

Autonome Systeme und neue Paradigmen

Autonomic Computing,
Organic Computing und Verwandte

Selbst-Eigenschaften

Neue Strategien

- Selbstorganisation
- Neue Formen der Informationsverarbeitung
 - Autonomic Computing
 - Organic Computing
 - Proactive Computing
 - Recovery-Oriented Computing
- „Autonome Systeme“

Selbst-Eigenschaften

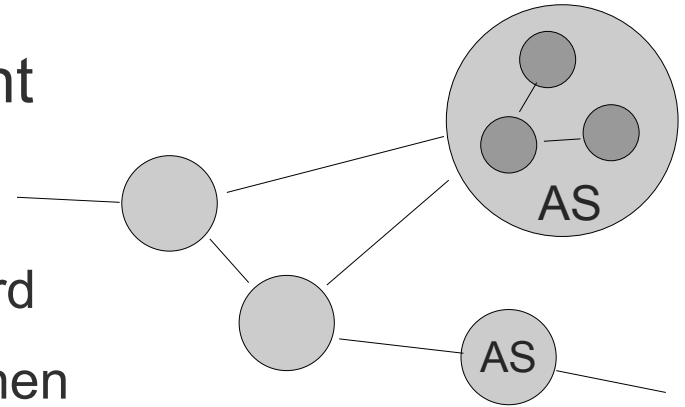
- Zahlreiche Begriffe, die ähnlichen Ursprung haben
 - Selbstorganisation
 - Selbstverwaltung
 - selbstheilend
 - selbsterklärend
 - ...

• Zusammengefasst: Selbst-Eigenschaften

Self-* properties
Self properties

Autonome Systeme

- Begriff aus der Vernetzung bekannt
- Autonomes System
 - Teilnetz, das als Einheit verwaltet wird
 - Kann aus weiteren Teilnetzen bestehen
 - Verwaltet von einer Firma oder einem Anbieter
 - Verbunden über Exterior Gateway Protocol mit anderen AS
- Internet = Verbindung vieler AS (autonome Systeme)



Diese Autonomie ist nicht gemeint.

Selbstorganisation

- Organische Systeme müssen fehlertolerant sein
 - „Es ist sehr wahrscheinlich, dass ein System mit der Komplexität lebender Organismen, das auf der Philosophie beruht, jeden Fehler fangen, erklären und korrigieren zu müssen, keine Millisekunde laufen würde.“
- Fehlertoleranz durch Anpassung
 - Tritt ein Problem auf, wird automatisch seine Schwere analysiert.
 - Wenn möglich, ignorieren und weiterarbeiten
 - Ist der Fehler von Bedeutung, umgehen und zu beheben versuchen
 - Fortsetzen der Verarbeitung, bis die Gesamtfunktion durch die Änderungen nicht mehr gewährleistet werden kann.



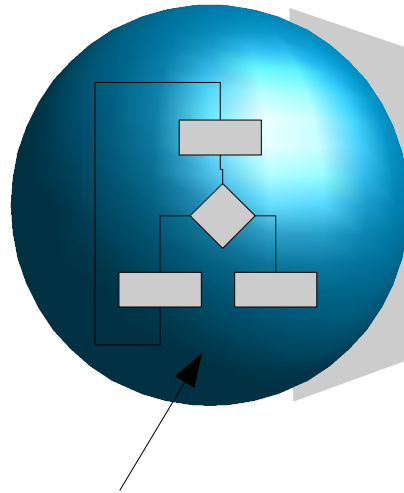
Zitate und Ideen von
John von Neumann

Selbstorganisation

- Reorganisation
 - Natürliche Organismen müssen sich angesichts der auftretenden Fehler flexibel zeigen
 - Umstrukturierung kann erforderlich sein
 - Auch eine Selbst-Beobachtung / Selbstkontrolle ist notwendig

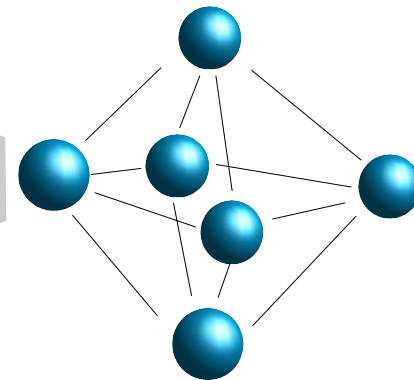
Agenten und Selbstorganisation

Mikroskopische Sicht



Autonomes
Verhalten

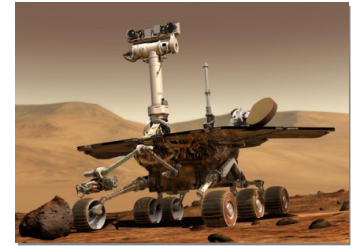
Makroskopische Sicht



Organisation

Selbstorganisation

- Szenarien für Selbstorganisation
 - Ubiquitous Computing (UbiComp)
 - Mobile Ad-hoc-Networks (MANETs)
 - Entfernte, mobile und autonome Roboter
 - Als Verbund auftretend, nicht alleine (→Autonomie)
 - Komplexe und skalierbare Anwendungen von „interplanetarer“ Größenordnung
 - Schar von Raumsonden, die miteinander kommunizieren
- Immer dann, wenn Systeme sehr groß, sehr verteilt und sehr komplex werden



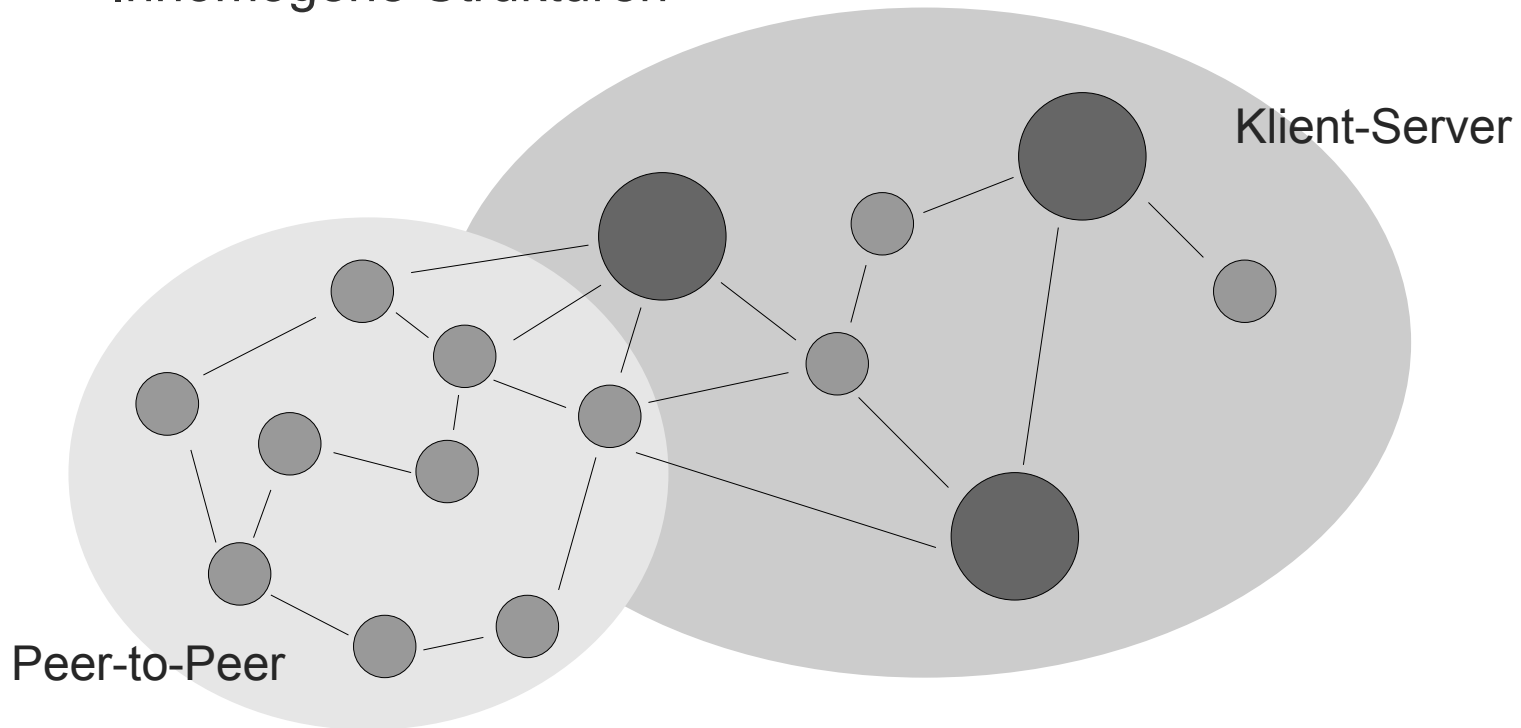
Wachsende Komplexität der Netze

- Größe der Netze und Zahl der vernetzten Geräte nimmt stetig zu
 - IPv6: Neue Möglichkeiten des großen Adressraums



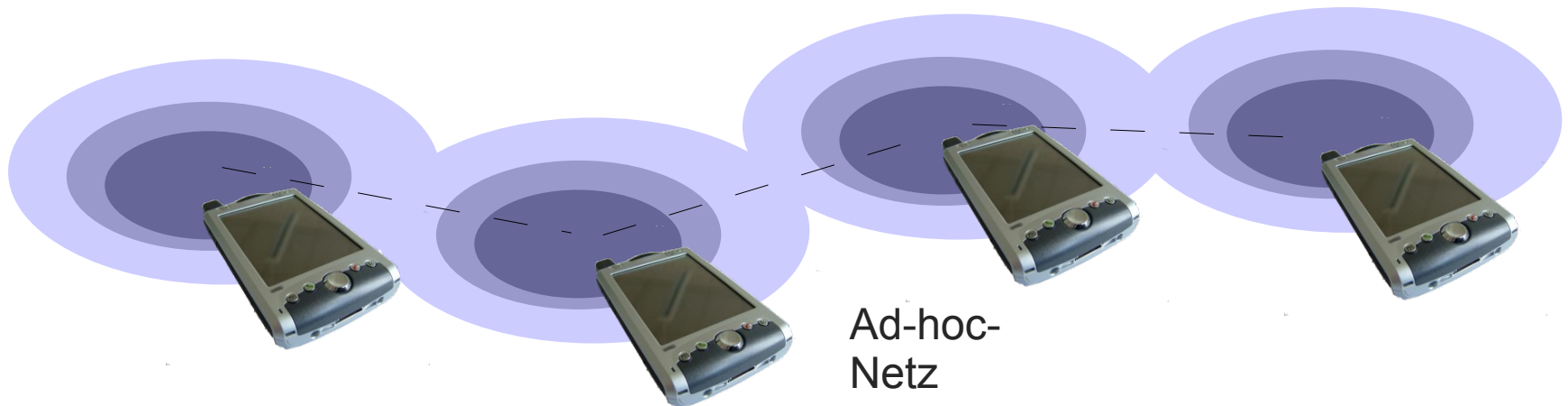
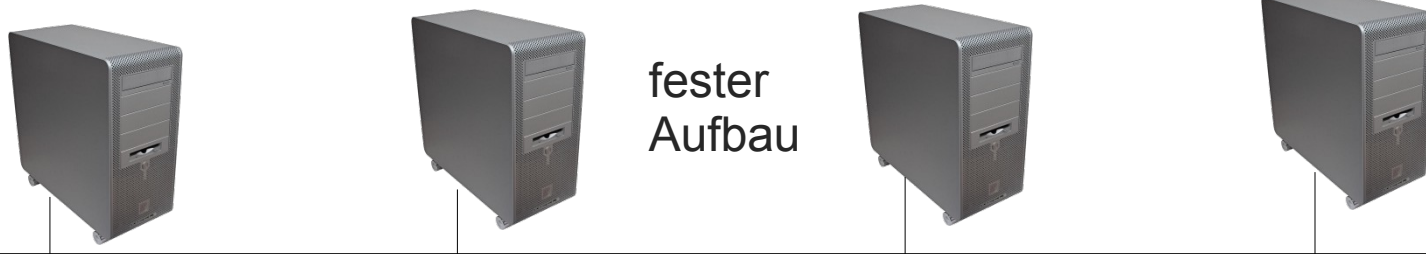
Wachsende Komplexität der Netze

- Struktur der Netze wandelt sich
 - Inhomogene Strukturen



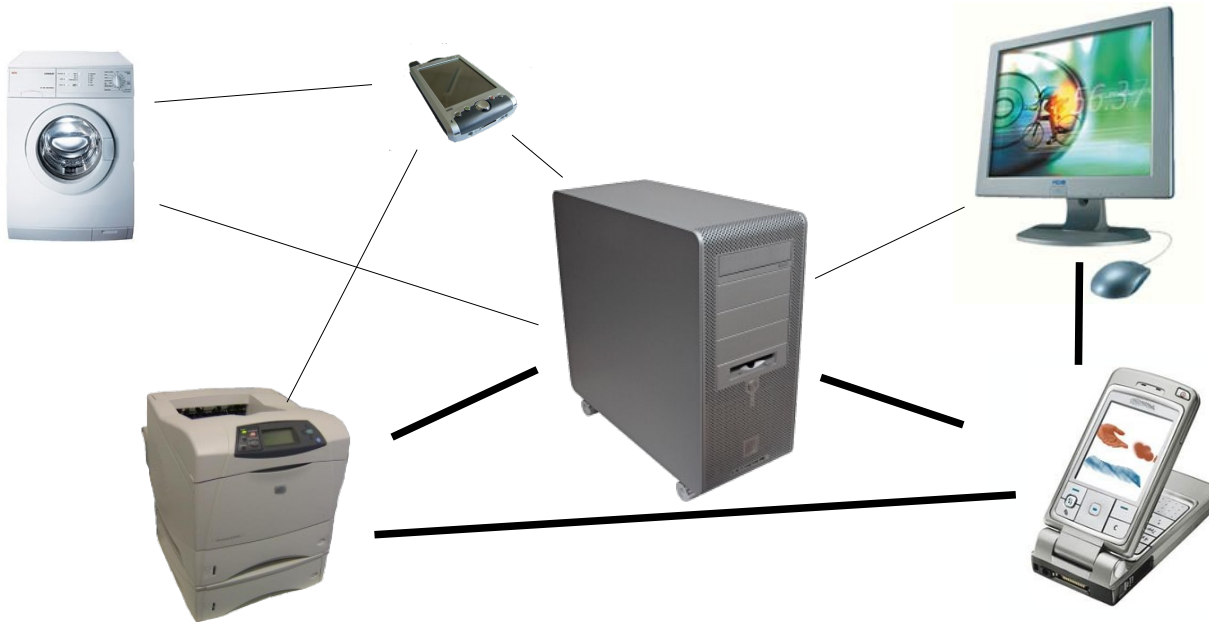
Wachsende Komplexität der Netze

- Errichtung der Netze ändert sich



Wachsende Komplexität der Netze

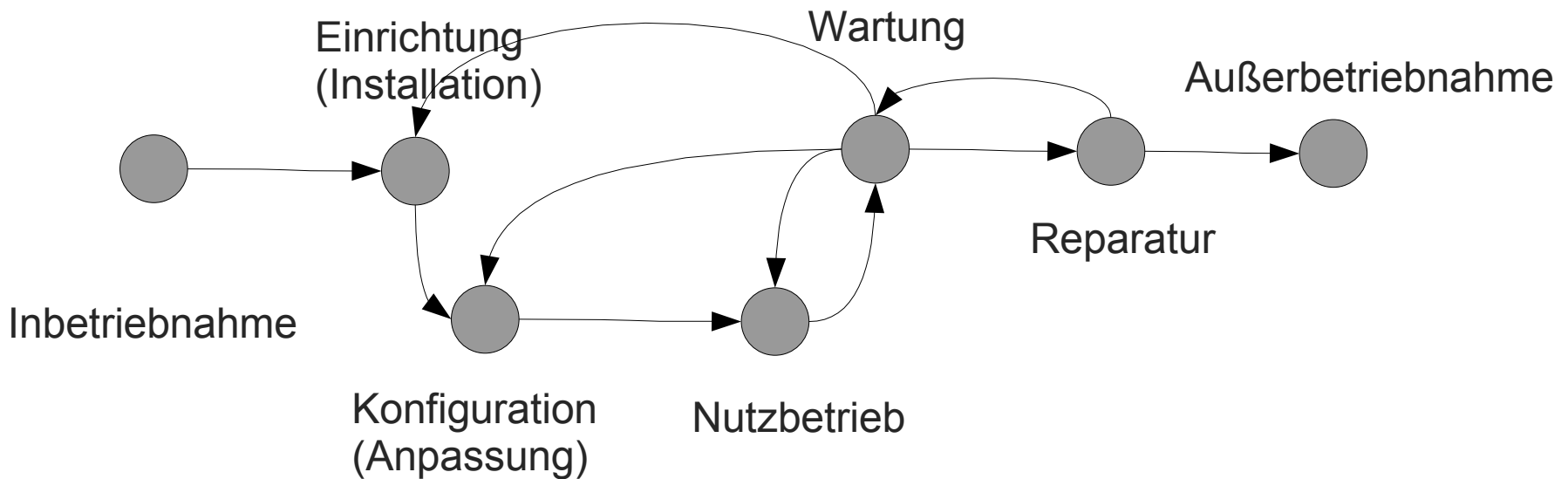
- Vernetzungsmöglichkeit für unterschiedliche Geräte



Jede der Kanten steht für ein denkbares oder heute schon verfügbares Szenario!

Wachsende Komplexität

- Komplexer Betriebszyklus



Warum Selbstorganisation?

- Wachsende Komplexität ist ein Faktum
- Damit einhergehend: Überforderung
 - des Anwenders
 - des Administrators
 - des Anwendungsentwicklers
 - des Planers (von Netzstrukturen, Anwendungseinsatz usw.)
- Der Computer sollte alles einfacher machen, aber...

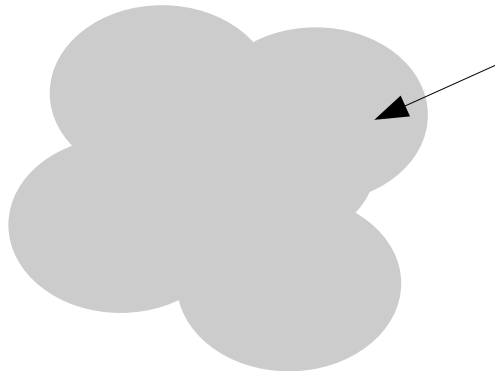
Wir erreichen die Grenzen dessen, was ein Mensch bewältigen kann und bewältigen will.

Selbstorganisation

- Ziel der Selbstorganisation
 - Erledigen von Routineaufgaben
 - Der unsichtbare Computer / das unsichtbare Netz
 - Bereitstellung von Diensten, nicht Bereitstellung von Möglichkeiten
 - Daraus Entlastung des Anwenders, verbunden mit
 - besserer Auslastung des Anwenders mit Nutzaufgaben
 - geringerem Stress durch Wegfall ungeliebter Aufgaben
 - höherer Ausfallsicherheit durch automatisierte Vorgänge und Ausschalten des „Unsicherheitsfaktors“ Mensch

Selbstorganisation

- Begriff sehr diffus
 - Ziel ist „irgendwie“ klar, aber ...
 - wie **identifiziert** man Selbstorganisation? (→ ähnlich dem Problem der Definition von Agenten)

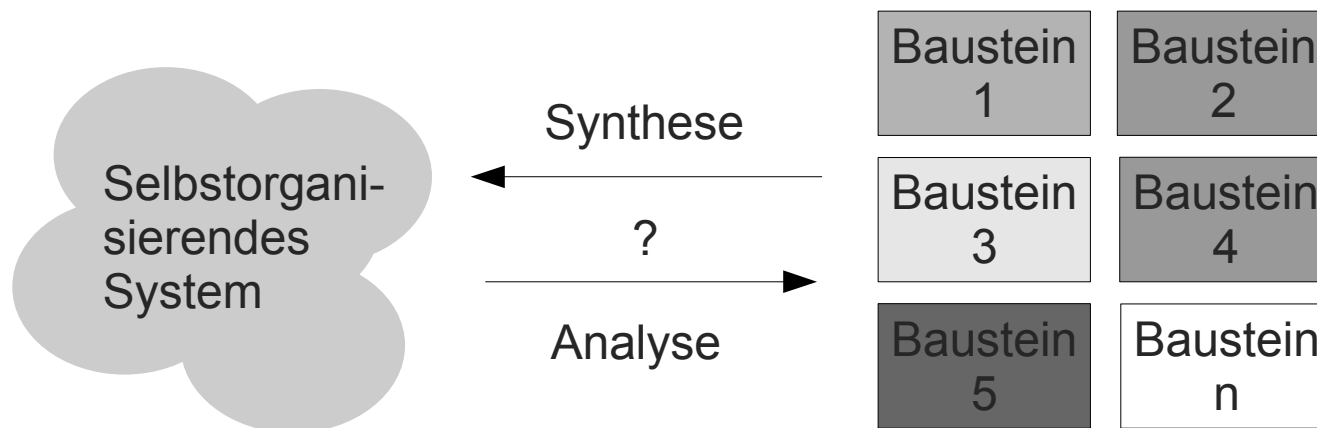


Das ist ein selbstorganisierendes System, weil es ...

Gibt es einen Konsens in der Begrifflichkeit?

Selbstorganisation

- Wie **analysiert** man Selbstorganisation?
 - Gibt es „Komponenten“ der Selbstorganisation?
- Wie **synthetisiert** man Selbstorganisation?
 - Gibt es Kochrezepte zum Bau?
 - ESOA: Engineering of Self-Organizing Applications

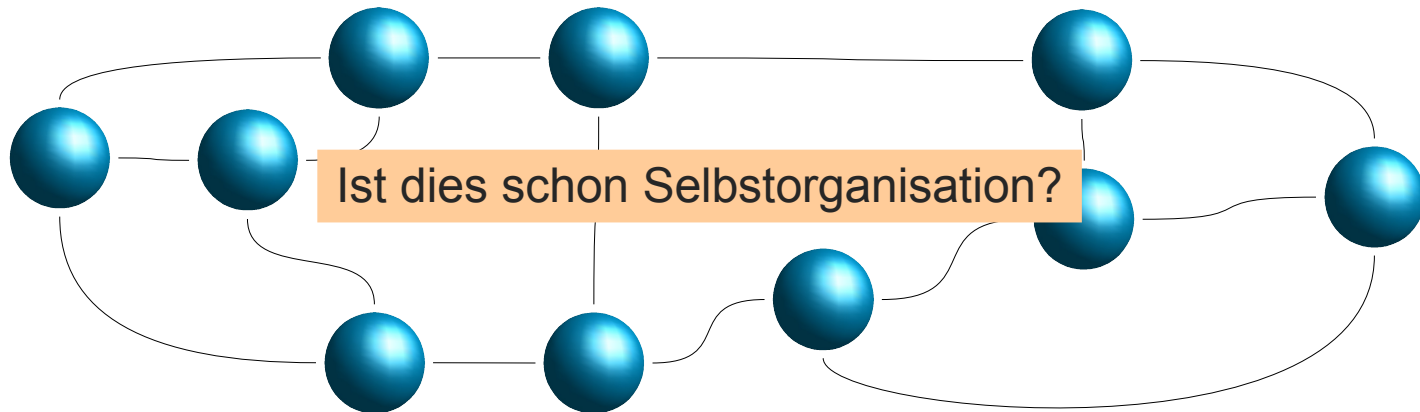


Das Dilemma der Selbstorganisation

- Theoretisches Dilemma
 - Die Begriffe sind nicht eindeutig gefasst.
 - Da keine Einigkeit herrscht, können nur schwer Schlussfolgerungen gezogen werden.
 - Klar scheint lediglich:
 - Es muss eine gewisse Menge von Individuen betrachtet werden, die eine gemeinsame Aufgabe erledigen und dabei eine makroskopisch erkennbare Struktur/Aufgabenverteilung/Kooperation bilden.
- Praktisches Dilemma
 - Wie groß ist diese Menge?
 - Ist sie zu groß, wie kann man dann überhaupt ein Szenario realisieren?

Die Natur der Selbstorganisation

- Beispiel
 - Auftrag an Agent: Beginne mit deiner Arbeit erst, wenn du drei andere Kollegen kennen gelernt hast. Erkläre dich bereit, wenn du drei andere Agenten kennst.
 - Warte, bis sich alle als bereit erklären und beginne dann mit deiner Arbeit.



Selbst-Eigenschaften

- Selbstorganisation
 - Bildung von Strukturen
- Selbstverwaltung
 - Selbständige Aktionen zur Aufrechterhaltung der Funktion (insb. des Netzes)
- Selbstkonfiguration
 - Automatische Einstellung von Parametern, um sich der Umgebung anzupassen
 - Wenn dies permanent geschieht, spricht man von (Selbst-)Adaption (oder Selbstanpassung)

Selbst-Eigenschaften

- Gibt es einen gemeinsamen Kern aller dieser „Selbst“-Eigenschaften?
 - Gibt es Systeme, die solche Eigenschaften bevorzugt ausbilden?
 - Sind diese Eigenschaften voneinander unabhängig?
- Wenn ja, gibt es konstruktive Wege zur Gewinnung dieser „Selbst“-Eigenschaften?
 - Oder kann man sie nur identifizieren?
- Erforschung dieser Konzepte unter dem Titel **Self-* properties**
 - auch bekannt als self-properties oder Selbst-Eigenschaften

Selbst-Eigenschaften

Self-adaptiveness (Selbstanpassung)
Self-analysis (Selbstanalyse)
Self-assembly (Selbstaufbau)
Self-awareness (Selbsterkenntnis)
Self-consciousness (Selbstbewusstsein)
Self-contained (abgeschlossen, unabh.)
Self-control (Selbststeuerung)
Self-creation (Selbstschöpfung)
Self-defense (Selbstverteidigung)
Self-description (Selbstbeschreibung)
Self-determination (Selbstbestimmung)
Self-development (Selbstentwicklung)
Self-enhancement (Selbstverbesserung)
Self-examination (Selbstprüfung)
Self-explaining (Selbsterklärend)
Self-governance (Selbstherrschaft)
Self-governing (Selbstregierend)
Self-healing (Selbtheilend)

Self-inspection (Selbstinspektion)
Self-management (Selbstverwaltung)
Self-modifying (Selbstmodifizierend)
Self-monitoring (Selbstkontrolle)
Self-organization (Selbstorganisation)
Self-organized Criticality (SOC/SOK)
Self-optimization (Selbstoptimierung)
Self-protection (Selbstschutz)
Self-reconfiguration (Selbstkonfiguration)
Self-reference (Selbstreferentialität)
Self-regeneration (Selbsterhaltung)
Self-regulation (Selbstregulierung)
Self-rejuvenation (Selbstverjüngung)
Self-repair (Selbstreperatur)
Self-replication (Selbstkopie)
Self-reproduction (Selbstreproduktion)
Self-similarity (Selbstähnlichkeit)
Self-stabilization (Selbststabilisierung)
Self-tuning (Selbsteinstellung)

Emergenz

- Weiteres Konzept selbstorganisierender Systeme
 - Ein Phänomen nennt man **emergent**, wenn es nicht (mit vertretbarem Aufwand) möglich ist, das Phänomen anhand der an seiner Entstehung oder Erhaltung beteiligten Faktoren als zu erwartendes Ziel zu erkennen.
 - Typische Idee: Das Ganze ist mehr als die Summe seiner Teile

Warnung: Logische Falle!

Was heißt „erkennen“, was heißt „vertretbarer Aufwand“?
Hängt das Urteil von unserem Erkenntnisstand ab?

Emergenz

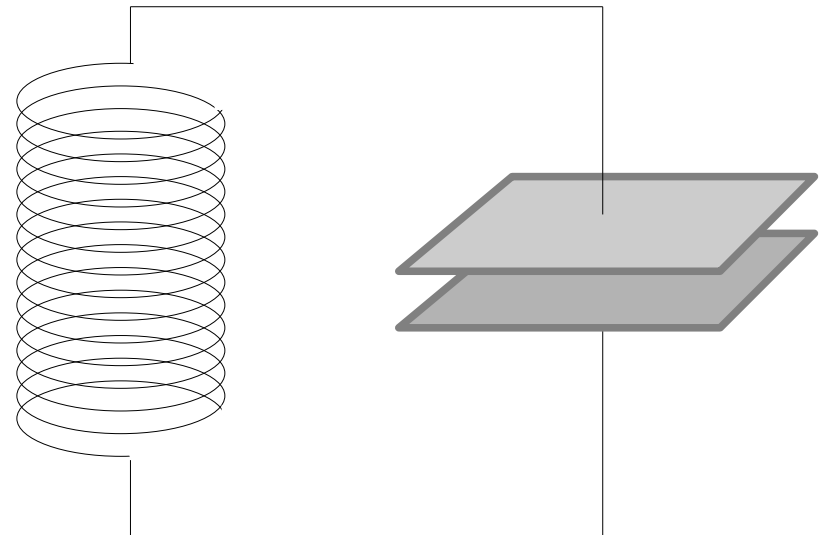
- Natürliche Phänomene
 - Warum bilden sich sechszackige, sternförmige Gebilde aus gefrorenen Wassertropfen?
 - Was bestimmt den maximalen Böschungswinkel eines Haufens?
 - Sandhaufen
 - Schnee (Lawinen!)



Zenon von Elea:
Es gibt gar keine
Haufen.
(„Beweis“ durch
Induktion.)

Emergenz

- Physikalische Phänomene
 - System beginnt zu schwingen, obwohl dies bei keiner Komponente direkt als Eigenschaft vorhanden ist
 - Schwingfrequenz ist nur bestimmbar, wenn das System vollständig ist



Emergenz

- Auch soziologisch
 - Warum verhalten sich Gruppen manchmal so völlig anders als ihre Individuen?
 - Warum verhalten sich Individuen in solchen Gruppen so völlig anders als in Isolation?



Love Parade, Berlin

Ist dies für Multiagentensysteme vielleicht auch von Relevanz?

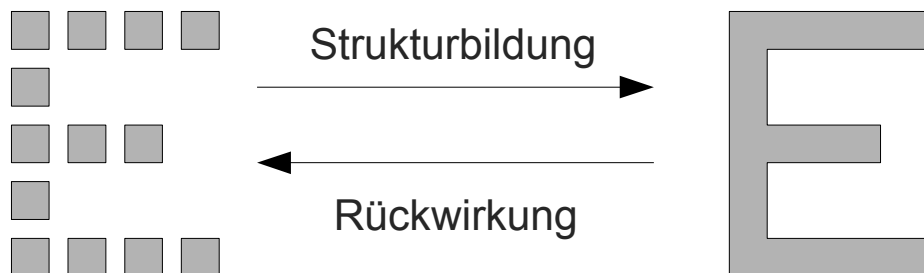
Das heißt: Müssen wir erwarten, dass eine Gruppe von Agenten – mit hinreichender Freiheit in ihren Aktionen – neuartige und unerwartete Verhaltensmuster ausbildet?

Emergenz

- Noch schwieriger zu fassen als der Begriff Selbstorganisation
 - da Emergenz etwas bekundet, das auf einer Unfähigkeit gründet (die Wirkung zu erklären)
- Betrachtung von Mikro- und Makroebene
 - Mikroebene: beherrscht von den Eigenschaften der Individuen
 - Makroebene: Struktur lässt eine Eigenschaft „emergieren“, die möglicherweise Auswirkung auf die Umgebung hat
- Eine emergente Eigenschaft muss persistent sein
 - sonst ist es sinnlos, von einer Eigenschaft zu sprechen
 - sie kann aber auch Ausdruck eines dynamischen Gleichgewichts sein

Emergenz

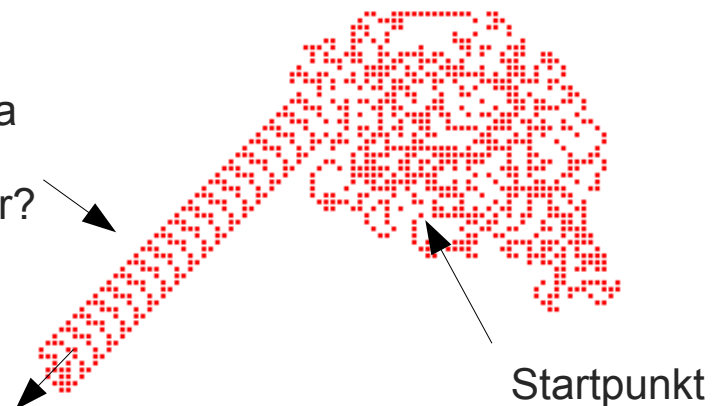
- Strukturbildung
 - geht von den lokalen Eigenschaften der Individuen aus
 - erzeugt neue Strukturen auf der Makroebene
- Rückwirkung
 - geht von den globalen Eigenschaften der Gruppe aus
 - beschränkt den Freiheitsgrad auf der Mikroebene



Emergenz in der IT

- Langtons Ameise (auch *Turmite* genannt)
 - Turingmaschine mit zweidimensionalem „Band“
 - Wenn du auf einem weißen Feld stehst, färbe es rot, drehe dich um 90° nach links und gehe einen Schritt vor
 - Wenn du auf einem roten Feld stehst, färbe es weiß, drehe dich um 90° nach rechts und gehe einen Schritt vor

Wieso entsteht nach etwa
10000 Iterationsschritten
diese geradlinige Struktur?



Die Turmite

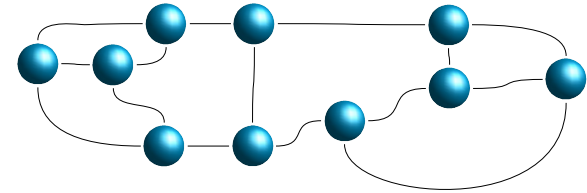
- Mikroebene
 - Definition des Verhaltens der Turmite (für jeden Schritt)
- Strukturbildung
 - Visuell erfassbares Konstrukt
 - hängt wesentlich von der Vorbelegung des Bandes ab
 - Aktionen der Turmite wirken sich auf das Band aus
- Makroebene
 - Bildung der Ameisenstraße
 - Emergentes Verhalten entsteht dadurch, dass die Umgebung so lange modifiziert wird, bis sie dieses Verhalten der Turmite erzwingt (Rückwirkung über die Umgebung)

Emergenz in der IT

- Weiteres prominentes Beispiel: Neuronale Netze
 - Regeln für das Netz sind einfach und vorgegeben
 - Das Netz wird mit Eingaben versorgt und lernt
 - Rückwirkung durch Verfahren wie *Backpropagation*
 - Irgendwann emergiert das gewünschte Verhalten
 - Man kann nicht anhand der einzelnen Neuronen erklären, wie das Gesamtverhalten entsteht
- Natürliches Vorbild
 - Gehirn: Mechanismen sind ziemlich klar
 - Verstand: Bislang ein Rätsel

Emergenz in der IT

- Die Agenten aus dem Beispiel bilden eine Struktur



- Ist diese Struktur emergent?
 - Nein, denn die Regeln sind klar vereinbart, und das Ergebnis ist daraus ableitbar.
 - Konkrete Struktur ist nicht vorhersehbar, aber auch nicht als Eigenschaft erwartet.
- Ist diese Struktur ein Zeichen von Selbstorganisation?
 - Schon eher, denn die Methodik wird lokal definiert (anstatt einer zentralen Instanz den Auftrag zu geben)
 - Begriff der Selbstorganisation nicht überzeichnen. Selbstorganisation als solche ist keine Magie!

Die Erzeugung von Emergenz

- Gibt es eine Möglichkeit, Emergenz zu bewirken?
 - Nicht konstruktiv (im klassischen Sinne), sonst wäre es „nur“ Selbstorganisation, keine Emergenz.
 - Aber wenn nicht konstruktiv, wie soll man sie in einem Programm bewirken?
- Wozu möchte man überhaupt Emergenz bewirken?
 - Beispiele
 - Unklarer Problemraum: Man kennt kein „Patentrezept“.
 - Optimierungen
 - Höherer Grad an Selbstorganisation und „Eigenleben“

Die Erzeugung der Emergenz

- Voraussetzung für die Bildung von Emergenz
 - Es gibt Individuen mit bestimmten Eigenschaften
 - Es ist definiert, wie Individuen miteinander wechselwirken
 - Individuen können Autonomie aufweisen (müssen aber nicht)
 - Es gibt eine Gruppe dieser Individuen
 - Die Gruppe als solche kann erkennbare Eigenschaften tragen
 - Solche Eigenschaften müssen abstrakt sein, also nicht einfach die Summe der Eigenschaften der Individuen
 - Eigenschaften der Gruppe müssen prinzipiell auf die Individuen unmittelbar oder mittelbar Einfluss haben
 - diese Eigenschaften, so sie sich denn bilden, wären dann die emergenten Eigenschaften

Emergenz als Gefahr?

- Die Kehrseite
 - Emergenz ist eine Eigenschaft von Gemeinschaften autonomer Komponenten, die per definitionem nicht vorhersehbar ist.
 - Damit ist sie in traditionellen Entwicklungsstrategien nicht eingeplant und ihre Folgen sind nicht abschätzbar
 - Das Auftauchen muss gemeinhin als unvorhergesehen erachtet werden.

Damit könnte das Auftauchen von Emergenz als Unfall, schlimmstenfalls als Katastrophe enden.

Es könnte also möglicherweise gerade so sein, dass man Selbstorganisation anstreben soll, aber Emergenz verhindern muss.

Proactive Computing

Eine neue Informatik

- Interaktive Computer – wie lange noch?
 - „Seit fast 40 Jahren dominiert J.C.R. Lickleaders Idee des interaktiven Umgangs mit Computern die Informatik. Auch wenn diese menschenzentrierte Forschungsrichtung beträchtlichen Erfolg hatte, wird das interaktive Modell mit dem Wachstum der Netze nicht mehr Schritt halten können, wenn die Zahl der Computer die Zahl der Anwender um den Faktor 100 bis 1000 übertrifft.“
- Kritik: Veralteter Ansatz
 - Zu viele Geldmittel fließen in die Erforschung interaktiver Verarbeitung (Standard-PCs, Standard-Informatik)
 - Die Zahl der eingebetteten Systeme ist wesentlich höher als die Zahl der PCs

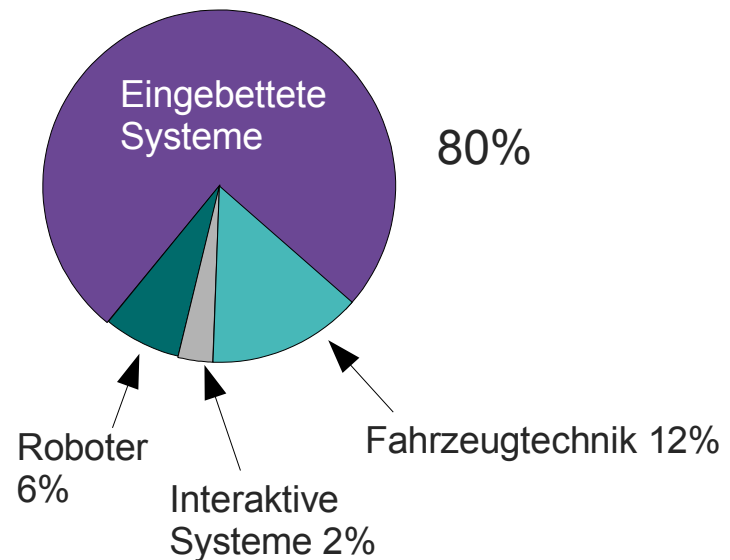
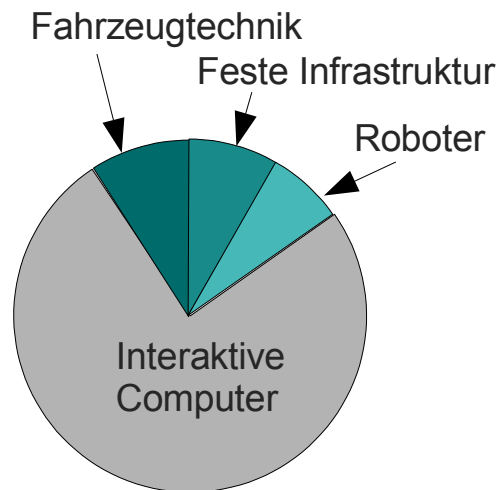


Ideen von David Tennenhouse, Vizepräsident der Corporate Technology Group von Intel

im Magazin „Communications of the ACM“, Mai 2000

Informatikforschung und Realität

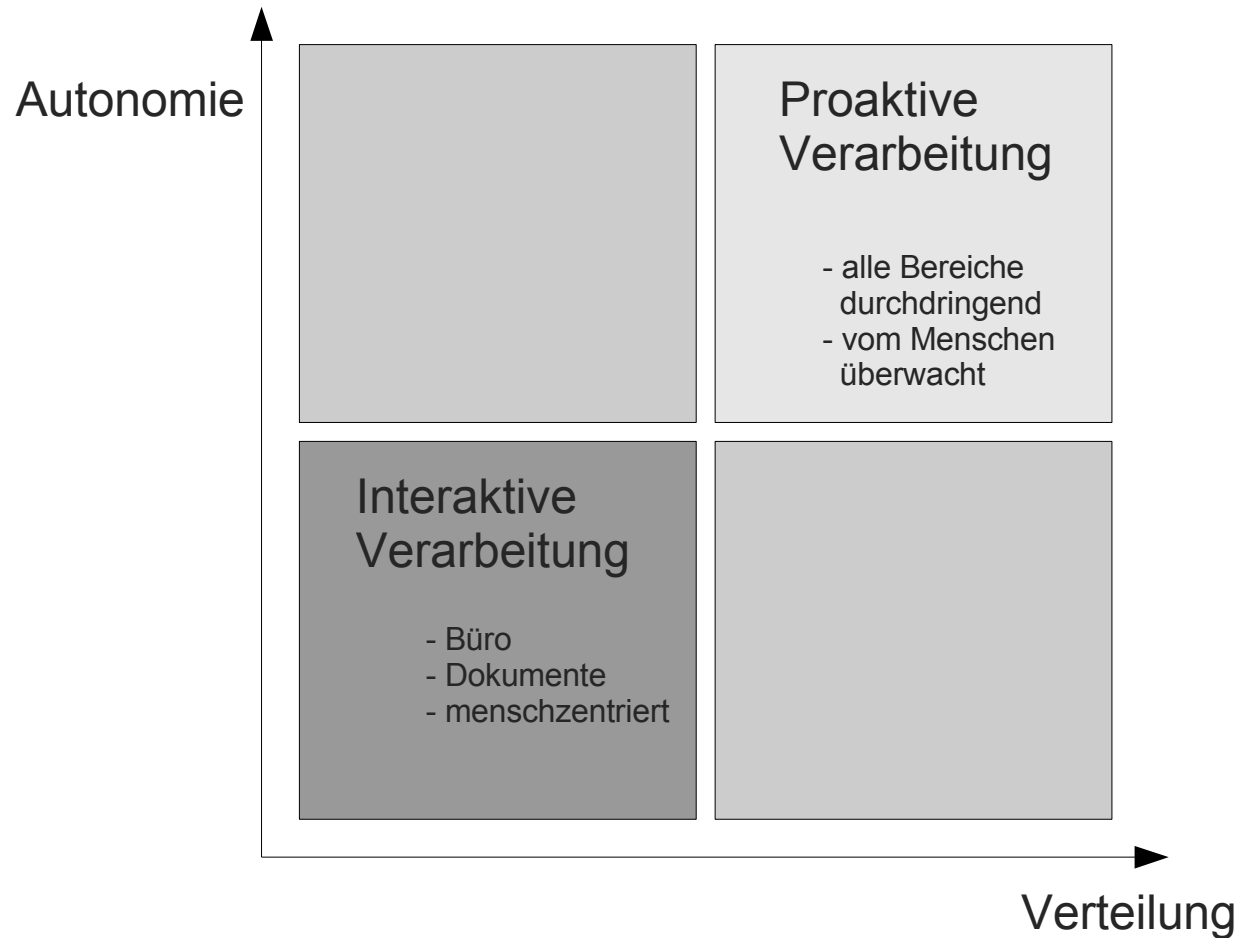
Fokus der Informatik



Verteilung der Prozessoren

Zahlen von D. Tennenhouse, 2000

Ubiquitous Computing



Ubiquitous:
allgegenwärtig

also nicht nur in
PCs, sondern auch
in anderen Geräten
und dort, wo wir
keine IT erwarten

Darstellung
nach Tennenhouse

Schritte zu einer neuen Informatik

- Getting Physical
 - Die Systeme sind immer stärker mit ihrer Umgebung verbunden
 - Sensoren, Aktoren als Schnittstelle zwischen Computer und der Welt
 - Bisherige Gebiete
 - Verbesserung der Mensch-Maschine-Interaktion (Bildschirm, Tastatur, Maus, Gestik, Haptik, ...)
 - Die Welt besteht (für IT-Systeme) nicht nur aus Menschen
 - Neuartige Sensoren und Aktoren verbinden die Systeme mit der Umwelt

Schritte zu einer neuen Informatik

- Software näher an die Natur rücken
 - etwa Software-gesteuerte Radio(de)modulation statt festgelöteter Elektronik (vgl. digitale Filter)
 - Annäherung der Verarbeitung an die physikalische Welt
- MEMS: Micro-Electromechanical Systems
- Digitale Signalprozessoren zu eingeschränkt
 - synchrones Programmiermodell
- Verbesserung durch Einführung asynchroner Verarbeitung und damit Entkopplung (faster-than-real)
 - auch Auswirkungen auf Design von Betriebssystemen

Schritte zu einer neuen Informatik

- Getting Real(-time)
 - Unmittelbare Reaktion auf Änderungen in der Umgebung
 - Informationsverarbeitung wird durch die Langsamkeit des Menschen behindert
 - Verarbeitungszeit zwischen Eingabe und Reaktion muss schneller als die Reaktion des Menschen sein
 - erlaubt Vorhersagen über die Reaktion des Systems noch vor Eintreten der Situation

Schritte zu einer neuen Informatik

- Getting Out (of the Loop)
 - Der Mensch verschwindet aus dem Regelkreis und steht *über* dem Regelkreis
 - Überwachungsschnittstelle statt Benutzungsschnittstelle
 - Verminderung des menschlichen Beitrags im Softwareentwicklungsprozess
 - Neue Abstraktionen für eingebettete Systeme
 - vgl. die Bedeutung der Abstraktion „Transaktion“ für zahlreiche Gebiete der klassischen IT
 - kann die Erschaffung von Lösungen für eingebettete Systeme beschleunigen und die Wiederverwendbarkeit fördern

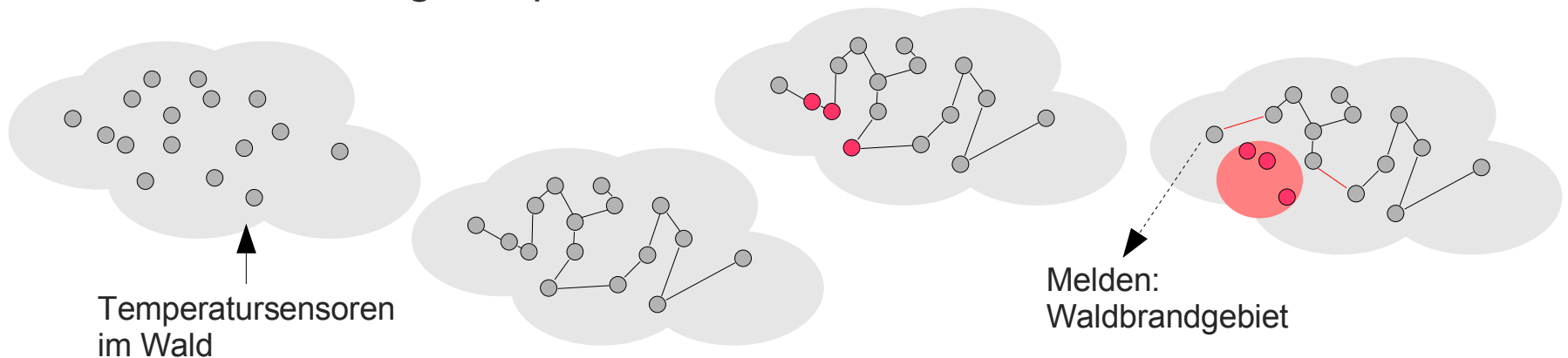
Sensornetze

- Sensorknoten als Miniaturcomputer
 - Sensorknoten sind autonom, haben eigene Stromversorgung
 - zurzeit meist Batterie-betrieben, aber auch Solarenergie denkbar
 - Drahtlose Verbindung, aber geringe Reichweite
 - Sensornetz muss den Ausfall einzelner Sensoren verkraften



Sensornetze und Selbstorganisation

- Ad-hoc-Vernetzung
 - Sensoren werden im Gelände verstreut
 - keine vorgegebene Vernetzung der Sensoren
 - selbständige Bildung eines Netzes
 - selbständige Reparatur des Netzes



Sensornetze und Proactive Computing

- Informatik muss sich gemäß dem Proactive Computing viel stärker den Ad-hoc-Netzen widmen
 - Neue Strategien der Vernetzung
 - „Energiebewusste Algorithmen“
- „Es ist Zeit, den Sieg der IT im Büroumfeld zu verkünden.“
Tennenhouse

Autonomic Computing

Autonomic Computing

- Begriff geprägt von IBM
 - Treibende Kraft in den Bemühungen des AC
- **Autonomic Nervous System = Vegetatives Nervensystem**
 - im Gegensatz zum somatischen Nervensystem, das für willkürliche Aktionen zuständig ist
 - steuert die lebenswichtigen Funktionen des Körpers ohne willentliche Einflussnahme
 - Also nicht mit „autonomous“ verwechseln!

Ansatz des Autonomic Computing

- Systeme (insb. Netze) haben primäre Funktionen zu erfüllen
- Darüber hinaus gibt es weitere Funktionen zu realisieren, u.a.
 - Sicherheit
 - Verlässlichkeit
 - Konfiguration / Umkonfiguration
 - Anpassung
- Autonomic Computing soll das System weitestgehend von menschlicher Administration unabhängig machen
- Anwender/Anwendungen können sich auf ihre Aufgaben konzentrieren
 - „focus on high-level activities“

Autonomic Computing

- Vier Basiseigenschaften
 - Selbst-konfigurierend (Self-Configuring)
 - um sich automatisch an verschiedene, teilweise noch unbekannte Situationen anzupassen
 - Selbst-optimierend (Self-Optimizing)
 - Permanente Überwachung der eigenen Komponenten und Justierung der Abläufe, um gegebene Ziele zu erreichen
 - Selbst-heilend (Self-Healing)
 - Eigenständige Problembehebung; Erkennen von (potenziellen) Problemen und Wählen von Alternativen
 - Selbst-schützend (Self-Protecting)
 - dabei verschiedene mögliche Angriffe erkennen und abwehren



Weitere Charakteristiken

- Selbstkenntnis
 - Eigene Komponenten müssen eine eigene Identität besitzen. Das System muss über sich auf verschiedenen Ebenen Bescheid wissen; seien es Komponenten, Zustand, Kapazität, Verbindungen; oder ob Ressourcen optimal genutzt werden.
- Adaption
 - Seine Umgebung analysieren und seine eigenen Regeln so anpassen, dass eine optimale Interaktion möglich wird. Es muss sich mit anderen Systemen um die Nutzung von Ressourcen bewerben.

Weitere Charakteristiken

- Interoperabilität
 - muss mit anderen Systemen zusammenarbeiten können und nicht auf proprietäre Lösungen basieren, sondern offene Standards nutzen
- Kapseln der Komplexität
 - muss die optimale Ressourcennutzung planen, ohne dabei seine eigene Komplexität offenzulegen. Es muss den Anwender von den Details der Ressourcenplanung verschonen.

IBMs Beispiel für Autonomic Computing

- IBM „eServer“
 - Beispiel für Intel-basierte Plattformen
 - Selbstkonfiguration, u.a.
 - Dynamische Partitionierung
 - Massenkongfiguration
 - Automatische Software-/Firmwareaktualisierungen (BIOS, Gerätetreiber usw.)
 - Selbstinstallation und -konfiguration mehrfach replizierter verteilter Systeme (Remote Deployment Manager)

Angaben laut IBM

IBMs Beispiel für Autonomic Computing

- Selbstheilung, u.a.
 - Speicherspiegelung
 - Echtzeitdiagnose
 - „Chipkill“-Speicher, Hot Spare Memory
 - Automatischer Server-Neustart
 - Watchdog-Timer für automatischen Neustart
 - IBM Director
 - Automatisches Systemzustandsmanagement
 - Software-Verjüngung zur Vorbeugung von Softwarefehlern
 - Kapazitätsmanager zur Feststellung von Engpässen
 - Automatische Korrekturmaßnahmen
 - Electronic Service Agent: "call home"

Angaben laut IBM

IBMs Beispiel für Autonomic Computing

- Selbstoptimierung, u.a.
 - Automatische Erkennung von Engpässen und Auflösung
 - Dynamisches Clustering von Servern
 - IBM Director
 - Automatische Aktionspläne (Event Action Plans) in Reaktion auf Alarm, automatische Scheduler, Prozesssteuerungswerkzeuge
 - Aktiver PCI-Manager zur E/A-Optimierung

Angaben laut IBM

IBMs Beispiel für Autonomic Computing

- Selbstverteidigung, u.a.
 - Fehlervorhersageanalyse mit automatischer Korrekturaktion
 - Automatische Bestimmung der Gerätebezeichnung (FRU-Zahl)
 - Web-basierter Agent mit SSL
 - für Internetübertragungssicherheit
 - Kerberos
 - Authentifiziert Dienstanfragen im Netz
 - Digitale Zertifikate
 - Verschlüsselung

Angaben laut IBM

IBMs Beispiel für Autonomic Computing

- Auch im Bereich Software: Beispiel Lotus Angaben laut IBM
 - Selbstkonfiguration
 - Automatische Verteilung von Entwurfselementen, Aktivitätsverfolgung, automatische Aktualisierung von Verzeichniselementen über Administrationswerkzeug
 - Selbstheilung
 - Analysewerkzeug kann Vorhersagen treffen
 - Selbstoptimierend
 - Analysewerkzeug ermöglicht Lastverteilung auf Serverclustern und verteilte Applikationsausführung
 - Selbstschutz
 - Verschlüsselung, sichere Authentifikation und Zugangssteuerungsmanagement, digitale Signaturen usw.

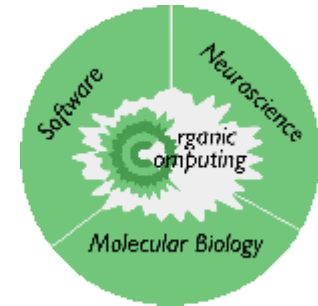
Autonomic Computing

- Kritik an der Definition
 - IBM hat den Begriff für sich „vereinnahmt“: Autonomic Computing hat nun eine feste Bedeutung
 - Alternative Ansätze lassen sich nun hier nicht mehr unterbringen
 - z.B. Konzepte des Informationsflusses ähnlich biologischer Prinzipien
- Kritik am Prinzip
 - Aufbau von Systemen als Kombination vieler eigenständiger Regelkreise
 - Interferenz zwischen Regelkreisen noch weit gehend unerforscht

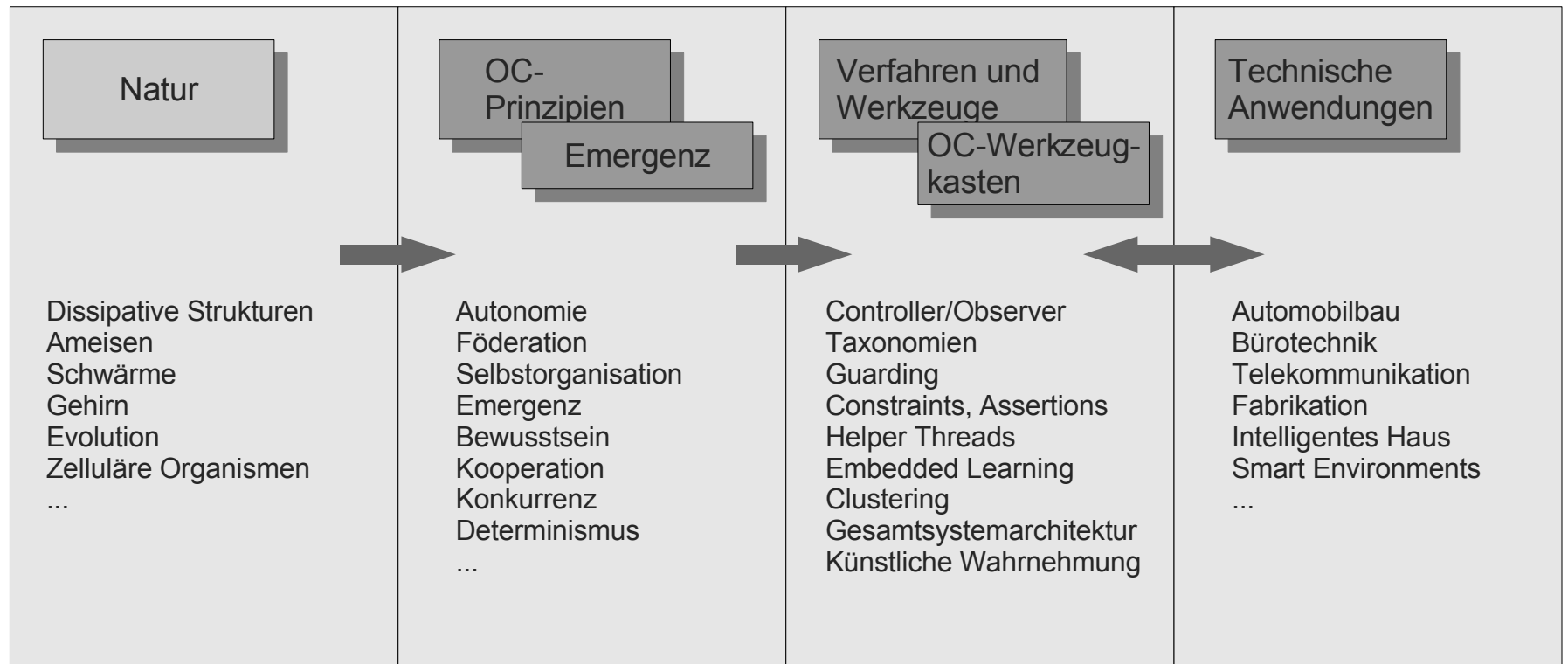
Organic Computing

Organic Computing

- Initiative aus Deutschland
 - Universitäten Bochum, Hannover, Lübeck und andere
- Ähnlich dem Autonomic-Computing-Ansatz
 - Fokus von AC eher auf selbstorganisierende Netze
- Prinzip
 - Lernen von der Natur; Natur als Vorbild; „technische Nutzung natürlicher Prinzipien“
 - Zusammenführen von Erkenntnissen aus dem Bereich der Molekularbiologie, Neuroinformatik, praktische/technische Informatik
 - Selbstheilung, Selbstschutz, Selbstoptimierung, Selbstkonfiguration, Selbsterklärung
 - Explizite Nutzung von Emergenz
- Soll im Gegenzug Ergebnisse an die beitragenden Gebiete liefern können



Organic Computing



← Hauptarbeitsgebiete des OC →

Organic Computing

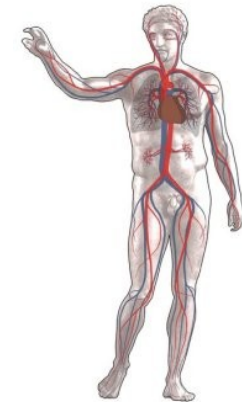
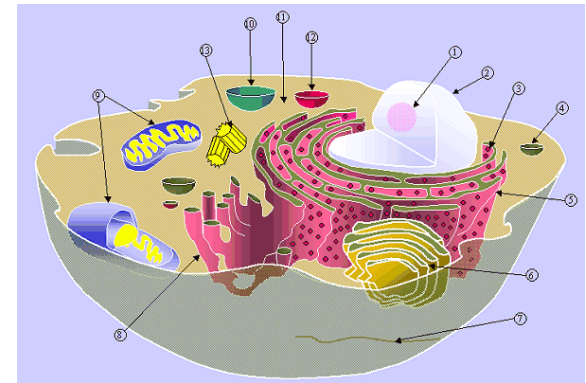
- Herausforderungen
 - **Theorie:** Mangel an grundlegendem Verständnis und an Metriken zur quantitativen Beurteilung von Phänomenen der Selbstorganisation und Emergenz
 - **Zielgerichtete Prozesse:** Nicht nur die Entwicklung statischer Strukturen studieren, sondern auch die Organisation zielgerichteter Prozesse. Keine Theorie vorhanden, welche die Interaktion zahlreicher autonomer, adaptiver Prozesse behandelt.
 - **System-Architekturen:** Wie erhält man stabile, nützliche Strukturen über Selbstorganisation? Methodiken?
 - Observer-Controller-Architektur als neues Schlagwort
 - **Funktionssicherheit:** Wie verhindert man Fehlentwicklungen im Lernvorgang?

Organic Computing

- Herausforderungen
 - **Embedded Learning:** Wie kann man A-priori-Wissen in den Lernvorgang einbauen, um das Lernen effizienter zu gestalten?
 - **Flexible Interaktion:** Verbesserung der Wahrnehmungsfähigkeit, um Interaktion mit Umwelt und Anwender zu verbessern
 - **Selbsterklärung:** Entwickler müssen Einblick in die Funktion bewahren können
 - was ein (künstliches) neuronales Netz nicht leistet

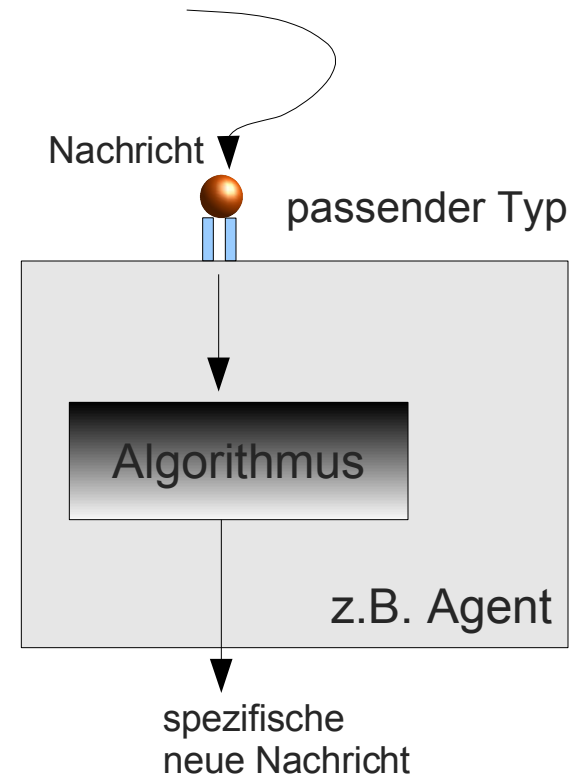
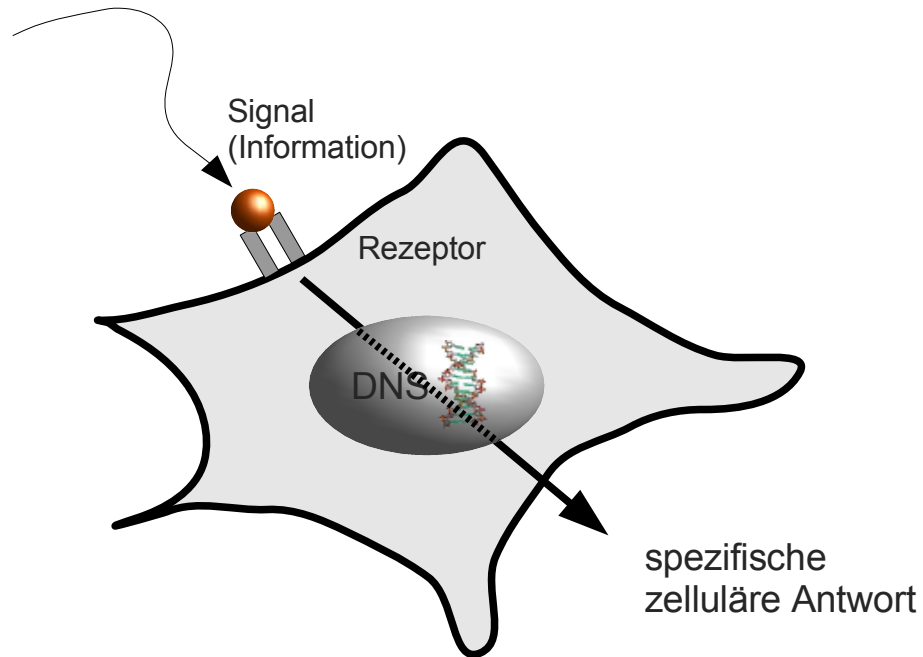
Biologisch inspirierte Systeme

- Nachbau des Organismus
 - Identifikation von Systembestandteilen als abgeschlossene, autonome Einheiten (Zellen, Organe, ...)
 - Aufbau von hierarchisch strukturierten, „künstlichen Organismen“
- Kommunikation ähnlich wie in der Biologie
 - Zellen kommunizieren über die Blutbahn
 - Nachrichten können ebenso zwischen den künstlichen Organen verbreitet werden

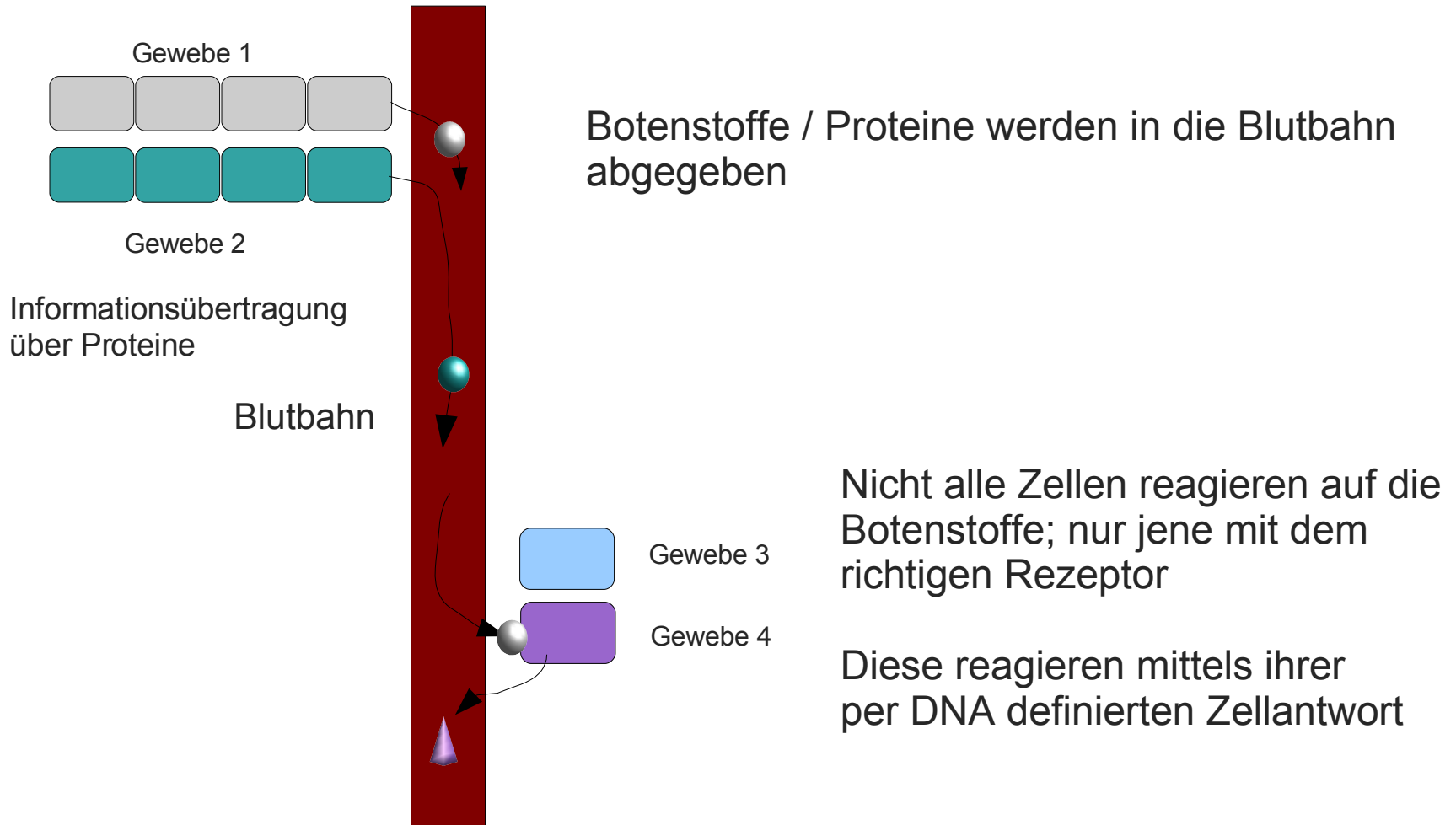


Biologisch inspirierte Computer

- Zell-Metapher

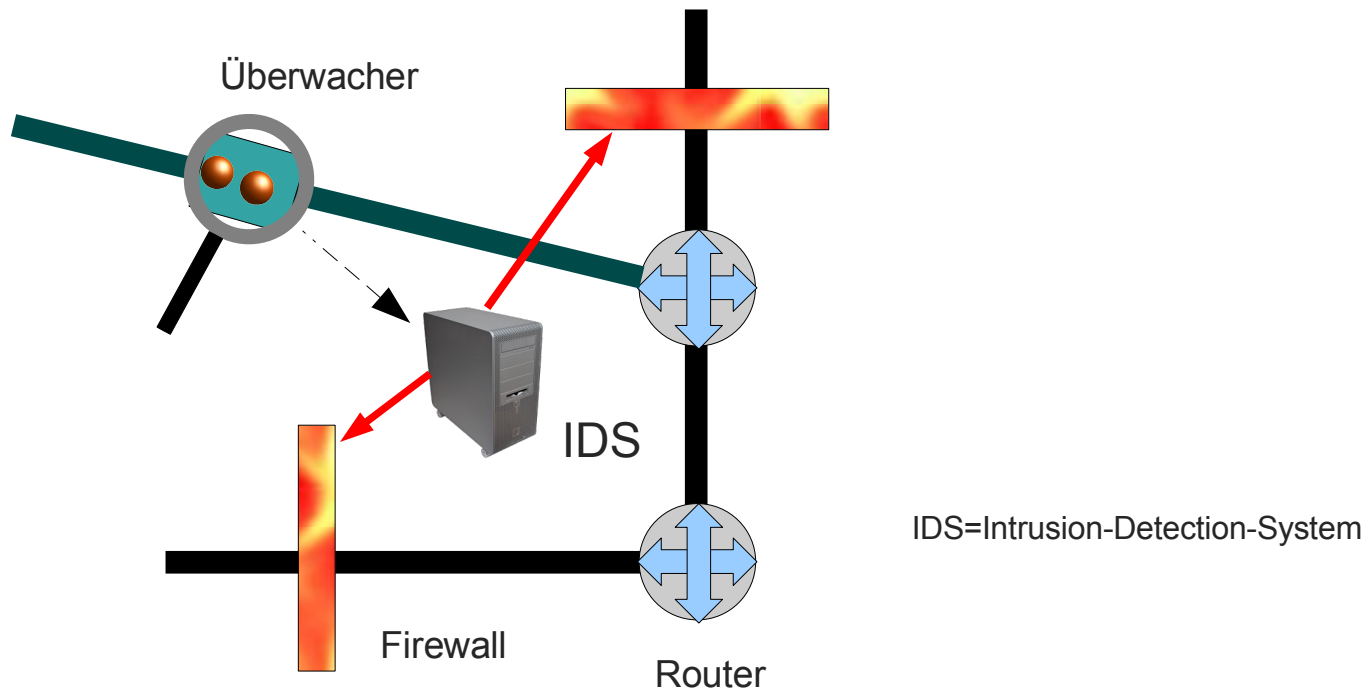


Biologisch inspirierte Computer



Biologisch inspirierte Computer

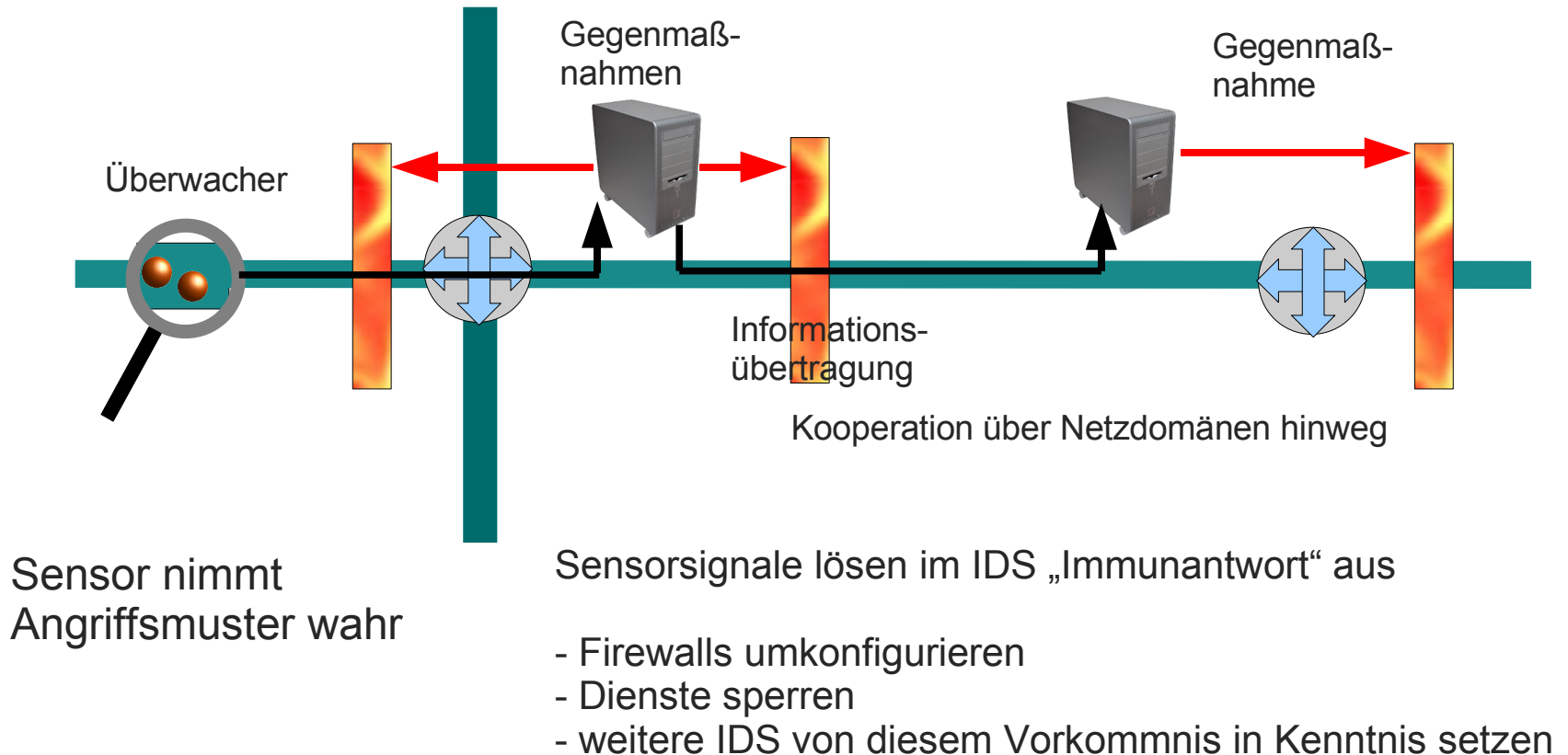
- Selbst-Schutz durch künstliche Immunsysteme
(ARTificial Immune SysTem: Aickelin, Greensmith, Twycross, 2004)



Biologisch inspirierte Computer

- Das natürliche Immunsystem erkennt unbekannte Eindringlinge und Fremdkörper
- Daraus Analogien
 - Intrusion-Detection-Systeme (IDS) agieren als Sensoren.
 - Im Falle des Einbruchs an einer Stelle des Firmennetzes melden diese ihre Beobachtungen und charakteristische Muster des Angriffs an verschiedene Stellen im Netz, insbesondere an Router und Firewalls.
 - Auch hier werden die Erkenntnisse der IDS an alle Interessenten im Netz propagiert - ganz ähnlich der Blutbahn.
 - Propagierung könnte über Multicast oder über Tupelraum-ähnliche Nachrichtensysteme geschehen

Biologisch inspirierte Computer



Biologisch inspirierte Computer

- Prinzip
 - Erkennung von Eindringlingen und Fremdkörpern („Intrusion Detection“) durch Detektoren (im Körper durch B- und T-Zellen)
 - Alarmierung bei Erkennung von fremdartigen Einheiten
 - Bekämpfung der erkannten Fälle
- Wie erkennt man etwas Unbekanntes?
 - Detektoren für Selbst/Nicht-Selbst bzw. harmlos/schädlich
 - Unterscheidung in der Regel einfach: Selbst = bekannte, gewohnte, normale Dinge; Nicht-Selbst/Fremd = unbekannte, ungewohnte Anomalie

Biologisch inspirierte Computer

- Negative Selektion: Zensur von fehlerhaften Detektoren
 - Große Vielfalt von Detektoren produzieren
 - alle, die auf systemeigene Signale ansprechen, eliminieren
 - alles, was übrig bleibt, sind Detektoren für fremde Signale
- Jedoch übliche Probleme der Medizin
 - neu zu installierende Software immer unbekannt,
 - falscher Alarm, Autoimmun-Angriffe, Allergien ...

Recovery-Oriented Computing

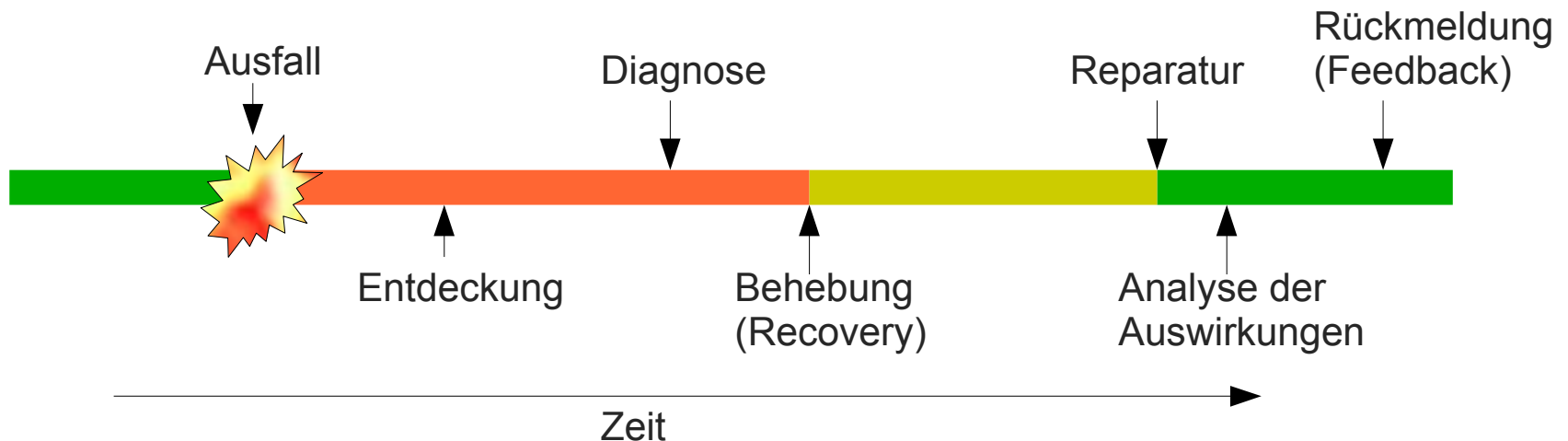
Recovery-oriented Computing

- Neuer Ansatz: RoC
 - führend: Universitäten Berkeley & Stanford
- Grundidee
 - Störungen/Fehler/Ausfälle unvermeidlich
 - Erwarte Fehler: Wenn Fehler auftritt, Schaden minimieren
 - schnelle (Selbst-)Diagnose und Wiederherstellung

Selbst-Heilung ist möglich!

Recovery-oriented Computing

- Übliche Strategie
 - betroffenes Teilsystem neu starten, am besten noch bevor Fehler auftritt.
 - Fehler-Toleranz durch Redundanz, Replikation, Fehler-Detektoren, ...

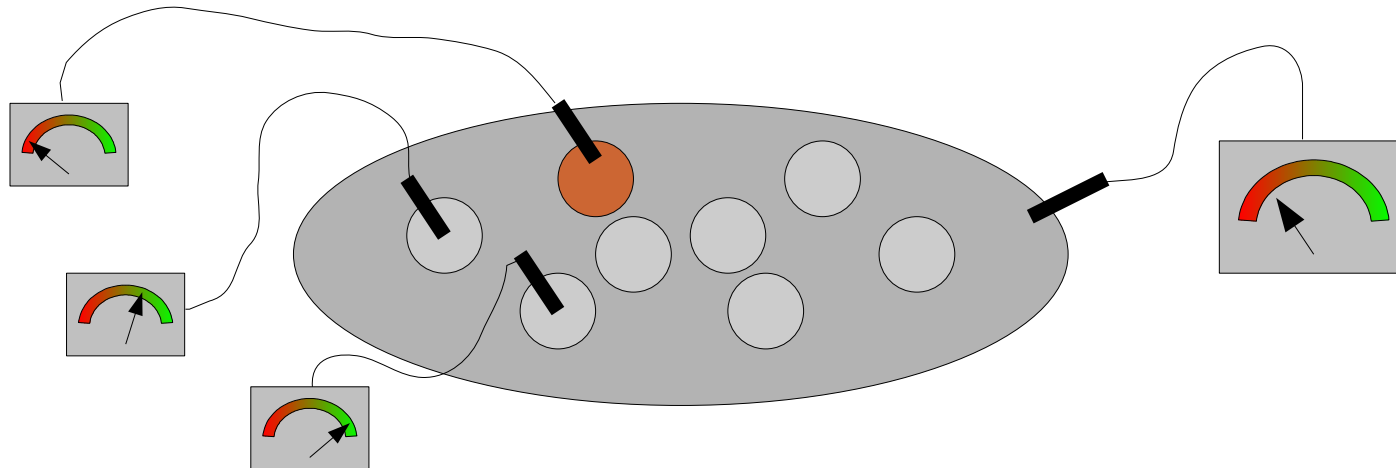


Recovery-oriented Computing

- Zweck dieses Ansatzes
 - Man kann nicht alle Fehlerquellen von vornherein ausschließen
 - Man muss Software so erstellen, dass sie Fehler verkraftet
 - Die Software soll (wie der Organismus) dafür sorgen, dass sie selbständig möglichst schnell wieder ihre alte Leistungsfähigkeit erreicht
 - Der Schaden/die Kosten eines Ausfalls sollen minimiert werden
- Voraussetzungen
 - Lebenszyklusabhängigkeiten zwischen den Komponenten minimieren
 - Jede Komponente kann unabhängig neu gestartet werden: *Micro Reboot*

Selbstverjüngung

- Ähnlich dem RoC-Ansatz
 - System beurteilt fortlaufend seine Performanz und die seiner Komponenten
 - Wenn eine Komponente „altert“, wird sie erneuert/aufgefrischt
 - Alterung: Längere Antwortzeiten durch wachsende Puffer / Tabellen / Speicherleck

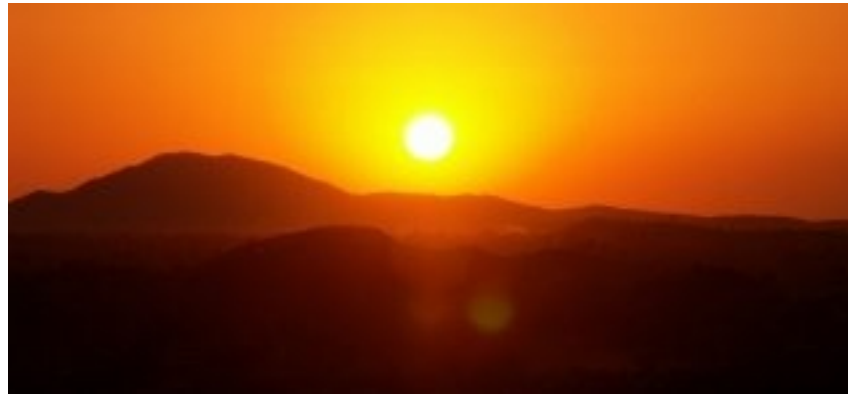


Der Weg in die Zukunft

- Ubiquitous Computing (auch Pervasive Computing)
 - allgegenwärtige Computer
 - Ausstatten der Umwelt mit IT
- Ambient Intelligence
 - Umgebung, die sich auf den Anwender anpasst

Der Weg in die Zukunft

- Intelligenz und Autonomie als Mittel zur Gestaltung selbstorganisierender Systeme



Ist unsere Gesellschaft bereit für eine Durchdringung unseres Lebens durch intelligente, autonome Systeme?