Simulation von Sensornetzwerken

Dennis Pfisterer, Alexander Kröller

TECHNISCHE UNIVERSITÄT
CAROLO-WILHELMINA
ZU BRAUNSCHWEIG

I. Simulationen

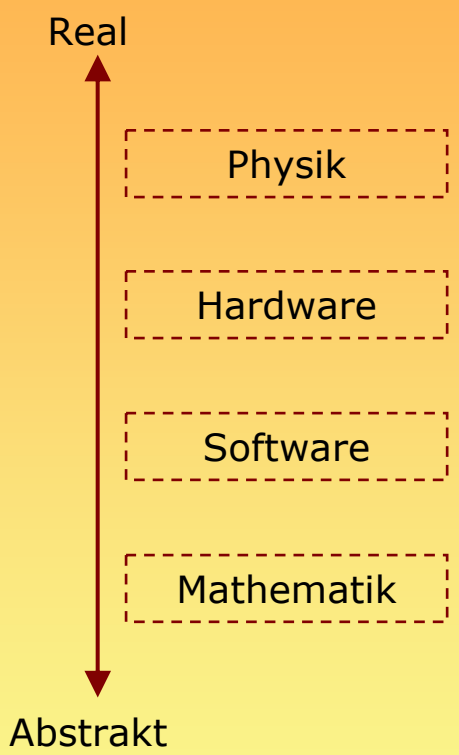
Simulationen von Sensornetzen

- Sehr hohe Anzahl
- Knappe Ressourcen
  - Energie
  - Rechenleistung
  - Speicher
- Kein OSI Stack
- Fehleranfällig

Simulationen

- Unterschiedlichste Vorstellungen von Simulationen
- Art der Simulation ist abhängig von
  - Fragestellung
  - Abstraktionsebene
  - Denkweise
  - Herangehensweise

Simulationen

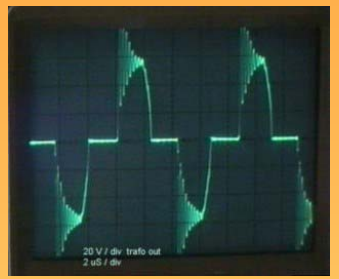


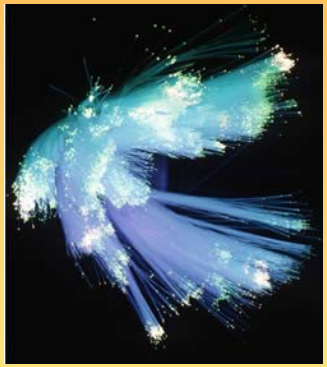
- Unterschiedliche Professionen

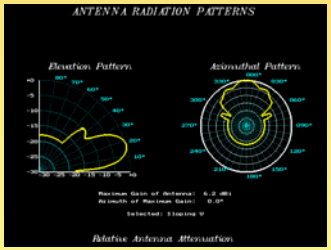
  haben verschiedenen Fokus

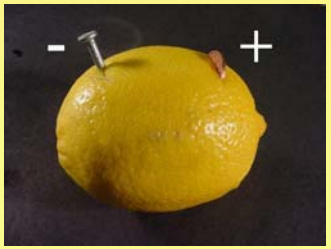
  - Betrachtung von Phänomenen der echten Welt
  - Entwicklung von Hard-/Software
  - Abstrakte Sicht der Mathematik

Simulationen: Physik









- Simulation der „echten“ Welt
- Physik betrachtet
  - Antennen
  - Energieversorgung
  - Übertragungsmedien
  - Werkstoffe
  - …

Simulationen: Einsatzgebiete

Simulationen: Hardware

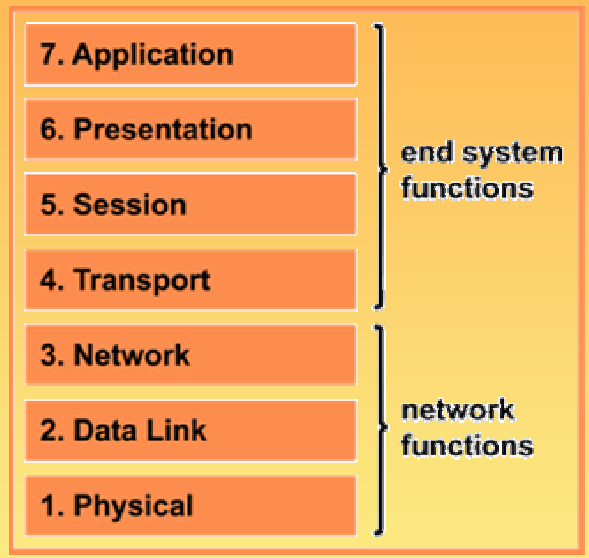
- Simulation/Emulation der Hardware
- Fokus auf
  - Prozessoren
  - Chips
  - Miniaturisierung
  - Funktechnik
  - Sensoren
  - Aktoren
  - …

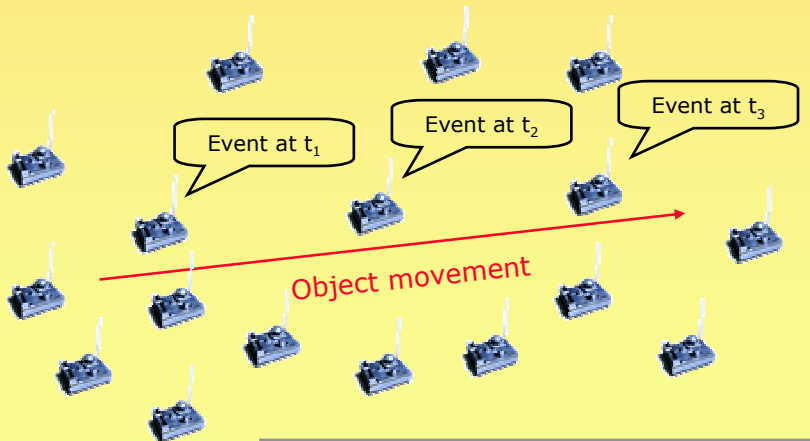
Simulationen: Einsatzgebiete

Simulationen: Software

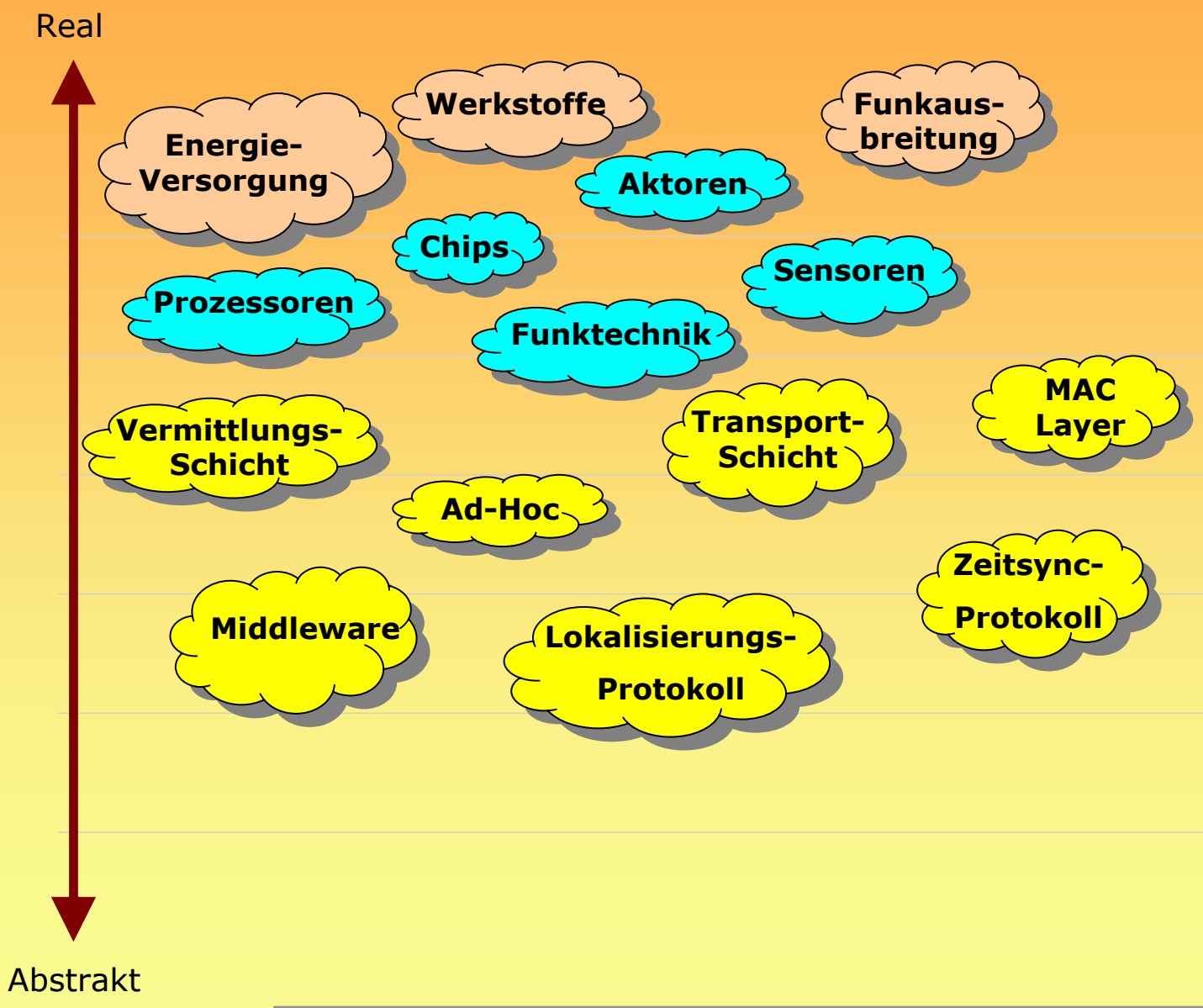
- Simulation aller Schichten des ISO/OSI Layers

- Anwendungen
  - MAC Layer
  - Protokolle
  - Middleware
  - Target Tracking, …



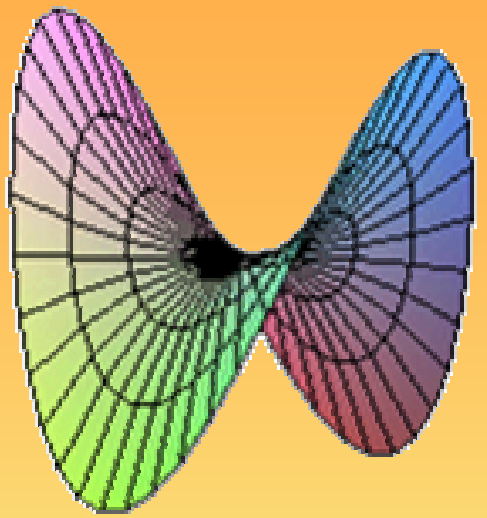


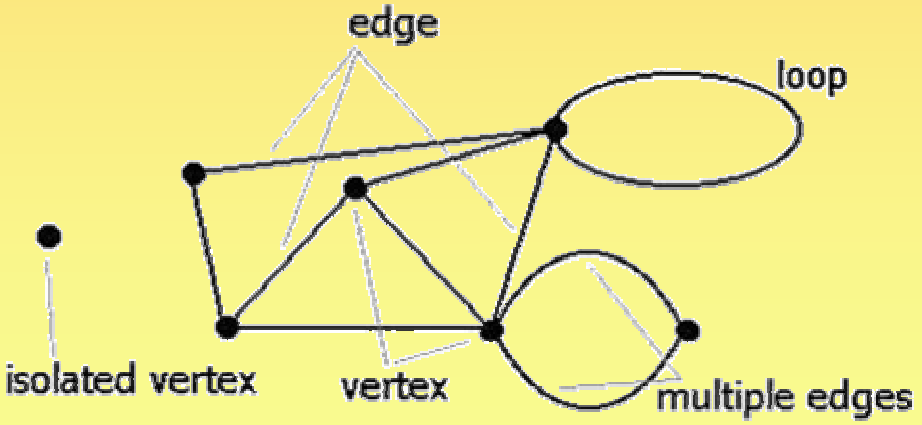
Simulationen: Einsatzgebiete



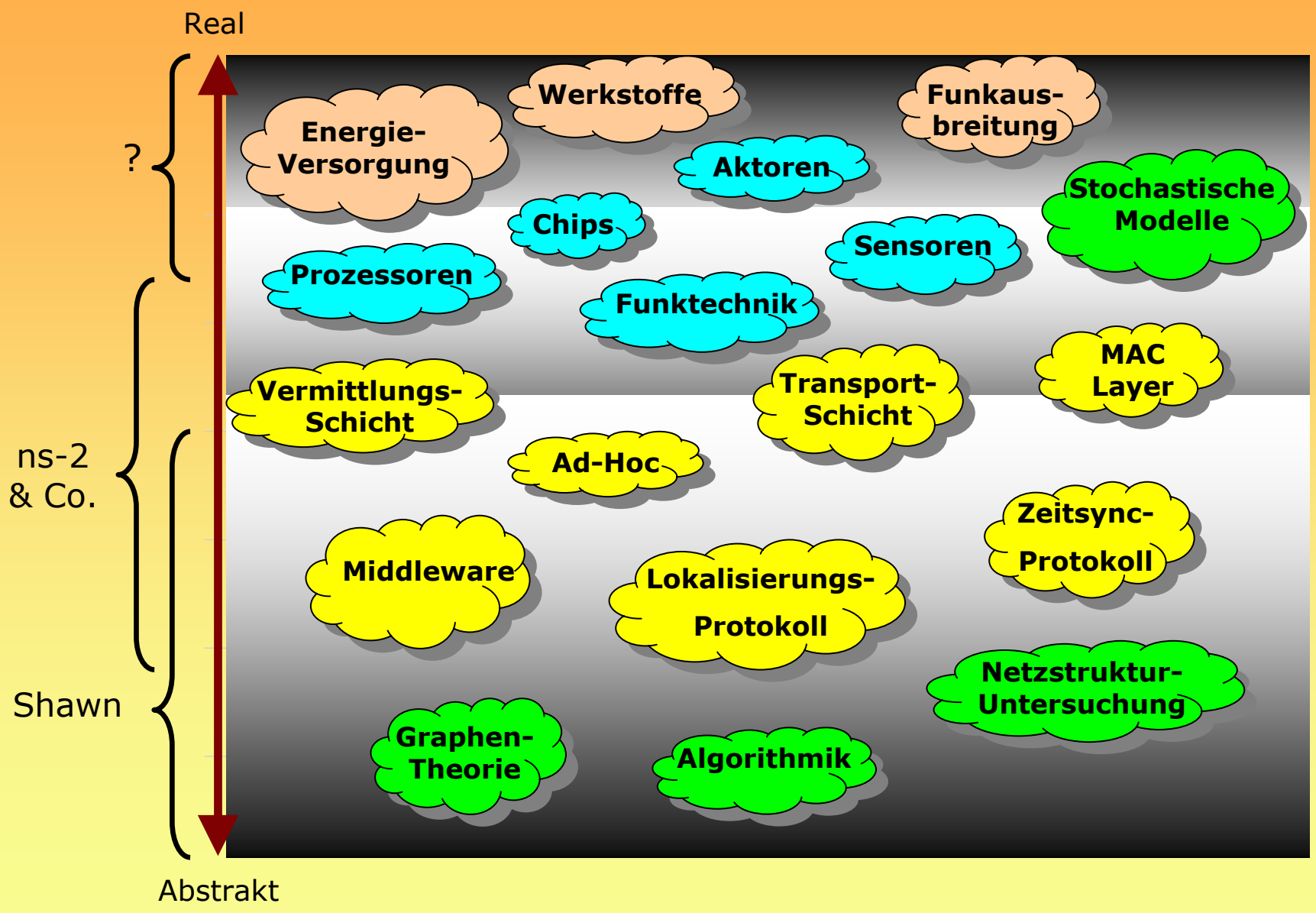
Simulationen: Mathematik

- Abstrakte Sicht auf Probleme
- Analyse der Netzstruktur
- Formulierung von Algorithmen
- Beweisbare Aussagen

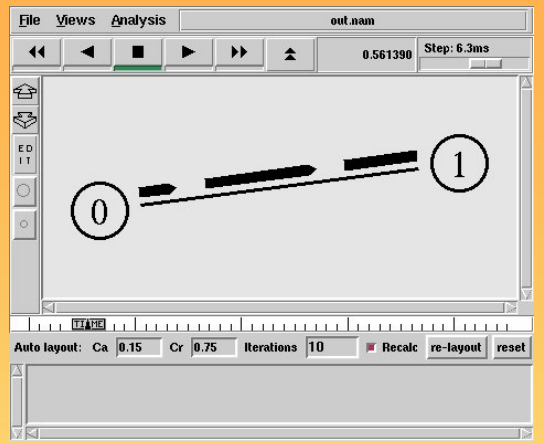




Simulationen: Einsatzgebiete



ns-2



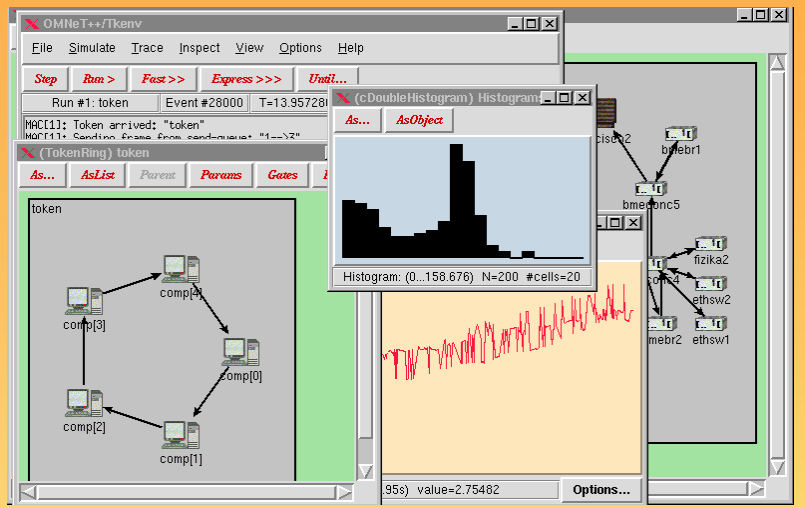
- Quasi—Standard für

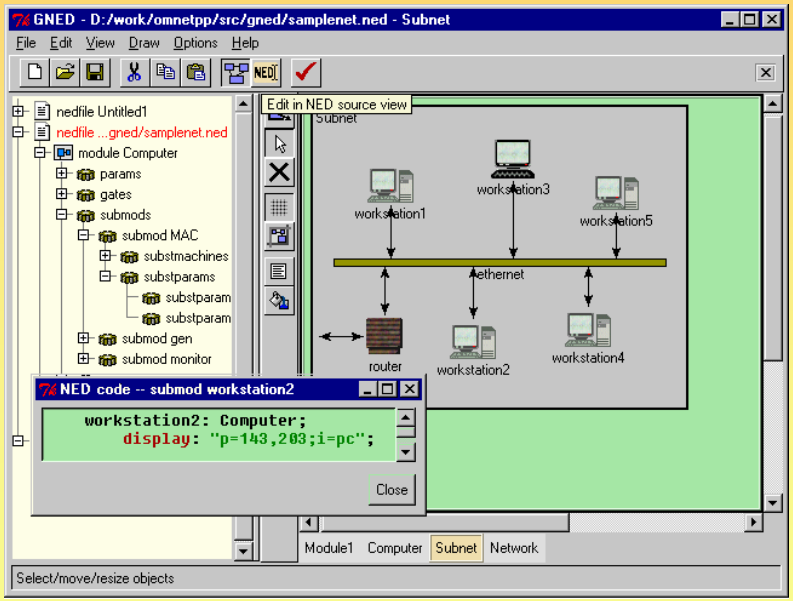
  ISO/OSI 1—4 Simulationen
- Viele Protokolle bereits integriert
- Erweiterungen für Sensornetzwerke
- Flexible Steuerung durch OTcl Skripte
- Detaillierte Tracefiles und Visualisierungstool „nam“
- Hohe Einarbeitungszeit, recht komplex
- Nicht für hohe Knotenzahlen ausgelegt (< 16k)

(http://www.isi.edu/nsnam/ns, 1989)

OMNet++

- Graphischer Netzwerkeditor und Visualisierung
- Komponentenarchitektur
- Folgt dem OSI Schichtenmodell
- Kombination aus C++ und eigener Sprache (NED)
- Frei für wissenschaftliche Anwendungen





(http://www.omnetpp.org, 1994)

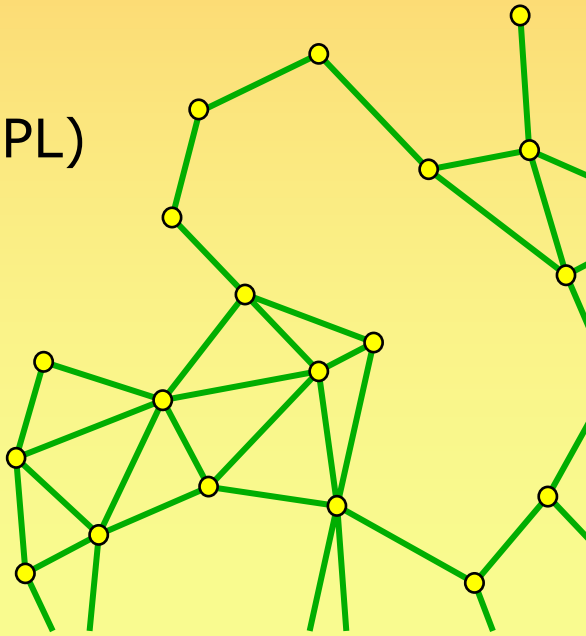
Zusammenfassung erster Teil

- Breite Spanne der Simulationsgegenstände
- Kein Simulator deckt das ganze Spektrum ab
- Eigenen Fokus finden
- Wahl des Simulators muss zu eigenen Zielsetzungen passen
- Interpretation und Schlussfolgerungen hinterfragen und in Kontext setzen

II. Shawn

Shawn: Überblick

- In dem Projekt SwarmNet entwickelt
- In C++ implementiert
- Verfügbar unter Open-Source-Lizenz (GPL)
- Mithilfe und Verwendung willkommen
- Homepage: http://www.swarmnet.de



Shawn: Ziele

Effekte der Dinge simulieren

Nicht: Die Dinge simulieren

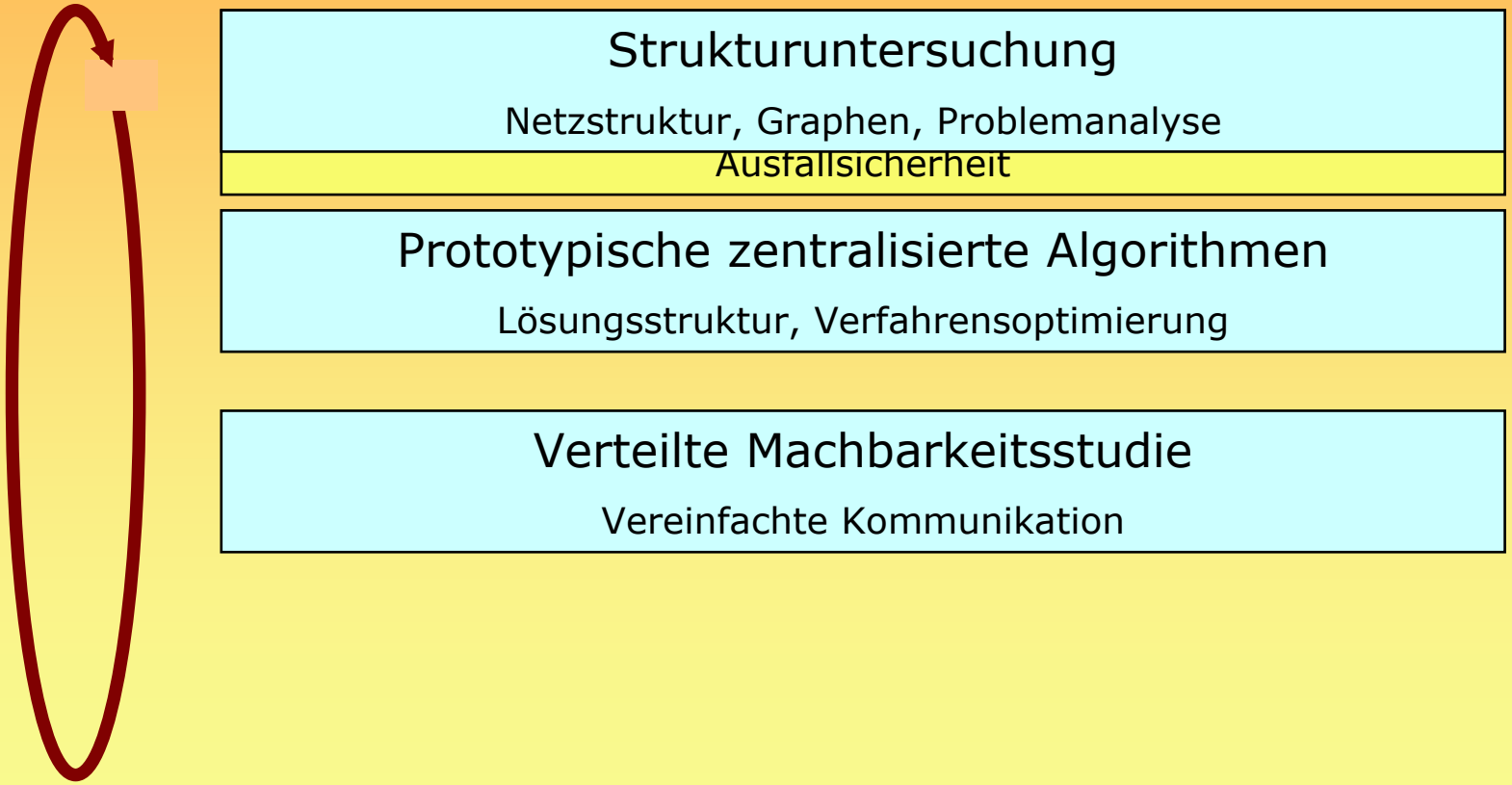
Simulation auch sehr großer Netze

Nicht: Klein und detailliert

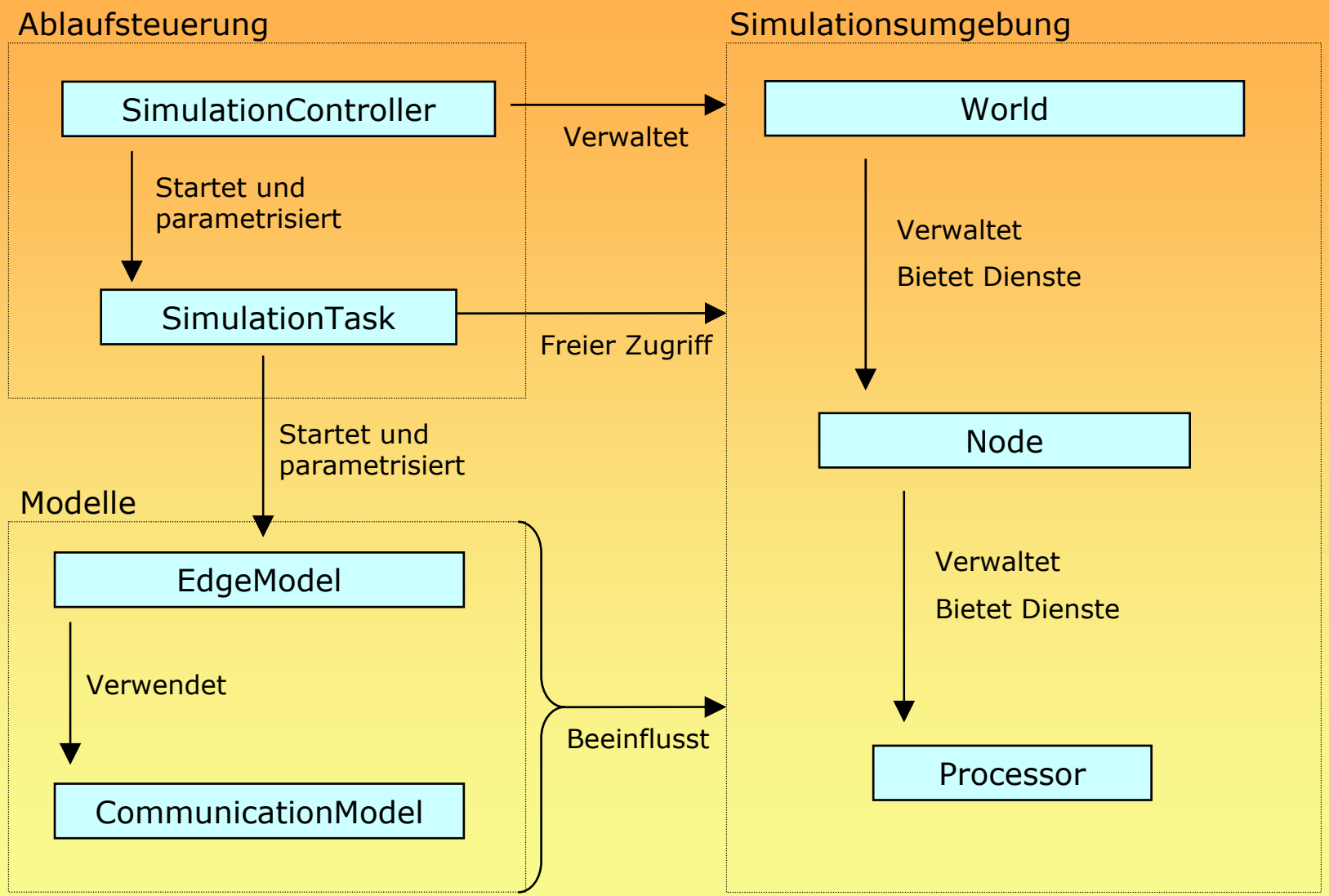
Freie Modellwahl

Nicht: Vorgegebene Strukturen

Shawn: Entwicklungszyklus

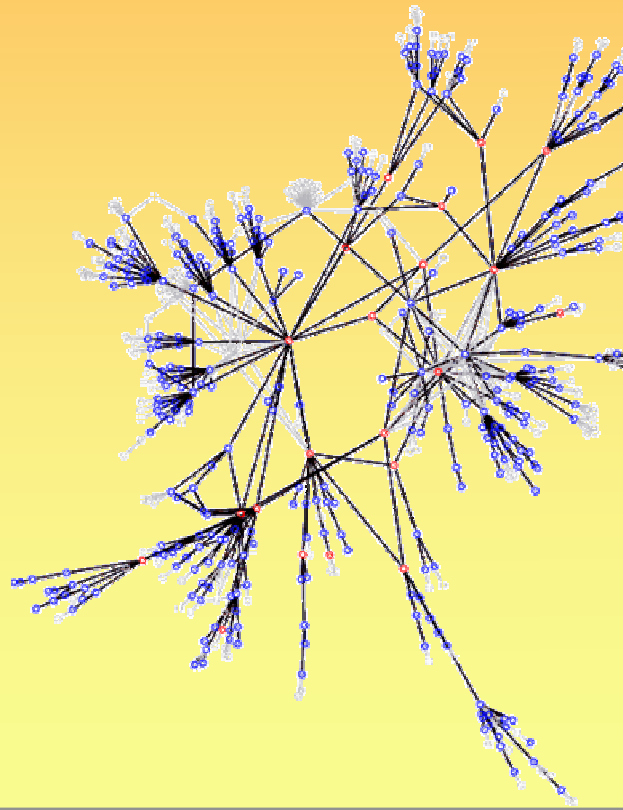


Shawn: Zusammenspiel



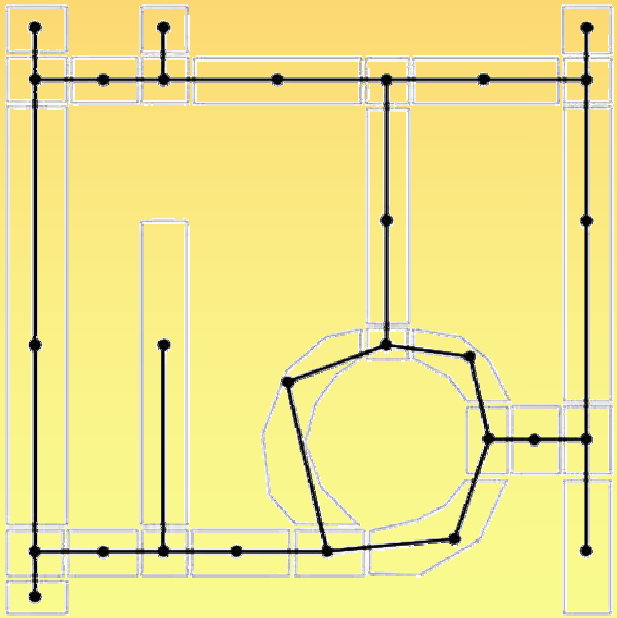
Shawn: Interface CommunicationModel

- Bestimmt welche Knoten kommunizieren können
  - can_communicate_bidi(u,v)
- Mögliche Implementierung:
  - Unit Disk Graph
  - Probabilistisch
  - Vorgegebene Liste

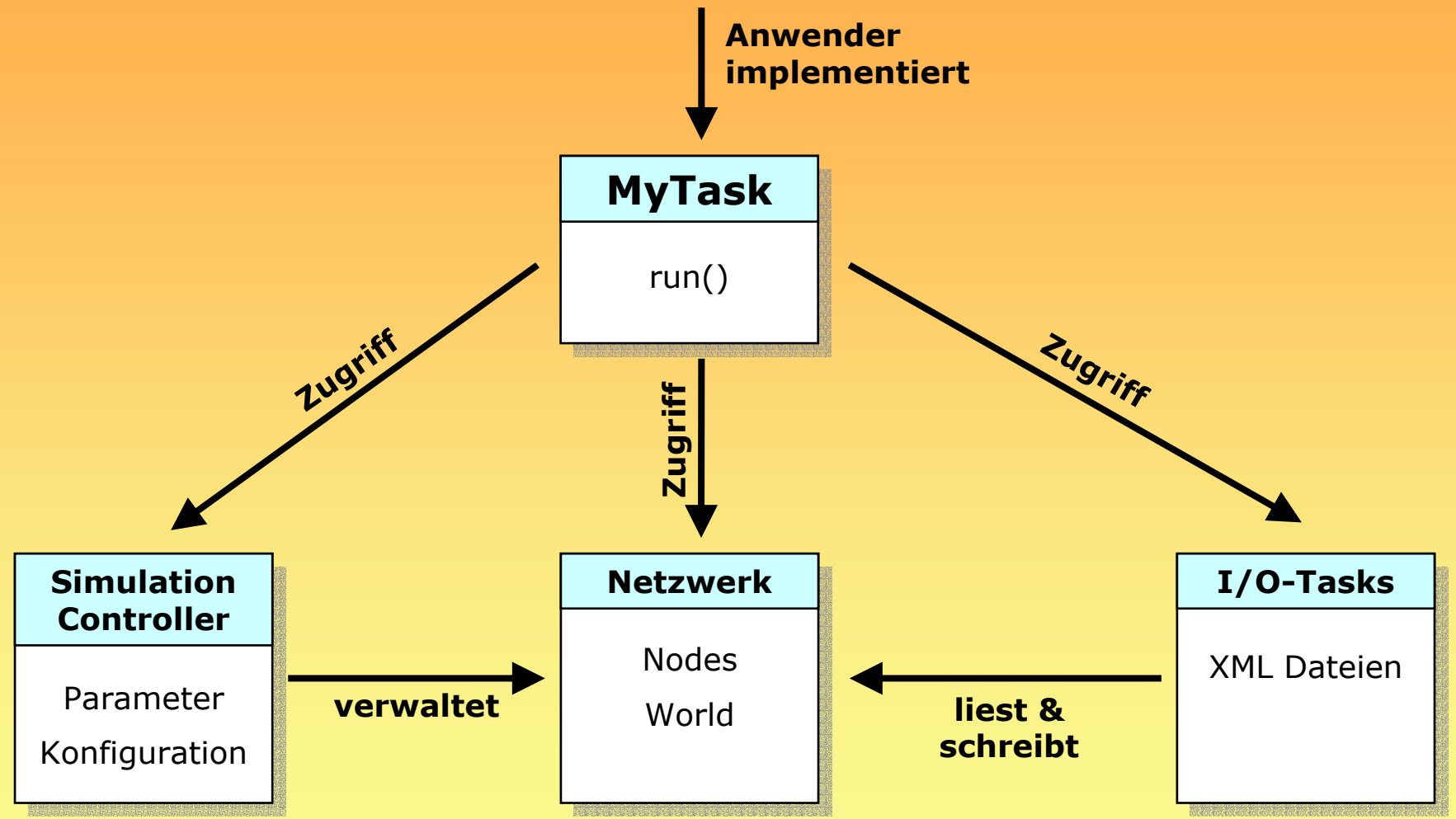


Shawn: Interface EdgeModel

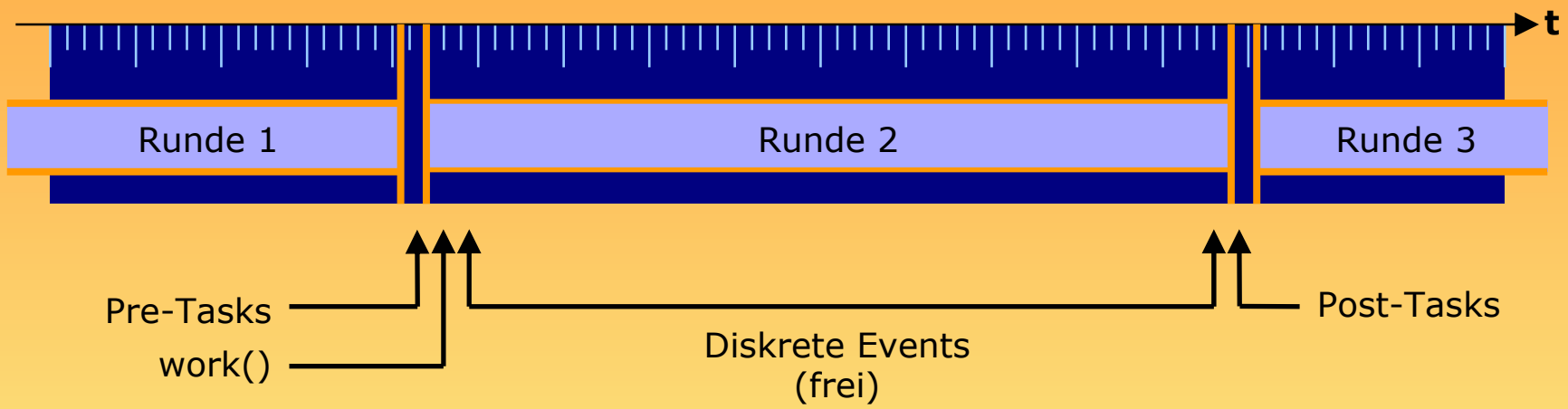
- Zugriff auf direkte Nachbarschaft
- Verwendet CommunicationModel
- Mögliche Implementierung:
  - Abspeichern des kompletten Graphen
  - Kein Caching, stets Neuberechnung
  - Gitterbasiert



Shawn: Zentralisierte Programmierung

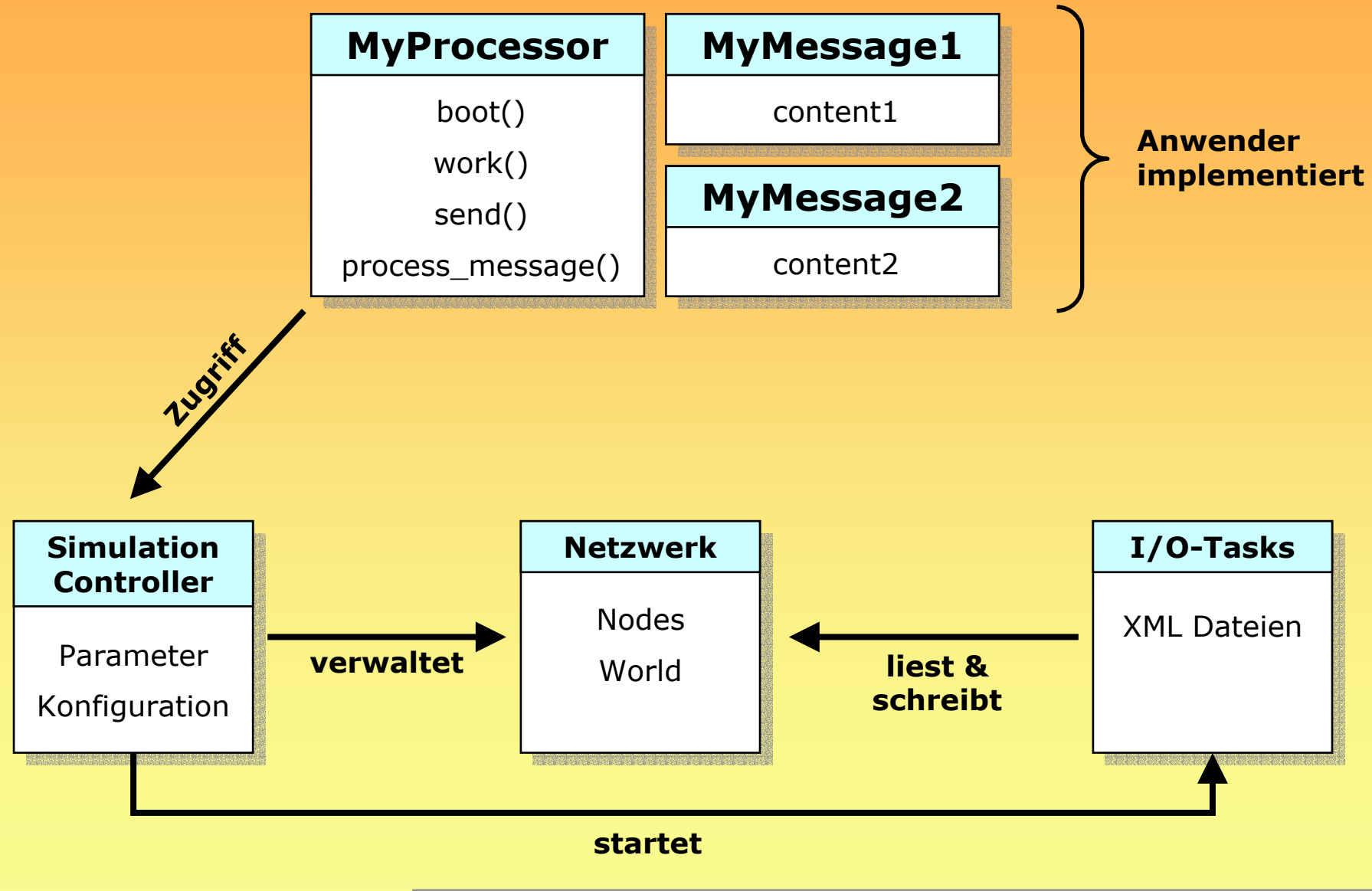


Shawn: Timing



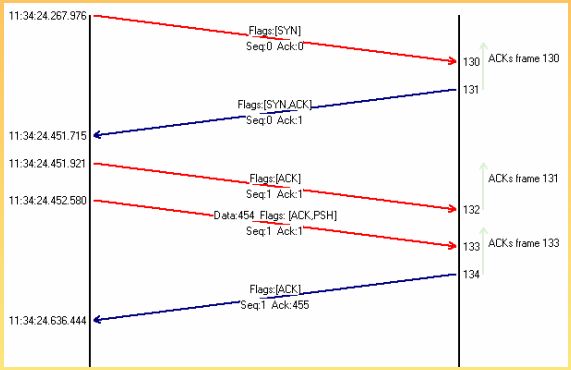
- Simulationen laufen in Runden ab
- Freie Wahl des Timings
  - Synchronisiert (via Pre-/Post-Tasks, work())
  - Völlig frei (via Events)

Shawn: Verteilte Programmierung



Shawn: Aktuelle und zukünftige Arbeiten

- Mehr Modelle
  - TransmissionModel
    - » Regelt Nachrichtenzustellung
    - » Kann z.B. Effekte von MAC Layern simulieren
  - Mobilitätsmodell
- Mehr Auswertungen
  - Detailliertere Logfiles
  - Visualisierungsframework
- Integration von
  - CGAL, TNT
  - ns-2 Dateiformaten



III. Live