ist jedoch nur bei sehr kleinen Welten sinnvoll zu verwenden.

```

310 random_location(X,Y):-
311     universe_bounders_2d(MaxX,MaxY),
312     random(0,MaxX,X),
313     random(0,MaxY,Y).
314
315 draw_events:-
316     sim_time(Tick),
317     universe_bounders_2d(MaxX,MaxY),
318     between(0,MaxX,C1),write('\n'),format('X~|~`0t~w~5| | ', C1),
319     ( between(0,MaxY,C2),
320         (draw_event(Tick,C1,C2) )
321         ),fail;true.
322
323 draw_event(Tick, X,Y):-
324     create_world_module:event(_ID, _Type, Tick, [X, Y]) -> write('0');
           ↪ write('+').
325
326 export_law_visualisation:-
327     findall([A,B], lawtree(A,B), Chain),
328     run_id(Run_ID),
329     switch_dir(run),
330     open("lawtree_visualisation.csv",write,Stream),
331     csv_write_stream(Stream, [ row('Source', 'Target') ], [ ] ),
332     forall(member(Pair,Chain),
333         ( Pair = [A,B], csv_write_stream(Stream, [ row(A, B) ], [ ] ))
           ↪ ,
334     close(Stream),
335     switch_dir(simulation).

```


Die weiteren Prädikate werden zur Generierung von Ereignissen und dem Update der Welt zu jedem Tick verwendet. Diese sind notwendig, um etwa Triggerbedingungen für Ereignisse zu prüfen oder diese auszuführen.

```

336 in_range(Base,List):-
337     Base = [X,Y],
338     MX is X-1,
339     PX is X+1,
340     PY is Y+1,
341     MY is Y-1,
342     List = [ [MX,MY], [MX,Y], [MX, PY],
343             [X,MY], [X,Y], [X, PY],
344             [PX,MY], [PX,Y], [PX, PY] ].
345
346 in_extended_range(Base,List):-
347     Base = [BX,BY],
348     LowX is BX -2,
349     LowY is BY -2,
350     HX is BX +2,
351     HY is BY +2,
352     findall([X,Y],(between(LowX,HX,X), between(LowY,HY,Y)),List).
353
354 nextto(ID1,ID2):-
355     create_world_module:event(ID1,Type1,Tick,C1),
356     create_world_module:event(ID2,Type2,Tick,C2),
357     in_range(C1,Range1),
358     sublist([C2],Range1).
359
360 neighborevents(ID,Neighborevents):-
361     findall(ID2, nextto(ID,ID2),Neighborevents ).

```

Für die Optimierung der Laufzeit können Events durch `update_range/0` vorberechnet werden. Dies geht jedoch zu Lasten des Speichers und sollte mit Bedacht eingesetzt werden. Bei größeren Simulationen oder einer hohen Anzahl von Events, kann der Speicher schnell volllaufen, da Relationen zwischen Events exponentiell steigen können.

```

364 /*
365 update_range:-
366     %retractall(utils:events_in_range/3),
367     sim_time(Tick),
368     writeln('... update range'),
369     forall(create_world_module:event(ID1,Type1,Tick,C1),
370           (in_range(C1,Range),
371            assertz( events_in_range(C1,Range,Tick) ) ) ).
372 */

```

```

373
374 /* %%% Sequentielle Version %%%
375 update_range:-
376     %retractall(utils:events_in_range/3),
377     sim_time(Tick),
378     writeln('... update range'),
379     forall(create_world_module:event(ID1,Type1,Tick,C1),
380         (in_range(C1,Range),
381             assertz( events_in_range(C1,Range,Tick) ) ) ).
382 */
383
384 update_range.
385
386 /*
387 update_range:-
388     writeln('... update range list'),
389     %retractall(utils:events_in_range/3),
390     sim_time(Tick),
391     retractall(utils:events_in_range(_,_,_)),
392     findall([ID1,Tick,Location],create_world_module:event(ID1,Type1,Tick,
393         ↪ Location), EventList),
394     concurrent_maplist(update_range_events, EventList),
395     %forall(member(EV, EventList), update_range_events(EV)),
396     writeln('... finish update range list').
397
398 update_range_events(Event):-
399     Event = [ID1,Tick,Location1],
400     in_range(Location1,Range),
401     findall(ID2, (member(Location, Range), create_world_module:event(ID2,
402         ↪ Type2, Tick, Location2)), List ),
403     assertz( utils:events_in_range(ID1,List,Tick) ).
404 */
405
406 number_of_events_in_tick(Tick, NumberEvents):-
407     findall(ID1,create_world_module:event(ID1,Type1,Tick,C1),List),
408     length(List,NumberEvents).

```

Durch `relative_to_absolut_location/3`. können absolute Positionen zu relativen Koordinaten in die jeweiligen Bezugssysteme umgerechnet werden.

```

408 relative_to_absolut_location(RelativeLocation,BaseLocation,Location):-
409     BaseLocation = [BaseX,BaseY],
410     RelativeLocation = [RelativeX, RelativeY],
411     TargetX is BaseX + RelativeX,
412     TargetY is BaseY + RelativeY,

```



```

448 %%% Other %%%
449 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Mittels `prop_call/2` lässt sich ein Aufruf bzw. eine Prolog-Abfrage zu einer gegebenen Wahrscheinlichkeit auslösen.

```

450 % executes predicate with defined probability
451 prop_call(Prop, Call):-
452     Prop = [H,E],
453     Ex is E+1,
454     random(H,Ex,P),
455     P is (E)-> call(Call);
456     true.

```

7.4 Modul: create_world_event.pl

```

1
2 /** <module> Create world (Events)
3     This module creates a world, consisting of events and laws
4     */
5
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 %%%                               Module create_world
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9 % This module creates a world, consisting of events and laws
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 %%%                               Create a world with
12 %%% "initialisation_experiment,initialisation_run,create_world."
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 %%% Module configuration %%%
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19 :- module(create_world_module,
20     [ create_world/0,
21       remove_world/0,
22       update_world/0,
23       lawtree/2,
24       lawtree/0,
25       list_of_event_types/1
26     ] ).
27
28 :- dynamic( [

```

```

29     lawtree/2,
30     eventtype/2,
31     event/4,
32     list_of_event_types/1,
33     ids_of_all_law_preconditions/1,
34     ids_of_all_law_targets/1,
35     state_of_the_universe/2,
36     types_in_universe/2
37 ] ).

```

```

38 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39 %%% Create world %%%
40 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Über `create_world_stand_alone/0` wird eine Welt ohne spezifische Parameter erzeugt. Dies ermöglicht auch die Erzeugung einer Welt außerhalb eines dedizierten Simulationslaufs.

```

1 create_world_stand_alone:-
2     initialisation_experiment,
3     initialisation_run,
4     create_world.

```

Das Prädikat `create_world/0` erzeugt die Welt. Es werden die Konfigurationen geladen, initiale Ereignisse und Gesetze erzeugt und dann Listen vorbereitet, um spätere Abfragen zu beschleunigen.

```

1 create_world:-
2     init_config,
3     event_generator,
4     sow_the_seeds_for_laws_and_events,
5     create_lawtree,
6     export_law_visualisation,
7     performance_optimisation.
8
9 event_generator:-
10    world_generation(Type,File),
11    event_generator(Type, File).
12
13
14 event_generator(generator, _):-
15    create_event_types,
16    create_initial_events,
17    create_laws.
18
19 event_generator(predefined, File):-
20    include(File).

```

Die Optimierung über `performance_optimisation/0` erzeugt Listen für häufig verwendete deterministische Abfragen.

```

1 performance_optimisation:-
2     find_IDs_of_all_law_preconditions(_),
3     find_IDs_of_all_law_targets(_).

```

Ereignistypen werden mittels `create_event_types/0` geschaffen und eine Liste aller Typen angelegt.

```

1 create_event_types:-
2     findall(Goal, create_event_types(Goal), Goals),
3     cores(Cores),
4     concurrent(Cores,Goals,[] ),
5     update_list_of_event_types.
6
7 create_event_types(Goal):-
8     number_of_types(NumberOfTypes),
9     ( between(1,NumberOfTypes,C),
10    Goal = create_event_type).

```

Die initialen Ereignisse zu t_0 werden mittels `create_initial_events/0` erzeugt. Auf dieser Grundlage werden dann auch die Gesetze erzeugt.

```

1 create_initial_events:-
2     findall(Goal, create_initial_events(Goal), Goals),
3     cores(Cores),
4     concurrent(Cores,Goals,[] ).
5
6 create_initial_events(Goal):-
7     number_of_events(NumberOfEvents),
8     ( between(1,NumberOfEvents,C),
9     Goal = create_event ).

```

Durch `create_laws/0` werden die Gesetze geschaffen. Diese transferieren die Welt von einem Zustand zum Zeitpunkt t_n in den Folgezustand t_{n+1} .

```

1 create_laws:-
2     findall(Goal, create_laws(Goal), Goals),
3     cores(Cores),
4     concurrent(Cores,Goals,[] ).
5
6 create_laws(Goal):-
7     number_of_laws(NumberOfLaws),
8     ( between(1,NumberOfLaws,C),
9     Goal = create_law).

```

Die Gesetze bilden eine Baumstruktur. Diese ergibt sich über die Bezugnahme durch Ausgangs- und Zielzustände. Mittels `create_lawtree/0` wird ein entsprechender Baum aufgebaut.

```

1 create_lawtree:-
2     findall(Goal, create_lawtree(Goal), Goals),
3     cores(Cores),
4     concurrent(Cores,Goals,[] ).
5
6 create_lawtree(Goal):-
7     number_of_followup_laws(NumberOfLaws),
8     ( between(1,NumberOfLaws,C),
9       Goal = lawtree).
10
11 init_config:- true.
12
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 %%% Remove world - Will be called after every run %%%
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Zum Ende eines Simulationslaufs wird die Welt vollständig aus dem Speicher entfernt. Dies geschieht mittels `remove_world/0`.

```

1 remove_world:-
2     remove_config,
3     remove_event_types,
4     remove_events,
5     remove_listings,
6     remove_laws.
7
8 remove_config:-
9     retractall(lawtree(_,_)).
10
11 remove_events:-
12     retractall(event(_,_,_)).
13
14 remove_event_types:-
15     retractall( eventtype(_,_ ) ),
16     retractall( state_of_the_universe(_,_ ) ),
17     retractall( types_in_universe(_,_ ) ).
18
19 remove_laws:-
20     retractall(ids_of_all_law_preconditions(_ ) ),
21     retractall(ids_of_all_law_targets(_)),
22     retractall(law_pre(_,_)),
23     retractall(law_target(_,_)).

```

Zusätzlich können auch die Ereignisse direkt aus dem Speicher gelöscht werden, was jedoch Einfluss auf die Simulationsauswertung haben kann.

Dazu steht `remove_old_universe/0` zur Verfügung.

```

1 remove_old_universe:-
2     sim_time(Tick),
3     retractall(event(_ID,_Type,Tick)).
4
5 remove_listings:-
6     retractall(list_of_event_types(_)).
7     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8     %% Update world - will be called once every tick %%
9     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Zu jedem Tick wird die Welt mittels `update_world/0` aktualisiert. Zunächst werden die Listen zum Zustand und der Typen aktualisiert und dann die Regeln angewendet. Optional können dann alte Zustände zu t_{n-1} aus dem Speicher gelöscht werden.

```

1 /*
2 update_world:-
3     time(state_of_the_universe(_State)),
4     time(types_in_universe(_Types)),
5     time(update_universe),
6     time(remove_old_universe), writeln('*****').
7 */
8
9 update_world:-
10    state_of_the_universe(_State),
11    types_in_universe(_Types),
12    update_universe,
13    remove_old_universe.
14
15 update_universe:-
16    cores(Cores),
17    findall(Goal,create_world_module:update_universe(Goal),Goals),
18    concurrent(Cores, Goals, []).
19
20 update_universe(Goal):-
21    universe_bounders_2d(BounderX,BounderY),
22    between(1, BounderX, C1),
23    ( between(1, BounderY, C2),
24      ( % thread_create_in_pool(threadpool , check_shapes([C1,C2]), _ ,
25        ↪ [])
26        % check_shapes([C1,C2])
27        Goal = check_shapes([C1,C2])
28      )
29    ).

```



```

28
29 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30 %%% Listings %%%
31 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

An diversen Stellen der Simulation werden Listen zum Zustand der Welt abgerufen. Diese Informationen zu den Regeln, Ereignissen, deren Typen und Positionen sind rechenintensiv und werden deswegen jeweils vorgehalten und nur bei Veränderung aktualisiert. Dies geschieht mittels der folgenden Prädikate.

```

32 update_list_of_event_types:-
33     retractall(list_of_event_types(_),
34     findall(ID, eventtype(ID,_), List),
35     asserta(list_of_event_types(List)).
36
37 state_of_the_universe(State):-
38     sim_time(Tick),
39     findall([ID, Type], event(ID,Type,Tick, Location), State),
40     asserta(state_of_the_universe(State, Tick)).
41
42 state_of_the_universe_with_location(State):-
43     sim_time(Tick),
44     switch_dir(run),
45     open('event_location.csv', append, Stream),
46     csv_write_stream(Stream,[row("ID","Type","Tick","X","Y")], [] ),
47     forall(event(ID,Type,Tick, [X,Y]),
48     csv_write_stream(Stream,[row(ID,Type,Tick,X,Y)], [] )),
49     close(Stream),
50     switch_dir(simulation).
51
52 types_in_universe(TypesSet):-
53     sim_time(Tick),
54     findall(Type, event(ID,Type,Tick, Location), Types),
55     list_to_set(Types,TypesSet),
56     asserta(types_in_universe(TypesSet, Tick)).
57
58 find_all_preconditions(PreLaws):-
59     findall([ID,Precondition], law_pre(ID,Precondition), PreLaws).
60
61 find_IDs_of_all_law_preconditions(ListOfLawPre):-
62     retractall(ids_of_all_law_preconditions(_)),
63     findall(ID, law_pre(ID,Content), ListOfLawPre),
64     assertz(ids_of_all_law_preconditions(ListOfLawPre)).
65
66 find_IDs_of_all_law_targets(ListOfLawT):-
67     retractall(ids_of_all_law_targets(_)),

```

```

68     findall(ID, law_target(ID,Content), ListOfLawT),
69     assertz( ids_of_all_law_targets(ListOfLawT) ).
70
71     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
72     %% Create types and events %%
73     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Ereignisse und Ereignistypen bestehen aus einem Bezeichner (ID) und einer optionalen Liste weiterer Eigenschaften. Zu Beginn der Simulation werden diese einmalig mittels `create_event_type/0` und `create_event/0` erzeugt. Dabei wird eine Anzahl von Typen und dann Ereignissen mit entsprechenden Typen erzeugt. Die Ereignisse erhalten dann über `random_type/1` und `random_location/2` zufällige Typen und Orte.

```

1  create_event_type:-
2      id("EVENTTYPE",ID),
3      asserta(eventtype(ID,_Options)).
4
5  create_event:-
6      id("EVENT",ID),
7      simulation:sim_time(Tick),
8      random_type(Type),
9      random_location(X,Y),
10     asserta(event(ID, Type, Tick, [X,Y])).
11
12  create_specific_event(Type,Tick):-
13     id("EVENT",ID),
14     random_location(X,Y),
15     assertz(event(ID, Type, Tick, [X,Y])).
16
17  random_type(Type):-
18     list_of_event_types(ListOfTypes),
19     random_member(Type, ListOfTypes).
20
21     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22     %%% Create Laws %%%
23     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Gesetze bzw. Regeln werden über `create_law` erzeugt.

```

1  create_law:-
2      base_law(LawID, Types, Complexity),
3      % random_sublist(Complexity, Types, Precondition),
4      % random_sublist(Complexity, Types, TargetState),
5      random(0, Complexity, ComplexityPre),
6      random(0, Complexity, ComplexityTarget),

```

```

7   findall(ElementPre, ( between(0,ComplexityPre, C),build_element(
      ↪ ElementPre) ), ElementsPre),
8   assertz(law_pre(LawID,ElementsPre)),
9   findall(ElementTarget, ( between(0,ComplexityTarget, C),build_element
      ↪ (ElementTarget) ), ElementsTarget),
10  assertz(law_target(LawID,ElementsTarget)).

```

Für die Ausgangs- und Zielzustände werden sogenannte Elemente verwendet. Diese beschreiben einen Objekttypen zu einer relativen Position. Ein Element wird mittels `build_element/0` geschaffen.

```

1  build_element(Element):-
2    list_of_event_types(Types),
3    random_member(Type,Types),
4    random_relative_location(Location),
5    Element = [Type,Location].

```

Der Abhängigkeitsbaum der Regeln wird über `lawtree/0` erzeugt. Dabei wird eine Kette von Gesetzen über Ausgangs- und Zielzuständen aufgebaut. Dazu wird zunächst ein Ausgangsgesetz erzeugt und dann eine Verkettung aufgebaut.

```

1  lawtree:-
2    base_law(LawID, Types, Complexity),
3    findall([TID,TState], law_target(TID,TState), ChainDomain),
4    random_member(ChainTuple, ChainDomain),
5    ChainTuple = [SourceID, ChainPrecondition],
6    assertz( lawtree(SourceID, LawID) ),
7    assertz(law_pre(LawID,ChainPrecondition)),
8
9    random(0, Complexity, ComplexityTarget),
10   findall(ElementTarget, ( between(0,ComplexityTarget, C),build_element
      ↪ (ElementTarget) ), ElementsTarget),
11   assertz(law_target(LawID,ElementsTarget)).

```

Das Ausgangsgesetz wird über `base_law/3` erzeugt.

```

1  base_law(LawID, Types, Complexity):-
2    uuid(UUID),
3    concat("LAW",UUID,LawID),
4    list_of_event_types(Types),
5    law_complexity(MaxComplexity),
6    random(1,MaxComplexity,Complexity).

```

Für die Anwendung der Gesetze werden Punkte in der Welt auf den vom Gesetz bestimmten Ausgangszustand getestet. Dieser Zustand wird als Shape bezeichnet.

```

7  check_shapes(BaseLocation):-
8    BaseLocation = [X,Y],

```

```

9     ids_of_all_law_preconditions(ListOfLawPre),
10    foreach( member(LawID,ListOfLawPre), check_law_shape(LawID,
        ↪ BaseLocation)).

```

Über `check_law_shape/2` wird geprüft, ob sich ein Gesetz auf eine gegebene Örtlichkeit anwenden lässt. Wenn der Ausgangszustand für dieses Gesetz an einem Ort erfüllt ist, wird der Zielzustand herbeigeführt und das Gesetz somit ausgeführt.

```

1  check_law_shape(LawID,BaseLocation):-
2     law_pre(LawID,Content),
3     forall(member(Element, Content), check_element(Element,BaseLocation))
        ↪ ->
4     apply_law_location(LawID, BaseLocation); true.
5
6  check_element(Element, BaseLocation):-
7     Element = [Type, RelativeLocation],
8     sim_time(Tick),
9     relative_to_absolut_location(RelativeLocation,BaseLocation,Location),
10    event(ID, Type, Tick, Location),
11    retract(event(ID, Type, Tick, Location)).
12
13  apply_law_location(LawID, BaseLocation):-
14    law_target(LawID, Content),
15    addlog(LawID, apply_law, _ , _),
16    sim_time(Tick),
17    addlog(LawID, law, apply_law,Tick),
18    forall(member(Element,Content), create_event_with_location(Element,
        ↪ BaseLocation)).
19
20  create_event_with_location(Element, BaseLocation):-
21    Element = [Type, RelativeLocation],
22    sim_time(Tick),
23    NewTick is Tick + 1,
24    id("event",EventID),
25    relative_to_absolut_location(RelativeLocation,BaseLocation,Location),
26    addlog(EventID, event, create_event_with_location,_),
27    assertz(event(EventID, Type, NewTick, Location)).
28
29  create_event_with_location_actual_tick(Element, BaseLocation):-
30    Element = [Type, RelativeLocation],
31    sim_time(Tick),
32    id("event",EventID),
33    relative_to_absolut_location(RelativeLocation,BaseLocation,Location),
34    addlog(EventID, event, create_event_with_location,_),
35    assertz(event(EventID, Type, Tick, Location)).
36

```

```

37 sow_the_seeds_for_laws_and_events:-
38     findall(Prelaw, law_pre(ID, Prelaw), IDs),
39     forall(member(ID, IDs), apply_prelaw(ID)).
40
41 apply_prelaw(ListPreCond):-
42     law_pre(ID, Content),
43     random_location(X,Y),
44     BaseLocation = [X,Y],
45     forall(member(Element,Content),
46         ↪ create_event_with_location_actual_tick(Element, BaseLocation)).
47     ↪

```

7.5 Modul: agents.pl

```

1
2 /** <module> Agents definition
3
4 */
5
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 %%% Prolog Database and definition: Agents %%%
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
9 :-module(agents,
10     [ agent/3,
11       belief/6,
12       create_domains/0,
13       construct_agent/0,
14       random_belief_for_each_agent/0,
15       specific_belief_for_set_of_agents/2,
16       create_new_agents/0,
17       create_new_agents_p/0,
18       check_if_agent_still_active/1,
19       agent_active/1,
20       agent_activity/0,
21       check_agent_active/2
22     ]).
23
24 :-dynamic([belief/6,agent/3,research_domain/2]).

```

Mittels `create_new_agents/0` wird eine zufällige Anzahl neuer Agenten erzeugt.

```

1 create_new_agents:-
2     random(1,10,Rnd),

```

```

3     foreach(between(0,Rnd,_),create_new_agents_p).
4
5 create_new_agents_p:-
6     random(0,10,Rnd),
7     ( Rnd > 3,
8     construct_agent );
9     true.
10
11 agent_activity:-
12     list_of_active_agents(ListOfActiveAgents),
13
14     forall(member(AgentID,ListOfActiveAgents), agent_active(AgentID)).

```

Über `check_if_agent_still_active/1` wird geprüft, ob der gegebene Agent im Aktuellem Tick noch aktiv ist oder bereits als inaktiv markiert wurde.

```

1 check_if_agent_still_active(Agent):-
2     agent(Type,Agent,Properties),
3     ( Inactive = Properties.get(inactive) -> (fail);
4     (true) ).

```

Das Alter eines gegebenen Agenten wird über `agent_age/2` bestimmt. Dabei bezeichnet *Alter* die Zahl der Ticks, welche der Agent seit seiner Erzeugung bis zum Moment der Abfrage aktiv ist.

```

1 agent_age(Agent,Age):-
2     sim_time(Tick),
3     agent(Type,Agent,Properties),
4     Created = Properties.get(created),
5     Age is Tick - Created.

```

Die folgenden Prädikate prüfen, ob der Agent aktiv ist oder aufgrund von gesetzten Parametern als inaktiv gekennzeichnet wird.

```

7 set_agent_inactive(Agent):-
8     agent(Type,Agent,Properties),
9     retract(agent(Type,Agent,Properties)),
10    sim_time(Tick),
11    NewProp = Properties.put(inactive,Tick),
12    assertz(agent(Type,Agent,NewProp)).
13
14 agent_active(Agent):-
15    agent_age(Agent,Age),
16    (check_agent_active(Agent, Age), fail; true).
17
18 /*check_agent_active(Agent, Age):-
19    Age > 45 -> set_agent_inactive(Agent); true.*/

```

Agenten, welche weniger als 6 oder mehr als 44 Ticks aktiv sind, werden dabei unterschieden und anders gewichtet.

```

21 check_agent_active(Agent, Age):-
22     Age < 6 *->
23     (
24         random(0,100,Random),
25         Random < 20,
26         set_agent_inactive(Agent)
27     );
28     Age >= 45 *->
29     (
30         Probability is (0.05*(Age-45)^2+55),
31         random(0,100,Random),
32         Random > Probability,
33         set_agent_inactive(Agent)
34     );
35     (
36         Probability is (0.5*Age),
37         random(0,100,Random),
38         Random < Probability,
39         set_agent_inactive(Agent)
40     ).
41
42 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
43 %%% Construct Agents %%%
44 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Über `construct_agent/0` wird ein Agent ohne spezifische Eigenschaften erzeugt. Dieser erhält dann einen eindeutigen Bezeichner und zufällige Eigenschaften.

```

1 % Constructing simple agent; no special attributes
2 construct_agent:-
3     select_agent_type(AgentType),
4     id("AGENT", AgentID),
5     random_simple_name(SimpleName),
6     sim_time(Tick),
7     default_resources(Res),
8     select_random_domain(DomainID),
9     determine_interests(DomainID, Interests),
10    random(1,10,RndAbility), Ability is (RndAbility/10),
11    assertz( agent(AgentType,AgentID,properties{created:Tick, resources:Res,
        ↪ reputation:1.0, ability:Ability, action_queue:[],
        ↪ research_domain:[DomainID], list_of_interests: Interests,
        ↪ simplename:SimpleName} ) ).

```

Dabei bestimmt `select_agent_type/1` den Agententypen. Diese können durch `type` \leftrightarrow /1 definiert werden.

```

1 select_agent_type(AgentType):-
2   list_of_types(List),
3   random_member(AgentType,List).

```

Das Forschungsinteresse wird über `determine_interests/2` festgelegt. Dabei werden eine Zahl von Interessensgebieten durch Ereignistypen beschrieben.

```

1 determine_interests(DomainID, Interests):-
2   research_domain(DomainID, EventTypes),
3   length(EventTypes, NumberOfTypes),
4   maxInterest(MaxInterest),
5   NumberIntersts is round( ( NumberOfTypes / 100 ) * MaxInterest) +1,
6   random_sublist(NumberOfTypes,EventTypes, Interests).
7
8 % provides list of types; types might be dynamic
9 list_of_types(List):-
10  findall(Type,agents:types(Type),List).
11
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 %%% Gives Agents set of random beliefs for initialisation %%%
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Zum Start eines Simulationslaufs erhält jeder der initialen Agenten eine Überzeugung.

```

15 % gives each existing agent a random belief; used for initalisation
16 random_belief_for_each_agent:-
17   find_all_agents(List_of_Agents),
18   find_all_facts(List_of_Facts),
19   % Give every Agent a random belief
20   ( member(Agent,List_of_Agents),
21     %random_element_from_list(List_of_Facts,FactID),
22     random_member(FactID, List_of_Facts),
23     % uuid(UUID),
24     sim_time(Tick),
25     assertz( belief(Agent, Tick, fact, BeliefContent, Domain, Degree) ),
26     fail ; true
27   ).
28
29 % gives a specific set of agents a random belief
30 specific_belief_for_set_of_agents(Set_Of_AgentIDs,FactID):-
31   ( member(Agent,Set_Of_AgentIDs),
32     sim_time(Tick),
33     assertz( belief(Agent, Tick, fact, FactID, Domain, Degree) ),

```



```

34     fail ; true
35   ).
36
37 random_belief_for_agent(AgentID):-
38   find_all_facts(List_of_Facts),
39   %random_element_from_list(List_of_Facts,FactID),
40   random_member(FactID,List_of_Facts),
41   sim_time(Tick),
42   assertz( belief(AgentID, Tick, ContentType, FactID, Domain, Degree) ),
43   fail ; true.
44
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46 %%% Agent Types %%%
47 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Durch `types/1` werden die Typen von Agenten definiert.

```

1 types(student).
2 types(researcher).
3 types(funder).

```

Die Liste von Überzeugungstypen wird durch `belief_types/1` beschrieben.

```

1 belief_types(Types):-
2   Types = [fact, hypothesis, observation, publication].
3
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 %%% Research Domain %%%
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Forschungsbereiche werden über `research_domain/2` beschrieben. Ein Forschungsbereich definiert sich dabei aus einer Liste von Ereignistypen, welche Gegenstände dieser Disziplin sind. Die folgenden Prädikate dienen entsprechender Erzeugung.

```

1 create_domains:-
2   number_of_domains(NumberOfDomains),
3   between(0,50, C),
4   (create_random_research_domain(Domain)),fail;true.
5
6 create_random_research_domain(Domain):-
7   list_of_event_types(EventTypes),
8   number_of_types(Types),
9   random(1,Types,Random),
10  random_sublist(Random, EventTypes, Sublist),
11  create_research_domain(Sublist,Domain).
12
13 create_research_domain(ListOfTypes, Domain):-
14  id("RESDOM", DomainID),

```

```

15   Domain = research_domain(DomainID, ListOfTypes),
16   asserta(Domain).
17
18 select_random_domain(Domain):-
19   findall(DomainID, research_domain(DomainID, ListOfTypes), Domains ),
20   random_member(Domain, Domains).

```

7.6 Modul: actions.pl

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% This module contains the action definition
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  :- module(actions,
6     [
7     data/7,
8     get_interest/2,
9     perform_actions/0,
10    plan/2,
11    plan_actions/0,
12    plan_action/1,
13    perform_action/1,
14    publication/6,
15    series/2,
16    find_independent_data_series/2,
17    theory/6 ]).
18
19 :- use_module('action_definition.pl').
20 %:- use_module('agents.pl').
21
22 :- dynamic([publication/6, traitor/2, action_queue/2, data/7, log/6,
23            ↪ model/3, pot_model/3, theory/6, publication_rejected/5, citation
24            ↪ /3]).
25
26 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27 %% Basic functions
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Über `find_valid_actions_for_agent/2` wird eine Liste möglicher Aktionen für einen gegebenen Agenten erzeugt.

```

1 find_valid_actions_for_agent(Agent, Actions):-
2   agent(Type, Agent, _Attributes),

```

```

3     findall(Action, ( action_type(Action,Types), member(Type,Types) ),
        ↪ Actions).

```

Aktionen werden mittels `add_action_to_queue/2` in eine Liste der geplanten Aktionen des Agenten eingefügt.

```

1 add_action_to_queue(AgentID, Action):-
2     sim_time(Tick),
3     ActionContent = execute_action(Action,AgentID,Tick),
4     agent(Type, AgentID, Attributes),
5     ActionQueue = Attributes.get(action_queue),
6     append([ActionContent],ActionQueue,NewActionQueue),
7     update_attributes(AgentID,action_queue,NewActionQueue).

```

Mittels `get_interest/2` wird ein zufälliges Interessensgebiet des Agenten aus einer Liste von Interessensgebieten gewählt.

```

1 get_interest(AgentID, EventType):-
2     agent(Type,AgentID, Attributes),
3     Interests = Attributes.get(list_of_interests),
4     random_member(EventType,Interests).

```

Bisher von dem Agenten erfasste Daten können über `dump_all_data/1` verworfen werden. Dies ist etwa nützlich, wenn ein Agent mit den bisher erhobenen Daten keine Ergebnisse erzielt hat und einen neuen Ansatz wählt oder das Forschungsinteresse wechselt und die bisher erhobenen Daten nicht verwertet werden können.

```

1 dump_all_data(AgentID):-
2     forall(actions:data(DataID,AgentID,Tick,observation_data,Location,
        ↪ EventID),
3         retract(actions:data(DataID,AgentID,Tick,
            ↪ observation_data, BaseType, Location,EventID)
            ↪ )).

```

Erhobene Daten können mittels `archive_data/1` archiviert werden. Diese bleiben dann erhalten, werden aber für kommende Analysen nicht verwendet. Jedoch kann der Agent diese Daten wieder reaktivieren oder die Forschungsarbeit damit nachweisen.

```

1 archive_data(Data):-
2     Data = actions:data(DataID,Agent1,Tick1,observation_data, BaseType
        ↪ , Location,EventIDs1),
3     retract(actions:data(DataID,Agent1,Tick1,observation_data,
        ↪ BaseType, Location,EventIDs1)),
4     assertz(actions:data(DataID,Agent1,Tick1,archive_data, BaseType,
        ↪ Location,EventIDs1)).

```

Durch `find_matching_types/2` werden Ereignistypen zu einer Liste von spezifischen Ereignissen mittels der IDs bestimmt.

```

1 find_matching_types(ListEventIDs,EventTypes):-
2     findall(EventType,find_matching_type(ListEventIDs,EventType),
3         ↪ EventTypes).
4 find_matching_type(ListEventIDs,EventType):-
5     member(EventID,ListEventIDs),
6     create_world_module:event(EventID,EventType,_,_).

```

Über `find_matching_typeslocations/3` werden die Typen und Position zu einer gegebenen Liste von Ereignissen zurückgegeben.

```

1 find_matching_typeslocations(ListEventIDs, Base, EventTypesLocations):-
2     findall(TypeLocation,find_matching_typerelocation(ListEventIDs, Base
3         ↪ ,TypeLocation),EventTypesLocations).
4 find_matching_typerelocation(ListEventIDs,Base, [EventType,RelativLocation
5     ↪ ]):-
6     member(EventID,ListEventIDs),
7     create_world_module:event(EventID,EventType,_,Location),
8     absolut_to_relativ_location(Location,Base, RelativLocation).
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %% Plan actions
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die folgenden Prädikate planen Aktionen und definieren dazu entsprechende Strategien. Es können Strategien zur Auswahl der Aktionen entsprechend der Agententypen oder zufällig gewählt werden. Über `plan_actions/0` wird der Planungsprozess für alle aktiven Agenten angestoßen.

```

1 plan_actions:-
2     list_of_active_agents(ListOfActiveAgentIDs),
3     forall( member(AgentID,ListOfActiveAgentIDs), plan_action(AgentID
4         ↪ )).

```

`plan_action/1` plant dann eine Aktion für einen gegebenen Agenten.

```

1 plan_action(AgentID):-
2     write_diary(AgentID, "plan_action"),
3     (action_planing(random) *-> random_action(AgentID);
4     action_planing(strategy) *-> strategic_action(AgentID)).
5
6 %plan_action(AgentID):-strategic_action(AgentID).

```

```

7
8 plan_random_action_for_agent(AgentID):-
9     find_valid_actions_for_agent(AgentID,List_Of_Actions),
10    random_member(Action,List_Of_Actions),
11    add_action_to_queue(AgentID, Action).
12
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 %%% Action strategy
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Über `strategic_action/1` werden den Agenten mögliche Aktionen entsprechend des Typs zurückgegeben.

```

1 strategic_action(AgentID):-
2     agent(Type,AgentID,_Dict),
3     strategic_action(AgentID,Type).

```

Mittels der folgenden Prädikate wird die Strategie für den Agententyp der Studierenden definiert. Diese können in der hier gezeigten Strategie lernen und zu einem Forschenden übergehen.

```

4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 %%% Strategy student
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 strategic_action(AgentID,student):-
8     graduation(AgentID)*->
9         change_type(AgentID, researcher);
10    (
11        publication(_A,_B,_C,_D,_E,_F)*->
12        plan(read_publication_rnd, AgentID);
13        msg_with_simple_name(AgentID,'nothing to learn')
14    ).
15 change_type(AgentID, NewType):-
16    msg_with_simple_name(AgentID,'changes Type'),
17    agents:agent(Type,AgentID,Dict),
18    retractall(agents:agent(Type,AgentID,Dict)),
19    asserta(agents:agent(NewType,AgentID,Dict)).
20
21 graduation(AgentID):-
22    agent_age(AgentID,Age),
23    random(4,7,Threshold),
24    Age > Threshold.

```

Analog wird die Strategie für die Forschenden definiert: In diesem Beispiel können diese publizieren, Daten analysieren, Publikationen lesen und Forschungsreihen durchführen.

```

25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26 %%% Strategy Researcher
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28 strategic_action(AgentID,researcher):-
29     plan(publish, AgentID),
30     plan(analyze_data, AgentID),
31     plan(read_publication_rnd,AgentID),
32     plan(experiment_series,AgentID).
33
34 sufficient_bel_for_publish(AgentID):-
35     findall(Theory,
36     ( actions:belief(AgentID, Tick, theory, Theory, Domain, Str), Str
37       ↪ >= 0.5 ),
38     List ),
39     length(List, Len), Len >1.

```

Die Forschungsförderer lesen in diesem Beispiel Publikationen und bilden so Überzeugungen, welche diese in Reviewprozessen benötigen.

```

39 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40 %%% Strategy funder
41 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
42 strategic_action(AgentID,funder):- plan(read_publication_rnd,AgentID).
43
44 enough_resources(AgentID, Threshold):-
45     agent(_Type,AgentID,Dict),
46     Res = Dict.get(resources),
47     Res > Threshold.
48
49 debug_actions(AgentID):-
50     list_of_active_agents(ListOfActiveAgentIDs),
51     sim_time(Tick),
52     %assertz(actions:debugcall(Tick, ListOfActiveAgentIDs, AgentID)),
53     add_action_to_queue(AgentID, debug).
54
55 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
56 %%% Prepare planned actions
57 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
58 % Plan and prepare actions and add them to the queue of the agents

```

Durch `plan/2` werden Aktionen vorbereitet. So können etwa Aktionen definiert werden, welche Vorbedingungen stellen oder auch Folgeaktionen verursachen. Ein Bei-

spiel ist etwa eine Serie von Experimenten oder die Analyse einer solchen Serie in einer Reihe von Daten.

```

1  plan(validation_type1, AgentID):- add_action_to_queue(AgentID,
    ↪ validation_type1).
2
3  plan(funding, AgentID):- add_action_to_queue(AgentID, funding).
4
5  plan(publish, AgentID):- add_action_to_queue(AgentID, publish_rnd).
6
7  plan(create_theory_type1, AgentID):- add_action_to_queue(AgentID,
    ↪ create_theory_type1).
8
9  plan(observation_types, AgentID):- add_action_to_queue(AgentID,
    ↪ observation_types).
10
11 plan(publish_types, AgentID):- add_action_to_queue(AgentID, publish_types
    ↪ ).
12
13 plan(select_new_interest, AgentID):- add_action_to_queue(AgentID,
    ↪ select_new_interest).
14
15 plan(analyze_data, AgentID):- add_action_to_queue(AgentID,
    ↪ create_theory_rnd).
16 /*    find_independent_data_series(AgentID, SeriesList) *->
17         plan(create_theory_type2, AgentID);
18         plan(create_theory_type1, AgentID). */
19
20 plan(funding, AgentID):- add_action_to_queue(AgentID, funding).
21
22 plan(read_related_publication, AgentID):- true.
23
24 plan(read_publication_rnd, AgentID):- add_action_to_queue(AgentID,
    ↪ read_publication_rnd).
25
26 plan(read_publication, AgentID):- true.
27         % add_action_to_queue(AgentID, read_publication).
28
29 plan(create_theory_type2, AgentID):- add_action_to_queue(AgentID,
    ↪ create_theory_type2).
30
31 plan(experiment_series, AgentID):-
32         msg_with_simple_name(AgentID, 'plan: experiment_series'),
33         random(2, 5, Random),
34         (    between(0, Random, C),

```



```

29 %%% Perform actions
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Über `perform_actions/0` werden alle aktiven Agenten aufgerufen die nächste geplante Aktion in dem laufendem Tick auszuführen.

```

1 perform_actions:-
2     list_of_active_agents(ListOfActiveAgents),
3     length(ListOfActiveAgents, Len), writeln(Len),
4     forall(member(Agent,ListOfActiveAgents), perform_action(
5         ↪ Agent)).
6
6 perform_action(AgentID):-
7     msg_with_simple_name(AgentID,'performs an action'),
8     agent(Type,AgentID, Attributes),
9     ActionQueue = Attributes.get(action_queue),
10    perform_action(AgentID,ActionQueue).

```

Mittels `perform_action/2` wird dann jeder Agent konkret aufgerufen und die nächste geplante Aktion ausgeführt. Sollte keine Aktion geplant werden, so wird an dieser Stelle eine neue Aktion entsprechend der gewählten Strategie geplant.

```

1 perform_action(AgentID,[]):-
2     msg_with_simple_name(AgentID,'empty queue'),
3     execute_action(plan_actions,AgentID,_Options).
4
5 perform_action(AgentID, Queue):-
6     Queue \= [],
7     Queue = [Action|Rest],
8     msg_with_simple_name(AgentID,'has something in queue'),
9     agent(Type,AgentID, Attributes),
10    Action = ..[_ ,Act,_,_],
11    price(Act, Price),
12    agent_resources(AgentID, Resources),
13    ( Price < Resources ->
14    perform_action(AgentID, Queue, Price, resources);
15    perform_action(AgentID, Queue, Price, no_resources) ).

```

Wenn die Aktion ausgeführt wird, wird diese aus der Liste der geplanten Aktionen entfernt und werden ggf. Ressourcen des Agenten verbraucht.

```

16 perform_action(AgentID, Queue, Price, resources):-
17     Queue = [Action|Rest],
18     update_attributes(AgentID,action_queue,Rest),
19     msg_with_simple_name(AgentID,Action),
20     Action = ..[_ ,Act,_,_],

```


21

Die hier beschriebenen Prädikate zeigen die zentrale Logik der Aktionen, welche die Agenten ausführen können. Zunächst werden alle Teilaktionen beschrieben, welche die Agenten anwenden, um Publikationen zu lesen.

```

22 execute_action(read_publication,AgentID,_Options):-
23     get_interest(AgentID,Interest),
24     find_related_publications(Interest,PublicationsAuthors),
25     PublicationsAuthors \= [] ->
26     ( select_publication(PublicationsAuthors, Publication),
27       update_belief_with_publication(publication, AgentID, Publication
28         ↪ ) );
29     no_publication(AgentID, Interest).
30 no_publication(AgentID, Interest):-
31     write_diary(AgentID, "no publication in this area - new research
32       ↪ topic!"),
33     update_belief_with_publication(research_topic, AgentID,Interest).
34 update_belief_with_publication(publication, AgentID, Content):-
35     sim_time(Tick),
36     asserta(actions:belief(AgentID, Tick, theory, Content, Domain,
37       ↪ 0.5)).
38 update_belief_with_publication(research_topic, AgentID, Content):-
39     sim_time(Tick),
40     asserta(actions:belief(AgentID, Tick, research_topic, Content,
41       ↪ Domain, 0.5)).
42 find_related_publications(Interest,PublicationsAuthors):-
43     findall([PublicationID, AuthorList],
44       (actions:publication(PublicationID,Journal,
45         ↪ AuthorList,Tick,theory,Content),
46         types_in_theory(Content,Interest)),
47       PublicationsAuthors).
48 types_in_theory(Theory,Type):-
49     unfold_theory(Theory, UnfoldedTheory),
50     UnfoldedTheory = [BaseType, Types, Relation],
51     Type = BaseType;
52     member(Type, Types).
53
54 select_publication(PublicationsAuthors, Publication):-
55     find_reputation(PublicationsAuthors, ReputationList),
56     select_best_publication(ReputationList, Publication).

```

```

57
58 find_reputation(PublicationsAuthors, SortReputation):-
59     findall( [PubID, AutID, Reputation],
60         ( member(PubAut, PublicationsAuthors),
61           find_reputation_from_list(PubAut, Reputation) ),
62         List),
63     sort(Reputation, SortReputation).
64
65 select_best_publication(ReputationList, Publication):-
66     sort(ReputationList, SortReputationList),
67     last(SortReputationList, Pub),
68     Pub = [Rep, AutID,Publication].
69
70 find_reputation_from_list(PubAut, Reputation):-
71     PubAut = [PubID, AutList],
72     findall([Rep, AutID, PubID],
73         ( member(AutID, AutList), agent_reputation(AutID, Rep) ),
74         Reputation).
75
76 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
77 %%% Action: Create Theory (Type 1)
78 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Darauf folgt die Definition der Aktionen für die Erzeugung von Theorien (Typ 1), wobei zunächst geprüft wird, ob die Grundvoraussetzungen erfüllt sind.

```

79 execute_action(create_theory_type1, AgentID, _Options):-
80     msg_with_simple_name(AgentID, 'action: create_theory_type1'),
81     agent(AgentType, AgentID, _Dict),
82     get_interest(AgentID, BaseType),
83
84     get_observed_events(AgentID, BaseType, DataObjects),
85     ( fitting_data(AgentID, BaseType, DataObjects)->
86         msg_with_simple_name(AgentID, 'create_theory_type1:
87             ↪ ... new Theory');
88         no_fitting_data(AgentID, BaseType, DataObjects) ).

```

Mittels `fitting_data/3` wird unterschieden, ob genug Daten vorhanden sind oder mittels `no_fitting_data/3` erst weitere Daten erhoben werden müssen.

```

1 no_fitting_data(AgentID, BaseType, DataObjects):-
2     msg_with_simple_name(AgentID, 'create_theory_type1: ... no fitting
3         ↪ data!'),
4     plan(select_new_interest, AgentID),
5     plan(observation_types, AgentID).
6 fitting_data(AgentID, BaseType, DataObjects):-

```

```

7      DataObjects \= [],
8      determine_types_from_eventids(DataObjects, Types),
9      multi_intersection(Types, CommonOccurence), % potential model
10     relative_neighborevents_type_location(DataObjects,
11         ↪ TypeLocationList),
12     multi_intersection(TypeLocationList, CommonTypes), % RelationModel
13     % Number of occurrences of unique Types in TypeLocList
14     distribution_of_elements(TypeLocationList, CommonRelations),
15     select_most_likely_candidate(CommonRelations, Relations),
16     create_theory(AgentID, t1, BaseType,Types, Relations, Theory),
17     sim_time(Tick),
18     assertz(actions:belief(AgentID, Tick, theory, Theory, Domain, 0.5)
19         ↪ ).

```

Die eigentliche Theorie wird dann mittels create_theory/5 erzeugt.

```

1  create_theory(AgentID, TType, BaseType,Types, Relations, TID):-
2      id('POTMOD',PotModID),
3      id('MOD',ModID),
4      id('THEORY',TID),
5      asserta(actions:model(ModID, PotModID, Relations)),
6      asserta(actions:pot_model(PotModID,BaseType, Types)),
7      asserta(actions:theory(AgentID, TType, TID, Link, PotModID, ModID)
8          ↪ ).
9
10 get_observed_events(AgentID, BaseType, DataObjects):-
11     findall(
12         [BaseType, BaseLocation, DataContent],
13         actions:data(DataID, AgentID, _Tick, _DataType, BaseType,
14             ↪ BaseLocation, DataContent),
15         DataObjects).
16
17 determine_types_from_eventids(DataObjects, Types):-
18     findall(TypesList,
19         ( member([_BaseType, _BaseLocation, DataContent],
20             ↪ DataObjects),
21             find_types(DataContent,TypesList)),
22         Types).
23
24 relative_neighborevents_type_location(DataObjects, TypeLocationList):-
25     findall([Type,RelLocation],
26         ( member([_BaseType, BaseLocation, EventList],
27             ↪ DataObjects),
28             member(Event, EventList),
29             create_world_module:event(Event, Type, _, Location),

```

```

26         absolut_to_relativ_location(Location, BaseLocation,
27             ↪ RelLocation)),
28         TypeLocationList).
29 select_most_likely_candidate(CommonRelations, HighestValue):-
30     sort(CommonRelations, CommonRelationsSort),
31     last(CommonRelationsSort, HighestValue).
32
33 unfold_theory(TheoryID, UnfoldedTheory):-
34     actions:theory(_AID, _Type, TheoryID, _Link, PotModID, ModID),
35     actions:model(ModID, PotModID, Relation),
36     actions:pot_model(PotModID, BaseType, Types),
37     UnfoldedTheory = [BaseType, Types, Relation].
38
39 find_types(EventIDs, EventTypes):-
40     findall(Type, (member(Event, EventIDs), create_world_module:event(
41         ↪ Event, Type, _, _)), EventTypes).
42
43 %%% Action: Theory Type 2
44 %%%

```

Theorien vom Typ 2 werden auf andere Weise erzeugt: Es wird nach Mustern in den Daten gesucht. Dies geschieht mittels der Suche nach Gemeinsamkeiten bzw. Schnittmengen über `find_commons_in_t1/2`.

```

1 execute_action(create_theory_type2, AgentID, _Options):-
2     msg_with_simple_name(AgentID, 'action: create_theory_type2'),
3     (find_independent_data_series(AgentID, PotentialSeriesList),
4     select_base_type(PotentialSeriesList, BaseType),
5     filter_series(PotentialSeriesList, BaseType, SeriesList),
6     find_commons_in_t1(SeriesList, ComCommonsT1),
7     find_commons_in_t2(SeriesList, ComCommonsT2),
8     write_theory(AgentID, ComCommonsT1, ComCommonsT2));
9     msg_with_simple_name(AgentID, 'action: no theory t2 found').
10
11 write_theory(AgentID, ComCommonsT1, ComCommonsT2):-
12     findall(TypeT1, member([TypeT1, _], ComCommonsT1), TypesT1),
13     findall(TypeT2, member([TypeT2, _], ComCommonsT2), TypesT2),
14     append(Types1, Types2, Types),
15     create_theory(AgentID, type2, BaseType, Types, [ComCommonsT1,
16         ↪ ComCommonsT2], TID).
17
18 find_commons_in_t1(SeriesList, CommonsT1):-
19     filter_t1(SeriesList, T1List),
20     extract_type_location(T1List, TypeLocationList),

```

```

20     multi_intersection(TypeLocationList,CommonsT1).
21
22 find_commons_in_t2(SeriesList, CommonsT2):-
23     filter_t2(SeriesList,T2List),
24     extract_type_location(T2List, TypeLocationList),
25     multi_intersection(TypeLocationList,CommonsT2).
26
27 filter_t1(SeriesList,T1List):-
28     findall(T1, member([_P,[T1,_T2]],SeriesList), T1List).
29
30 filter_t2(SeriesList,T2List):-
31     findall(T2, member([_P,[_T1,T2]],SeriesList), T2List).
32
33 extract_type_location(TList, TypeLocationList):-
34     findall(TypeLocation,
35             (
36                 member(DID,TList),
37                 data(DID,_AID,_Tick,observation_data,_BaseType,
38                     ↪ Location,Observation),
39                 actions:find_matching_typeslocations(Observation,
40                     ↪ Location,TypeLocation)),TypeLocationList).
41
42 select_base_type(SeriesList,SelectedBaseType):-
43     find_independent_data_series(AgentID,SeriesList),
44     findall(BaseType,member([BaseType,_],SeriesList),Types),
45     distribution_of_elements(Types,Dist),
46     sort(Dist,Sort),
47     last(Sort,[_,_,SelectedBaseType]).
48
49 filter_series(PotentialSeriesList,BaseType,SeriesList):-
50     findall([BaseType,Content],member([BaseType,Content],
51     ↪ PotentialSeriesList),SeriesList).
52
53 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
54 %%% Action: Publish
55 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die Publikationen der Agenten werden über eine Reihe von Aktionen erzeugt und schließlich über publish/3 publiziert. In diesem Prozess sind auch Review-Verfahren und Co-Autorschaften abgebildet.

```

1 execute_action(publish, AgentID, _Options):-
2     msg_with_simple_name(AgentID,'action: publish'),
3     select_bel_to_publish(AgentID, Theory)*->
4     (
5         review_draft_pub(AgentID, Theory, Result),
6         get_observed_events(AgentID, BaseType, DataObjects),
7         exec_publish([AgentID], theory, Theory, Result)

```



```

7         );
8         msg_with_simple_name(AgentID,'action: no fitting content to
          ↪ publish').
9
10    exec_publish([AgentID], theory, Theory, accepted):-
11        publish([AgentID], theory, Theory),
12        publish([AgentID], data, DataObjects).
13
14    exec_publish([AgentID], theory, Theory, rejected):-
15        msg_with_simple_name(AgentID,'action: rejected').
16
17    select_bel_to_publish(AgentID, Theory):-
18        findall(
19            actions:belief(AgentID, Tick, theory, Theory, Domain, Str),
20            (actions:belief(AgentID, Tick, theory, Theory, Domain, Str),Str>0.4)
          ↪ ,
21        BellList),
22        random_member(actions:belief(_, _, theory, Theory, _, _), BellList).
23
24    review_draft_pub(AgentID, Theory, Result):-
25        agents:agent(T,AgentID, Dict),
26        AuthorDomain = Dict.get(research_domain),
27        select_pub_reviewer(AgentID, AuthorDomain, Reviewer)*->
28        conduct_pub_review(non-blind, Reviewer, AgentID, Theory, Result);
29        (random_reviewer(Reviewer),conduct_pub_review(non-blind, Reviewer,
          ↪ AgentID, Theory, Result)).
30
31    conduct_pub_review(non-blind, ReviewerID, AgentID, Theory, Result):-
32        agents:agent(_,AgentID, Dict),
33        agents:agent(_,ReviewerID, RevDict),
34        AuthorReputation = Dict.get(reputation),
35        AuthorAbility = Dict.get(ability),
36        Threshold = 1.5,
37        check_draft_soundness(Theory, ReviewerID, Soundness),
38        Review is (AuthorReputation + AuthorAbility) + Soundness,
39        pub_review_result(AgentID, Threshold, Review, Result).
40
41    pub_review_result(AgentID, Threshold, Review, accepted):-
42        Review > Threshold,
43        write_diary(AgentID, "Publication accepted"),
44        msg_with_simple_name(AgentID,'action: publication accepted').
45
46    pub_review_result(AgentID, Threshold, Review, rejected):-
47        Review =< Threshold,
48        write_diary(AgentID, "Publication rejected"),

```


Über `read_publication_rnd` können Agenten eine zufällig ausgewählte Publikation lesen und bewerten.

```

1 execute_action(read_publication_rnd,AgentID,_Options):-
2     agent(_,AgentID,_),
3     findall(ID, actions:publication(ID,Journal,Authors,Tick,Type,
4         ↪ Content),Pubs),
5     length(Pubs, NP),
6     select_publication_rnd(AgentID,NP).
7
8 select_publication_rnd(AgentID,NP):-
9     NP<2, writeln('no publications to read').
10
11 select_publication_rnd(AgentID,NP):-
12     NP>=2,
13     agent(_,AgentID, Dict),
14     LOI = Dict.get(list_of_interests),
15     %get_interest(AgentID,Interest),
16     %select_topic(Interest, TID),
17     select_topic_list(LOI, TID),
18     find_publication(TID,Publication),
19     check_for_plagiatism(Publication);
20     read_publication_rnd(AgentID, Publication).
21
22 check_for_plagiatism(Publication).

```

Publikationen mit mehreren Autoren werden über `double_publication/3` entsprechend zugeordnet.

```

1 double_publication(Publication, Author1, Author2):-
2     actions:publication(Publication, Journal, Authors1,_, Tick1,TID),
3     actions:publication(Publication, Journal, Authors2,_, Tick2,TID),
4     intersection(Authors1,Authors2, Int),
5     Tick1 \= Tick2,
6     Int = [].
7
8 read_publication_rnd(AgentID, []).
9
10 read_publication_rnd(AgentID, Publication):-
11     Publication \=[],
12     update_belief_with_publication(publication, AgentID, Publication),
13     actions:publication(Publication, Journal, Authors,_, theory_rnd,
14         ↪ TID),
15     actions:theory(AuthorID,rnd,TID,_,PotM,M),
16     agents:agent(_,AgentID, Dict),
17     agents:agent(_,AuthorID, AuthorDict),

```

```

17     Interests = Dict.get(list_of_interests),
18     Domain = Dict.get(research_domain),
19     AutInterests = AuthorDict.get(list_of_interests),
20     AuthorDomain = AuthorDict.get(research_domain),
21     intersection(Domain,AuthorDomain, Dom),
22     intersection(Interests,AutInterests, Int),
23     append(Dom,Int,Match),
24     exec_citation(AgentID,AuthorID,Publication,Match).
25
26 exec_citation(AgentID,AuthorID,Publication, []).
27
28 exec_citation(AgentID,AuthorID,Publication,Match):-
29     Match \=[],
30     actions:plan(publish, AgentID),
31     cite_rnd(AgentID,AuthorID,Publication).
32
33 cite_rnd(AgentID,CitedAuthor,Publication):-
34     asserta(actions:citation(AgentID, CitedAuthor,Publication)).
35
36 select_topic(Interest, Theory):-
37     ( findall(PotID, actions:pot_model(PotID,Interest,_),PotMods),
38       random_member(PotMod,PotMods),
39       findall(TID, actions:theory(CreatorID, rnd, TID,_,PotMod, Mod),
40         ↪ TIDs ),
41       random_member(Theory, TIDs) ); Theory = [].
42
43 select_topic_list(InterestList, Theory):-
44     ( findall(PotID, (member(Interst, InterestList), actions:
45       ↪ pot_model(PotID,Interest,_),PotMods),
46       random_member(PotMod,PotMods),
47       findall(TID, actions:theory(CreatorID, rnd, TID,_,PotMod,
48         ↪ Mod), TIDs ),
49       random_member(Theory, TIDs) ); Theory = [].
50
51 find_publication(TID,Publication):-
52     findall(PubID, actions:publication(PubID, high_impact, Authors,_,
53       ↪ theory_rnd,TID),HPubs),
54     findall(PubID, actions:publication(PubID, mid_impact, Authors,_,
55       ↪ theory_rnd,TID),MPubs),
56     findall(PubID, actions:publication(PubID, low_impact, Authors,_,
57       ↪ theory_rnd,TID),LPubs),
58     (random_member(Publication,HPubs);
59      random_member(Publication,MPubs);
60      random_member(Publication,LPubs);

```

```

55     ( findall(PubID, actions:publication(PubID, _, Authors,_,
56         ↪ theory_rnd,TID),Other), random_member(Publication,Other));
57     Publication=[],!).
58     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59     %% Action: Publish (random content)
60     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die Simulation bietet auch die Möglichkeit Publikationen zufallsgesteuert durchzuführen. Dabei werden zufällig gewählte Daten oder Theorien eines Agenten publiziert.

```

61 execute_action(publish_rnd, AgentID, _Options):-
62     msg_with_simple_name(AgentID,'action: publish'),
63     findall(TID, theory(AgentID, rnd, TID, Link, _, _),PotT),
64     plagiarize(AgentID);
65     publish_rnd_no_j(AgentID,PotT).
66
67 publish_rnd_no_j(AgentID,[]):- writeln('nothing to publish').
68
69 publish_rnd_no_j(AgentID,PotT):-
70     PotT \= [],
71     random_member(SelectedTheoryID,PotT),
72     publish_rnd(AgentID, oa, SelT, 1.01, 1 , 0, SelectedTheoryID).
73
74 publish_rnd(AgentID, Journal, SelT, RepFactor,Result,Threshold,
75     ↪ SelectedTheoryID):-
76     Result >= Threshold,
77     exec_publish_rnd([AgentID], Journal, theory_rnd,SelectedTheoryID,
78     ↪ accepted),
79     gain_reputation(AgentID, RepFactor).
80
81 publish_rnd(AgentID, Journal, SelT, RepFactor,Result,Threshold,
82     ↪ SelectedTheoryID):-
83     Result < Threshold,
84     rejected_rnd(AgentID,Result, SelectedTheoryID, Journal),
85     reduce_reputation(AgentID).

```

Dabei besteht auch die Möglichkeit eine Publikation von anderen Agenten zu übernehmen. Mittels `plagiarize/1` kann ein Plagiat erzeugt werden. Der entsprechende Agent wählt dafür eine zufällige Publikation aus und veröffentlicht diese unter eigenem Namen.

```

1 plagiarize(AgentID):-
2     sim_time(Tick),
3     Tick > 110,

```

```

4      actions:traitor(Traitor,TTick),!,
5      Tick > TTick + 10,
6      asserta(actions:traitor(AgentID, Tick)),
7      write_diary(AgentID, "plagiarize"),
8      findall(Publications, actions:publication(PublicationID,_Journal,
9          ↪ Authors,_, Type,_Content), ListOfPubs),
10     random_member(PublicationID, ListOfPubs),
11     asserta(actions:publication(PublicationID,_Journal,[AgentID],Tick,
12     ↪ Type,Content)).
13
14 quality_publication_rnd(AgentID,SelectedTheoryID,QualityPublication):-
15     agents:agent(_,AgentID, Dict),
16     Ability = Dict.get(ability),
17     actions:theory(AgentID, rnd, SelectedTheoryID,_,_,ModID),
18     actions:model(ModID, PotModID, Quality),
19     QualityPublication = Quality * Ability + 1.
20
21 rejected_rnd(AgentID,Result,SelectedTheoryID, Journal):-
22     sim_time(Tick),
23     asserta(actions:publication_rejected(AgentID, Tick, Result,
24     ↪ SelectedTheoryID, Journal)).
25
26 select_journal(Journal, Threshold,RepFactor):-
27     random_member([Journal,Threshold,RepFactor,Type],
28     [ [low_impact,4,1.01,closed], [mid_impact, 10,1.05,closed],
29     ↪ [high_impact,25,1.2,closed] ] ).
30
31 rnd_rew(QualityPublication,AgentID,Result):-
32     select_reviewer_rnd(AgentID, ReviewerList),
33     findall(Score, (member(Rev, ReviewerList), review_rnd(AgentID, Rev
34     ↪ , QualityPublication, Score)), Scores),
35     sum_list(Scores,Result).

```

Die Begutachtung ist in diesem Fall ebenso mittels review_rnd/4 zufallsgesteuert.

```

1 select_reviewer_rnd(AgentID, ReviewerList):-
2     agents:agent(_,AgentID, Dict),
3     Domain = Dict.get(research_domain),
4     ( findall(RevID, ( agents:agent(_,RevID, RevDict),
5         RevDomain = RevDict.get(research_domain),
6         intersection(Domain, RevDomain, IntDom),
7         IntDom \=[] ),
8         PotReviewerList),
9         random_sublist(3,PotReviewerList,ReviewerList)) ;
10    (list_of_active_agents(Agents),
11    random_sublist(3, Agents, ReviewerList )).

```

```

12
13 review_rnd(AgentID, Rev, QualityPublication, Score):-
14     agents:agent(_,AgentID, Dict),
15     agents:agent(_,Rev, RevDict),
16     InterstsAgent = Dict.get(list_of_interests),
17     Reputation = Dict.get(reputation),
18     InterstsRev = RevDict.get(list_of_interests),
19     append(InterstsAgent,InterstsAgent,AllInt),
20     sort(AllInt,SortAllInt),
21     length(SortAllInt,LenSortAllInt),
22     intersection(InterstsAgent,InterstsRev,IntersectionInt),
23     length(IntersectionInt,LenInt),
24     Match is LenInt/LenSortAllInt*100,
25     Score is (QualityPublication)+(Match/10)*Reputation.
26
27 conduct_pub_review_rnd(non-blind, ReviewerID, AgentID, Theory, Result):-
28     agents:agent(_,AgentID, Dict),
29     agents:agent(_,ReviewerID, RevDict),
30     AuthorReputation = Dict.get(reputation),
31     AuthorAbility = Dict.get(ability),
32     Threshold = 1.5,
33     check_draft_soundness(Theory, ReviewerID, Soundness),
34     Review is (AuthorReputation + AuthorAbility) + Soundness,
35     pub_review_result(AgentID, Threshold, Review, Result).
36
37 exec_publish_rnd([AgentID], Journal, theory_rnd, Theory, accepted):-
38     publish([AgentID], theory_rnd, Journal, Theory).
39
40 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
41 %%% Action: Create Theory (random content)
42 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Ebenso bietet die Simulation die Möglichkeit, Theorien nicht nur auf Grundlage des vorgestellten Modells zu Theoriefindung zu erzeugen, sondern diese aus zufällig zusammengestellten Theorieelementen aufzubauen. Genaugenommen erfüllen solche Konstrukte nicht die Anforderungen an eine Theorie, da diese keine Daten zur Erfüllung aufweisen können. Dennoch bietet dieses System die Möglichkeit, solche Konstrukte zum Vergleich mit anderen Mechanismen heranzuziehen.

```

43 execute_action(create_theory_rnd, AgentID, _Options):-
44     msg_with_simple_name(AgentID, 'action: create_theory_random'),
45     agent(AgentType, AgentID, Dict),
46     get_interest(AgentID, BaseType),
47     Ability = Dict.get(ability),

```


Diese Aktion erlaubt es Agenten, Publikationen über das einfache Vorkommen von Ereignistypen zu publizieren.

```

84 execute_action(publish_types, AgentID, _Options):-
85     msg_with_simple_name(AgentID, 'action: publish observation of event
      ↪ types'),
86     findall(Observation, actions:data(DataID,AgentID,Tick,ContentType,
      ↪ _, _, Observation), ObservedTypes),
87     publish([AgentID], observed_types, ObservedTypes).
88
89 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
90 %%% Action: Funding (randomized, without strategy)
91 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Für die Antragsstellung auf Forschungsförderung stehen der Simulation ebenfalls eine Zufallsvariante und ein Simulationsmodell zur Verfügung, in welcher die Entscheidung über einen Forschungsantrag dann entsprechend zufallsbasiert abläuft.

```

92 execute_action(simple_funding, AgentID, _Options):-
93     msg_with_simple_name(AgentID, 'action: funding'),
94     agent(Type,AgentID, Attributes),
95     Resources = Attributes.get(resources),
96     Reputation = Attributes.get(reputation),
97     maxFunding(MaxFunding),
98     maxInvestment(MaxInvestment),
99     random(0,MaxFunding,ResourcesRandom),
100    random(0,MaxInvestment,Investment),
101    NewResources is (round(ResourcesRandom * Reputation) + Resources)
      ↪ - (round(Investment-Reputation)),
102    %addlog(AgentID, action, generating_resources,NewResources),
103    update_attributes(AgentID, resources, NewResources).
104
105 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
106 %%% Action: Funding (strategy: review process)
107 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Im komplexeren Fall wird ein Antrag gestellt, über welchen dann mittels eines Gutachterverfahrens entschieden wird. Die Begutachtung wird durch Agenten durchgeführt und basiert auf der Vorhabensbeschreibung und der Einschätzung zu den Antragstellern.

```

108 execute_action(funding, AgentID, _Options):-
109     msg_with_simple_name(AgentID, 'action: request funding'),

```

Zunächst wird `write_proposal/2` aufgerufen, um den Antrag zusammenzustellen.

```

1 write_proposal(AgentID, Proposal),

```

Der Antrag wird dann in das Gutachterverfahren gegeben.

```

2      review(AgentID, Proposal, Result),
3      format("Result: ~w\n", [Result]),

```

Das Ergebnis des Gutachtens entscheidet dann über eine Finanzierung oder Ablehnung.

```

4      Result = "funding_recommended" ->
5      execute_funding(AgentID);
6      reject_funding(AgentID).

```

Der Antrag wird über `write_proposal/2` erzeugt. Neben den inhaltlichen Aspekten, des Fachbereiches und der Fragestellung fließen hier ebenso die Reputation und Eignung des Antragsstellers ein.

```

1 write_proposal(AgentID, Proposal):-
2     id(prop, PropID),
3     agent(Type, AgentID, Dict),
4     Subject = Dict.get(list_of_interests),
5     Domain = Dict.get(research_domain),
6     Ability = Dict.get(ability),
7     Reputation = Dict.get(reputation),
8     Quality is (Ability + Reputation)/2,
9     Proposal = [Subject, Domain, Quality].

```

Das Gutachterverfahren wird dann mittels `review/3` angestoßen.

```

1 review(AgentID, Proposal, Result):-
2     Proposal = [Subject, Domain, Quality],
3     select_reviewer(Domain, ListOfReviewerIDs)->
4         conduct_nonblind_review( AgentID, Proposal,
5             ↪ ListOfReviewerIDs, Result);
6     new_subject(AgentID, Proposal, ListOfReviewerIDs, Result).
7 conduct_nonblind_review(AgentID, Proposal, ListOfReviewerIDs, Result):-
8     writeln(ListOfReviewerIDs),
9     Proposal = [Subject, Domain, Quality],
10    Threshold = 0.7,
11    findall(Fitting,
12            ( member(Rev, ListOfReviewerIDs),
13              reviewer_bias(Rev, Subject, Fitting) ),
14            Results),
15    length(Results, LenResults),
16    sum_list(Results, SumResults),
17    originality(AgentID, Subject, OriginalityScore),
18    familiarity(AgentID, Subject, FamiliarityScore),

```

```

19     Result is ((SumResults/LenResults*Quality)+OriginalityScore)+
        ↳ FamiliarityScore,
20     Result>Threshold *->
21         Result = "funding_recommended" ;
22         Result = "funding_rejected".

```

Das Gutachterverfahren erzeugt aus dem Antragssteller, dem Antrag und den Gutachtenden eine Entscheidung.

```

23 conduct_blind_review(non-blind, AgentID, ProposalID, ListOfReviewerIDs,
        ↳ Result):-
24     Proposal = [Subject, Domain, Quality],
25     Threshold = 0.5,
26     findall(Fitting,
27             ( member(Rev, ListOfReviewerIDs),
28               reviewer_bias(Rev, Subject, Fitting) ),
29             Results),
30     Result is (SumResults/LenResults*Quality),
31     Result>Threshold *->
32         Result = "funding_recommended" ;
33         Result = "funding_rejected".

```

Die Gutachtenden können dem Antrag positiv oder ablehnend gegenüberstehen. Diese Bewertung wird durch die persönlichen Interessen der Gutachtenden bestimmt.

```

34 reviewer_bias(ReviwerID, PropSubject, Fitting):-
35     agent(Type, ReviwerID, Dict),
36     Subject = Dict.get(list_of_interests),
37     subset(PropSubject, Subject)->Fitting =1; Fitting =0.
38
39 new_subject(AgentID, Proposal, ListOfReviewerIDs, Result):-
40     Proposal = [Subject, Domain, Quality],
41     originality(AgentID, Subject, OriginalityScore),
42     familiarity(AgentID, Subject, FamiliarityScore),
43     Evaluation is (Quality + OriginalityScore) + FamiliarityScore,
44     Threshold = 0.5,
45     Evaluation>Threshold ->
46         Result = "funding_recommended" ;
47         Result = "funding_rejected".
48
49 originality(AgentID, Subject, Originality):-
50     number_of_related_publications(AgentID, Subject,
        ↳ NumberRelatedPublications),
51     number_of_all_publications(AgentID, NumberOfAllPublications),
52
53     (      NumberRelatedPublications >0,

```

```

54         ShareRelatedPubs is (NumberRelatedPublications /
55             ↪ NumberOfAllPublications) * 100,
56         Originality is 100 - ShareRelatedPubs );
57     Originality = 0.
58 number_of_related_publications(AgentID, Subject,
59     ↪ NumberRelatedPublications):-
60     findall(RelatedPublicationID,
61         ( actions:publication(RelatedPublicationID,_Journal,Authors,
62             ↪ _Tick,_Type,_Content),
63             not(member(AgentID, Authors)),
64             unfold_theory(Theory, Content), Content=[Subject|_Rest] ),
65             RelatedPublications),
66     length(RelatedPublications,NumberRelatedPublications).
67 number_of_all_publications(AgentID, NumberOfAllPublications):-
68     findall(RelatedPublicationID,
69         (actions:publication(RelatedPublicationID,_Journal, Author,
70             ↪ _Tick,_Type,_Content),
71             not(member(AgentID, Authors))),
72         AllPublications),
73     length(AllPublications,NumberOfAllPublications).
74 familiarity(AgentID,Subject,FamiliarityScore):-
75     findall(RelatedPublicationID,
76         (actions:publication(RelatedPublicationID,_Journal, Authors,
77             ↪ _Tick,_Type,_Content),
78             member(AgentID, Authors),
79             unfold_theory(Theory, Content), Content=[Subject|_Rest] ),
80             RelatedPublications),
81     length(RelatedPublications,NumberRelatedPublications),
82     NumberRelatedPublications >0 ->
83     FamiliarityScore = NumberRelatedPublications / 10;
84     FamiliarityScore = 0.
85 reject_funding(AgentID):-
86     msg_with_simple_name(AgentID,'funding rejected'),
87     write_diary(AgentID, "Funding rejected"),
88     reduce_reputation(AgentID).
89 reduce_reputation(AgentID):-
90     agent(Type,AgentID, Attributes),
91     Reputation = Attributes.get(reputation),
92     NewReputation is Reputation * 0.7,
93     update_attributes(AgentID, reputation, NewReputation).

```

```

94
95 select_reviewer(Domain,Reviewer):-
96     findall(FID, ( agent(funder, FID, FDict), Domain = FDict.get(
97         ↪ research_domain ) ), PossibleReviewer),
98     length(PossibleReviewer,Len),
99     Len > 1 ->
100     random(1,Len,NumberReviewer),
101     random_sublist(NumberReviewer, PossibleReviewer, Reviewer),
102     length(Reviewer,NBRev),
103     NBRev > 1.
104
105 execute_funding(AgentID):-
106     msg_with_simple_name(AgentID,'execute approved'),
107     agent(Type,AgentID, Attributes),
108     Resources = Attributes.get(resources),
109     maxFounding(MaxFounding),
110     random(0,MaxFounding,ResourcesRandom),
111     NewResources is ResourcesRandom +Resources,
112     gain_reputation(AgentID),
113     write_diary(AgentID, "Funding approved"),
114     %addlog(AgentID, action, generating_resources,NewResources),
115     update_attributes(AgentID, resources, NewResources).
116
117 gain_reputation(AgentID):-
118     agent(Type,AgentID, Attributes),
119     Reputation = Attributes.get(reputation),
120     NewReputation is Reputation * 1.2,
121     update_attributes(AgentID, reputation, NewReputation).
122
123 gain_reputation(AgentID, Factor):-
124     agent(Type,AgentID, Attributes),
125     Reputation = Attributes.get(reputation),
126     NewReputation is Reputation * Factor,
127     update_attributes(AgentID, reputation, NewReputation).
128
129 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
130 %%% Action: Observe event types
131 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die Beobachtung von Ereignistypen erfolgt über die folgenden Prädikate. Bei einschlägigen Beobachtungen werden diese dann ebenso analysiert und publiziert. Die Entscheidung darüber, ob die Beobachtung dafür hinreichend ist, findet in den entsprechenden Aktionen selbst statt.

```

131
132 execute_action(observation_types, AgentID, _Options):-

```

```

133     msg_with_simple_name(AgentID,'action: observation (type)'),
134     agent(Type,AgentID, Dict),
135     sim_time(Tick),
136     events_in_tick(Tick) *->
137     (
138         observe_location(Tick, PossibleObservation),
139         extract_types(PossibleObservation,Types),
140         analyse_observation(Dict,Types, Observation),
141         publish_observation_types(AgentID,Tick,Observation));
142     writeln('... no events to observe.').
143
144 events_in_tick(Tick):-
145     create_world_module:event(_,_,Tick,_),!.
146
147 publish_observation_types(AgentID,Tick, []).
148
149 publish_observation_types(AgentID,Tick,Observation):-
150     Observation \= [],
151     ContentType = observed_types,
152     uuid(DataID),
153     assertz(actions:data(DataID,AgentID, Tick, ContentType, _, _,
154         ↪ Observation)),
155     assertz(actions:belief(AgentID, Tick, observed_types, [Observation
156         ↪ ], Domain, 0.7)),
157     plan(publish_types, AgentID).
158
159 analyse_observation(Dict, [], _):-
160     writeln('... not types.').
161
162 analyse_observation(Dict, Types, Observation):-
163     Types \= [],
164     Ability = Dict.get(ability),
165     length(Types,NumberOfPossibleObservedTypes),
166
167     NumberOfObservedTypes is (round(Ability *
168         ↪ NumberOfPossibleObservedTypes)),
169     random_sublist(NumberOfObservedTypes,Types,Observation).
170
171 observe_location(Tick,PossibleObservation):-
172     findall(EID, create_world_module:event(EID, _,Tick,_), EIDs)*->
173     (
174         random_member(EventID,EIDs),
175         neighborevents(EventID,PossibleObservation) );
176     writeln('... no events to observe.').
177
178 extract_types([],_).
179
180

```


Die Validierung von Theorien ist abhängig von der Art ebendieser. Die folgenden Prädikate beschreiben den Prozess für die Validierung von Theorien des Typs 1.

```

8 execute_action(validation_type1, AgentID, _Options):-
9     msg_with_simple_name(AgentID,'action: Theory Type 1 validation'),
10    select_t1_theory(AgentID, TheoryID),
11    TheoryID \= []*->
12    theory_t1_validation(AgentID, Theories);
13    msg_with_simple_name(AgentID,'action: no theory for validation').
14
15 select_t1_theory(AgentID, TheoryID):-
16    findall(TheoryID,
17    (
18        actions:belief(AgentID, Tick, theory, TheoryID, Domain, Str
19        ↪ ),Str>0.4)*->
20        actions:theory(_,type1,TheoryID,_,_,_), Type1TheoryList),
21    Type1TheoryList \= [] *->
22    random_member(TheoryID,Type1TheoryList);
23    (
24        TheoryID = [],
25        msg_with_simple_name(AgentID,'action: no fitting
26        ↪ belief')).

```

Über `theory_t1_validation/2` wird eine Theorie anhand eigens beobachteter Daten validiert.

```

1 theory_t1_validation(AgentID, TheoryID):-
2     actions:theory(CreatorID,TheoryType,TheoryID,_Link,_PotMod,_Mod)
3     ↪ *->
4     (
5         unfold_theory(TheoryID,[Type,T1,T2]),
6         actions:data(DataID_T1, AgentID, PTick_T1,ContentType_T1,
7         ↪ Type, Location_T1, Observation_T1)*->
8         (
9             T2 = [Count,Dist,[Elety,ElePos]],
10            (
11                member(Elety,Observation_T1)*->
12                theory_t1_validation(AgentID, TheoryID,
13                ↪ validated);
14                theory_t1_validation(AgentID, TheoryID,
15                ↪ not_validated)));
16            msg_with_simple_name(AgentID,'action: no data');
17            msg_with_simple_name(AgentID,'action: no fitting
18            ↪ theory').
19
20 theory_t1_validation(AgentID, TheoryID, validated):-
21    msg_with_simple_name(AgentID,'action: Theory Type 1 validated'),
22    retractall(actions:belief(AgentID, Tick, theory, TheoryID, Domain,
23    ↪ Str)),
24    asserta(actions:belief(AgentID, Tick, theory, TheoryID, Domain,
25    ↪ 1.0)),

```



```

16     actions:theory(CreatorID,_TheoryType,TheoryID,_Link,_PotMod,_Mod),
17     gain_reputation(CreatorID).
18
19 theory_t1_validation(AgentID, TheoryID, validated):-
20     msg_with_simple_name(AgentID,'action: Theory Type 1 validated'),
21     retractall(actions:belief(AgentID, Tick, theory, TheoryID, Domain,
22         ↪ Str)),
23     asserta(actions:belief(AgentID, Tick, theory, TheoryID, Domain,
24         ↪ 0.1)),
25     actions:theory(CreatorID,_TheoryType,TheoryID,_Link,_PotMod,_Mod),
26     reduce_reputation(CreatorID).
27
28 %%% Action: Validation Experiment

```

Alternativ zur Validierung einer Theorie mittels bereits beobachteter Daten, kann auch ein Experiment dafür geplant und durchgeführt werden.

```

29 execute_action(validation_experiment, AgentID, _Options):-
30     msg_with_simple_name(AgentID,'action: conduct validation
31         ↪ experiment'),
32     agent(Type,AgentID, Attributes),
33     sim_time(Tick),
34     select_theory_from_bel(AgentID, Theory),
35     set_up_experiment(Theory, Tick, Location),
36     set_up_observation(Theory, Tick, Location, Observation),
37     analyze_experiment(Observation),
38     publish_experiment_results(AgentID, Tick, Results).
39
40 select_theory_from_bel(AgentID, Theory):-
41     findall(actions:belief(AgentID, Tick, theory, T, Domain, Str),
42         actions:belief(AgentID, Tick, theory, T, Domain, Str
43         ↪ ),
44         ListOfBel),
45     random_member(actions:belief(AgentID, _, theory, Theory, _, _),
46         ↪ ListOfBel).
47
48 set_up_experiment(TheoryID, Tick, Location):-
49     actions:theory(_AID,_TP, TheoryID,_Link,PotModID, ModID),
50     actions:model(ModID, PotModID, Relations),
51     writeln(Relations).
52
53 set_up_observation(Theory, Tick, Location, Results):-
54     %observe location in tick +1
55     true.

```

```

53
54 analyze_experiment(Observation):-true.
55
56 publish_experiment_results(AgentID, Tick, Results):- true.
57
58 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59 %% Action: Observation of events
60 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Die Beobachtung von Ereignissen wird über mehrere Zeitschritte durchgeführt. Entsprechend teilt sich die folgende Aktion in zwei Phasen: Es wird eine Beobachtung geplant und dann über die folgenden Zeitschritte fortgesetzt.

```

61 execute_action(observation_of_events, AgentID, _Options):-
62     msg_with_simple_name(AgentID,'action: observation (events)'),
63     agent(Type,AgentID, Attributes),
64     sim_time(Tick),
65     % Check if there is an ongoing observation
66     (ongoing_experiment(AgentID, Tick)*->
67         % If there is an ongoing observation -> continue with same
68         ↪ type
69         continue_observation(AgentID, ListOfBaseEvents);
70     % If this is a new obs. experiment, get interests
71     make_new_observation(AgentID, ListOfBaseEvents)),
72     % conduct observation
73     observation(AgentID, Tick, ListOfBaseEvents).

```

Durch `ongoing_experiment/2` wird geprüft, ob ein laufendes Experiment bzw. eine Beobachtungsreihe fortgesetzt wird. Dazu wird überprüft, ob in dem Zeitschritt $t_i = t - 1$ bereits Daten durch den Agenten erhoben wurden. Wenn dies der Fall ist, wird der Prozess weitergeführt.

```

1 ongoing_experiment(AgentID, Tick):-
2     PTick is Tick - 1,
3     actions:data(DataID,AgentID, PTick,_ContentType, _BaseType,
4         ↪ _Location, _Observation),!.
5     %addlog(AgentID, subaction, ongoing_experiment,_).
6
7 observe_event(AgentID, Event, Tick):-
8     neighborevents(Event,Observation),
9     create_world_module:event(Event, BaseType,_Tick, Location),
10    ContentType = observation_data,
11    uuid(DataID),
12    assertz(actions:belief(AgentID, Tick, ContentType, DataID, Domain,
13        ↪ 1.0)),

```

```

12     assertz(actions:data(DataID,AgentID, Tick, ContentType, BaseType,
13           ↪ Location, Observation)),
        addlog(AgentID, action, doing_empirical_research,_).

```

Wenn in $t_i = t-1$ entsprechende Daten vorliegen, wird mittels `ontinue_observation` ↪ /2 die Beobachtungsreihe weitergeführt.

```

1  continue_observation(AgentID, ListOfEventIDs):-
2      msg_with_simple_name(AgentID,'action: continue_observation'),
3      sim_time(Tick), PTick is Tick -1,
4      actions:data(DataID,AgentID, PTick,_ContentType, BaseType,
5           ↪ Location, _Observation),
6      findall(ID, create_world_module:event(ID, EventType, PTick,
7           ↪ Location), ListOfEventIDs),
        addlog(AgentID, subaction, continue_observation,_).

```

Sollten bisher noch keine Beobachtungen vorliegen, wird über `make_new_observation` ↪ /2 eine neue Beobachtung angestoßen.

```

1  make_new_observation(AgentID, Observation):-
2      msg_with_simple_name(AgentID,'action: make_new_observation'),
3      addlog(AgentID, subaction, make_new_observation,_),
4      sim_time(Tick),
5      get_interest(AgentID, EventType),
6      findall(ID, create_world_module:event(ID, EventType, Tick,
7           ↪ Location), Observation).

```

Die eigentliche Beobachtung erfolgt dann über `observation/3`. Das Ergebnis ist eine Liste von beobachteten Ereignissen.

```

1  observation(AgentID, Tick, ListOfBaseEvents):-
2      agent(_, AgentID, Attributes ),
3      Ability is Attributes.get(ability),
4      length(ListOfBaseEvents,ListOfBaseID ),
5      Scope is round(ListOfBaseID*Ability),
6      random_sublist(Scope, ListOfBaseEvents, Observation),
7      cores(NumberOfCores),
8      findall(Goal,observation_concurrent(Goal,Observation, AgentID,
9           ↪ Event,Tick),Goals),
        concurrent(NumberOfCores, Goals, []).
10
11 observation_concurrent(Goal, Observation, AgentID,Event,Tick):-
12     member(Event,Observation),
13     Goal = observe_event(AgentID,Event,Tick).

```

Literatur

- Abbott, B. P. u. a. (11. Feb. 2016). »Observation of Gravitational Waves from a Binary Black Hole Merger«. In: *Physical Review Letters* 116.6. ISSN: 0031-9007, 1079-7114. DOI: 10/gcp5km.
- Ahrweiler, Petra (2011). »Modelling Theory Communities in Science«. In: *Journal of Artificial Societies and Social Simulation* 14.4. ISSN: 1460-7425. DOI: 10.18564/jasss.1833.
- Andreas, Holger und Frank Zenker (Aug. 2014). »Basic Concepts of Structuralism«. In: *Erkenntnis* 79, S. 1367–1372. ISSN: 0165-0106, 1572-8420. DOI: 10/f22ktq.
- Baker, Monya (7. März 2016). »Statisticians Issue Warning over Misuse of P Values«. In: *Nature* 531.7593, S. 151–151. ISSN: 0028-0836, 1476-4687. DOI: 10/gc3f63.
- Balzer, Wolfgang (2003). »Wissen und Wissenschaft als Waren«. In: *Erkenntnis* 58.1, S. 87–110. ISSN: 01650106. DOI: 10/fsjq9n.
- (2015). »Scientific Simulation as Experiment in Social Science«. In: *Philosophical Inquiry* 39.1, S. 26–37. ISSN: 1105-235X. DOI: 10/f27vp8.
- Balzer, Wolfgang und Karl-Rudolf Brendel (2018). *Theorie der Wissenschaften*. 1. Auflage 2018. Wiesbaden: Springer VS. ISBN: 978-3-658-21221-6, 978-3-658-21222-3.
- Balzer, Wolfgang, Daniel Kurzawe und Klaus Manhart (2014). *Künstliche Gesellschaften mit PROLOG*. Göttingen: V&R unipress. ISBN: 978-3-8471-0332-5, 3-8471-0332-6.
- Balzer, Wolfgang und Klaus Manhart (2011). »A Social Process in Science and Its Content in a Simulation Program«. In: *Journal of Artificial Societies and Social Simulation* 14.4. ISSN: 1460-7425. DOI: 10/gdtmv9.
- (Aug. 2014). »Scientific Processes and Social Processes«. In: *Erkenntnis* 79.8, S. 1393–1412. ISSN: 0165-0106, 1572-8420. DOI: 10/gdm28x.
- Balzer, Wolfgang, Carles Ulises Moulines und Joseph D. Sneed (1987). *An Architectonic for Science: The Structuralist Program*. Dordrecht: Dordrecht u.a., Reidel, ISBN: 978-94-010-8176-4, 978-94-009-3765-9.
- Balzer, Wolfgang und Dieter Will (2019). »Eine Theorie der Wissenschaftlichen Gutachtersysteme«.
- Bauer, Martin W. (2017). »Kritische Beobachtungen zur Geschichte der Wissenschaftskommunikation«. In: *Forschungsfeld Wissenschaftskommunikation*. Hrsg. von Heinz Bonfadelli, S. 17–39.
- Beall, Jeffrey (15. Juni 2017). »What I Learned from Predatory Publishers«. In: *Biochemia Medica* 27.2, S. 273–278. ISSN: 1330-0962, 1846-7482. DOI: 10/gbmsg6.
- (2018). »Predatory Journals Exploit Structural Weaknesses in Scholarly Publishing«. In: *4open* 1, S. 1. ISSN: 2557-0250. DOI: 10/gfj97t.

- Beisbart, Claus und Stephan Hartmann (2011). »Computersimulationen in der Angewandten Politischen Philosophie – ein Beispiel«. In: *Deutsches Jahrbuch Philosophie. Lebenswelt und Wissenschaft*. Deutsches Jahrbuch Philosophie 2. Hrsg. von C. F. Gethmann, S. 1151–1162.
- Collaboration, The ATLAS (Sep. 2012). »Observation of a New Particle in the Search for the Standard Model Higgs Boson with the ATLAS Detector at the LHC«. In: *Physics Letters B* 716.1, S. 1–29. ISSN: 03702693. DOI: 10/jb7. arXiv: 1207.7214.
- Cyranoski, David (2009). »Cyranoski, D. Woo Suk Hwang convicted, but not of fraud.« In: *Nature* 461, 1181. DOI: <https://doi.org/10.1038/4611181a>.
- Deutsche Forschungsgemeinschaft, Hrsg. (2013). *Vorschläge zur Sicherung guter wissenschaftlicher Praxis: Denkschrift ; Empfehlungen der Kommission »Selbstkontrolle in der Wissenschaft« = Proposals for safeguarding good scientific practice: memorandum ; recommendations of the Commission on Professional Self Regulation in Science*. Weinheim: Wiley-VCH. ISBN: 978-3-527-33703-3, 3-527-33703-2, 978-3-527-33703-3.
- Dohmen, Dieter und Lena Wrobel (März 2018). *Entwicklung der Finanzierung von Hochschulen und außeruniversitären Forschungseinrichtungen seit 1995*. Endbericht einer Studie für Deutscher Hochschulverband. Berlin: Forschungsinstitut für Bildungs- und Sozialökonomie.
- Doublet, David R. (2003). »Der hermeneutische Zirkel: Über Grenzen für die Interpretation und Bedingungen für das Verstehen«. In: *Theorien und Methoden in den Sozialwissenschaften*. Hrsg. von Stein Ugelvik Larsen und Ekkart Zimmermann. 1. Aufl. Wiesbaden: Westdt. Verl, S. 61–77. ISBN: 978-3-531-13995-1, 978-3-322-80451-8.
- Forschungsgemeinschaft, Deutsche (Sep. 2019). *Guidelines for Safeguarding Good Research Practice. Code of Conduct*. DOI: 10.5281/zenodo.3923602.
- Frigg, Roman und Julian Reiss (Aug. 2009). »The Philosophy of Simulation: Hot New Issues or Same Old Stew?« In: *Synthese* 169.3, S. 593–613. ISSN: 0039-7857, 1573-0964. DOI: 10/cxd2ms.
- Fuhrin, Katharina (2013). *Der prominente Wissenschaftler*. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-02639-4, 978-3-658-02640-0. DOI: 10.1007/978-3-658-02640-0.
- Gärdenfors, Peter (2005). *The Dynamics of Thought*. Synthese Library v. 300. Dordrecht ; New York: Springer. ISBN: 978-1-4020-3398-8.
- (7. Apr. 2020). »Events and Causal Mappings Modeled in Conceptual Spaces«. In: *Frontiers in Psychology* 11. ISSN: 1664-1078. DOI: 10/ggr54d.
- Gibbons, Michael u. a. (1994). *The New Production of Knowledge: The Dynamics of Science and Research in Contemporary Societies*. London ; Thousand Oaks, Calif: SAGE Publications. ISBN: 978-0-8039-7793-8, 978-0-8039-7794-5.

- Giere, Ronald N. (1997). *Explaining Science: A Cognitive Approach*. 4. Auflage. Science and Its Conceptual Foundations Series. Chicago, Ill.: The Univ. of Chicago Press. ISBN: 978-0-226-29206-9.
- (2009). »Is Computer Simulation Changing the Face of Experimentation?« In: *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition* 143 (1), S. 59–62. DOI: 10/d75zzb. JSTOR: 27734388.
- Gilbert, G. Nigel und Klaus G. Troitzsch (2005). *Simulation for the Social Scientist*. 2. Auflage. Maidenhead, England ; New York, NY: Open University Press. 295 S. ISBN: 978-0-335-21600-0.
- Gläser, Jochen (2006). *Wissenschaftliche Produktionsgemeinschaften - Die Soziale Ordnung der Forschung*. Campus Forschung Bd. 906. Frankfurt/Main ; New York: Campus. 421 S. ISBN: 978-3-593-38186-2.
- Gramelsberger, Gabriele (2010). *Computerexperimente: Zum Wandel der Wissenschaft im Zeitalter des Computers*. Science Studies. Bielefeld: Transcript. 313 S. ISBN: 978-3-89942-986-2.
- Hagenhoff, Svenja u. a. (2007). *Neue Formen der Wissenschaftskommunikation: eine Fallstudienuntersuchung*. Göttingen: Niedersächsische Staats- und Universitätsbibliothek. ISBN: 978-3-938616-75-8, 3-938616-75-X.
- Hartmann, Stephan (1996). »The World as a Process. Simulations in the Natural and Social Sciences.« In: *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*. Hrsg. von Rainer Hegselmann, Ulrich Mueller und Klaus G. Troitzsch. Dordrecht: Springer Netherlands, S. 77–100. ISBN: 978-90-481-4722-9, 978-94-015-8686-3.
- Haug, Charlotte J. (17. Dez. 2015). »Peer-Review Fraud — Hacking the Scientific Publication Process«. In: *New England Journal of Medicine* 373.25, S. 2393–2395. ISSN: 0028-4793, 1533-4406. DOI: 10/gdwtpw.
- Hempel, Carl G. (15. Jan. 1942). »The Function of General Laws in History«. In: *The Journal of Philosophy* 39.2, S. 35. ISSN: 0022362X. DOI: 10/d3f8fc.
- Herb, Ulrich (2015). *Open Science in der Soziologie. Eine interdisziplinäre Bestandsaufnahme zur offenen Wissenschaft und eine Untersuchung ihrer Verbreitung in der Soziologie*. Schriften zur Informationswissenschaft 67. Glückstadt: Werner Hülsbusch.
- Hofmann, Solveig (2009). *Dynamik sozialer Praktiken. Simulation gemeinsamer Unternehmungen von Frauengruppen*. Wiesbaden: VS Verlag für Sozialwissenschaften / GWV Fachverlage GmbH, Wiesbaden. ISBN: 978-3-531-91807-5.
- Hornbostel, Stefan (1997). *Wissenschaftsindikatoren: Bewertungen in der Wissenschaft*. Opladen: Westdeutscher Verlag. 355 S. ISBN: 978-3-531-12908-2.
- Ihrig, Martin und Klaus G Troitzsch (2013). »An Extended Research Framework for the Simulation Era«. In: *Proceedings of the Emerging Me&S Applications in Industry & Academia Modeling and Humanities Symposium*. EAIA and Math '13. San Diego,

- California: Society for Computer Simulation International, S. 9. ISBN: 978-1-62748-039-0.
- Internationale Expertenkommission zur Evaluation der Exzellenzinitiative (Jan. 2016). *Internationale Expertenkommission zur Evaluation der Exzellenzinitiative - Evaluation der Exzellenzinitiative Endbericht der Internationalen Expertenkommission*. Endbericht. Berlin: Institut für Innovation und Technik (iit), S. 67.
- Jantzen, Benjamin C. (Okt. 2016). »Discovery without a 'Logic' Would Be a Miracle«. In: *Synthese* 193.10, S. 3209–3238. ISSN: 0039-7857, 1573-0964. DOI: 10.1007/s11229-015-0926-7.
- Knorr-Cetina, Karin und Rom Harré (2012). *Die Fabrikation von Erkenntnis: zur Anthropologie der Naturwissenschaft*. Erw. Neuaufl., 3. Aufl. Suhrkamp Taschenbuch Wissenschaft 959. Frankfurt am Main: Suhrkamp. 357 S. ISBN: 978-3-518-28559-6.
- Krause, Peter und Torger Möller (2008). »Vorwort: Die Förderinitiative Wissen für Entscheidungsprozesse - Forschung zum Verhältnis von Wissenschaft, Politik Und Gesellschaft«. In: *Wissensproduktion und Wissenstransfer: Wissen im Spannungsfeld von Wissenschaft, Politik und Öffentlichkeit*. Hrsg. von Renate Mayntz u. a. Science Studies. Bielefeld: Transcript, S. 11–19. ISBN: 978-3-89942-834-6.
- Krohn, W. und G. Küppers (1990). »Selbstorganisation Der Wissenschaft«. In: *Selbstorganisation: Aspekte einer wissenschaftlichen Revolution (Wissenschaftstheorie, Wissenschaft und Philosophie 29)* 29, S. 303–325.
- Kronick, David A. (1962). *A History of Scientific and Technical Periodicals: The Origins and Development of the Scientific and Technological Press, 1665-1790*. New York: Scarecrow Press. 274 p.
- Kuhn, Thomas S. (1962). *The Structure of Scientific Revolutions*. 3. Aufl. (1996). Chicago: University of Chicago Press. 212 S. ISBN: 978-0-226-45807-6, 978-0-226-45808-3.
- Lange, Marc (1. Sep. 2013). »What Makes a Scientific Explanation Distinctively Mathematical?«. In: *The British Journal for the Philosophy of Science* 64.3, S. 485–511. ISSN: 0007-0882, 1464-3537. DOI: 10/gftk2c.
- Langley, Pat u. a., Hrsg. (1987). *Scientific Discovery: Computational Explorations of the Creative Processes*. Cambridge, Massachusetts, London: MIT Press. 357 S. ISBN: 978-0-262-12116-3, 978-0-262-62052-9.
- Latour, Bruno und Steve Woolgar (2013). *Laboratory Life: The Construction of Scientific Facts*. Princeton: Princeton University Press. ISBN: 978-0691028323.
- Masterman, Margeret (1974). »Die Natur eines Paradigmas«. In: *Kritik und Erkenntnisfortschritt*. Hrsg. von Imre Lakatos und Alan Musgrave. Wissenschaftstheorie Wissenschaft und Philosophie 4. London: Vieweg, S. 59–88. ISBN: 978-3-322-90613-7.
- Milojević, Staša, Filippo Radicchi und John P. Walsh (11. Dez. 2018). »Changing Demographics of Scientific Careers: The Rise of the Temporary Workforce«. In: *Pro-*

- ceedings of the National Academy of Sciences* 115,50, S. 12616–12623. ISSN: 0027-8424, 1091-6490. DOI: 10/gfpgwt.
- Morrison, Margaret (März 2009). »Models, Measurement and Computer Simulation: The Changing Face of Experimentation«. In: *Philosophical Studies* 143,1, S. 33–57. ISSN: 0031-8116, 1573-0883. DOI: 10/d9rsc6.
- Moulines, Carles Ulises (1979). »Theory-Nets and the Evolution of Theories: The Example of Newtonian Mechanics«. In: *Synthese* 41,3, S. 417–439. DOI: 10/dksqb2.
- (2008). *Die Entwicklung der modernen Wissenschaftstheorie (1890 - 2000)*. LIT. ISBN: 978-3-8258-8965-4.
- (Aug. 2014). »Intertheoretical Relations and the Dynamics of Science«. In: *Erkenntnis* 79,8, S. 1505–1519. ISSN: 0165-0106, 1572-8420. DOI: 10/gdm28t.
- Nowotny, Helga, Peter Scott und Michael Gibbons (2003). »Introduction: Mode 2' Revisited: The New Production of Knowledge«. In: *Minerva* 41,3, S. 179–194. DOI: 10/frphm4.
- O'Connor, Cailin und Justin Bruner (4. Nov. 2017). »Dynamics and Diversity in Epistemic Communities«. In: *Erkenntnis*. ISSN: 0165-0106, 1572-8420. DOI: 10/gdqbpw.
- O'Connor, Cailin und James Owen Weatherall (2017). »Scientific Polarization«. In: *SSRN Electronic Journal*. ISSN: 1556-5068. DOI: 10/gdqbpz.
- Organisation for Economic Co-Operation and Development (2016). *OECD Factbook 2014: Economic, Environmental and Social Statistics*. ISBN: 978-92-64-23256-3.
- Parthey, Heinrich (2013). »Phantasie in der Forschung und Kriterien der Wissenschaftlichkeit«. In: *Kreativität in der Forschung*. Hrsg. von Thomas Heinze u. a. 2012. Berlin: wvb, Wiss. Verl, S. 9–28. ISBN: 978-3-86573-693-2.
- Pluchino, Alessandro, Alessio Emanuele Biondo und Andrea Rapisarda (Mai 2018). »Talent vs Luck: The Role of Randomness in Success and Failure«. In: *Advances in Complex Systems* 21 (03n04), S. 1850014. ISSN: 0219-5259, 1793-6802. DOI: 10/gd82dv. arXiv: 1802.07068.
- Pluchino, Alessandro, Giulio Burgio u. a. (11. Jan. 2019). »Exploring the Role of Interdisciplinarity in Physics: Success, Talent and Luck«. In: *PLOS ONE* 14,6. Hrsg. von Alejandro Raul Hernandez Montoya. DOI: 10.1371/journal.pone.0218793. arXiv: 1901.03607 [physics].
- Price, Derek J. de Solla (1971). *Little Science, Big Science*. 3. paperback print. Columbia Paperback 62. New York: Columbia Univ. Press. 118 S. ISBN: 978-0-231-08562-5.
- Rao, Anand S. und Michael P. Georgeff (1995). »BDI Agents: From Theory to Practice«. In: *Proceedings, First International Conference on Multi-Agent Systems (ICMAS-95)*, S. 312–319.
- Rüsen, Jörn (2013). *Historik: Theorie der Geschichtswissenschaft*. Köln: Böhlau Verlag. 322 S. ISBN: 978-3-412-21110-3.

- Saam, Nicole J. (2015). »Simulation in den Sozialwissenschaften«. In: *Handbuch Modellbildung und Simulation in den Sozialwissenschaften*. Hrsg. von Norman Braun und Nicole J. Saam. Wiesbaden: Springer Fachmedien Wiesbaden, S. 61–95. ISBN: 978-3-658-01164-2. DOI: 10.1007/978-3-658-01164-2_3.
- Smith, Richard (2006). »The Trouble with Medical Journals«. In: *Journal of the Royal Medicine* 99, S. 115–119. DOI: 10.1258/jrsm.99.3.115.
- Sneed, Joseph D. (1971). *The Logical Structure of Mathematical Physics*.
- Spier, Ray (Aug. 2002). »The History of the Peer-Review Process«. In: *Trends in Biotechnology* 20.8, S. 357–358. ISSN: 01677799. DOI: 10/d26d8b.
- Statistisches Bundesamt (Destatis) (28. Nov. 2017). *Schnellmeldungsergebnisse der Hochschulstatistik zu Studierenden und Studienanfänger/-innen - vorläufige Ergebnisse - Wintersemester 2017/2018*. Wintersemester 2017/2018. Statistisches Bundesamt (Destatis), S. 13.
- (7. Sep. 2018). *Studierende an Hochschulen. Wintersemester 2017/2018*. Fachserie 11 Reihe 4.1. Statistisches Bundesamt (Destatis).
- (2016). *Bevölkerung und Erwerbstätigkeit. Zusammenfassende Übersichten Eheschließungen, Geborene und Gestorbene 1946 - 2015*. Wiesbaden: Statistisches Bundesamt (Destatis).
- Suppes, Patrick (1957). *Introduction to Logic*. The University Series In Undergraduate Mathematics. Cincinnati, New York, Chicago, Millbrae, Dallas: Von Nostrand Reinhold Company.
- Takahashi, Shingo, David L. Sallach und Juliette Rouchier, Hrsg. (2007). *Advancing Social Simulation: The First World Congress*. 1. Auflage. New York: Springer. 354 S. ISBN: 978-4-431-73150-4.
- Taubert, Niels (2017). »Formale wissenschaftliche Kommunikation«. In: *Forschungsfeld Wissenschaftskommunikation*. Hrsg. von Heinz Bonfadelli u. a. Wiesbaden: Springer Fachmedien Wiesbaden, S. 125–139. ISBN: 978-3-658-12897-5, 978-3-658-12898-2. DOI: 10.1007/978-3-658-12898-2_7.
- Taubert, Niels und Peter Weingart (2016). »Wandel des wissenschaftlichen Publizierens – eine Heuristik zur Analyse rezenter Wandlungsprozesse«. In: *Wissenschaftliches Publizieren – Zwischen Digitalisierung, Leistungsmessung, Ökonomisierung und medialer Beobachtung*. Hrsg. von Peter Weingart und Niels Taubert. Berlin: De Gruyter Akademie Forschung, S. 3–38.
- Taubert, Niels C u. a. (2018). *The Future of Scholarly Publishing Open Access and the Economics of Digitisation*. Baltimore, Maryland: Project Muse. ISBN: 978-1-928331-54-4.
- Thagard, Paul (1988). *Computational Philosophy of Science*. Cambridge, Mass: MIT Press. 240 S. ISBN: 978-0-262-20068-4.

- Thagard, Paul und Fred W. Kroon (29. Mai 2006). »Emotional Consensus in Group Decision Making«. In: *Mind & Society* 5.1, S. 85–104. ISSN: 1593-7879, 1860-1839. DOI: 10.1007/s11299-006-0011-5.
- Triggle, Chris R. und David J. Triggle (2007). »What Is the Future of Peer Review? Why Is There Fraud in Science? Is Plagiarism out of Control? Why Do Scientists Do Bad Things? Is It All a Case of: "All That Is Necessary for the Triumph of Evil Is That Good Men Do Nothing?"« In: *Vascular Health and Risk Management*, S. 16.
- Tröger, Jochen (2004). »Drittmittel aus der Sicht der Universität«. In: *Drittmittelleinwerbung — Strafbare Dienstpflicht?* Hrsg. von Brigitte Tag, Jochen Tröger und Jochen Taupitz. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 5–14. ISBN: 978-3-642-17055-3. DOI: 10.1007/978-3-642-17055-3_1.
- Troitzsch, Klaus G. (1997). »Social Science Simulation – Origins, Prospects, Purposes«. In: *Simulating Social Phenomena*. Hrsg. von Rosaria Conte, Rainer Hegselmann und Pietro Terno. Lecture Notes in Economics and Mathematical Systems 456. Berlin, New York: Springer, S. 42–54. ISBN: 978-3-540-63329-7.
- (2017). »Axiomatic Theory and Simulation: A Philosophy of Science Perspective on Schelling's Segregation Model«. In: *Journal of Artificial Societies and Social Simulation* 20.1. ISSN: 1460-7425. DOI: 10.18564/jasss.3372.
- (6. Feb. 2019). »Axiomatisation and Simulation«. In: *Information* 10.2, S. 53. ISSN: 2078-2489. DOI: 10/gfwrdd.
- Wajnborg, Charles-David u. a. (2004). »A Structuralist Approach Towards Computational Scientific Discovery«. In: *Discovery Science* Volume 3245, S. 412–419. DOI: 10/drxvtw.
- Wasserstein, Ronald L. und Nicole A. Lazar (2. Apr. 2016). »The ASA's Statement on p-Values: Context, Process, and Purpose«. In: *The American Statistician* 70.2, S. 129–133. ISSN: 0003-1305, 1537-2731. DOI: 10/bc4d.
- Winterhager, Nicolas (2015). *Drittmittelwettbewerb im universitären Forschungssektor*. Organization & Public Management Research. Wiesbaden: Springer VS. 287 S. ISBN: 978-3-658-07187-5, 978-3-658-07188-2.
- Wooldridge, Michael J. (2000). *Reasoning about Rational Agents*. Intelligent Robotics and Autonomous Agents. Cambridge, Mass: MIT Press. 227 S. ISBN: 978-0-262-23213-5.
- Zacharias, P., H. Peter und Sven Bingert (Juli 2011). »Investigation of Mass Flows in the Transition Region and Corona in a Three-Dimensional Numerical Model Approach«. In: *Astronomy & Astrophysics* 531. ISSN: 0004-6361, 1432-0746. DOI: 10/d7243r.
- Zilsel, Edgar (1976). »Die sozialen Ursprünge der neuzeitlichen Wissenschaft«. In: *Die sozialen Ursprünge der neuzeitlichen Wissenschaft*. Hrsg. von Wolfgang Krohn. 1. Aufl. Suhrkamp-Taschenbuch Wissenschaft 152. Frankfurt am Main: Suhrkamp.

Zollman, Kevin (2011). »Computer Simulation and Emergent Reliability in Science«. In: *Journal of Artificial Societies and Social Simulation* 14.4. ISSN: 1460-7425. DOI: 10/gdm9bx.

Stichwortverzeichnis

- Agenten, 83
- Aktion, 92
- Aktionen, 85, 93, 111
- Analyse, 109
- Antrag, 69
- Antragsverfahren, 100
- arXiv, 59

- Basiskonfiguration, 109
- Basissimulation, 109
- Beispiel: Entwicklung der Agenten, 110
- Beispiel: Publikationsformen, 112
- Beobachtung, 102

- Datengrundlage, 58
- Datenpublikation, 98
- double-blind peer review, 57
- Drittmittelförderung, 46

- Empirische Beobachtung, 103
- Ereignis, 35, 90
- Event, 35

- Fakt, 30
- Forschende, 45, 47
- Förderer, 48
- Förderung, 68

- Gesetz, 36, 88
- Gutachter, 56, 69, 102
- Gutachterverfahren, 101, 102
- Gute wissenschaftliche Praxis, 55

- Handlung, 72, 85, 92, 118
- Handlungen, 50
- Horn-Klausel, 30
- Hypothese, 38

- Interessensgebiet, 105

- Kommunikation, 52, 53

- Maschinelles Lernen, 21
- Methode, 15, 27
- Modell, 29

- Namen, 86
- Netze, 49

- Objekt, 38
- Open Access, 114

- Plagiat, 107
- Planung, 85
- Prolog, 23, 29
- Publikation, 53, 98, 99, 116
- Publikationsformen, 112
- Publikationsnetzwerke, 58
- Publikationsprozess, 55, 98

- Quasiordnung, 31

- Regel, 30
- Reputation, 72
- Ressourcen, 118
- Review, 99, 101, 114
- Reviewverfahren, 56, 99

- Simulation, 15
- Simulationsebenen, 7
- Simulationskern, 78
- Simulationskonfiguration, 84
- Simulationslauf, 34
- Simulationsschritt, 82
- Simulationsumgebung, 77
- Simulationszeit, 84
- Simulierte Welt, 34
- Strategie, 94
- Strukturalismus, 29
- Studierende, 48

- Theorie, 28, 29, 32, 95
- Theorie Typ 1, 96
- Theorie Typ 2, 97
- Theoriefindung, 33, 95
- Täuschung, 106

- UUID, 86

- Validierung, 105

- Welt, 35, 87
- Wissen, 50, 118
- Wissenschaft, 63
- Wissenschaftsgemeinschaften, 47, 49
- Wissenschaftspolitik, 44

- Zitation, 116

Abbildungsverzeichnis

1.1	Überblick über die Ebenen	8
1.2	Mögliche Varianten der Theoriebildung durch empirische Beobachtung	9
2.1	Definition der Ereignisse	35
2.2	Gesetze bestehend aus einem Ausgangszustand und einem Zielzustand	36
2.3	Struktur der Gesetze einer simulierten Welt	37
2.4	Definition der Objekte, Typen und Eigenschaften	38
3.1	Studierende pro Jahrgang.....	44
3.2	Verhältnis zwischen Personal und Studierenden	45
3.3	Rollen in der Wissenschaft	47
3.4	Relative Entwicklung wachsender Kosten von Publikationen in Bezug zur Zahl von Forschenden und Budgets für Bibliotheken.....	55
3.5	Bewertungen und Gutachten.....	58
3.6	Anstieg der Publikationen pro Jahr	59
3.7	Netzwerk der Co-Autoren	61
4.1	Forschung und Gesellschaft.....	65
4.2	Fördervolumen der EU-Forschungsprogramme	67
4.3	DFG-Antragsverfahren als Prozess.....	68
5.1	Start der Simulation.....	78
5.2	Schema der äußeren Schleife.....	78
5.3	Zusammenfassung der Simulationsergebnisse.....	81
5.4	Schema innerhalb eines Ticks	82
5.5	Aktionen und Ressourcen	86
5.6	Abbildung von Ereignissen im Raum	90
6.1	Wahrscheinlichkeit zum Ausstieg nach Agentenalter in %.....	110
6.2	Häufigkeitsverteilung und Gesamtzahl der Agenten pro Lauf	112
6.3	Verteilung der Aktionen der Forschenden <i>Maggie51</i>	113
6.4	Absolute Anzahl von Publikationen und Ablehnungen nach Journal	115
6.5	Die Verteilung der Erfolgsrate der Publikation nach Journaltypen.....	116
6.6	Die Anteil der Zitationen nach Journaltyp.....	117
6.7	Publikationen im Verhältnis zu Zitationen (mit und ohne Peer-Review)	117
6.8	Vergleich zweier Fundingstrategien: Obergrenze bei der Bewilligung von 20 Ressourceneinheiten vs. 6 Einheiten	119
6.9	Aktionen des unterfinanzierten Agenten <i>DeMarcus59</i>	119

Tabellenverzeichnis

3.1	Verfügbare Metadaten	60
3.2	Publikationen pro Autor.....	60
5.1	Agentenkonfiguration.....	84
5.2	Simulationskonfiguration	85
5.3	Weltengenerierung mit Ereignissen	87
5.4	Weltengenerierung mit Objekten.....	88
5.5	Durch Aktionen erzeugte Prädikate	93
6.1	Standardkonfiguration der simulierten Welt.....	109
6.2	Ressourcen der Agenten	110
6.3	Verteilung des Alters.....	111
6.4	Auszug aus den Aktionen eines Agenten	111
7.1	Übersicht über die Module der Simulation.....	123

Wie sind Wissenschaft und Gesellschaft verzahnt und welche Auswirkung haben soziale Prozesse auf die Entwicklung wissenschaftlicher Theorien? Dieses Buch beschreibt einen Rahmen für die übergreifende Betrachtung der Wissenschaft. Von der Wissenschaftstheorie ausgehend werden soziologische wie gesellschaftswissenschaftliche Perspektiven in einer Computersimulation zusammengeführt.

Eine eigens in Prolog entwickelte Simulationsumgebung bildet die Grundlage für konkrete Simulationsexperimente und zeigt so Querbeziehungen zwischen verschiedenen Betrachtungsebenen mit entsprechenden Einflussfaktoren auf wissenschaftliche Prozesse. Gefragt wird zum Beispiel danach, welche Auswirkungen Plagiate auf die Theoriebildung haben oder inwiefern Publikationsprozesse oder die Verteilung von Forschungsbudgets Einfluss nehmen.

Betrachtet man aktuelle Tendenzen, etwa die wachsende Zahl von Publikationen sowie die zunehmende Bedeutung von Drittmitteln in der Forschung, wird die gesellschaftliche Relevanz dieser Themen deutlich. Entsprechend bietet der in diesem Buch vorgestellte Rahmen nicht nur eine Brücke zwischen verschiedenen Betrachtungsebenen, sondern auch das Potential für ein Werkzeug zur Steuerung wissenschaftlicher Prozesse und Rahmenbedingungen.

Daniel Kurzawe studierte Logik und Wissenschaftstheorie, Computerlinguistik und Philosophie an der Ludwig-Maximilians-Universität München, wo er dann auch 2020 promovierte. An der Staats- und Universitätsbibliothek Göttingen forscht und entwickelt er informationswissenschaftliche Methoden und digitale Forschungsinfrastrukturen.

44,90 €
ISBN 978-3-487-16307-9



www.olms.de