

# Eine zellbiologisch inspirierte Sichtweise auf selbst-organisierende und emergente Systeme

## Die Experimentierumgebung SelOnEx

Gerhard Fuchs und Stefanie Mika

Universität Erlangen-Nürnberg  
Lehrstuhl für Informatik 7 (Rechnernetze und Kommunikationssysteme)  
Martensstr. 3, 91058 Erlangen, Deutschland  
[gerhard.fuchs@informatik.uni-erlangen.de](mailto:gerhard.fuchs@informatik.uni-erlangen.de)  
[stefanie.u.mika@stud.informatik.uni-erlangen.de](mailto:stefanie.u.mika@stud.informatik.uni-erlangen.de)  
<http://www7.informatik.uni-erlangen.de>

**Kurzfassung.** Die Themen Selbstorganisation und Emergenz sind aktuelle Forschungsthemen in verschiedensten Fachgebieten. Daraus resultieren viele unterschiedliche Definitionen dieser beiden Phänomene. Im nachfolgenden Artikel beschreiben wir eine durch Studien im Bereich Zellbiologie inspirierte Sichtweise auf Systeme, die aus vielen miteinander kommunizierenden Teilsystemen bestehen. Das grundlegende Paradigma besteht darin, dass Informationen die in diesem System ausgetauscht werden ein Teilsystem bilden, das ein eigenständiges Verhalten aufweist. Wir unterscheiden zwischen dem Knotensystem (bestehend aus Aktivitätsträgern) und dem Signalsystem (bestehend aus Informationsträgern) und untersuchen die Interaktion dieser beiden Systeme, insbesondere deren gegenseitige Beeinflussung. Aus dem vorgestellten Konzept resultieren Arbeitshypothesen (Unterscheidung, Koordination, Dualismus und Rekursion), die uns als Grundlage für die Untersuchung der Mechanismen von Selbstorganisation und Emergenz dienen. Weiterhin stellen wir die Java-basierte Experimentierumgebung SelOnEx („Selbst-organisierende Experimente“) vor, die das eben dargestellte Prinzip in Form einer diskreten Simulation am Beispiel von Elementaren Zellulären Automaten in einer ersten Ausbaustufe umsetzt.

**Schlüsselwörter:** Selbstorganisation, Emergenz, Elementare Zelluläre Automaten, Experimentierumgebung SelOnEx

## 1 Einleitung

Die koordinierte Interaktion von vielen einzelnen Komponenten ist ein wesentlicher Forschungsaspekt in der Informatik. Sensornetze, Roboterschwärme, Peer to Peer Netze, Multiagentensysteme, das Internet sind nur einige Beispiele, bei denen sehr viele Einzelkomponenten als ein ganzes System fungieren sollen. Hieraus ergibt sich eine Komplexität, die neue Herausforderungen bei der Programmierung hervorbringt. Um diese zu bewältigen wird derzeit intensiv Forschung in

den Bereichen Selbstorganisation und Emergenz betrieben. Tom de Wolf u.a. z.B. unterscheidet in ihrem Artikel [1] diese beiden Konzepte strikt und sieht in deren Kombination eine Möglichkeit große Multiagentensysteme zu konstruieren. Hartmut Schmeck sieht in seinem Artikel über Organic Computing [2] ebenfalls erheblichen Forschungsbedarf in den Bereichen Selbstorganisation und Emergenz. Ein aus vielen Einzelteilen bestehendes Gesamtsystem soll mit möglichst wenigen Eingriffen von Außen eine Aufgabe lösen, aber kontrollierbar bleiben.

Um diese Mechanismen zu verstehen und sie technisch nutzbar zu machen wurden schon viele verschiedene qualitative und quantitative Definitionen von Selbstorganisation (z.B. [3], [4], [5]) und Emergenz (z.B. [6], [7], [8], [9]) vorgestellt, ein allgemeines formales Modell der Funktionsweisen dieser beiden Phänomene fehlt aber bisher noch.

Ein gern verwendetes Vorbild für die Konstruktion komplexer Systeme ist die Natur, die es z.B. beim Menschen geschafft hat eine riesige Anzahl von Zellen zu einem gesamten funktionierenden Organismus zu vereinen. Bei dem Studium zellbiologischer Vorgänge haben wir festgestellt, dass einzelne Zellen mittels Botenstoffen kommunizieren. Das besondere hierbei ist, dass die Botenstoffe ein eigenständiges Verhalten aufweisen, was uns zu einer Sichtweise auf Systeme gebracht hat, bei der ein Gesamtsystem in zwei miteinander interagierende Teilsysteme (Knotensystem, Signalsystem) aufgeteilt wird. Darauf aufbauend haben wir Arbeitshypothesen formuliert und untersuchen die allgemeinen Mechanismen dieser beiden Phänomene.

Der Artikel gliedert sich wie folgt: Abschnitt 2 führt einige Definitionen von Emergenz und Selbstorganisation auf, Abschnitt 3 gibt einen Überblick über das grundlegende Paradigma unserer Sichtweise. Die Experimentierumgebung SelOnEx (diskretes Simulationsframework) wird im Abschnitt 4 am Beispiel Elementarer Zellulärer Automaten vorgestellt. Abschnitt 5 enthält die aus den ersten Erkenntnissen resultierenden Arbeitshypothesen und stellt diese in Bezug zu einigen in Abschnitt 2 aufgeführten Artikeln. Abschnitt 6 fasst diesen Artikel zusammen und gibt einen Ausblick.

## 2 Arbeiten auf diesem Gebiet

### 2.1 Emergenz

Das Wort „Emergenz“ stammt vom Lateinischen Wort „emergere“, auftauchen, zum Vorschein kommen [10], ab. Es gibt viele Versuche Emergenz zu definieren (einzelne Fachgebiete haben eigene Definitionen), aber bisher noch kein eindeutiges Ergebnis. Allgemein lässt sich sagen, dass Emergenz ein Verhalten, ein Effekt, eine Eigenschaft eines Systems ist, das sich aus dem Verhalten seiner Komponenten nicht vorhersagen oder bestimmen lässt. Ein Beispiel für Emergenz ist die Tatsache, dass ein Goldatom an sich nicht metallisch glänzend ist, Gold, eine Ansammlung von Goldatomen, schon.

Joris Deguet u.a. finden in ihrem Artikel über verschiedene Emergenzdefinitionen einige Gemeinsamkeiten in den Definitionen [7]. So ist ein Ereignis, das

einfach erscheint, ein Kandidat für die Bezeichnung emergent. Alle Vorkommen geschehen in der Dynamik eines Systems, bei welchem mindestens zwei Ebenen oder Sprachen unterschieden werden. Doch sie bemerken auch, dass Emergenz nicht immer klar definiert wird.

Eric Bonabeau und Jean-Louis Desalles legen mehr Wert auf das Erkennen des Phänomens der Emergenz [11]. Für sie muss ein System aus einer höheren und einer niedrigeren Ebene bestehen, die jeweils kein Wissen über den Aufbau der anderen Schicht besitzen. Emergent sind nun die Phänomene, die die obere Schicht entdeckt.

Jochen Fromm sieht Emergenz in seinem Buch „The Emergence of Complexity“ [6] als Transfer von Komplexität. Er unterscheidet zwischen temporärer Emergenz und Emergenz, die zu einem dauerhaften Anstieg von Komplexität führt. Emergenz ist das Vorkommen von emergenten Eigenschaften und Strukturen auf einem höheren Level der Organisation oder Komplexität.

Christian Müller-Schloer und Bernhard Sick erläutern in ihrem Artikel [9] verschiedene Definitionen von Emergenz, diskutieren und kommentieren diese in Hinblick auf eine einheitliche Definition von Emergenz im Bereich intelligenter, technischer Systeme. In Hinblick auf eine technische Beschreibung von Emergenz postulieren sie eine Definition zwischen schwachem und starkem Emergentismus.

All diese Definitionen von Emergenz sind zwar beobachtbar, aber nicht messbar. Deshalb definieren Moez Mnif und Christian Müller-Schloer quantitative Emergenz als den in einem System auftretenden Entropieunterschied  $\Delta H$  [8]. Als notwendige Bedingungen für Emergenz nennen sie die Existenz einer großen Population von interagierenden Elementen ohne zentrale Kontrolle, deren Interaktionen nur auf lokalen Regeln basieren. Im makroskopischen Verhalten treten neue Eigenschaften auf, die auf der Mikro-Ebene nicht vorhanden sind. Das entstehende makroskopische Muster wird als Ordnung wahrgenommen und muss ohne äußere Beeinflussung, also selbst-organisiert, zustande kommen.

## 2.2 Selbstorganisation

Klaus Herrmann u.a. zeigen in ihrem Artikel [3] einige andere typische Eigenschaften natürlicher, selbst-organisierender Systeme auf. Diese sind z.B. das Vorhandensein einer großen Anzahl redundanter Komponenten, dezentrale Eigenkoordination anstelle eines zentralen Koordinators, das Zusammenspiel von positivem und negativem Feedback um Strukturen aufzubauen und zu erhalten, das Entstehen von komplexem, globalem Verhalten aus vielen lokalen Interaktionen und ein hoher Grad an Anpassungsfähigkeit. Des Weiteren geben sie eine formale Definition die eine Klassifizierung in selbst-organisierende und nicht selbst-organisierende Softwaresysteme ermöglicht und von der Art der Beschreibung des Systems abhängt. Ihre Definition eines selbst-organisierenden Softwaresystems lautet:

„A self-organizing software system is an adaptive software system that adapts to fit its environment by changing its structure. This structure at the global level arises solely from numerous interactions among the

lower-level components of the system. Moreover, the rules specifying the interactions among the components are executed using only local information, without reference to the global structure.“

Gero Mühl u.a. teilen Systeme ebenfalls in Klassen ein [4]. Sie unterscheiden zwischen „adaptive“  $C_{ada}$ , „self-manageable“  $C_{sma}$ , „self-managing“  $C_{smg}$  und „self-organizing“ Systemen  $C_{sog}$  und postulieren  $C_{ada} \supseteq C_{sma} \supseteq C_{smg} \supseteq C_{sog}$ . Für sie muss ein selbst-organisierendes System „self-managing“ und „structure-adaptive“ sein und außerdem eine dezentralisierte Kontrolle aufweisen.

Emre Çakar u.a. erfassen in Ihrem Artikel [5] Selbstorganisation quantitativ. Sie unterscheiden zwischen Selbstorganisation mit und ohne zentraler Kontrolle. Der von ihnen definierte Grad der Selbstorganisation, kann statisch oder dynamisch sein und setzt die Beeinflussbarkeit des Systems von Außen ins Verhältnis zu der von Innen. Im statisch Fall werden die interne, und externe Variabilität (Anzahl der Bits die nötig sind um alle Konfigurationen des Systems zu Beschreiben), im dynamischen Fall das Integral der Kontrollflüsse betrachtet.

### 3 Prinzip

#### 3.1 Inspiration durch Analyse der Zellbiologie

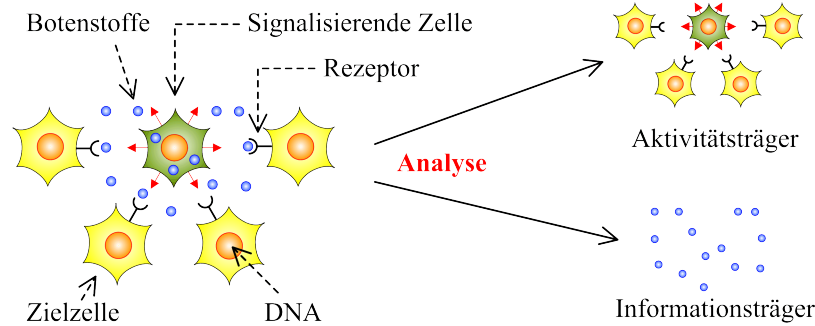
Zellbiologie ist ein faszinierendes Gebiet der Biologie. Im Laufe der Evolution hat es die Natur z.B. beim Menschen geschafft, ca. 100 Billionen Zellen selbst-organisierend zu koordinieren und zu einem funktionierenden Ganzen zu vereinen. Abbildung 1 zeigt unsere abstrakte Sichtweise der Funktionsweise der Zellen. Detailliertere Informationen zur Zellbiologie können z.B. [12] entnommen werden.

Soll ein aus mehreren Zellen bestehender Organismus eine Aufgabe lösen, sendet eine signalisierende Zelle Botenstoffe in das System, die von Zielzellen mit Hilfe von Rezeptoren empfangen werden. Die Botenstoffe übermitteln eine Information, sind also Informationsträger. Sie legen fest was getan werden soll und koordinieren somit die Zellen. Die Zellen entscheiden anhand der in ihren DNAs gespeicherten Informationen autonom wie sie mit der Information der Botenstoffe umgehen, wie sie darauf reagieren und tragen Ihren Teil zur Erfüllung der Aufgabe bei (Aktivitätsträger).

Es resultiert ein aus Zellen bestehendes ausführendes System, das für die Erledigung der Aufgaben zuständig ist und ein aus Botenstoffen bestehendes koordinierendes System, das angibt was getan werden soll. Auf molekularer Ebene bilden die Botenstoffe ebenfalls ein eigenes System, dass nicht statisch ist, sondern sich mit der Zeit verändert. Beispielsweise können einige Botenstoffe mit der Zeit zerfallen.

#### 3.2 Abstrahierung des Prinzips - Divide et impera

Das im Abschnitt 3.1 geschilderte Prinzip, wird in Abbildung 2 abstrahiert wiedergegeben. Die Zellen werden nachfolgend nur noch Knoten, Botenstoffen nur noch Signale genannt.



**Abb. 1.** Abstrakte Analyse der Zellbiologie. Ein Gesamtsystem bestehend aus Zellen und Botenstoffen lässt sich in Aktivitätsträger und Informationsträger aufteilen.

Ein Gesamtsystem, welches aus mehreren mittels Signale kommunizierenden Knoten besteht, kann bezüglich des Orts diskretisiert werden **(1)**. Ein Knoten kann sich an genau einer Position befinden, was in der Abbildung mit einem Kreis symbolisiert wird. Ein Signal kann sich über mehrere Positionen erstrecken, was mit einer Menge von gleichfarbigen Kreuzen dargestellt wird. Die Positionen im System lassen sich mit Koordinaten eindeutig adressieren. An einer Position können sich mehrere Knoten und Signale gleichzeitig befinden. Im nächsten Schritt **(2)** wird das Gesamtsystem in zwei Teilsysteme, das Knoten- und das Signalsystem zerlegt. Hieraus ergibt sich folgende Terminologie:

- Ein **System**  $S$  besteht aus (mehreren) **Knoten**  $k$ , die mittels **Signalen**  $s$  kommunizieren können.
- Eine Abstraktionsebene höher: Ein **Gesamtsystem**  $\mathcal{G}$  setzt sich aus einem **Knotensystem**  $\mathcal{K}$  und einem **Signalsystem**  $\mathcal{S}$  zusammen. Diese beiden Teilsysteme können sich gegenseitig beeinflussen.

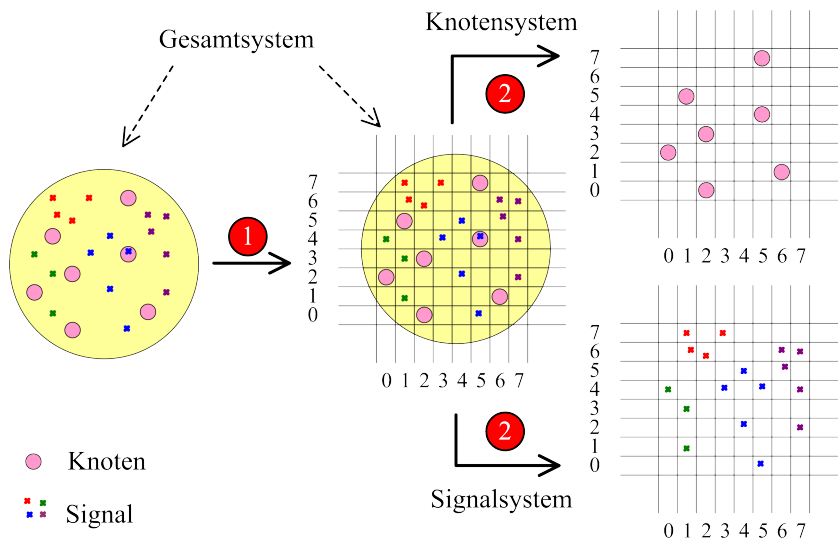
$$\mathcal{G} := (\mathcal{K}, \mathcal{S})$$

Um ein Verhalten beschreiben zu können muss der zeitliche Verlauf berücksichtigt werden. Dies geschieht durch das Einfügen einer diskreten Systemzeit. Der Zustand zu einem Zeitpunkt  $t$  wird wie folgt definiert:

$$\mathcal{G}_t := (\mathcal{K}_t, \mathcal{S}_t)$$

## 4 Die Experimentierumgebung SelOnEx

Die im Abschnitt 3.2 beschriebene Aufteilung ist sehr generisch gehalten. Die Experimentierumgebung SelOnEx setzt das Prinzip am Beispiel von zellulären Automaten um. Das Hauptaugenmerk liegt darauf, dass mit der Experimentierumgebung zu jedem diskreten Zeitschritt ein Abbild des Zustands des Gesamtsystems  $\mathcal{G}$ , also vom Knotensystem  $\mathcal{K}$  und Signalsystem  $\mathcal{S}$  gemacht und in einer



**Abb. 2.** Abstrakte Sichtweise zellbiologischer Abläufe. Zellen werden zu Knoten, Botenstoffe zu Signalen. Ein Gesamtsystem kann in diskrete Gebiete aufgeteilt werden, die über Koordinaten eindeutig adressierbar sind (1). Das Gesamtsystem wird im nächsten Schritt in ein Knotensystem und ein Signalsystem (2) aufgeteilt. Die beiden Systeme haben ein unterschiedliches Verhalten.

mysql-Datenbank gespeichert werden kann. Die Auswertung der Experimente erfolgt im Nachhinein. Derzeit haben wir eine Visualisierung mit SVG und eine Animation mit SVG + JavaScript implementiert.

#### 4.1 Formale Beschreibung

Die nachfolgenden Definitionen können nicht nur für Elementare Zelluläre Automaten verwendet werden. Andere Beispiele müssen noch implementiert und getestet werden. Weiterhin müssen die Grenzen der hier beschriebenen Definitionen ermittelt werden was zu anderen, abstrakteren Definitionen führen kann.

**Definition 1 (Signalsystem).** Das Signalsystem  $\mathcal{S}$  ist eine endliche Menge von Signalen  $s$ .

**Definition 2 (Signal).** Ein Signal  $s$  ist ein 2-Tupel der Form

$$s := (\text{typ}, \mathcal{P})$$

$\text{typ}$  Typ des Signals (beliebiger Text)

$\mathcal{P}$  Menge an Punkten  $(x, y)$  an denen das Signal wirkt

**Definition 3 (Knotensystem).** Ein Knotensystem  $\mathcal{K}$  ist eine endliche Menge von Knoten  $k$ .

**Definition 4 (Knoten).** Ein Knoten  $k$  ist ein 5-Tupel der Form

$$k := (\text{typ}, x, y, z, \mathcal{Z})$$

$\text{typ}$  Typ des Knotens (Text)  
 $x$   $x$ -Koordinate des Knotens im System  
 $y$   $y$ -Koordinate des Knotens im System  
 $z$  Aktueller Zustand des Knotens  
 $\mathcal{Z}$  Aktueller Zustandsautomat

**Definition 5 (Zustandsautomat).** Ein Zustandsautomat  $\mathcal{Z}$  ist eine endliche Menge von Transaktionen  $T$ .

**Definition 6 (Transaktion).** Eine Transaktion  $T$  ist ein 5-Tupel der Form

$$T := (z, z', \text{sig}^+, \text{sig}^-, \text{akt})$$

geschrieben als:

$$z \xrightarrow[\text{akt}]{\text{sig}^+/\text{sig}^-} z'$$

$z$  Ausgangszustand (beliebiger Text)  
 $z'$  Endzustand (beliebiger Text)  
 $\text{sig}^+$  Menge an Signalen  $S$ , die sich alle an der Position des Knotens befinden müssen, damit die Transaktion stattfinden kann  
 $\text{sig}^-$  Menge an Signalen  $S$ , von denen eines an der Position des Knotens ausreicht um die mögliche Ausführung der Transaktion zu verhindern  
 $\text{akt}$  Geordnete Menge an Aktionen  $A$  die beim Übergang der Reihe nach ausgeführt werden

**Definition 7 (Aktion).** Eine Aktion  $A$  ist eine Funktion, die

- einen (oder mehrere) Knoten  $k_{\text{neu}}$  in das Knotensystem  $\mathcal{K}$  einfügen kann:

$$\mathcal{K} \rightarrow \mathcal{K} \cup k_{\text{neu}}$$

- einen Knoten  $k$  beliebig manipulieren kann:

$$k := (\text{typ}, x, y, z, \mathcal{Z}) \rightarrow k := (\text{typ}', x', y', z', \mathcal{Z}')$$

- ein (oder mehrere) Signale  $s_{\text{neu}}$  in das Signalsystem  $\mathcal{S}$  einfügen kann:

$$\mathcal{S} \rightarrow \mathcal{S} \cup s_{\text{neu}}$$

- eine beliebige Kombination der oben dargestellten Möglichkeiten bewirken kann.

## 4.2 Ablauf der Simulation

Listing 1.1 zeigt den Ablauf der Simulation, die in einer *while*-Schleife abläuft (1). Zu Beginn eines jeden Durchlaufs wird die Systemzeit erhöht (2). Es gilt, dass alle Knoten des Knotensystem zum Zeitpunkt  $time$  ( $\mathcal{K}_{time}$ ) für ihre Entscheidungen einen Schnappschuss (*snapshot*) des Signalsystems zum Zeitpunkt  $time$  ( $\mathcal{S}_{time}$ ) verwenden (3). Dieser dient als Grundlage für die Entscheidungen der Knoten im Knotensystem (4) und bleibt für diesen Zeitpunkt immer gleich. Die bei der Aktualisierung des Knotensystems erzeugten Signale (*update*) werden dem Signalsystem für dessen Aktualisierung als Parameter übergeben (5). Nach der Aktualisierung werden  $\mathcal{K}_{time}$  und  $\mathcal{S}_{time}$  in der Datenbank gespeichert (6,7). Der Endpunkt der Simulation ist erreicht, wenn alle Knoten des Knotensystems inaktiv sind. Dies wird in Zeile (8) überprüft.

Listing 1.1. Ablauf der Simulation

```

1 while (!stop) {
2     time++;
3     signale = signalSystem.snapshot();
4     update = knotenSystem.nextStep(signale);
5     signalSystem.nextStep(update);
6     db.signalToDB();
7     db.knotenToDB();
8     stop=knotenSystem.hasFinished();
9 }

```

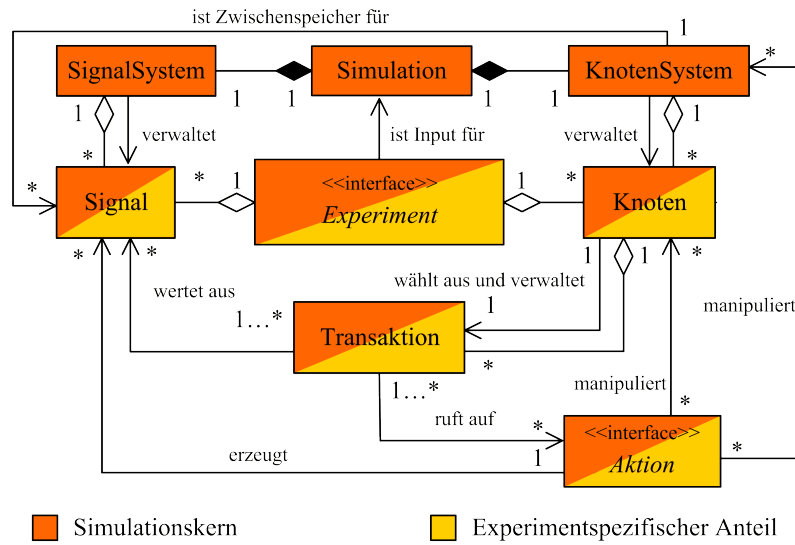
## 4.3 Klassendiagramm des Kerns der Implementierung

Abb. 3 beschreibt den Kern der Experimentierumgebung. Hierbei zeigen die unterschiedlichen Färbungen ob es sich um den Simulationskern (**Simulation**, **SignalSystem**, **KnotenSystem**) oder experimentspezifische Anteile (**Aktion**, **Experiment**, **Knoten**, **Signal**, **Transaktion**) handelt. Der Simulationskern ist für jedes Experiment gleich, der experimentspezifische Anteil variabel. D.h. die entsprechenden Interfaces (**Aktion** und **Experiment**) müssen realisiert, bzw. die Membervariablen und abstrakten Methoden von **Knoten**, **Signal**, **Transaktion** belegt, bzw. implementiert werden.

Nachfolgend steht der Zusammenhang der einzelnen Bausteine. Das zentrale Element ist die **Simulation**-Klasse, welche genau ein **SignalSystem** und genau ein **KnotenSystem** enthält. Das **Experiment** muss realisiert werden, besteht aus allen **Knoten** und **Signalen** der Ausgangskonfiguration (die Anzahl ist beliebig) und dient als Input für die **Simulation**. Die **Knoten** werden vom **KnotenSystem**, die **Signale** vom **SignalSystem** verwaltet.

Ein **Knoten** kann mehrere **Transaktionen** enthalten, welche Dynamik ermöglichen. Sie werten die **Signale** der Umgebung aus und rufen die **Aktionen** auf, welche in der aktuellen Situation ausgeführt werden müssen. Eine **Aktion** kann die **Knoten** und das **KnotenSystem** manipulieren. Werden **Signale** erzeugt, müssen diese im **KnotenSystem** zwischengespeichert werden.

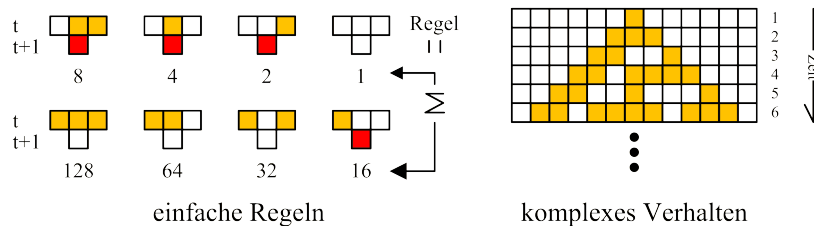




**Abb. 3.** Zusammenhänge der einzelnen Klassen. Der Simulationskern ist für alle Experimente gleich. Hierbei handelt es sich um Klassen, die immer unverändert bleiben. Beim experimentspezifischen Anteil werden Klassen und Interfaces des Simulationskerns modifiziert, bzw. implementiert. Das Zusammenspiel der Klassen kann den Assoziationen entnommen werden.

#### 4.4 Beispiel Elementare Zelluläre Automaten

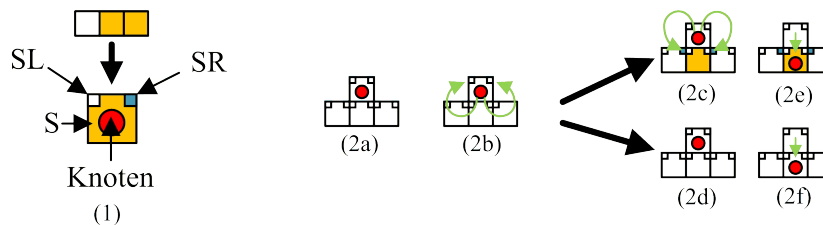
Stephen Wolfram beschreibt in seinem Buch „A new kind of science“ [13] eine Möglichkeit mit einfachen Regeln, komplexes Verhalten zu erzeugen und sieht darin die Basis für einen eigenen Forschungszweig. Als einfachstes Beispiel nennt er Elementare Zelluläre Automaten (Abb. 4). Ein Raum wird in Zellen aufge-



**Abb. 4.** Elementare Zelluläre Automaten. Mit einfachen Regeln wird ein komplexes Verhalten erreicht. Hier wird „Regel 30“ dargestellt.

teilt, die zwei Zustände (markiert, nicht markiert) einnehmen können. Die Zellen entscheiden an Hand ihres eigenen Zustandes und dem ihrer beiden Nachbarn

zum Zeitpunkt  $t$ , welchen Zustand sie im nächsten Zeitpunkt  $t + 1$  einnehmen. Betrachtet man das Gesamtsystem von außen, erkennt man beim zeitlichen Verlauf ein komplexes Verhalten. Für diesen Typ von Zellulären Automaten gibt es 256 Mögliche Verhalten. Wolfram benennt sie nach der Summe der der einzelnen Regeln. Bei der Umsetzung dieser Elementaren Zellulären Automaten mit SelOnEx wird die oben dargestellte „flache“ Beschreibung in das Knoten- und Signalsystem transformiert. Hierzu haben wir Knoten als Aktivitätsträger eingeführt (Abb. 5) und ihnen die Möglichkeit gegeben, Signale zu versenden. Knoten können sich nur an diskreten Positionen im Raum befinden **(1)**. An jeder Position befinden sich alle für die Entscheidung des Knotens nötigen Signale (wie ist der lokale Zustand und der meiner beiden Nachbarn, gekennzeichnet mit S, SL und SR). Der Ablauf eines Simulationsschritts wird in **(2a)** bis **(2f)** dargestellt. Entsprechend dem mit den Regeln festgelegten Verhalten teilt sich der Ablauf in **(2c,e)** - Zelle markiert - bzw. **(2d,f)** - Zelle nicht markiert - auf.



**Abb. 5.** Realisierung der Elementaren Zellulären Automaten mit SelOnEx. (1) Statischer Fall: Es werden 3 Signale und ein Knoten berücksichtigt. S = lokales Signal, SL = Signal beim linken Nachbarn, SR = Signal beim rechten Nachbarn. Ein ausgefülltes Quadrat bedeutet Signal vorhanden, ein leeres, das Gegenteil. (2) Übergang zum nächsten Zustand: (2a) Ausgangssituation; (2b) Analyse der lokal vorhandenen Signale; je nach Ergebnis wird die nachfolgende Position und die der Nachbarn markiert (2c) oder nicht (2d); Bewegung des Knotens um einen Schritt nach vorne (2e/2f).

## 5 Resultierende Arbeitshypothesen

Bei den nachfolgenden Vergleichen werden der Lesbarkeit halber nur die Erstautoren genannt.

**Unterscheidung Emergenz / Selbstorganisation:** Emergenz und Selbstorganisation sind als verschiedene, voneinander unabhängige Phänomene zu untersuchen. Hier stimmen wir mit der von Tom de Wolf in [1] erläuterten Ansicht überein.

**Existenz eines Koordinierenden Anteils (bei Selbstorganisation):** Beim Phänomen der Selbstorganisation schließen wir ein koordinierendes Element

nicht aus. Dieses kann wie bei den untersuchten Zellen erst erzeugt werden. Hier koordinieren sehr viele (Signale) einige wenige (Knoten). Wird das Signalsystem entfernt findet keine Selbstorganisation statt, d.h. es existiert eine zentrale Schwachstelle. In diesem Punkt unterscheidet sich unsere Meinung von denen von Klaus Herrmann in [3] und Gero Mühl in [4] beschriebenen. Emre Çakar unterscheidet in [5] zwischen „self-organisation with / without central control“. Hier ergänzen wir „im Knotensystem“, da das Signalsystem - zumindest in der Zellbiologie - immer koordiniert.

**Dualismus Signal / Knoten (bei Emergenz):** Für Emergenz müssen zwei Ebenen berücksichtigt werden. Bei unserer Sichtweise sind das das Knoten- und das Signalsystem. Wichtig hierbei ist, dass die Signale wie Knoten, und Knoten wie Signale betrachtet werden können. Dieser Punkt konnte in keinem der im Abschnitt 2 vorgestellten Arbeiten gefunden werden.

**Rekursive Definition von System:** Für die Untersuchung dieser beiden Phänomene ist eine gemeinsame Definition eines Systems  $\sigma$  wichtig. Unsere derzeitige Definition ist rekursiv und lautet in Anlehnung an die holistische Interpretation der Quantenphysik wie folgt: Ein aus mehreren Komponenten bestehendes System  $\sigma$  ist mehr ( $\Delta$ ), als nur die Summe der Einzelteile  $\mathcal{T}$ , kurz:  $\sigma := (\mathcal{T}, \Delta)$ .  $\mathcal{T}$  ist eine Menge von Systemen  $\sigma$  (wie definiert). Das fehlende Abbruchkriterium ist uns bewusst und muss sich zeigen. Diese Arbeitshypothese ist eine Abstrahierung des vorgestellten Prinzips ( $\sigma \hat{=} \mathcal{G}$ ,  $\mathcal{T} \hat{=} \mathcal{K}$ ,  $\Delta \hat{=} \mathcal{S}$ ). Die in Abschnitt 2 aufgeführten Artikel verwenden keine rekursive Definition von „System“.

## 6 Zusammenfassung und Ausblick

In diesem Artikel haben wir ausgehend von einigen Definitionen von Emergenz und Selbstorganisation unsere zellbiologisch inspirierte Sichtweise auf Systeme vorgestellt. Wir teilen ein aus mehreren, miteinander interagierenden Komponenten bestehendes System in ein Knotensystem (Aktivitätsträger) und ein Signalsystem (Informationsträger) auf. Beide Systeme haben ein eigenständiges Verhalten und können miteinander interagieren. Weiterhin haben wir die Experimentierumgebung SelOnEx in ihrer ersten Ausbaustufe vorgestellt und die Umsetzung des Prinzips am Beispiel Elementarer Zellulärer Automaten demonstriert. Das Verhalten von Systemen ist auf unterschiedliche Weise beschreibbar. Die bei SelOnEx gewählte Methode ist nur ein erster Ansatz, eine andere Beschreibung kann nötig werden.

Auf den gewonnenen Erkenntnissen haben wir unsere 4 Arbeitshypothesen Unterscheidung, Koordination, Dualismus und Rekursion erläutert und mit den vorgestellten Definitionen von Emergenz und Selbstorganisation verglichen. Die rekursive Definition ist in Anlehnung an die Quantenphysik gewählt, da hier durch die Einführung komplexer Zahlen die Beschreibungsmöglichkeiten der klassischen Physik stark erweitert worden sind. Was die Aufteilung in ein Knotensystem (vgl. Realteil) und ein Signalsystem (vgl. Imaginärteil) bringt, werden wir untersuchen. Dies wird auch zeigen in wie weit SelOnEx einen Mehrwert

gegenüber anderen mehrschichtigen Simulation wie z.B. NetLogo [14] und Repast [15] aufweist.

## Literatur

1. De Wolf, T., Holvoet, T.: Emergence versus self-organisation: Different concepts but promising when combined. In Brueckner, S., Di Marzo Serugendo, G., Karageorgos, A., Nagpal, R., eds.: *Engineering Self Organising Systems: Methodologies and Applications*. (May 2005) 1–15
2. Schmeck, H.: Organic computing - a new vision for distributed embedded systems. In: *ISORC '05: Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)*, Washington, DC, USA, IEEE Computer Society (2005) 201–203
3. Herrmann, K., Werner, M., Mühl, G.: A Methodology for Classifying Self-Organizing Software Systems. *International Transactions on Systems Science and Applications* **2**(1) (2006) 41–50
4. Mühl, G., Werner, M., Jaeger, M.A., Herrmann, K., Parzyjegla, H.: On the definitions of self-managing and self-organizing systems. In Braun, T., Carle, G., Stiller, B., eds.: *KiVS 2007 Workshop: Selbstorganisierende, Adaptive, Kontext-sensitive verteilte Systeme (SAKS 2007)*, Bern, Switzerland, VDE Verlag (March 2007) 291–301
5. Çakar, E., Mnif, M., Müller-Schloer, C., Richter, U., Schmeck, H.: Towards a quantitative notion of self-organisation. In: *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore (September 2007) 4222–4229
6. Fromm, J.: *The Emergence of Complexity*. kassel university press GmbH, Kassel (2004)
7. Deguet, J., Demazeau, Y., Magnin, L.: Elements about the emergence issue, a survey of emergence definitions. In: *Proceedings of the European Conference on Complex Systems (ECCS)*, Paris, France (November 2005)
8. Mnif, M., Müller-Schloer, C.: Quantitative emergence. In: *Proceedings of the IEEE Mountain Workshop on Adaptive and Learning Systems (SMCals/06)*, Logan, U.S.A. (July 2006)
9. Müller-Schloer, C., Sick, B.: Emergence in organic computing systems: Discussion of a controversial concept. In: *Keynotes ATC 06, 3rd International Conference on Autonomic and Trusted Computing*, Wuhan and Three Gorges, China (September 2006)
10. Brockhaus: *Brockhaus die Enzyklopädie*. 20 edn. Volume 6. F.A Brockhaus GmbH, Leipzig, Mannheim (1996)
11. Bonabeau, E., Dessalles, J.L.: Detection and emergence. *Intellectia* **2**(25) (1997)
12. Lodish, H., Berk, A., Zipursky, S.L., Matsudaira, P., Baltimore, D., Darnell, J.: *Molekulare Zellbiologie*. 4 edn. Spektrum Akademischer Verlag, Heidelberg, Berlin (2001)
13. Wolfram, S.: *A new kind of science*. Wolfram Media, Champaign, IL (2002)
14. Tisue, S., Wilensky, U.: *Netlogo: A simple environment for modeling complexity*. (2004)
15. North, M., Howe, T., Collier, N., Vos, R.: The repast symphony runtime system. In: *Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms*, Argonne, IL USA (October 2005)