Inhalt

[1. Introduction 2](#_Toc40276956)

[Motivation 2](#_Toc40276957)

[Beispiele für wachsende Komplexität 2](#_Toc40276958)

[Bekannte Ausfälle mit erheblicher Auswirkung – wegen Komplexität 2](#_Toc40276959)

[Natur als Inspiration 3](#_Toc40276960)

[Wie wird Komplexität in der Natur angegangen? 3](#_Toc40276961)

[Definition Organic Computing 3](#_Toc40276962)

[Aspekte 3](#_Toc40276963)

[Natur als Inspiration 3](#_Toc40276964)

[Systeme 3](#_Toc40276965)

[Terminologie 4](#_Toc40276966)

[OC Systeme bauen 4](#_Toc40276967)

[Paradigmenwechsel 4](#_Toc40276968)

[Grundsätzliche Methoden 4](#_Toc40276969)

[Applikationen 4](#_Toc40276970)

[Related Fields 4](#_Toc40276971)

[Aussicht 4](#_Toc40276972)

[History 4](#_Toc40276973)

[Zusammenfassung 4](#_Toc40276974)

[Fragen 4](#_Toc40276975)

[2. Self-organised Order 5](#_Toc40276976)

[Wassertempel auf Bali 5](#_Toc40276977)

[Selbst-organisierte Ordnung 6](#_Toc40276978)

[Ameisen 6](#_Toc40276979)

[Emergenz 7](#_Toc40276980)

[Beispiele 7](#_Toc40276981)

[Definitionen 7](#_Toc40276982)

[Quantifizierung von Emergenz 7](#_Toc40276983)

[Quantifizierung von Emergenz – ein verfeinerter Ansatz 7](#_Toc40276984)

[Fazit 7](#_Toc40276985)

[Fragen 7](#_Toc40276986)

# 1. Introduction

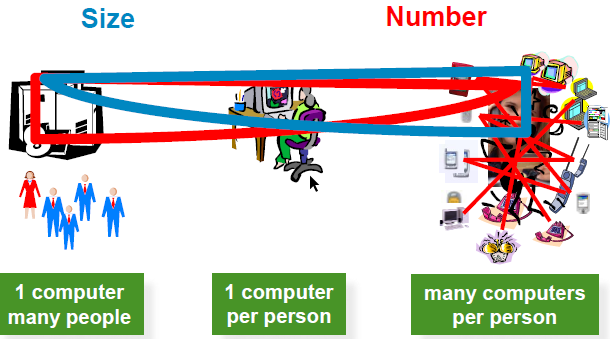
Ziele:

Studenten sollen folgendes können:

* Motivation für OC beschreiben
* Beispiele für steigende Komplexität in Informationssystemen geben
* Erklären wie natürliche Prozesse als Inspiration dienen
* Zusammenfassen welche Aspekte von OC gedeckt werden

## Motivation

Trends: Immer mehr Geräte, immer stärkere Rechner, Fehler durch gegenseitige Beeinflussung



Beobachtung: Hohe Komplexität führt zu Fehlern, verbundene Systeme schwerer zu managen. Unmöglich, alle Situationen vorherzusagen, da dynamische Umgebung und unbekannter Configspace

## Beispiele für wachsende Komplexität

* Moore’s law: Komplexität integrierter Schaltkreise mit minimalen Kosten verdoppelt sich alle 2 Jahre
* Hosts at the internet: `94 0, `07 500 Mio, `15 1 Milliarde durch IP erreichbare Hosts
* Request for Comments
* Glass‘ law: IT-Komplexität hängt indirekt mit Funktionalität zusammen – 25% funktionelles Wachstum erhöht Komplexität um 100%

## Bekannte Ausfälle mit erheblicher Auswirkung – wegen Komplexität

* Skype 2007: Sicherheitskritisches Update von Microsoft hatte Reboot vieler Computer weltweit zu Folge. Ergebnis: Distributed Denial of Service
* Google Mail 2009: Wartungsarbeiten in europäischem Rechenzentrum hatte Verantwortlichkeitsverteilung an umliegende Zentren zufolge. Ergebnis: Kaskadierende Ausfälle von Computern und Rechenzentren.
* EON 2006: Hochspannungsleitung wurde für vorbeifahrendes Schiff abgeschalten, Stromausfälle in ganz Europa

Traditionelle Konzepte, Prozesse und Methoden müssen abgeschafft werden. Die beobachteten Fehler, Ausfälle und der steigende Administrationsaufwand bestätigen das.

## Natur als Inspiration

Komplexität

* … ist ein gewöhnliches Naturphänomen
* Viele gut angepasste Lösungen beobachtbar
* Gemeinsamer Nenner: Komplexitätsminderung durch Zusammenarbeit autonomer Entities

Natürliche Systeme

* … sind selbst hoch komplexe Systeme
* … entwickelten sich über Milliarden von Jahre
* … zeigen self-\* properties
* … sind veränderbar, flexibel, robust gegen Störeinflüsse, belastbar, optimiert

Inspiration ist nicht Nachahmung, sondern Transfer grundlegender Mechanismen

### Wie wird Komplexität in der Natur angegangen?

* System besteht potenziell aus riesigen Sammlungen von Individuen
* Agieren selbst ohne zentrale Kontrolle
* Jeder hat self-\* Eigenschaften: self-learning, self-adaption, self-protection
* Entscheidungen werden basierend auf local knowledge getroffen
* Interagieren und kooperieren miteinander -> Makrolevel verhalten
* Manche Eigenschaft haben Emergenz Effekt
* Jedes Indivuum ist self-motivated und folgt eigenen Zielen

## Definition Organic Computing

* Paradigmenwechsel in Syseng
* Ziele: Designtime- zu Runtimeentscheidung, von Ingenieur- zu Systemverantwortlichkeit
* Viele kleine statt riesiger Strukturen
* Erlaubt autonome Entscheidungen
* Diese basieren auf lokaler Kenntnis, ohne Zugriff auf globale Welt
* Verteilte Entitys interagieren und kooperieren
* Feedback Mechanismus für jede Entity, um zu lernen
* Systeme bestehen aus autonomen Subsystemen: Diese haben Aktuatoren und Sensoren, interagieren miteinander und Umgebung, keine globale Kontrolle nötig
* Die resultierenden OC Systeme müssen: sich organisieren, adaptiv sein und optimales Verhalten lernen

## Aspekte

### Natur als Inspiration

* Selforga und adap meistert Komplexität -> Prinzipien transferieren

### Systeme

* OC ist Systemengineering, definiere das System und seine Orga, kläre die Komplexität

### Terminologie

* Spezifiziere Grundterme wie Selforga, Robustheit, Autonomie

### OC Systeme bauen

* Konzepte für individuelle, kontextbewusste Systeme mit organischen Eigenschaften bauen,
* Makrosicht auf Ökosysteme

### Paradigmenwechsel

* Von Designtime zu Runtime
* Resultierender OC Meta Design Prozess

### Grundsätzliche Methoden

* Spezialisierte Technologie wird für self\*-Properties gebraucht
* Am Wichtigsten: Autonomes lernen und Wissensmodellierung in Laufzeit
* Optimierungstechniken, Interaktionschemata, gemeinsame Einflüsse erkennen

### Applikationen

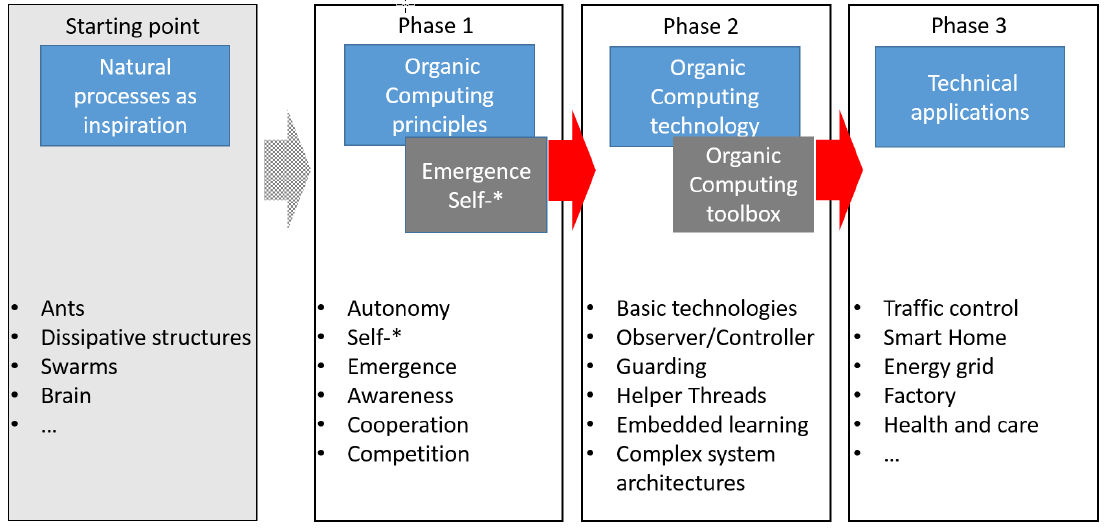
* Technische Systeme erfüllen bestimmten Zweck
* Verkehr, Datenkommunikation, Energienetz, distributed computing, smart cameras

### Related Fields

### Aussicht

* Trend mehr Komplexität
* Jetzige und künftige Herausforderungen für OC Forschung?

## History



## Zusammenfassung

* Trend zur allgegenwärtigen Vernetzung, Komplexität, Outages, Von der Natur lernen, worum geht’s bei OC, wichtige Aspekte

### Fragen

* Was ist die Motivation bei OC?
* Beispiele für steigende Komplexität in Informationssystemen?
* Warum können natürliche Prozesse als Inspiration hinsichtlich Komplexitätsbewältigung dienen?
* Welche Aspekte werden von OC abgedeckt?

# 2. Self-organised Order

## Wassertempel auf Bali

Bewässerungssystem für Reisanbau

* Hauptfaktoren sind Wasserzirkulation und Wechsel zwischen Feucht- und Trockenperioden
* Ziele sind gute PH-Werte, Mikroorganismen, Mineralstoffverteilung, Unkrautvernichtung, Schädlingsbekämpfung, Temperaturstabilisierung
* Problem: Permanente Bewässerung führt zu sehr hohem Wasserbedarf, deshalb muss das Wasser geteilt werden
* Die Farmer wollen ihre Ernte maximieren und müssen hierzu folgende 4 Dinge sequenziell aufeinander abstimmen: Erntephase, Pflanzenart/gruppe, Bewässerung, Trocknen

Wie stellt man optimale Sequenz fest?

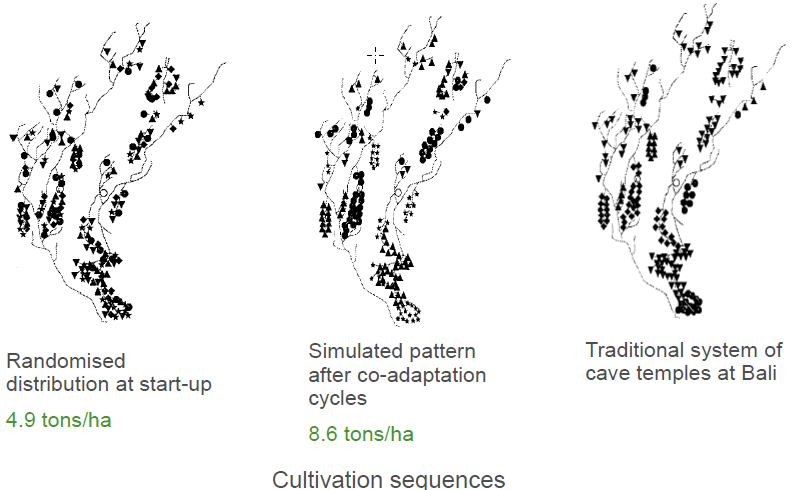
* Trial and error vs. Planning, globale oder lokale Koordination, Gesamtlösung für local brauchbar?

Hypothese:

* Koordinationsalgorithmus, synchron, lokal, wie der beste Nachbar, Co-Adaption

Hypothese verifizieren – Simulation:

* Die Ernte wird als Funktion der Kultivierungssequenz und der Umgebung modelliert.
* Zufällige Initialisierung als Start



Welche Schlüsse kann man ziehen?

* Netzwerk enthält lokale und self-motivated Famer (nodes, agents)
* Kooperation führt zu globalen „optimalen“ Mustern -> Emergenzeffekt
* Bottomup entwickelnde Netzwerke sind adaptiv und reagieren auf Umgebung (zb Regenarmut)
* Erfolg hängt von der Fähigkeit der Nodes lokale Info zu sammeln und darauf zu reagieren ab
* Keine zentrale Autorität nötig: Dezentrales System
* Keine externe Autorität nötig: Selbst-organisierendes System
* Co-adaption verlangt hohe Agentenpopulation
* Agenten entscheiden selbst aber koordinieren sich regelmäßig mit Nachbarn: Semi-Autonom
* System lernt und passt sich in Evolutionszirkeln an
* Zufällige Variationen bei Evolutionsschritten

### Selbst-organisierte Ordnung

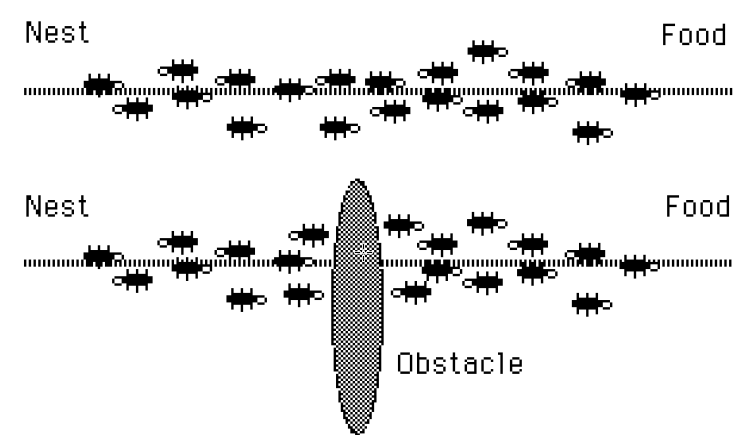
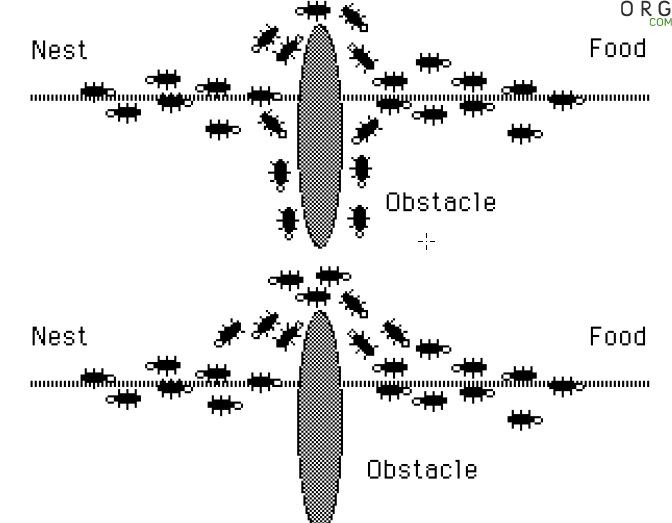
* Strukturiertes System mit globalem Muster entsteht
* Niemand ist dafür verantwortlich, hat globale Sicht oder plant

Mikrolevel: Farmer mit lokaler Sicht, die miteinander interagieren -> Aktionen beeinflussen Makrolevel

Makrolevel: Globale Sicht, Struktur, Muster, „optimale“ Lösung

## Ameisen

* Verteilen Arbeit: Soldaten, Bauarbeiter, Gärtner, Fortplanzung
* Arbeiten zusammen, um Brücken zu bauen
* Bauen Pilze an für Nahrung
* Finden den kürzesten Pfad zum Futter
* Priorisieren anhand Distanz und Erreichbarkeit
* Dynamische Adaption teilnehmender Ameisen, basierend auf Anzahl, gelagertes Futter, nahe Futterquellen und Rivalen

* Ameisen erkunden kurzen und langen Weg, aber finden trotz Kurzsichtigkeit den kurzen
* Folgen dem Weg mit stärkster Pheromonkonzentration, der kürzeste Pfad ist ihnen egal

## Emergenz

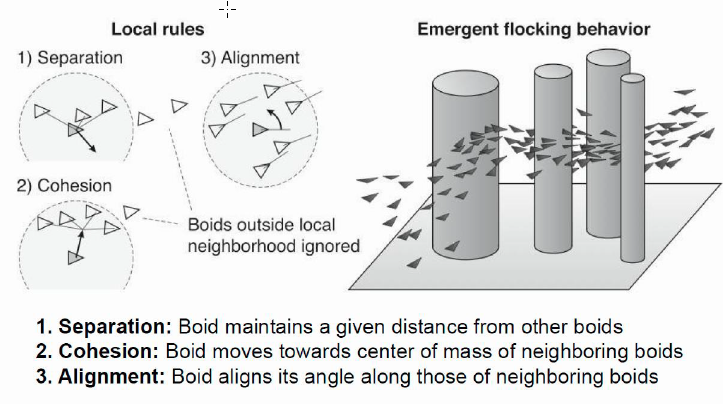
Etwas entsteht einfach – Kürzester Pfad, Schutz, Rollenverteilung, Gärten.

Diese Dinge können nicht anhand einzelner Entitys vorausgesagt werden. Sie entstehen durch Multiple, autonome, homogene, selbstmotivierte und selbstorganisierte Individuen, die miteinander kommunizieren. Keine zentrale Autorität

### Beispiele

**Schwärme**

Fisch- oder Vogelschwärme agieren als Organismus ohne Anführer. Fische wirken als riesiger Fisch, Vögel sparen Ressourcen.



**Termiten**

Bauen Strukturen mit verschiedenen Kanälen und Räumen ohne Einzelintelligenz oder Plan. Emergenz durch lokale und globale Interaktion. Um sicherzugehen, dass sich die Struktur an lokale Gegebenheiten anpasst, ist ein random Verhalten der Individuen von Nöten.

**Menschliches Gehirn**

* Mikro-Level: Neuron, sehr einfaches Verhalten, single Bit
* Makro-Level: Hirn, milliarden Neurone, unendlich anspruchsvolles, kompliziertes Verhalten
* Das Untersuchen einzelner Neuronen kann das Gesamtbild nicht vorhersagen!

**London Millenium Bridge**

Starke und stabile Brücke, die kurz nach Eröffnung wegen starker Schwingung geschlossen werden muss.

* Fußgänger haben sich unbewusst in Phase und Frequenz zueinander und zur Brücke synchronisiert und so für starke Schwingungen gesorgt

**Schweißroboter**

* Mehrere Roboter in Fabrik, je höher Spannung desto besser Schweißnaht
* Roboter schweißen simultan, Spannung fällt und führt zu Defekt
* Spannungsleitung wurde überwacht und Roboter dürfen nur schweißen, wenn Spannung hoch, das machte Problem noch schlimmer da alle synchronisierten
* Random delays eingebaut

**Festplatten**

* Simultane Datenbanksuchen führen zu vibrieren mehrerer Festplatten in einem Serverrig
* Vibrationen synchronisieren sich miteinander und führen zu Störungen

**Verkehrsstau**

* Individuelle Fahrer reagieren auf lokale Kleinigkeiten mit Delay, wirkt sich auf ganzen Verkehr aus

**Gehweg**

* Uni Michigan wollte Gehweg bauen aber wusste nicht wie, deshalb hat sie nur Gras gepflanzt und hat Schüler Wege selbst gehen lassen. Massenverhalten führte dann zu optimalem Pfad

**Sonstige**

* Social Networks, Milgram Experiment (Average number of hops bis Brief target erreicht, 6), Bacon-Number(shortest path Kevin Bacon <-> any actor: 3), Erdös Number (Distance in graph of paper coauthors, 4.7 und 7.3), 6 Hops zwischen Kebapbesitzer und Marlon Brandon.
* Average Number bei solchen Networks ist 6

## Definitionen

## Quantifizierung von Emergenz

## Quantifizierung von Emergenz – ein verfeinerter Ansatz

## Fazit

### Fragen