Der Weltenmotor

Paul Geisler

'Der Weltenmotor' ist ein Versuch, eine möglichst komplexe virtuelle 'Welt' aus dem Nichts zu erschaffen. Sie sollte in vielfältiger Weise unbegrenzt sein; so wie unsere reale, in der wir leben, uns erscheint.

Dieses Dokument ist nicht als faktisches Resultat zu lesen, eher als Arbeitsansatz. Insbesondere sind die wie üblich gedruckten **Definition**en natürlichsprachig und damit eher unpräziser Natur.

Einführung

Blutbäder, Kampfkommerz und oberflächliches Rumglitzern – viele Menschen sind von der Welt, in der sie mehr oder weniger unschuldigerweise leben, zu recht genervt. Der Bedarf an Alternativen ist also groß. Während er am einfachsten mit Träumen zu befriedigen ist, so ist er am ausgefeiltsten mit dem Computer zu umgarnen. Die Industrie kennt sich damit aus, und schickt so hunderttausende *kreativer* Menschen in ihren Studiohöllen an den die Tastaturen, dem Konsumenten mit ausgefeilten Weltenentwürfen zu schmeicheln. Die Begeisterung für diese Produkte ist groß; jedoch zu Unrecht: Jedes durchschnittliche Wohnzimmer stellt mehr Interaktionsmöglichkeiten zur Verfügung als ein gefeiertes *Computerspiel*. Wer in einem solchen auf die Idee kommt, seine Bude spontan zu berserken, wird im Spiel meist nicht viel Freude finden: die kreativen Menschen haben daran nicht gedacht. Wer einfach losrennt, stößt im *echten Leben* meist rasch an die Grenze seiner Kondition. Im Computerspiel dagegen oft an den Rand der Scheibe. Interessanterweise ist dieser meist gut versteckt, mann fällt nicht herunter, weil man ihn gar nicht erreicht. Mauern verhindern das. Um dieses Elend anzugehen, benötigen wir eine bessere Welt. Der *Weltenmotor* macht uns eine.

1 Der Weltenmotor - eine Lösungssicht

Ein zentrales Prinzip besteht in der Betrachtung aller Systeme von Seiten ihrer Lösung her. Der Weltenmotor ist darauf angewiesen.

Eine klassische Simulation eines (physikalischen, elektrischen, chemischen) *Systems* beruht in der Regel auf der Infinitesimalrechnung, es nützt die Möglichkeit, einen zukünftigen Systemzustand um so einfacher beschreiben zu können, je weniger weit wir in die Zukunft vorgreifen müssen. Die Grundlagen hierzu sind Linearisierung und Diskretisierung. Die Zeit und der Zustandsraum werden auf endliche, diskrete Räume reduziert. Statt unendlicher Zeitpunkte haben wir nur noch 'Jetzt' und 'Gleich'. Wenn man es ganz einfach haben will, sind die diskreten Zeitpunkte äquidistant, die Abbildungsvorschrift, die uns einen Zustand aus dem Vorhergehenen gewinnen läßt, dann von ihm unabhängig, sofern das System autonom war. Wenn wir von 'sehr kleinen' Zeitschritten ausgehen, und unser System für stetig (am besten stetig differenzierbar oder gar analytisch) halten, so liegt Linearisierung nahe:

$$\text{Welt(Gleich)} = \text{Welt(Jetzt + Augenblick)} = \text{Weltenfunktion(Welt(Jetzt), Augenblick)}$$

All diese Techniken der Numerik der dynamischen Systeme können Großes leisten, wenn man einer Abfolge von Systemzuständen interessiert ist, und was mehr als eine ablaufende Simulation kann man sich noch wünschen? Ich möchte hier einen neuen Begriff einführen, zunächst ihn aber nennen: Eine *globale Simulation*, selbsterklärender auch *Simulation wahlfreien Zugriffs* genannt.

Definition 1 (Simulation wahlfreien Zugriffs)**.** Eine *Simulation wahlfreien Zugriffs* ist ein dynamisches System mit einer geschlossenen, expliziten, zu jedem Zeitpunkt direkt unabhängig von den Lösungen zu anderen Zeiten berechenbaren Lösung.

Beispiel 2. *Das durch die simple Differentialgleichung (DGL) $\dot{x}(t) = x(t)$ erzeugte dynamische System hat eine eindeutige, geschlossene Lösung in Form der analytisch ermittelten Funktion $x(t) = ce^t$. Nun ist zwar zur Bestimmung eines Wertes eine Festlegung von c z.B. mit Hilfe eines Anfangswertes $x(0) = 1$ nötig, jedoch kann danach zu jedem Zeitpunkt t eine Auswertung erfolgen, ohne dass das Verhalten des Systems auf dem Intervall $(0, t)$ von irgendeinem Interesse sein muss.*

Diese Art von System hat einen unschätzbaren Vorteil: Beliebige Systeme erfordern zur Berechnung eines einzelnen exakten Lösungswertes die Bestimmung einer riesigen Schar weiterer, zumeist aller aus einer dichten Teilmenge des Raumes: Einer Trajektorie. Dies begrenzt die Anwendung auf endliche Räume über diskreten, endlichen Zeitintervallen bei endlicher Genauigkeit (Finite-Elemente-Methoden, zellulare Automaten). Die globale Simulation jedoch kann aus unendlichen Räumen schöpfen: Jede gewünschte Auswertung an einer Stelle zu einem Zeitpunkt kann vorgenommen werden, der 'unendliche Rest' interessiert nicht.

In dem wir den Begriff 'dynamisches System' breit wählen, 'gibt' zu 'vermutlich geben könnte' aufweichen und 'erzeugendes' zu 'der Lösung nahes erzeugendes' abschwächen, kommen wir, zum Leid des Mathematikers, in die algorithmische Realität. Durch die Existenz des nahen 'dynamischen Systems' erwerben wir uns das Recht, unsere berechenbaren Funktionen 'Lösung' zu nennen, und das billig, billig präsentieren dieser 'Lösung' 'Simulation'. Das ist ziemlicher Betrug am Betrachter, aber niemand kann leugnen, dass die Beschreibung ungehemmten Wachstums durch $x(t) = e^t$ irgendwie fälschlich sei, auch wenn er (und ich) noch nie von Differentialgleichungen gehört hätte.

Ziel wird es nun sein, ein Modell einer vorstellbaren, am besten unserer *Welt* zu schaffen, dass einer globalen Simulation genügt. Im Idealfall hieße dies, eine Funktion zu besitzen, die es auf einfachem Wege ermöglicht, jeden Ort des Universums zu jeder Zeit zuhause auf dem Fernsehschirm (oder zumindest in einem Supercomputer der U.S. Army) sichtbar macht. Dies könnte man durch geduldiges Sammeln von möglichen Blicken unter starken Einschränkungen des Definitionsbereiches machen. Mit einer vorgegeben Trajektorie im Phasenraum nennt man es *Kino*, mit einer etwas flexibleren *Computerspiel*. Wenn wir kleinere Räume als Bildraum wählen (Text etc.), haben wir z.B. einen *Roman* oder ein *Pornoheft*. Es ist unschwer zu erkennen, dass die dort abgebildete Welt auch nur mäßig mit der realen übereinstimmt.

Nichts größeres in dieser Beziehung leisten wollend, schlage ich jedoch vor, zumindest die Größe des Urbildraumes etwas flexibler zu gestalten. Ich dachte da an sagen wir mal eine Galaxie als Vorbild für Länge, Höhe und Breite, und eine unvorstellbare aber von jedem aufschreibbare Zeitspanne. Selbstredend, dass hier auf den anderen Gebieten Abstriche gemacht werden müssen. So erlaube ich zunächst die Darstellung 'einer Welt', nicht unserer Welt, allerdings mit Offenhaltung der Möglichkeit, unsere Welt doch irgendwo in ihr zu finden. Dazu spielen *Konstanten* eine wichtige Rolle, ob sie es nun wirklich sind oder nicht, als solche empfunden sind sie für unsere Wahrnehmung der Welt von essentieller Bedeutung: Allen voran die *Naturgesetze*. Ich spreche von ihnen in folgender Form:

Definition 3 (Naturgesetze)**.** Die *Naturgesetze* nennt man einen Satz von Regeln die den Fluß des von ihnen erzeugten dynamischen Systems eindeutig in- oder explizit definieren.

Damit können wir das Ziel neu beschreiben: Gesucht ist die Lösung einer globalen Simulation deren Naturgesetze die bekannten bzw. vorgegebenen approximieren.

2 Figur und Hintergrund

Es stellt sich die Frage, wieviele Informationen wir zum Errichten unserer Welt brauchen.

Beispiel 4. *Die bisherige Umsetung des Weltenmotors deklariert als* 'Daten' *nur einen Text, z.B:*

Dronus, Konus & Dronulli Inc.

der als einzige Basis für alle dann scheinbar zufälligen Ausstaffierungsmerkmale unserer Welt sorgt.

Nun wäre es jedoch naiv, zu behaupten, dieses wäre also der ganze Informationsgehalt des Universums. In der Tat gibt ja der Algorithmus, das *Programm* vor, wie ein Planet aufgebaut ist. Lediglich die Ausprägungen der mannigfaltigen Möglichkeiten hängt von der Datenbasis ab. Grob gesprochen wird hier versucht, nur die Grundlage für willkürliche Varianzen Daten zu nennen, der invariante Anteil, also so gesehen eine Auslegung der Naturgesetze, ist Programm.

Beispiel 5. *Zieht man das Computerspiel heran, so heißen obige Daten oftmals* Content*. Die Algorithmen sind* maschinencodiert *in den* Binarys *versteckt*[1]*. Content sind klassischerweise Grafiken (Bitmaps; Texturen), Sounds, Musik, 3D-Modelle, Texte und Leveldaten*[2]*. Die Binarys enthalten möglichst universelle Algorithmen. So wird in der Regel der Spielinhalt möglichst im Content abgelegt, die Binarys immer universellere Content-Darstellungs-Systeme, sogenante Engines*[3]*. Hier ist bereits der Ansatz zu finden, dass die Gesetze der Simulation als universell anerkannt in den 'festen' Binarys verschwinden, die flexible Ausprägung der Welt dann durch Content beschrieben wird. Gesetze sind hier immer weniger Dinge, zumindest meist aber: Der Raum ist dreidimensional, es gibt Licht und Schatten, meist wirkt Schwerkraft usw..*

[1] Dies ist die verbreitetste Form zur Zeit, 2006.
[2] Eine Art Verzeichnis, die Positionen und das Zusammenspiel der übrigen Daten beschreiben
[3] Dieses Wort inspiriert auch den Namen Weltenmotor

Wenn wir also immer weniger Daten fordern, auf der anderen Seite dabei aber Invariantes in das Programm packen, so bleibt die Frage: Was ist Inhalt, was ist Kontext? Ohne Kenntnis der deutschen Sprache und was 'Computerspiel' eigentlich ist, wäre ja auch dieser Text nicht interpretierbar. Das sehr komplexe Programm interpretiert also sehr wenig Inhalt. Hofstaedter behandelt diese Frage in seinem Buch [Hof79], er nennt es die Frage nach Figur und Hintergrund[4].

So bleibt uns nur, diese faulen Tricks hinzunehmen. Vielleicht nimmt man eine Trennung zwischen Programmierer und *Benutzer* vor. Der Benutzer ist dann in der Lage, durch Eingabe der Parole ein Universum auswählen, und so, man schreibe es in der Werbung, den gesamten Inhalt selbstzubestimmen. Vor dieser nüchternen Auffassung ist es auch nur legitim, wenn dann der Code selbst kleine Mengen 'Daten' enthält; z.B. benötigte Naturkonstanten. Die übrigen Daten bennnen wir:

Definition 6 (Grundparameter)**.** Die willkürlich wählbaren Daten, die unter allen möglichen den Naturgesetzen genügenden Strukturen ein bestimmtes Universum auswählen, nennen wir *Grundparameter*.

Unsere Grundparameter können nach obigem Beispiel etwa 'Dronus, Konus & Dronulli Inc.' sein, vielleicht noch ein Startzeit- und Raumpunkt dazu. Es bleibt der Rest:

Definition 7. Die Algorithmen und Daten, die versuchen ein möglichst breites Feld von Universen die den Naturgesetzen genügen zu erzeugen, heißen *Weltenmotor*. Die wenigen enthaltenen numerischen Daten nennen wir *Naturkonstanten*.

Diese unscharfe Definition wird vor allem durch die Vorverdauung des Naturgesetzes gründlich verwaschen: Natürlich sollte ein intelligenter Algorithmus die Notwendigkeit einer Regenrinne erkennen, sobald ihm klar ist, das a) die Menschen es gerne trocken haben, und b) die Schwerkraft kondensierendes Wasser nach unten fallen läßt. Da ich jedoch weit entfernt von der Lage bin, intelligente Algorithmen anzuwenden, teile ich dem Programm eben von Schrägdächern und Regenrinnen mit. Natürlich sind dies vorgekaute Daten. Leider. Sie sind in gewissem Sinne redundant. Aber es sollte das nötigste getan werden, sie möglichst vielfältig in der Ausübung ihrer Gesetzerfüllung zu halten. So ist eine Regenrinne sicher in vielen Formen und Variationen denkbar, der Weltenmotor sollte sich alle Mühe geben, diese wahr werden zu lassen: *Was möglich ist, ist auch irgendwo zu finden.*

3 Hierarchie

Wie finden wir uns nun in einer unendlichen Welt zurecht, stellen sie dar? Es ist plausibel, nach hierarchischen Strukturen der simulierten Welt zu fragen: Hierarchie ist immer ein Konzept des Unterteilens ('Teile und Herrsche') und läßt hoffen, dass sich der Fokus einer Betrachtung irgendwie beschränken läßt. In einer totalen Hierarchie mit nur einem Element an der Spitze erhalten wir einen *Baum*, es ist klar, daß jede partielle Betrachtung letztendlich in einem *Knoten* zusammengefaßt ist. Genauer:

[4] Hofstaedter benutzt hier die seltene Bedeutung von Hintergrund, die ich lieber mit Hintergrundwissen bzw. Kontext bezeichnen würde. Vgl. englisch *background*

Definition 8 (Baum[5] und Zubehör)**.** Ein Netz aus *Knoten, Objekten* der keine Kreise oder Schleifen enthält, nennt man einen *Baum.* Insbesondere kann man einen Knoten auswählen, von dem aus es dann genau einen *Pfad* zu jedem anderen Knoten gibt. Wenn man einen bestimmten Knoten als Ursprung auswählt, die *Wurzel*, so kann man alle anderen Knoten in Ebenen einteilen; nach der Anzahl der Knoten auf dem Weg zum Wurzelknoten. Alle Knoten die am Ende eines Weges stehen, nennen wir *Blätter.*

Im Extremlfall ist der Startknoten die oberste je zu betrachtende Hierarchiestufe. Besser allerdings ist es ein Knoten irgendwo im Hierarchiebaum. Bei der Darstellung kann man dann von einem Knoten ausgehend jeweils die Frage stellen, ob Dinge *oberhalb* oder *unterhalb* noch von Belang sind. In unserer realen Welt bekannte Hierarchien beschäftigen sich oft mit Macht, Organisation und Ort. In der Simulation wollen wir zunächst den Ort favorisieren: Ort trennt Dinge einfach und effizient. Im Extremfall nähmen wir den Ereignishorizont zur Hilfe, und glaubten Einstein, so können wir sagen, dass von einem Ereignis in einem Ort nur benachbarte betroffen werden, mit einer sich allerdings rasant mit Lichtgeschwindigkeit ausdehnenden Nachbarschaft. In unserem Erfahrungsraum wird es noch weniger: Die für uns schnellsten Signale sind elektrische Kommunikationen, die unmittelbaren jedoch beschränken sich auf Geh-, Fahr- Wurfgeschwindigkeit und unsere unmittelbare Umgebung, unsere Blickweite. Hier ist eine örtliche Hierarchie ein effizientes Werkzeug: In einem ganzen Universum ist der menschliche Blick eher in einer tiefen Hierarchieebene abgegolten. Um sich das klarzumachen ein

Beispiel 9. *Ein simple örtliche Hierarchie wäre*

$$\textit{Universum} \to \textit{Galaxie} \to \textit{Sonne} \to \textit{Planet} \to \textit{Stadt} \to \textit{Haus} \to \textit{Zimmer.}$$

Die für den Menschen direkt wahrnehmbaren Vorgänge laufen nah ab: Er sieht meist im Zimmer, selten aus dem Fenster die Stadt. Vielleicht scheint die Sonne auf ihn. Von der Galaxie bekommt er höchstens nachts etwas mit, wenn es dunkel ist: Einige wenige Lichtstrahlen lassen ihm die Sterne funkeln.

Die Welt ist also sehr viel kleiner, wenn man sie auf die Notwendigkeit ihrer Erfahrung durch den Menschen reduziert. Interessant ist jedoch die Tatsache, das die Galaxis nachts eine Rolle spielt; dies liegt am hohen 'Gewicht' der Sterne: Sie sind so groß, dass ihre Wirkung auch in einer fernen Nachbarschaft für den Menschen empfindbar ist. Dies legt nahe, dass wir uns nicht innerhalb einer Hierarchieebene auf unsere Umgebung beschränken können, sondern auch nach 'oben' und 'unten' forschen müssen. So ist in einer simplen Welt z.b. die Galaxie ein akzeptabler Abbruchpunkt, und nach 'unten' vielleicht ein kleiner Gegenstand. Die Atome interessieren uns nicht mehr. Welchen immensen Wert diese hierarchische Struktur der Welt hat, bemerken wir im nächsten Abschnitt.

3.1 Methoden der Auswertung

Wir wollen nun verschiedene Arten eine Struktur auszuwerten und darzustellen betrachten und feststellen wann Hierarchie nützlich ist. Wir stellen zunächst einige Begriffe zum Darstellen einer Figur über einem Hintergrund fest:

[5] Dieser hier schnell umrissene Begriff wird in der *Graphentheorie* definiert und ausführlich untersucht. Wir verwenden hier die gebräuchliche, dem Biologen unverständliche Variante, bei dem die Wurzel *oben*, das Blattwerk eher unten ist.

Definition 10 (Zeichnen)**.** Die Darstellung einer Figur durch einzeichnen in den Hintergrund nach ihrer gegebenen Parametrisierung nennen wir *Zeichnen*.

Dieser Begriff ist sehr intuitiv: Eine Linie wird ihrer Länge nach, eine Fläche vielleicht als Schraffur dargestellt, so man es mit dem Stift in der Hand täte.

Definition 11 (Abtasten)**.** Wir gewinnen die Darstellung der Figur, indem wir den gesamten Hintergrund nach ihr *abtasten*, und zu ihr zugehörige Teile einzeichnen.

Dieses Verfahren ist offensichtlich vor allem dann sinnvoll, wenn es zu jedem Ort des Hintergrundes etwas darzustellen gibt, z.B. beim *Texture Mapping* oder wenn die Geometrie der Figur nur räumlich implizit beschrieben werden kann.

Definition 12 (Rekursives Zeichnen)**.** Wenn wir eine Figur als figürliche Anordnung von Teilfiguren zeichnen, so nennen wir das *Rekursives Zeichnen*.

Hier scheint sich wieder Hierarchie einzuschleichen. Insbesondere kann eine Teilfigur vernachlässigt, oder durch eine einfachere ersetzt werden, wenn eine hinreichend grobe Darstellung ausreicht.

Im Weltenmotor finden alle diese Methoden ihre Verwendung. Am grundlegendsten ist jedoch das rekursive Zeichnen, dass, nur unbenannt, bereits zu Beginn des Kapitels angesprochen wurde. Wir wollen es nun eingehender betrachten. Figur und Teilfigur sind einfach Knoten unseres Baumes. Da wir jeden Knoten des Baumes als Wurzel bezeichnen können[6], benennen wir:

Definition 13 (Weltbaum, Weltwurzel)**.** Die Baumdarstellung unserer Welt nennen wir *Weltbaum*, wenn alle Objekte so beschaffen sind, dass jeder Knoten seine Kinder räumlich einschließt. Der Wurzelknoten dieses Baumes heißt dann *Weltwurzel*.

Nun sind wir nicht gewohnt, dass alle Objekte um uns herum vollständig ineinander stecken. Aber durch benennen neuer Objekte lässt sich dieser Zustand leicht herstellen:

Beispiel 14. *Eine kleine Welt, die nur unser Sonnensystem beinhaltet, hätte als Weltwurzel das Sonnensystem, als Kinder die Sonne und die Planetensysteme, diese wiederum ihre eigenen Planeten und alle seine Monde als Kinder. Die ...-system-Objekte sind nicht anfassbar, sie sind sie räumlich sehr leer (die Kinder machen nur wenig Volumen aus), dafür sind alle Objekte jetzt räumlich ineinander eingebettet.*

Wenn wir die Welt nun darstellen, können wir mit der Weltwurzel beginnen, jedes Kind auf *Sichtbarkeit* testen, und gegebenenfalls den entprechenden Teilbaum auf die selbe Weise behandeln. In der Praxis gibt es jedoch weitere Merkmale, die alternative Darstellungsreihenfolgen erforderlich machen. Wir betrachten hier nur das *Darstellen mit beliebiger Wurzel*: Der Weltbaum kann ja von jedem Knoten aus *traversiert* werden. Wir wählen also eine *Darstellungswurzel* aus. Eines ihrer Kinder ist nun im ursprünglichen Weltenbaum, der von der Weltenwurzel ausgeht, ein Elternknoten. Um der Struktur des Baumes zu folgen, müssen

[6] Man kann sich einen Baum als verknotete Schnüre vorstellen, der an seiner Wurzel gehalten herunter hängt. Greift man einen beliebigen Knoten heraus und fällt der Rest wieder herunter, so ändert sich an seiner Struktur nichts. Der Baum hat eine neue Wurzel bekommen.

wir im Weltbaum also mal auf-, und mal hinabgehen[7]. Sind zum Beispiel die räumlichen Beziehungen der Objekte durch Transformationen im Weltenbaum ausgedrückt, so müssen wir nun manchmal die Transformation invertieren. Dieses zunächst sehr umständlich anmutende Verfahren hat jedoch auch Vorteile: Wenn man die Darstellungswurzel als Fokus des Blickes betrachtet, so ist sie zum Beispiel zumeist ähnlich groß im Blickfeld. Das sichert uns eine ausreichende Genauigkeit der Darstellung, da Fehler bei den Transformationen sich nun erst in den weit über den Baum entfernten Knoten aufsummieren. Insbesondere ist die Quantisierung der Koordinaten immer an die Darstellungsgröße angepasst. Des weiteren ist die Auswahl des Abbrechpunktes der Darstellung plausibel: Kinder im Weltenbaum sind nahe, aber kleine Dinge, die irgendwann einfach zu klein werden, um sie zu sehen. Eltern im Weltenbaum sind große Dinge, die sich aber in zunehmender Entfernung befinden. Im Blickfeld sind sie daher zumeist auch kleine Dinge.

Beispiel 15 (Ergänzung zu Beispiel 9)**.** *Auf dem Tisch liegen noch staubige Bücher. Beginnen wir mit dem Tisch, dort sehen wir Bücher, den Staub sehen wir nur als graue Schicht, das Staubkorn sehen wir schon nicht mehr. Blicken wir dann aus dem Fenster, da ist die Stadt, der Himmel, einige große Sterne: Sie sind recht nah und hell. Ferne, kleine Sterne bilden wie Staub das Band der Milchstraße. Den einzelnen Stern sieht man hier nicht mehr.*

Wir sehen also, dass 'klein' und 'fern' fast den selben Effekt hat. So ist es gerechtfertigt, eine Darstellungswurzel zu wählen, und beides gleich zu behandeln. Der Weltenmotor verwendet ein Gemisch dieser Methoden: Zum einen nutzt er die Darstellungswurzel, um dem Raum praktisch auf das angesehene zu fokussieren, der Punkt höchster Genauigkeit liegt vor einem. Zum anderen durchsucht er diesen Baum zumeist nach den Weltbaumeltern zuerst, denn viele große Phänomene haben Folgen im kleinen: Die Sonne bestimmt, wohin der Schatten des Buches fällt. Zu seiner Darstellung ist also die vorhergehende Auswertung des Sonnenlichtes unabdingbar.

3.2 Unterteilungen

Ein Sonderfall von hierarchischen Strukturen sind fraktale Strukturen. Vor allem der Begriff der *Selbstähnlichkeit* legt nahe, dass es sich im gewissen Sinne um eine Hierarchie gleichartiger Dinge handelt. Wie solche Strukturen unserer Darstellung dienen sei in einem einfachen Beispiel aufgezeigt:

Beispiel 16. *Ein fraktaler Fluß Gegeben sei die Aufgabe, ein rechteckiges Stück platte Landschaft mit einem Fluß darin darzustellen. Folgende Ansätze bieten sich an:*

- *(*Zeichnung*) Wir betrachten den Fluß als eine zufällig schlängelnde Linie, die wir entlang seiner Länge in die Landschaft zeichnen.*
- *(*Abtastung*) Wir betrachten den Fluß als zufällige aber invertierbare Funktion eines Parameters, und stellen für jeden Punkt der Landschaft fest, ob er durch ihn verläuft.*

[7] Das Durchsuchen eine hierarchischen Baumes von einer von der ursprünglichen Wurzel aus verschiedenen ist zu realisieren, indem man den Elternknoten eben wie ein Kind behandelt. Beim rekursiven Traversieren merkt man sich nun jeweils den letzten besuchten Knoten. Diesen schließt man beim Durchlauf der Kinder aus, da er ja der Elternknoten der neuen Ansicht ist.

- *(*Rekursion*) Wir betrachten den Fluß als einen Bogenzug in der Landschaft, der sich bei genauerer Betrachtung aus mehreren verschiedenen Bögen zusammensetzt.*

Während die die Zeichnung nicht lokalisierbar ist, es muss die gesamte Landschaft in einer Auswertung gezeichnet werden, leidet die Abtastung am Mangel interessanter aber invertierbarer Funktionen. Die Rekursion kann also als verfeinernde Zeichnung in mehreren Ebenen verstanden werden. Ein kleines gerades ode einfach gebogenes Flußstück kann bei näherer Betrachtung durch mehrfache Kurven ersetzt werden.

3.3 Zufall im Griff

Wo wir so viel vom Zufall reden, ihm alles überlassen wollen, da sollten wir ihn auch uns gefügig machen. Wenn echter Zufall herrscht, so gäbe es keine Konsistenz: Die Frage 'Was sehe ich?' kann ich ja nicht in jedem Moment anders beantworten, zumal stillstehend, im stillen Zimmer sitzend. Unsere Welt ist neben all der Dynamik natürlich auch einer gestrengen Persistenz unterworfen. Wenn wir den Zufall haltbar machen wollen, können wir ihn konservieren, seine Entscheidungen abspeichern. Bewegung wir uns in der erschaffenen Welt, so wird immer mehr aus dem 'bekannten' geschöpft, nur wenn wir nie gesehene Orte betreten, darf der Zufall wieder richtig mitreden. Dieser Ansatz hat jedoch einen Makel: Der Bedarf an abgelegter Welt wächst beständig. Ließe sich der Zufall wiederholen, wäre viel gewonnen. Tatsächlich ist der synthetische Zufall des Computers, man ahnt es, so wie die Maschine, deterministisch. *Pseudo-Zufall.* Diese klassischerweise als Nachteil empfundene Trickserei kommt uns nun sehr zu Gute: Das scheinbar zufällige Universum ist deterministisch!

Üblicherweise erzeugt der Computer Zufallszahlen in Folge. Jede geforderte Zufallszahl ist tatsächlich der genaue Nachfolger der letzten Zahl in einer sehr verwirrenden, daher getrost als Zufall empfundenen Reihe. Wenn wir nun örtlich, hierarchisch determinierten 'Zufall' wollen, so ist natürlich eine einzige Reihe nicht grad sinnig, da wir ja, so wir jede benutze Zahlenfolge verbraucht haben, immer neue bekommen. Diesmal jedoch kommen wir ohne Speichern davon: Da, soll 'Zufall' einigermaßen ernst gemeint sein, irgendwann jede beliebige Zahl erhalten können, ist die Abfolge des künstlichen Zufalls so gebaut, alle erlaubten (d.h. für den Computer im gewünschten Format darstellbaren) Zahlen zu erhalten[8]. Tatsächlich bedeutet das, daß wir eine x-beliebige Zahl dahernehmen können, und sie ist garantiert in der Reihe der Zufallszahlen enthalten. Mehr noch, wir können den Computer nun zu dieser Zahl 'vorspulen' lassen, ganz so, als hätte *er selbst* sie uns zuletzt als 'Zufall' verkauft. Fragen wir ihn dann abermals nach dem Zufall, so bietet er uns die nächste Zahl der Folge an. Hoch deterministisch, aber sehen kann man das am Ergebnis nicht. Mit diesem Verfahren können wir also scheinbar zufällige Zahlenfolgen immer wieder exakt reproduzieren, ausgehend von einer einzigen Startzahl, der *Saat.* Jetzt können wir ohne Speichern alles zurückholen, wenn wir nur stets dieselbe Saat anbieten.

Für dieselben Dinge dieselbe Saat anzubieten scheint einfach; leider müssen wir auch für verschiedene Dinge verschiedene Saat anbieten, sollen sie nicht gleich werden. Hierzu müssen wir Orte, Dinge, Ereignisse eindeutig kennzeichnen, die doch selbst aus der Quelle des Zufalls stammen sollten. Wir wollen jedem Ding eine Identität geben, noch bevor es das Ding gibt. Denn erst mit dieser Identität können wir den Zufall anrufen, uns das Ding zu beschreiben. Dazu nutzen wir wieder die Hierarchie aus. Haben wir ein Ding, einen Hierarchieknoten, so

[8] Endlich mal kann man sich über die Endlichkeit aller Zahlen im Computer freuen!

können wir seine Identität nutzen, den Zufall zu säen, und ihn nach Existenz und Identitäten seiner untergeordneten Dinge, seiner *Kinder* zu fragen. Dazu schlage ich zwei mögliche Methoden vor: Die Pfad- sowie die Adressmethode.

Definition 17 (Pfadmethode)**.** Die *Pfadmethode* beschreibt ein Verfahren zur Gewinnung einer eindeutigen Saat unter Verwendung eines Hierarchie*pfades*, der mit Hilfe von *Schlüsselworten* eindeutig und mit Hilfe einer *Hashfunktion* zu einer einigermaßen eindeutigen Saat verwendet wird. Der Pfad für ein Objekt hat das Format

$$\textit{Schlüssel}_0\_i_0/\textit{Schlüssel}_1\_i_1/.../\textit{Schlüssel}_n\_i_n.$$

wobei i_j ein eindeutiger Enummerator der Kinder des übergeordneten Hierachieobjektes ist.

Beispiel 18. *Ein realistischer Pfad wäre*

$$\textit{Galaxis}\_1/\textit{Sektor}\_4/\textit{Sonnensystem}\_15/\textit{Planet}\_4/\textit{Stadt}\_12,$$

beliebig fortzusetzen.

Ein Codebeispiel zur Pfadmethode findet sich in Abschnitt 8.5. Nun ist die Erzeugung einer Saat mittels einer Hashfunktion keineswegs sicher, jede Hashkollision bedeutet Uneindeutigkeit. Allerdings ist nicht jede Uneindeutigkeit eine Gefahr: Da der Zufall ja verschiedenen Interpretationen unterliegt, ist eine Kollision unter Umständen garnicht auffällig, so wird das Bestücken einer Stadt mit Bürgern sicher ein derartig anderes Fazit aus den Zufallsereignissen ziehen wie das bestimmen von Lufttemperaturen einer Atmosphäre, das bezweifelt werden darf, ob ein Beobachter diesen Zusammenhang je bemerkt[9]. Betrachten wir nun das andere Verfahren, daß diesen Makel nicht trägt:

Definition 19 (Adressmethode)**.** Die *Adressmethode* erzeugt eine eindeutige Saat unter Verwendung von endlichen *Adressräumen* bekannter Größe, die zu einem gemeinsamen Raum gekreuzt werden. Die Saat, definiert als *Adresse*, ist gegeben durch

$$|A_n|...(|A_2|\textit{Adresse}_1 + \textit{Adresse}_2) + ...) + \textit{Adresse}_n = \sum_{j=1}^{n} \textit{Adresse}_j \prod_{i=j+1}^{n} |A_i|$$

wobei $A_1, ..., A_n$ die Adressräume der jeweiligen Adressen $\textit{Adresse}_j \in A_j$ beschreiben, $|A_j|$ ihre Größen.

Beispiel 20. *Wir wählen einfachheithalber als Adressraumgrößen stets 16 für die Räume Galaxien, Sektoren, Sonnensysteme, Planeten, Städte. Die zu Beispiel 18 äquivalente Adresse wäre dann:*

$$1 \cdot 16^4 + 4 \cdot 16^3 + 15 \cdot 16^2 + 4 \cdot 16 + 12 = 85.836$$

Das entstehende Kreuzprodukt der Adressräume sorgt für eine gesicherte Eindeutigkeit. Der Nachteil sind jedoch die endlichen, da konstant festgelegten Adressräume. Ihr Platz ist sehr begrenzt; wird der Zufall mit 'gebräuchlichen' 32- oder 64-bittigen Verfahren erzeugt, so erhält man im letzteren Fall gerade mal die Möglichkeit, 8 8-bittige Räume zu falten, eine

[9] Ein Gesetz, das Lufttemperatur mit einer bestimmten Verteilung der Bürger in der Stadt in Zusammenhang setzt, ist allerdings als ein skurriles Phänomen durchaus liebenswert.

Grenze die spätestens bei Straßenzügen und Galaxiehaufen eng wird. Man muss zwischen Hierarchie- und Adressgröße abwägen. Adressgrößen sind jedoch oftmals an Häufigkeiten geknüpft, die als normalverteilt anzunehmen sind. Unendliche Resultate sind die Möglichkeit, so dass ein kleiner Adressraum schnell eng wird. Eine weitere Schwäche ist die Adressnähe: Benachbarte Dinge haben ähnliche Adressen. Im Falle der Pfadmethode wurde dies durch die Hash-Eigenschaft der Hashfunktion zuverlässig verhindert. Die Nähe kann Probleme bereiten: Die von üblichen Pseudozufallsgeneratoren erzeugten Folgen sind danach konstruiert, möglichst 'unabhängige' aufeinanderfolgende Werte zu erzeugen. Die Unabhängigkeit der Nachfolger zweier ähnlicher Werte ist jedoch keineswegs garantiert. Ein Zufallsgenerator für die Adressmethode müßte also auch auf dieses Kriterium hin getestet bzw. konstruiert werden.

Die Bezeichnung Adressraum fordert dazu auf, den Raum der möglichen Zufälle wie einen Speicher zu betrachten. Wenn wir kollisionsfrei arbeiten wollen, so steht uns der Gesamtadressraum als größter Raum unabhängig auswertbarer Zufälle zur Verfügung. Wenn wir also den 'Speicherbedarf' unserer Welt bennenen, so können wir den Arbeitsspeicher als einen Cache annehmen, der nur die berechneten Ergebnisse des gerade betrachteten Ausschnittes bereithält. Er enthält einen Ausschnitt aus unserem 'Zufallsspeicher': Dieser ist ein virtueller Speicher, der herrlicherweise durch keinerlei Hardwarespeicher begrenzt ist, er hat eine Größe von 4GB (32 Bit), 281474 GB (48 Bit)[10] bzw. 1.8×10^{10} GB (64 Bit). Wow!

3.4 Abhängigkeiten

Beim Verteilen vom Eigenschaften wird man ziemlich schnell auf voneinander abhängige Werte stoßen. So hat ein Planet mitnichten eine rein willkürliche Temperatur, sie ist viel eher von dem Abstand zur Sonne, der Zusammensetzung seiner Atmosphäre, und vielen Details, die wir wieder mit Willkür behandeln, abhängig. Während hier der Fall klar ist, sind z.B. gesellschaftliche Zusammenhänge keineswegs frei von *Schleifen*: Eine prosperierende Stadt verfügt über eine gute Infrastruktur; eine gute Infrastruktur läßt die Stadt prosperieren. Diese zyklische Abhängigkeit, hier noch in einem einfachen paarweisen Fall geschildert, stellt uns vor ein Problem. Völlig wahlweise läßt sich einer der Werte willkürlich, der andere dann abhängig bestimmen. Da Schleifen in dynamischen Systemen zur Rückkoppelung führen, müssen wir eine geeignete explizite Lösung hierfür bieten. Hierfür bedienen wir uns aus der Klasse der Lösungen der linearen Differntialgleichungssysteme, worauf in Kapitel 5.1 genauer eingegangen wird. Um irgendwo Zufall einbringen zu können, kommen wir vermutlich nicht umhin, die Schleifen zu zerschlagen, und damit die Abhängigkeiten hierarchisch und eindeutig zu machen. Wir lösen die Schleifen bzw ganze Netze so auf, dass wir die Verbindungen mit dem meisten Zufallseinfluß zuerst kappen. Da wir mit Hierarchie einen gerichteten Graphen erhalten, ist es durchaus möglich, mehrere Parameter auf einen wirken zu lassen, ohne eine Schleife zu benötigen: Im Graphen bleibt ein Kreis, der jedoch keine Schleife ist. Die verlorengegangenen Beziehungen sind jetzt nicht so problematisch: Die meisten ehemals verbundenen Parameter werden immer noch stark von gemeinsamen Parametern beeinflußt, und sind damit stochastisch nach wie vor abhängig. Man mag sich nun fragen, wie wir mit den massig vielen unbekannt wirksamen Zusammenhängen umgehen, wo doch ein Hitler ausreicht,

[10]klassische Zufallsgeneratoren arbeiten üblicherweise mit dieser Zahl, der Platz auf 64 Bit ist für Zwischenergebnisse, der Überschuss über 32 Bit ermöglicht eine längere Zufallssequenz als das 32-bittige Ergebnis vermuten lässt.

um das Antlitz der Welt umzurühren? Hier halten wir uns an Asimov, dessen vorgeschlagene Wissenschaft der Psychohistorik (siehe [Fly91]) wir einfach als gegeben hinnehmen: Das Wirken eines jeden geht im statistischen Rauschen der Gesellschaft unter. Wenn doch mal jemand auffällig wird, so ist es nicht er, es ist der Zeitgeist. Keiner hat Schuld! So machen wir uns das Leben leicht. Der Weltenmotor lost einfach einen Fingerzeig aus: 'Da! Er war's!'. Und keiner hats gemerkt.

3.5 Glücksrad und Tortenschneiden

Wir wollen nun sehen, wie wir die berechneten Eigenschaften effizient an die abhängigen Strukturen applizieren können. Wenn man einige Häuser betrachtet, so ist die Frage ihrer Bewohnung zweifelsohne einigermaßen mit Klischees[11] in Griff zu bekommen: Solange alle Fenster heil sind, wohnen eben die üblichen Strukturen Familie, Paare oder Singles darin, ist das Gebäude verlassen, vermutlich niemand mehr. Wer jetzt interessiert ist, wieviele in der Straße, Stadt, Land, Erde,... wohnen, kommt um eine Volkszählung nicht herum, die bekanntermaßen oft nur mäßig genau ausgeht. Wenn wir uns jedoch erlauben, die Welt von 'oben' zu betrachten, oder zu gestalten, so können wir einen exakten Weg gehen - das *Glücksrad*-Prinzip. Nehmen wir an, wir wüssten um die exakte Anzahl der Weltbevölkerung. Nun verwenden wir eine statistische Verteilung, die die reale bekannte Bevölkerungsdichte approximiert - in einer zunächst groben Struktur. Nehmen wir zunächst die Kontinente. Wir teilen nun unser Glücksrad in entsprechend große Sektoren ein, einer für jeden Kontinent, die die Warscheinlichkeitsdichte für ein Individuum in diesem Kontinent zu leben, repräsentieren. Nun lassen wir jedes der 6 Milliarden Menschlein einmal drehen, und es wird sodann auf seinem Kontinent ausgesetzt. Vor Ort findet es jedoch gleich das nächste Glücksrad vor: Es muss seine Nation erlosen, sodann seinen Heimatort usw. Mit dem selben Verfahren verteilen wir dann Bodenschätze, Wohlstand, die sieben Weltwunder usw.

Ein Vorteil des Verfahrens liegt jetzt auf der Hand: Die Volkszählung gestaltet sich einfach. Denn das globale Glücksrad ist hervorragend, ja sogar exakt informiert, in welchen Erdteil es wieviele Personen verschickt hat. Unsere Volkszählung ist also in jeder Hierarchie-Stufe exakt.

Nun wollen wir den Personen ihre Individualität zunächst einmal absprechen. Da sie nun ununterscheidbar, und zunächst allein durch ihre Anzahl repräsentierbar sind, kann man den Glücksradvorgang durch simples *Tortenschneiden* ersetzen. Anstatt für jeden zu drehen, bestimmt man nur, wieviele jeweils den Wert eines Attributes erhalten, bis die Torte in endlich viele Klassen zerteilt ist.

Wenn wir nun eine Stadt betrachten, und die Einwohner bestimmen wollen, so brauchen wir nun nicht mehr jeden Straßenzug, jedes Haus aufzusuchen. Andersherum: Wir schauen uns diese Galaxis an, wieviele Leute sie auf die Erde wirft, wieviel in dieses Land, Stadt usw. Allerdings müssen wir immernoch in jeder Hierarchiestufe die ganze Torte zerschneiden, obwohl wir uns doch vielleicht nur für ein Stück interessieren. Gerade wenn es viele Stücke gibt also müßig. Eine Alternative ist jene Torten, die sehr viele Stücke geben, schlampig zu behandeln: Wie nehmen ein Tortenstück für sich raus, ohne die anderen zu erzeugen. Wenn sie nicht alle gleich groß sind, was meist der Fall ist, so ist die Verteilung der Stückgrößen geeignet zu wählen, dass die Gesamtgröße der Torte hinreichend vernünftig approximiert

[11] Der Mathematiker sagt *Statistik* dazu.

wird. Wie gewonnen, so verlieren wir an dieser Stelle die exakte Volkszählung wieder. Dafür wird der Aufwand der Berechnung von den Tortengrößen unabhängig, nur noch die Tiefe der Hierarchie spielt eine Rolle.

4 Vielfalt und Singularität

Viele Dinge in unserer Welt treten in eine großen Fülle von Variationen auf. In dem wir ein Ding jedoch benennen, geben wir gleichzeitig vor, dass es auch etliche *Invaranten* hat, die es eben von anderen Dingen unterscheiden, und einzigartig machen, *klassifizieren.* Die Variablen, die *Varianten* ermöglichen, streuen oft stark, der Designer macht sich sogar eine Kunst daraus, möglichst extreme Werte zu erkunden. Die Invarianten dagegen werden in einem Ingenieursprozess, der Evolution oder gar durch schlichte Physik beschränkt, oftmals derartig, dass man von einer Singularität in ihrer Verteilung sprechen kann.

Beispiel 21. *Ein Anspitzer kann rot, grün, durchsichtig sein, seine Farben verteilen sich, getrieben durch den menschlichen Geschmack, weit im Raum aller möglichen. Seine Funktion, und damit unser Wille, ihm den Namen 'Anspitzer' zu gönnen, hängt jedoch von einer extremen invarianten Auslegung seiner Form- und Materialgestaltung ab. Sicher würde ein zufälliger Tropfstein-Anspitzer keinen Anstoß geben, aber die durch den Menschen induzierte turbokapitalistische Massenproduktion dieses Dingelchens prägt der Verteilungsdichte für Formen des gesamten Universums eine empfindliche Singularität auf: Kegelförmig eingeschnittener Block, Metallklinge.*

Wir erwarten also in einer Welt einerseits Vielfalt, da wo sie um ihren Selbstzweck existiert, aber auch extreme Einfalt, da wo die Vielfalt durch Energieminimierung, Selektion oder den Ingenieur berührt wird. Vielfalt zu erzeugen ist der Erfolg jedes entropischen Verfahrens, sei es eine Batikwanne, ein Würfel, oder eine Monte-Carlo-Methode, vertreten z.B. durch den linearen Kongruenzgenerator (Siehe [Knu97, Abschnitt 3.2.1]). Einfalt jedoch beinhaltet Optimierung, Perfektion und offenbart sich in Wiederholung. Im Folgenden wird erläutert, wie man aus Entropie eine Invarianz herausschält.

Wenn wir unsere Welt aus einer Formel schaffen, die weit weniger komplex als die reale Welt ist, so stoßen wir schnell auf einfache Struktur, oder eben keine erkennbare Struktur. Einfache Struktur sind wir jedoch nur wenig gewohnt.. so sieht unsere Welt nicht aus! Sie überwältigt uns eher mit leicht an- um- und durchstrukturiertem Chaos, kleinen Strukturen, großen Strukturen, dazwischen aber eine Menge unstrukt. Also lassen wir unsere simplen Formel erstmal die Unstruktur. Wie also bekommen wir die Erfahrung der Invarianz zurück? Zellulare Automaten könnten –ein leichtes– lokal Struktur pflanzen. Doch der Mensch giert nach mehr – Strukturen sind sinnvoll, gestaltet, funktional. Wenn wir die Funktionalität nicht als Resultat erwarten, und das wäre töricht –der Computer wäre dann bereits Ingenieur– so müssen wir ihm auf die Sprünge helfen, ihm Struktur vorgeben, Invarianz abspeichern. Aber bitte nicht mehr als nötig! Offensichtliche Notwendigkeiten geben wir ein, Varianz lassen wir dem Zufall. Was bleibt sind unoffensichtliche Notwendigkeiten. Freiwillige Varianzen, noch nicht erklärt, oder durch Synergie durchgesetzt, Strukturen, die zwar erkennbar, aber für eine reale Welt uns (noch) nicht zwingend erscheinen. Diese Schmankerl gewinnen wir nun durch *Deentropisierung.*

Definition 22 (Deentropisierung)**.** *Deentropisierung* bezeichnet den Vorgang, die Warscheinlichkeitsdichte einer Variablen auf Singularitäten zu reduzieren.

Der Sinn ist gut an einem Beispiel zu erläutern:

Beispiel 23. *Ist die notwendige Struktur eines Gebrauchsgegenstandes durch den Ingenieur gegeben, so folgt die übrige durch das Design, und zumeist werden einige davon beliebt. Vielleicht kann man auch von 'Marke' sprechen. Es gibt dann wenige Varianten des Dinges, und jeweils davon sehr viele Exemplare, teilweise weltweit verbreitet.*

Wir werden nun sehen, dass eine Deentropisierung durch simple Quantisierung zu erreichen ist. Wenn wir eine exponentiell verteilte Größe auf Ganzzahlen runden, so entsteht eine grobe Box-Annäherung der Exponential-Verteilung. Die erste Box ist je nach Ansatzpunkt des Rundens ein festgelegter Anteil an der Gesamtheit. Des weiteren gibt es unendlich viele weiterer Ereignisse mit sinkender Wahrscheinlichkeit. Jetzt bedarf es nur noch einer hinreichend krummen Abbildung, um diese quantisierten Ereignispunkte zu vielfältigen Ereignissen zu machen. Die Rationalität kann hier sogar wieder beliebig eingeblendet werden: Je 'glatter' die Zurordnung eines Parameters zur Zufallsgröße, um so mehr erhalten wir eine Annäherung an unsere angenommene Verteilung. So läßt sich wildes Design des Anspitzers genauso ermöglichen, wie die kontrollierte Verteilung seiner Durchmessertoleranz. Ein Codebeispiel findet sich in Abschnitt 8.6.

5 Generatorfunktionen

> Um aus dem Zufall Struktur zu erheben, bedarf es einiger Mühe. Wir verpacken diese Struktur in Funktionen, die sich vom Zufall beeinflussen lassen, ihn aber mehr als Inspiration verwenden.

Definition 24 (Gemischte Position)**.** Eine Ortsangabe, die durch Verwendung vorhandener Struktur bewegte Punkte in einer konstanten Art beschreibt nennen wir *(gemischte* Position.)

Irgendwo vom All aus gesehen ist die Bezeichnung '15 km nördlich Erfurt' ein sehr bewegter Punkt, für uns hier unten ist er hinreichend konstant.

Definition 25 (Parametergenerator)**.** Ein *Parametergenerator* ist eine Funktion, die einen reellen Parameter ($r \in \mathbb{R}$) abhängig von stets vorhandenen Größen herstellt. Typische Argumente sind Zeit, Raum (meist in Sinne einer gemischten Postition).

5.1 Wellenfunktionen

Unter der arroganten Annahme, dass die meisten quantitativen Sachverhalte sich bereits durch lineare Abhängigkeiten 'plausibel' beschreiben lassen (s. Beispiel) macht es Sinn, als Parametergeneratoren die Lösungen linearer dynamischer Systeme zu verwenden. Diese werde ich hier allgemein Wellenfunktionen nennen, obwohl sie auch 'unwellige' Anteile, nämlich exponentielles Wachstum, haben. Die Parameter werden zumeist Zeit (*temporal*), Raum (*spatial*) oder sonstige bereits errechnete Werte (*funktional*) sein. Zyklen sind in der realen Welt ubiquitär, am Auffälligsten zum Beispiel der Wechsel von Aufschwungs- und Regressionsphasen (also eine temporale Wellenfunktion). Von der anderen Seite her kann man mit der Fourieranalyse alles zu Wellen machen, und dann nach dem urhebenden linearen System fragen. Bis auf wenige Rückschläge scheint 'Wissen' zumindest in unserem

Zeithorizont nicht rückläufig zu sein; hier wäre also ein großer exponentieller Anteil zu finden, ebenfalls temporal.

Eine Interpretation des dynamischen Universums als zellularer Automat (eigentlich nur ein dynamisches System mit quantisierten spatialen Lokalitäten) legt nahe, dass auch lineare Dynamik im Raum zu finden ist. Daher folgende Begriffe:

Definition 26 (Spatiale, temporale Welle, Unabhängigkeit)**.** *Spatiale Welle* nennen wir eine Funktion die als Resultat eines räumlich lineardynamischen Zusammenhangs erscheint. *Spatial Unabhängig* nennen wir eine Funktion die als Resultat eines Vorganges ohne räumlichen Zusammenhang zu entstammen scheint. *Temporale Welle* nennen wir eine Funktion ,die Resultat eines lineardynamischen Vorganges in der Zeit ist. *Temporal unabhängig* nennen wir eine Funktion die bezüglich der Zeit unabhängig ist.

Entsprechend kann man auch z.B. von einer spatial unabhängigen temporalen Welle sprechen usw. Spatiale Unabhängigkeit ist auch Resultat stark unstetiger Zellularität, so haben Planeten durch ihre lokal zum leeren All sehr verschiedenen Bedingungen auch untereinander wenig Zusammenhang, zumal ihr Ort sich stets ändert. Doch auch wenn sie sich grad nahe sind, ist wenig Ähnlichkeit zu erwarten.

Wir wollen nun verschiedene Wellenfunktionen angeben, und ihre Einsatzzwecke abstecken.

Fourierreihe Eine periodische Funktion f lässt sich unter hinreichenden Bedingungen durch eine Reihe von Sinus- und Kosinusfunktionen darstellen, deren Frequenzen ganzzahlige Vielfache der Grundfrequenz $\omega = 2\pi/T$ sind:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cdot \cos(n\omega t) + b_n \cdot \sin(n\omega t))$$

Diese Darstellung erlaubt die Berechnung beliebiger Funktionen als Wellenfunktionen und wird andererseits sehr einfach, wenn nur wenige Frequenzen vertreten sind. In höheren Dimensionen entstehen jedoch mehrfach indizierte, daher sehr viele Koeffizienten

Perlins Rauschen

Simplexrauschen

Wavelet-Techniken

5.2 Ereignisfunktionen

Nachdem wir quantitative Dinge bestimmen können, interessieren nun die ”kleinen Sprünge”, die sich in den großen Strömungen niederschlagen. Wir suchen nach der kleinsten Veränderungseinheit, die nicht weiter zerlegt wird. Wir akzeptieren eine (kleine) plötzliche Veränderung der Welt.

Würden wir einfach eine Liste in der Zeit gleichverteilter Ereignisse erzeugen, um danach in ihr zu suchen, so wär diese endlich; und die Zeit somit auch. Ein Verfahren hierzu ist die *Schlange*: Eine Reihung von *Segementen*. Auf den Segmenten werden die Ereignisse gleichverteilt platziert. Wenn jedes Segment zumindest ein Ereignis enthält, so kann von jedem Ereignis sein Vor- und Nachfolger gefunden werden, insbesondere kann zu jedem

Zeitpunkt das vor-, nachlaufende Ereignis mit einer kurzen Suche im Segment bestimmt werden. Eine Gleichverteilung über der ganzen Schlange erreicht man so natürlich nicht, kann sie aber approximieren: Je größer die Segmente, umso mehr nähert sich die Verteilung auch auf der Schlange einer Gleichverteilung an. Ist eine beliebige Dichtefunktion auf der Schlange gegeben, zB als Wellenfunktion, so kann diese auch stufenweise auf den Segmenten approximiert werden. Wie klein die Segmente dann sinnvollerweise werden, hängt vom Minimum der Dichtefunktion ab: Ihre Approximation ist nur dann gegeben, wenn das eine Pflichtereignis in jedem Segment immernoch in vernünftigen Konfidenzgrenzen liegt.

Möchte man Ereignisse beobachten, die auch eine bestimmte Zeitdauer des 'Bestehens' haben, also eine Start- sowie Endzeitpunkt, so wäre eine simple Plazierung in einer Schlange nicht sehr sinnvoll: Die Ereignisse könnten nicht überlappen, da bereits zum Zeitpunkt des Nachfolgers das vorhergehende nicht mehr bekannt, und damit auch nicht bestehen kann. Hierzu bietet sich ein 'Tracker' an, eine mehrspurige Schlange. Innerhalb der *Spuren* überlappen die Ereignisse nicht. Im Idealfall mit unendlichen vielen Spuren infinitesimal kleiner Dichte kann man sich so zB einer Poissonverteilung beliebig nahe bringen. Weniger Spuren verbiegen die Verteilung durch Beschränkung der Anzahl gleichzeitig möglicher Ereignisse, ermöglichen jedoch eine Auswertung in endlicher Zeit.

6 Lokalisation globaler Bewegung

Wir betrachten ein Auto in einer Stadt. Meistens steht es auf irgendeinem Parkplatz, manchmal bewegt es sich zu einem anderen Ort. Wissen wir über sein Verhalten bescheit, so meinen wir damit meist, dass es uns keine Schwierigkeit macht, zu jedem Zeitpunkt mitzuteilen, *wo* sich der Wagen befindet. Wenn wir jedoch eine Straße betrachten, und fragen; 'Welche Fahrzeuge sind grad hier?', so ist scheinbar das Untersuchen des Verhaltens jedes in Frage kommenden Autos unabdingbar. Wenn man sich mit den 'in Frage kommenden' nicht schon im Vornherein stark einschränken will, so stellt sich die Frage nach einem statistischen Algorithmus, der eindeutige Objekte auch auf einem lokalen Ausschnitt des für sie zur Verfügung stehenden Raumes ausfindig machen kann, mit einer Komplexität, die nicht von der Gesamtzahl der im Raum bewegten Objekte abhängt, sondern besser von der Dichte, also Anzahl der Objekte im betrachteten lokalen Gebiet.

Ein Vorschlag: Es erscheint zu sinnvoll, die Fahrzeuge mittels eines Parameters eindeutig zu machen. Ein gleichverteilter, zählender Parameter ermöglicht, die Fahrzeuge gleichmäßig auf ein lineares Gebiet zu verteilen. Zu einem anderen Zeitpunkt werden wir eine andere Konstellation haben, für die wir allerdings folgende Eigenschaften fordern:

1. *Verwebung*: Die Fahrlinien der Fahrzeuge haben keine lokalen Gemeinsamkeiten, sie können sich kreuzen.

2. *Intervallierung*: Das Fahrzeug kann sich bewegt haben, oder stillgestanden sein.

3. *Eindeutigkeit*: In der Vereinigung aller möglichen Gebiete sind alle Fahrzeuge verschieden

4. *Persistenz*: In der Vereinigung aller möglichen Gebiete sind zu jedem Zeitpunkt dieselben Fahrzeuge enthalten

Die Forderung (1.) schließt zunächst vor allem jegliche Betrachtung als Phasenraum eines Differentialgleichungssystems aus, da sie der Flußeigenschaft direkt widerspricht. Die letzten zwei Forderungen sind bei einer Fahrzeug-zentrierten Betrachtung s.o. selbstverständlich zu realisieren, bei einem wahlfrei zugreifenden Verfahren jedoch mühsam zu erarbeiten.

7 Inszenierung

> Egal wer, oder was die Welt erschaffen hat, sie ist, wie wir sie kennen. Wenn wir uns weitere Welten vorstellen, erschaffen, fremde Welten, dann sind sie doch meist nah an der, die wir kennen. Die Vorstellungskraft ist endlich.

Wenn wir den Zufall so weite Hand bei der Gestaltung lassen, so bleibt uns die Bestimmung der Invarianten (wie in Abschnitt 4 propagiert), letztendlich Mitten und Streuungen von Verteilungen, auch wenn einige sehr schmal sind. Feste Konstante sind auch nichts anderes: Ihre Streuung ist nur minimal. Wo gewinnen wir diese Daten? Hier hilft uns die Statistik. Unsere Stichprobe ist allerdings sehr klein, wir kennen ja nur die eine Welt. Da sagt die Statistik, dass sie dann unzufrieden ist. Aber sie sagt auch, dass der Mittelwert der Stichprobe dennoch mit größter Wahrscheinlichkeit auch jener der Verteilung ist. Daraus folgt nichts anderes, als die skurrile Annahme dass alle Welten im Mittel wie unsere sind. Das ist meisterlich egozentrisch vom Menschen, aber durchaus gerechtfertigt. Damit können wir nun die Realität beobachten, unsere Werte schätzen und festlegen. Bleibt noch ein Problem: Wir kennen nicht einmal die gesamte unsere Welt. Einige Dinge konnten die Astronomen und Physiker doch tatsächlich hier, auf der Erde, von der Erde aus bestimmen! Fantastisch. Diese Parameter nehmen wir begierig mit. Aber vieles anderes ist uns nicht bekannt... so müssen wir eine Auswahl treffen. Wenn wir an Orte kommen, wo der Mensch nie war, so soll die Streuung die Welt fremd und bunt machen. So kommen wir zu Dingen, die wir nicht kennen. Wer weiß, wie weit wir uns bewegen müssen, um die Grenze der Vorstellungskraft zu überschreiten[12]... Wenn wir Parameter gemessen haben, so sind der Extrapolation keine Grenzen gesetzt. Jede Ausprägungen kann weiter getrieben werden, bis nur Karikaturen von dem bleiben, was abgebildet wurde[13].

Wollen wir unser Welten-Modell dazu benutzten, vorhersehbares zu *inszenieren* so stoßen wir schnell an die Grenzen: Je weniger Informationen wir benutzen mußten, um die Welt zu formen, um so weniger haben wir im Griff wie sie wirkt. Ein wenig ist es wie mit der Urknall-Hypothese: der Urzustand des Universums war überaus primitiv; seine Naturgesetze jedoch hinreichend komplex, um ein verwirrendes dynamisches System aufstehen zu lassen, dass uns am Ende den Menschen beschert. An einem Haufen Energie in einem Punkt läßt sich der Mensch aber mit unserem Verstand wohl kaum ablesen. Gut.

Wenn wir also eine Geschichte in unserer künstlichen Welt inszenieren wollen, so haben wir kaum eine Chance, dies durch drehen an den Grundparametern zu tun. Entweder fügen wir vorgegebenes, aber nicht durch das System vorbestimmtes in selbiges ein, machen eine *Einblendung*, oder wir gehen auf die *Jagd*: Wenn wir einen Sonnenuntergang inszenieren wollen, so gibt es viele Orte, an denen einer stattfindet. Wollen wir einen grünen Sonnenuntergang, so ist, sofern dieser in unserer Welt möglich ist, auch er zu finden. Es müssen nur hinreichende

[12] Vermutlich jedoch ist der Computer zu klein dafür.

[13] Extrapolation als Quelle von Karikatur ist auch Thema von [ges].

Freiheiten in Raum und Zeit gelassen werden. Die Suche nach so einem Ereignis erledigt der *Jäger*.

7.1 Einfache Jagd

Im häufigsten Fall wird die Suche nach einen Ort und eine Zeit fahnden, an dem bestimmte Eigenschaften vorherrschen. Der Suchraum wird vornehmlich durch Radien in Raum und Zeit abgesteckt werden, jedoch sind beliebige zulässige Mengen sicherlich denkbar. Um jedoch eine kontinuierliche Inszenierung zu erhalten, ist Zeitintervall von jetzt bis ein wenig in der Zukunft natürlich sinnig. Da gewohnheitsgemäß ein Ortswechsel eine oftmals monoton mit der Entfernung ansteigende Dauer hat, ist Nähe ebenso eine sinnvolle Forderung.

Die Suche kann eine reine Existenzsuche sein, oder aber ein Optimierungsverfahren: Wir suchen den 'ersten' Sonnenuntergang innerhalb unseres Suchraumes nach irgendeiner willkürlichen Reihenfolge, oder wir suchen den frühesten Sonnenuntergang, bzw. den rötesten, längsten oder was auch immer ihn besonders romantisch macht. Die Zielfunktion wird im allgemeinen sehr unangenehm sein, da ja mannigfaltige orts- und zeitunabhängige stochastische Kräfte wirken (Siehe Kapitel 5.1). Die Algorithmen werden speziell en detail auf konkrete Suchgegenstände optimiert werden; der rote Sonnenuntergang wird sicher von einem roten Stern begünstigt, aber auch unsere weiße Sonne ist in der Lage dazu. Auch komplexe Suchen sind denkbar, z.B. die Suche nach einem Pfad zwischen bekannten Orten, der bestimmte Suchgegenstände passiert usw.

Um die Suche möglichst flexibel formulieren zu können, ist es sinnvoll, alle Attribute der Dinge einheitlich 'im Griff' zu haben. Insbesondere ist es, um jederzeit Suchen formulieren zu können sinnvoll, die Objekte zur Laufzeit des Programmes beliebig untersuchen zu können[14]. Vor einem solchen sind alle Parameter eines Datentyps gleich, und die Suchanfrage kann so mit Respekt auf alle gewünschten Parameter formuliert werden, ohne dass der Suchalgorithmus explizit auf die Feinheiten des Objekttyps eingehen muss.

7.2 Statistik

Eine zunächst nahe scheinende Anwendung der Suche ist die *Statistik*: Wie viele Sonnenuntergänge gibt es in unserem Suchgebiet? Wie Rot sind sie im Schnitt? Die Resultate lassen sich auf jeden Fall mit einer Absuche durch den Jäger finden. Aber weitere Optimierung drängen sich auf: Das Tortenschneiden aus Kapitel 3.5 ist eine Ursache für viele mögliche Abkürzungen. Approximationen über gefaltete Verteilungen ermöglichen unter Umständen einen frühen Abbruch in der durchsuchten Hierarchie.

7.3 Einblendung

Wie bereits im Kapitel 7 angesprochen, kann eine Inszenierung durch Einblendung erfolgen, d.h. es werden Parameter nahtlos abweichend von den auf den Grundparametern basierenden Resultaten gegeben. Im Extremfall bedeutet dies nahtloses Einfügen von Inhalten, feiner ist jedoch die Manipulation der Parameter in mittleren Schichten. Die Einblendung ermöglicht

[14] Um den Programmieraufwand zu reduzieren, der sich ergäbe wenn alle Attribute entweder im Suchalgorithmus individuell behandelt werden oder aber sich selbst jeweils geeignet präsentieren müßten, macht es Sinn, mit *Laufzeittypeninformation* zu arbeiten, ein Programmierkonstrukt, dass die Analyse von Datenstrukturen zur Laufzeit eines Programmes ermöglicht.

auch Interaktion der Welt mit dem Nutzer. Da es kaum möglich ist, die Grundparameter dahingehend zu manipulieren, werden wir versuchen, so 'hohe' Parameter wie möglich zu finden, die eine angemessene Reaktion der Welt auf die Benutzerinteraktion ermöglichen. Hier sind zwei verschiedene Ansätze möglich: Begrenzte und unbegrenzte Reaktion.

Definition 27. Die *begrenzte Reaktion* blendet Folgen einer Handlung ein, die jedoch räumlich und zeitlich begrenzt bleiben. Es stellt sich wieder ein Weltzustand rein auf den Grundparametern basierend ein.

Definition 28. Die *unbegrenzte Reaktion* simuliert die Folgen einer Handlung als Abweichung zum eingriffslosen Weltenfortlauf. Insbesondere ist sie räumlich und zeitlich unbegrenzt, durch Simulation kann sich die Abweichung mit der Zeit beliebig ausdehnen.

Man erkennt schnell, dass die leistungsfähigere zweite Variante radikale Folgen hat: Mit fortschreitender Zeit und weiterer Interaktion fällt ein zunehmender Teil des Universums in die Hand der echten Simulation, während der Job für den Weltenmotor immer weniger wird.

Im Fall der Definition 27 geht man am Besten von einem *stabilen* Universum aus, nach der *Störung* durch den Eingriff bewegt es sich wieder auf den 'natürlichen' Verlauf zurück. So ist auch ein 'vertuschendes' Universum denkbar, wo eine Aktion zunächst eine sich ausbreitende Reaktion verursacht, die dann aber (am besten unter Unaufmerksamkeit) wieder abebbt und verschwindet; z.B. eine abgehackte gaußsche Glockenkurve[15]. Nach endlicher Zeit haben wir uns den Folgen des Eingriffes entledigt.

Für eine authentische Erfahrung der Welt ist vermutlich ein gesunder Mittelweg zwischen diesen Verfahren erforderlich.

Beispiel 29 (Frontier)**.** *Das Computerspiel Elite II - Frontier (siehe [BB93]) bietet ein gemischtes Universum: In einer kleinen Datei von <1MB Größe befindet sich sowohl Programm als auch Inhalt. Das Universum ist jedoch scheinbar unendlich groß, über tausend Sternensysteme sind 'bewohnt'. Interessanterweise existiert unser Sonnensystem als exakte Wiedergabe in dem Spiel, inclusive einiger (weniger) Städte auf der Erde, die sich durch ihre Baudenkmäler identifizieren. Das Handeln des Spielers hat für das Universum kurzfristige Folgen: So ist innerhalb eines Sternensystems dynamische Bewegung von Raumschiffen zwischen den Planeten zu beobachten, durch Gefechte kann der Spieler auf sie einwirken. Folgen davon überleben jedoch nicht lange, Trümmer verlieren sich im All. Nur ein egozentrisches Häuflein Daten hat Bestand: Das Raumschiff des Spielers. Es folgt ihm durch die Sternensysteme, behält Ladung, Ausrüstung und Beschädigungen, der Bordcomputer speichert Aufträge für den Spieler, die dieser angenommen hat; wodurch der Eindruck erzeugt werden kann, der Auftraggeber sei irgendwo da draußen und warte auf ihn.*

8 Praxis: Schummeln und Betrügen

> Die Welt ist, was wir so sehen. Leider sehen wir schon immer eine. Wird uns eine andere angeboten, muss sie sich gleich an der einen messen. Das ist der lange Schatten auf der künstlichen Realität.

[15] Viele sinnvolle Funktionen auf kompaktem Träger sind als *Wavelets* bekannt.

Abbildung 1: Grafische Interpretation eines Lindenmayer-Systems

In diesem Kapitel soll zunächst eine Reihe von Generatoren vorgestellt werden, die aus Zahlen realistische Dinge machen, die die menschlichen Intuition für die Naturgesetze befriedigen.

Der Biologe Aristid Lindenmayer entwarf ein Kalkül, um eine Theorie biologischer Entwicklung aufzustellen. Ausgehend von der Beobachtung dass in einem wachsenden Organismus neue Gewebe aus sich teilenden Vorläuferzellen hervorgehen formulierte er eine Sprache, in der Strukturen durch wiederholte Ersetzung von Zeichenketten geschaffen werden. Diese sogenannten Lindenmyer-Systeme lassen sich anschaulich sehr gut an der Entwicklung von Pflanzen vorstellen, die man dementsprechend mit ihnen auch sehr gut beschreiben kann.

Beispiel 30. *Es sei ein Alphabet* $\{z, +, -, [,]\}$ *definiert. Die Ersetzungsregel* $\{z \rightarrow z[-z]z[+z]z\}$ *wird sukzessive auf das Axiom* z *angewendet, und liefert so*

$$z$$

$$z[-z]z[+z]z$$

$$z[-z]z[+z]z[-z[-z]z[+z]z]z[-z]z[+z]z[+z[-z]z[+z]z]z[-z]z[+z]z$$

usw.. Die Interpretation $\{z$*: Zweig,* $-$*:Rechtsknick,* $+$*: Linksknick sowie* $[,]$*: hierarchischer Teil*$\}$ *beschreibt eine pflanzenartige Form. Wird diese Figur mit kurzen Linienstückchen als Zweige gezeichnet, so ergibt sich Bild 30.*

Werfen wir einen Blick auf die 'echte Welt'. Da sind Systeme, verflochtene Beziehungen, Netzwerke. Wir wollen einige davon ansehen und sehen, wie ein Weltenmotor sie falsch, aber authentisch erscheinend nachbilden kann. Insbesondere werden wir Netze zerschneiden, wie bereits in Abschnitt 3.4 festgestellt.

8.1 Newton

Von einem ptolomäischen Weltmodell über Kepler und Newton zu Einstein: Jeder dieser Wissenschaftler zeigte die Fehler der Theorien seiner Vorgänger auf. Diese Fehler waren jedoch kein Versagen, sie waren vielmehr nur Abweichungen, die erst mit der Zeit meßbar und interessant wurden: Für die Frage nach dem Jahreszeiten war es egal, wer um wen kreist, Erde und Sonne. Um die Planeten mit dem Fernrohr zu erforschen war jedoch eine Berechnung ihrer Position sinnig, und Kepler konnte diese bieten. Doch erst Newton fragte nach dem Warum. Wäre nicht die Sonne unheimlich viel schwerer als alle anderen Planeten zusammen, hätte Kepler bereits versagt. So konnte Newton nachliefern, was man kaum sah. Einstein schaute noch genauer: Und siehe da, die Welt funktionierte doch ganz anders. Jeder dieser Fortschritte brachte eine ganz neue Ansicht der Welt hervor. Dennoch waren die jeweiligen Vorgängermodelle bereits für die meisten Probleme bis dato tauglich.

Für den Weltenmotor empfehle ich also Newton bis Kepler. Ich akzeptiere die Existenz Newtons Massenanziehung, ignoriere jedoch die kleinen Abweichung zum einfachen Kepler-Modell. Die Bewegungen der Gestirne läuft so sauber, explizit, hierarchisch ab:

$$\text{Sonne} \rightarrow \text{Planet} \rightarrow \text{Mond}.$$

Alles dreht sich um seinen Eltergestirn, die Massen der weiteren Monde wird jeweils vernachlässigt. Der Grund liegt auf der Hand: Bereits das *Dreikörperproblem* nach Newton ist nur noch numerisch, iterativ zu lösen. Mit Kepler hingegen können wir eine explizite Darstellung als hierarchische Bewegung erwarten. Das Netz ist zerschlagen, der Baum siegt. Dieses Modell ermöglicht die Auswertung der Planetenpositionen zu jeder Zeit, bedingt aber statische Sonnensysteme. Zunächst können wir uns auf solche beschränken, die Entstehung und das Vergehen selbiger ist auf derart langen Zeitskalen angesiedelt, so dass sie für übliche Weltenmotor-Anwendungen noch nicht benötigt werden.

8.2 Vereinfachte Wirkunsgeflechte

Um die Abhängigkeiten nach Abschnitt 3.4 aufzulösen, benutzen wir reduzierte *Wirkungsgeflechte*. Schon beim Aufstellen solcher Geflechte wird schnell klar, dass die Zusammenhänge nur unvollständig bekannt sind, und dennoch in beliebigem Detailreichtum ausgeführt werden können. Ein Beispiel für ein primitives Netz ist Abbildung 2. Durch Entfernen von Schleifen[16] und einiger Redundanzen[17] erhält man das Diagramm 3.

Wie werden nun die Wirkungen simuliert, die wir im Diagramm gerade entfernt haben? Dazu benutzen wir wieder die Lineare-Dynamik-Annahme des Systems. Die entfernten Schleifen entsprechen positiven und negativen Rückkoppelungen; deren Darstellung wir mit Wellenfunktionen (siehe Abschnitt 5.1) abdecken: Die negativen Rückkopplungen führen zu schwingenden Lösungen, die positiven zu exponentiellen, bei begrenzten Größen logistischen Lösungen (Frequenz f, Geeignete Konstante c)[18]:

$$\begin{array}{llll} \text{DGL} & \ddot{x}_s = -x_s & \dot{x}_e = x_e & \dot{x}_l = M - x_l \\ \text{Lösung} & x_s(t) = c\sin(ft) & x_e(t) = ce^{ft} & x_l(t) = \dfrac{M}{1+ce^{-fMt}} \end{array}$$

Die Begriffe 'erhöht' und 'senkt' werden meist multiplikativ umgesetzt. Die Werte werden dann in jedem Knoten aus den eingehenden Einflüssen ermittelt, als Produkt oder als Faktor für eine der dynamischen Lösungen.

[16] Eine entfernte Schleife ist z.B. $\{\text{Reichtum} \xrightarrow{erhöht} \text{Bevölkerungszahl} \xrightarrow{senkt} \text{Reichtum}\}$.

[17] Eine entfernte Redundanz $\{\text{Wissen} \xrightarrow{erhöht} \text{Haushöhe, Wissen} \xrightarrow{erhöht} \text{Bevölkerungsdichte} \xrightarrow{erhöht} \text{Haushöhe}\ \}$.

[18] Die Schreibweise $\dot{x} = x$ bedeutet nichts weiter als $x'(t) = x(t)$ für alle t und ist vor allem in der Physik gebräuchlich.

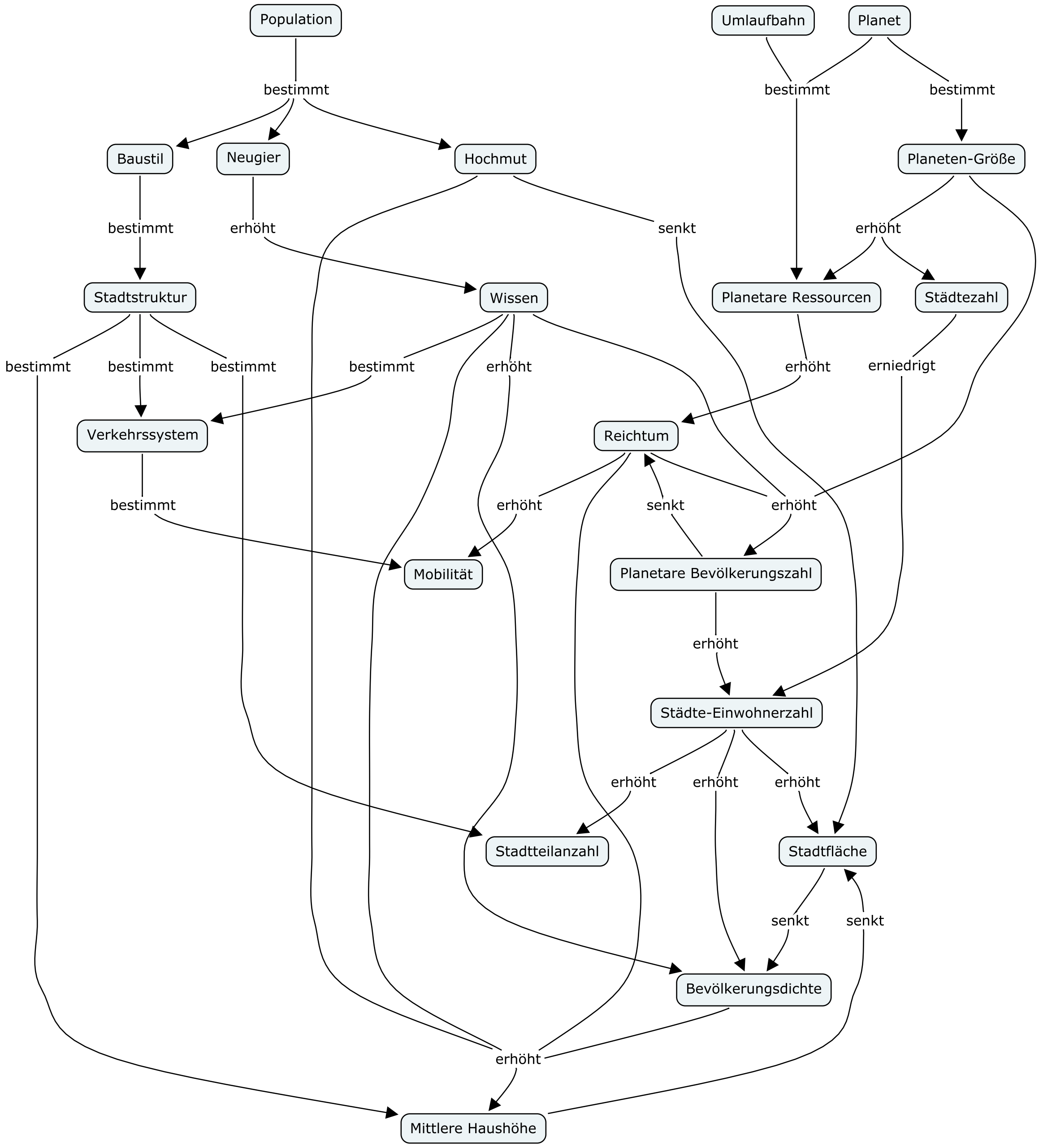

Abbildung 2: Behauptetes Wirkungsgeflecht für eine Stadt, die Beziehungen wurden oberhalb des Planetenbegriffes abgeschnitten, und gehen nur beispielhaft auf einige der für die Darstellung relevanten Details ein.

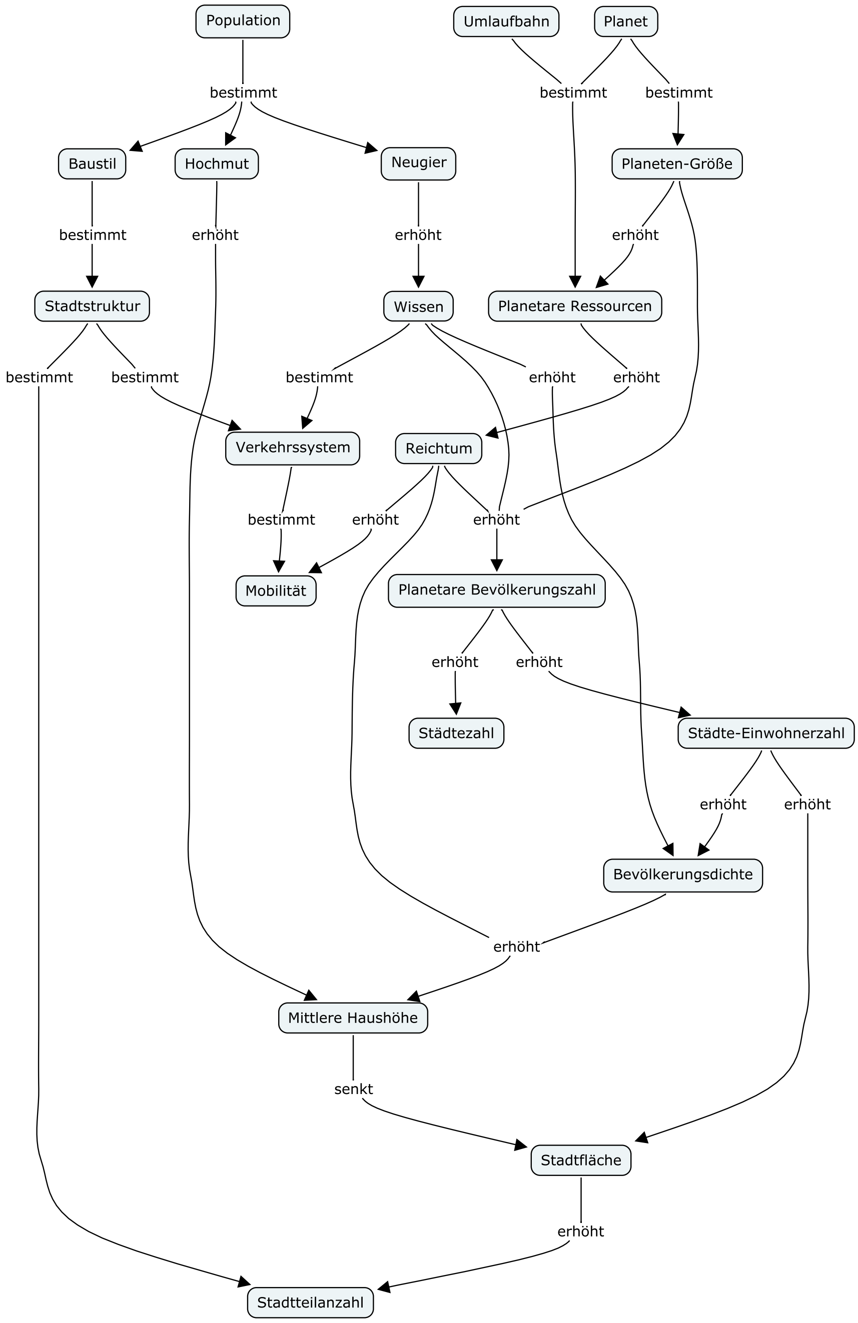

Abbildung 3: Vereinfachtes Wirkungsgeflecht für eine Stadt, Schleifen und ein Teil der Redundanzen entfernt.

8.3 Backen

Wenn man aus den Modellen konkrete Daten, zur Darstellung zumeist Gemoetrie erzeugt, so gibt es einen ganzen Haufen bekannter Kniffe und Verfahren um sie auf konkreter Hardware effizient darzustellen (vgl. auch Renderingverfahren). Die meisten davon werden ”unbewusst” bereits durch die Grafikbiliotheken erledigt, doch einige Dinge werden regelmässig dem Programmierer anvertraut. Üblicherweise gehört auch das *backen* dazu, das Herstellen möglichst weit verarbeiteter Daten. So wird man wenn man eine Kugel aus ihrem mathematischen Modell mittels eines geeigneten Algorithmus als Dreiecke darstellt, so wie es aktueller Hardware genehm ist, natürlich diese Darstellung nicht in jedem zu berechnenden Bild einer Sequenz vornehmen, sondern die resultierenden Dreiecke speichern, wenn möglich direkt im RAM der Grafikkarte. Diese sehr konkrete Verfahrensweise ist üblicherweise simpel und in den Grafikbibliotheken vorgesehen. Was bedeutet das für den Weltenmotor? Strukturen die man speichert, zeichnet man so als merhfach benutzbar und damit im gewissen Sinne statisch aus. Das ist natürlich ein Widerspruch zu einer dynamischen Welt, wie unsere eine ist. Jedoch: Stehen die Pyramiden nicht immernoch, und wer mag eine Veränderung an ihnen zu erkennen, während er davor steht? Das legt einen Ansatz zur Formalisierung des ”backens” nahe: Jedes Ding darf nur solange gespeichert werden, bis seine Änderung die Wahrnehmungsschwelle des Betrachters übersteigt, also beim Vollziehen als Sprung zu erkennen wäre. Doch wie lässt sich diese Schwelle festlegen, und wie kann man die Distanz der Veränderung bestimmen, ohne sie durchführen zu müssen? Hierzu kann man *Hinweise* sammeln, kleine Funktionen, die heuristisch bestimmt, Parameteränderungen ihren visuellen Wirkungen zuordnen.

Beispiel 31. *Ein Haus habe einen Parameter für sein Baudatum, danach verfällt es, ausgedrückt in vielen Details wie Moosbewuchs, Risse u.s.w. Das Moos wächst sehr langsam, solange kein Tornado es hinweg pustet, ist das Haus ziemlich statisch. Eine Hinweis-Funktion sollte nun für das Haus ein Zeitintervall geben können, in dem eine Darstellung gültig ist. Wenn man das Problem teilt, so ist diese Zeitspanne das Minimum der Teilspannen: Wächst das Moos oder reisst ein Riss merklich, so ist auch das Haus merklich verändert. Doch wenn man weit weg steht, so sieht man das Moos vielleicht garnicht. . .*

An diesem einfachen Beispiel sieht man bereits, dass sich eine derartige Optimierung durch das gesamte Modell zieht. Sie ist idealerweise sowohl von den Objekten, Blickwinkel- und Distanz, dem gewünschten Detailgrad etc. abhängig. Es besteht eine starke Analogie und gleichzeitig eine Erweiterung des hierarchischen Detailmodells der Welt: Raum und Zeitintervalle bestimmen die Berechnungsnotwendigkeiten. Während der Raum große Dinge sichtbar macht, sind es in der Zeit gerade die kurzen Intervalle, die eine Berechnung erforderlich machen. Am wenigsten Aufmerksamkeit verlangen sehr kleine, entfernte Dinge, die sich langsam entwickeln. Viel Aufmerksamkeit verlangen große, nahe Dinge, die sich rapide entwickeln.

Anhang

8.4 Programmschnipsel

Diese kleinen Fragmente in Java aus dem Sourcecode des Weltenmotors zeigen den aktuellen Stand, keineswegs einen Vorschlag wie ein Problem am Besten zu lösen ist.

8.5 Pfadzufallsgeneratoren

Der Zufallsgenerator im Anhängen einer Hierarchieebene (der Pfadgenerator nach Definition 17):

```
long appendSeed(String s){
    long hc1=hash(s);
    //thanks to Donald E. Knuth
    return ((seed^hc1) * 0x5DEECE66DL + 0xBL);
}
```

mit folgender Hashfunktion (`getBytes` gibt ein Array aller Zeichen des Strings als Bytes zurück, `rotateRight` rotiert die Bits einer Zahl um die im zweiten Argument gegebene Stellenanzahl):

```
final long m=0x9E3779B97F4A7C15L;

long hash(String s){
    long code=0;
    byte[] b=s.getBytes();
    for (int i=0; i<b.length;i++)
        code=Long.rotateRight(code,8)^(b[i]<<48);
    return code*m;
}
```

Der verwendete Multiplikator $\texttt{9E3779B97F4A7C15}_{16}$ stammt von [PE97].

Als schlechtes[19] Gegenbeispiel dazu die `hashCode`-Methode aus dem Java 1.5 SDK, `value` ist der Zeicheninhalt (Array von Characters); `offset` der Begin des Textes darin; `count` seine Länge:

```
public int hashCode() {
    int h = hash;
    if (h == 0) {
        int off = offset;
        char val[] = value;
        int len = count;
        for (int i = 0; i < len; i++) {
            h = 31*h + val[off++];
        }
        hash = h;
    }
    return h;
}
```

Furchtbar.

[19] Genügt keinen Bedingungen, die Kollisionen vermeiden. Ähnliche Strings, vor allem nur in den letzten Zeichen verschiedene, generieren nahe Hashcodes.

8.6 Exponentielle Deentropisierung

`getDiversity` erzeugt eine (z.Z: quadratnormalverteilte) qunatisierte Zufallsgröße:

```
public DiversityGod getDiversity(String seed, float diversity){
    double x=getQuadNormal(0,diversity); //Quadratnormalverteilte Zufallsgröße mit S
    double p=Distributions.quadnormald(diversity,x);
    int i=(int)x;
    return new DiversityGod (baseengine.appendSeed(seed+i),p);
}
```

`getQuadNormal(`μ`,`σ`)` erzeugt einen quadratnormalverteilten Zufallswert, `Distributions.quadnormald` wertet die Dichtefunktion selbiger aus. Der erzeugte `DiversityGod` enthält die neue Saat, und `p`: die Wahrscheinlichkeit, mit der ebendiese Saat entstand.

Literatur

[BB93] Braben, David und Ian Bell: *Elite II - Frontier.* 1993.

[Fly91] Flynn, Michael F.: *Einführung in die Psychohistorik.* In: *Die Foundation-Trilogie*, Seiten 827–904. Heyne, München, 1991.

[ges] *Gesichtsbieger.*

[Hof79] Hofstadter, Douglas: *Gödel, Escher, Bach - Ein endlos geflochtenes Band.* 1979.

[Knu97] Knuth, Donald E.: *The art of computer programming, volume 2 (3rd ed.): seminumerical algorithms.* Addison-Wesley Longman Publishing, 1997.

[PE97] Preiss, Bruno R. und P. Eng: *Data Structures and Algorithms with Object-Oriented Design Patterns in C++.* `http://www.brpreiss.com/books/opus4/html/page214.html` (19.09.06), 1997.