

# Zusammenarbeit zwischen Agenten durch Protokolle (Contract Net etc.)

Michael Gress (WI4237)

Fachhochschule Wedel - Informatikseminar SS2004

Dr. S. Iwanowski u. Professor Dr. U. Schmidt

## Vorbemerkungen:

### Teil 1:

*Zusammenarbeit zw. Agenten* durch Protokolle

→ Verteiltes Problemlösen (*Distributed Problem Solving*)

### Teil 2:

Zusammenarbeit zw. *Agenten und Protokolle(n)*

→ Contract Net-Protokoll, FIPA etc.

## Veranstaltungsüberblick:

1. Agenten und Multiagentensysteme
2. Verteiltes Problemlösen
3. Contract Net – Protokoll im Detail
4. FIPA
5. Zusammenfassung und Ausblick

# 1. Agenten und Multiagentensysteme

## Definition: Was ist eigentlich ein Agent???

Russell and Norvig 2003 „An agent is **anything** that can be viewed as **perceiving its environment** through sensors and acting upon that environment through actuators.“

IBM „Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of **independence or autonomy**, and in so doing, employ some knowledge or representation of the user's **goals or desires**.“

Maes 1995 „**Autonomous agents** are computational systems that inhabit some complex dynamic environment, sense and **act autonomously** in this environment, and by doing so **realize** a set of **goals or tasks** for which they are designed.“

## Definition: Was ist eigentlich ein Agent???

Smith, Cypher and Spohrer 1994 *“Let us define an agent as a persistent software entity dedicated to a **specific purpose**. 'Persistent' distinguishes agents from subroutines; agents have their **own ideas** about how to accomplish tasks, their **own agendas**. 'Special purpose' distinguishes them from entire multifunction applications; agents are typically much smaller.”*

Brustoloni 1991, Franklin 1995 *“**Autonomous agents** are systems capable of **autonomous, purposeful** action in the real world.“*

Wooldridge and Jennings 1995 *“An agent is a computer system that is situated in some environment, and that is capable of **autonomous action** in this environment in order to meet its design goals”*

FIPA 1998 *“An Agent is the **fundamental actor** in a domain. It combines one or more service capabilities into a unified and integrated execution model which can include access to external software, human users and communication facilities.”*

---

## Intelligente Agenten

Wooldridge and Jennings 1995 *“a hardware or (more usually) software-based computer system that enjoys the following properties:*

- autonomy**: *agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;*
  - social ability**: *agents interact with other agents (and possibly humans) via some kind of agent-communication language;*
  - reactivity**: *agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;*
  - pro-activeness**: *agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.“*
-

## „Sophisticated Agents“

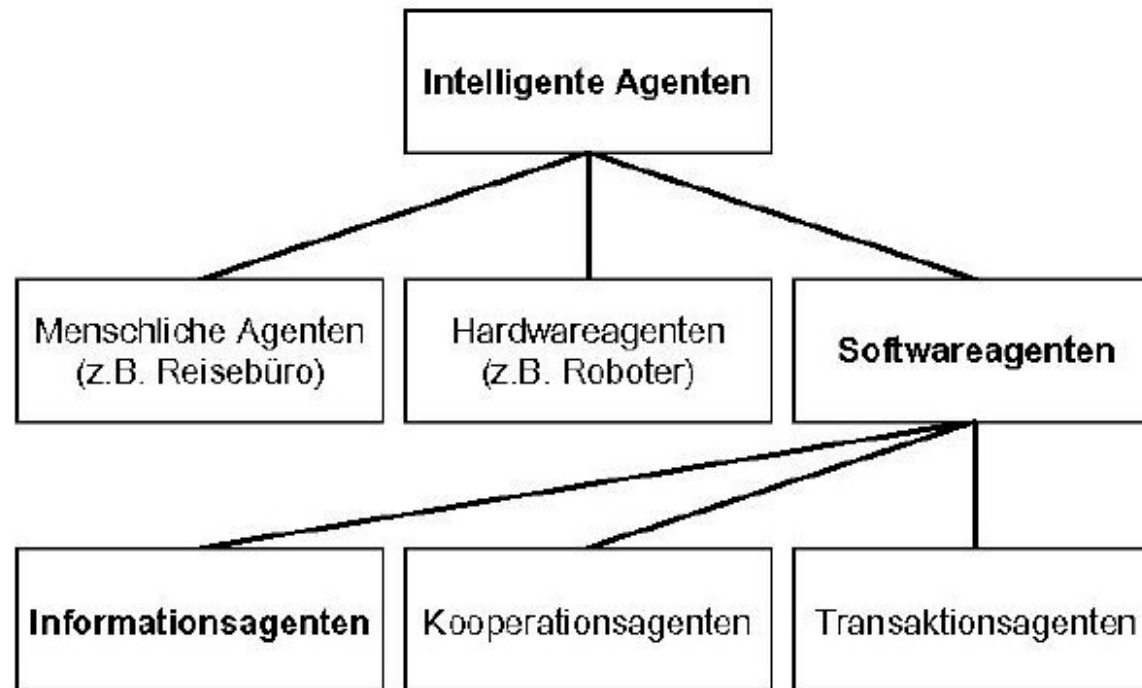
Lesser 2003

*“Agents are **autonomous**, heterogeneous, persistent, computing entities that have the ability to choose which task to perform and when to perform them.*

*Agents are rationally bounded, resource bounded, and have limited knowledge of other agents.*

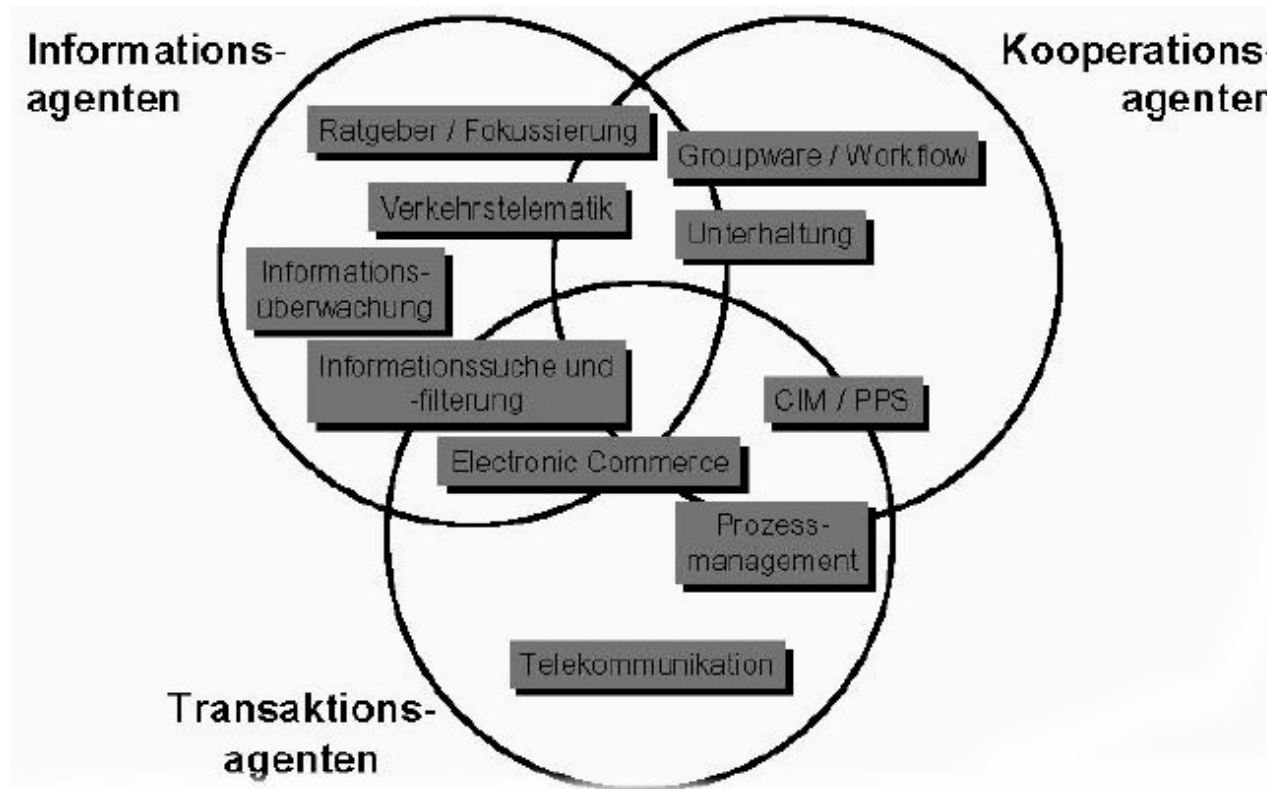
*Agents can perform tasks locally if they have sufficient resources and they may interact with other agents.”*

## Agentenkategorien (ein Versuch)...



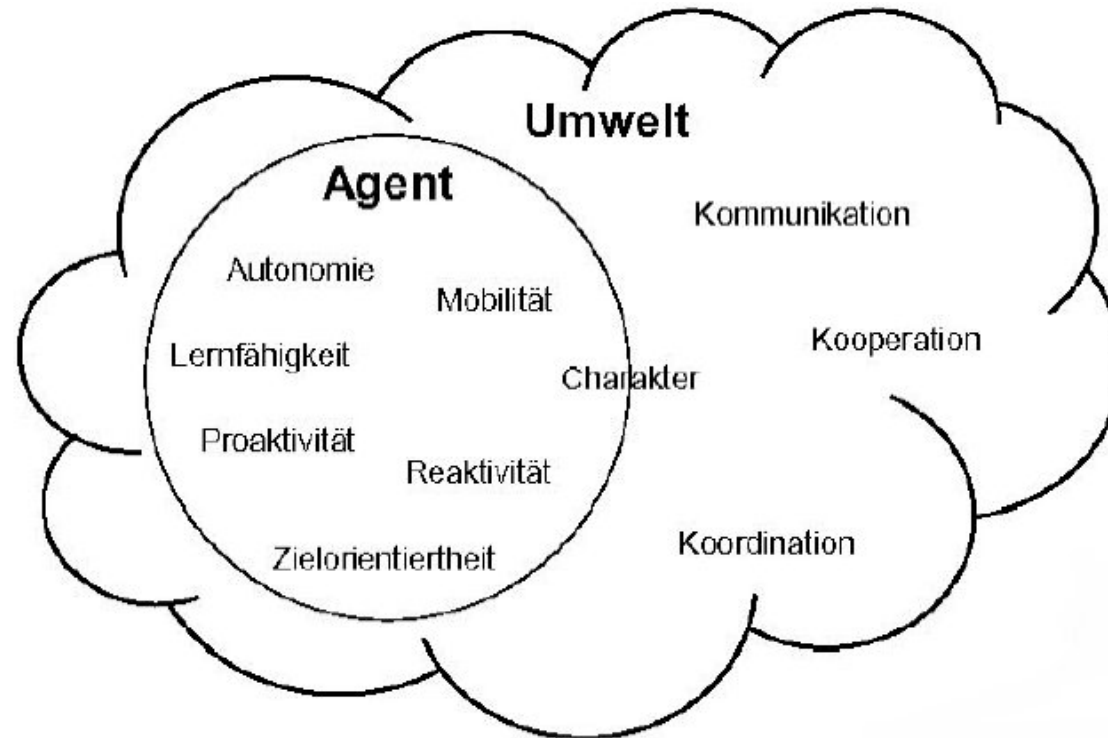
Quelle: W. Brenner et al., Intelligente Softwareagenten, 1998

# Anwendungsgebiete



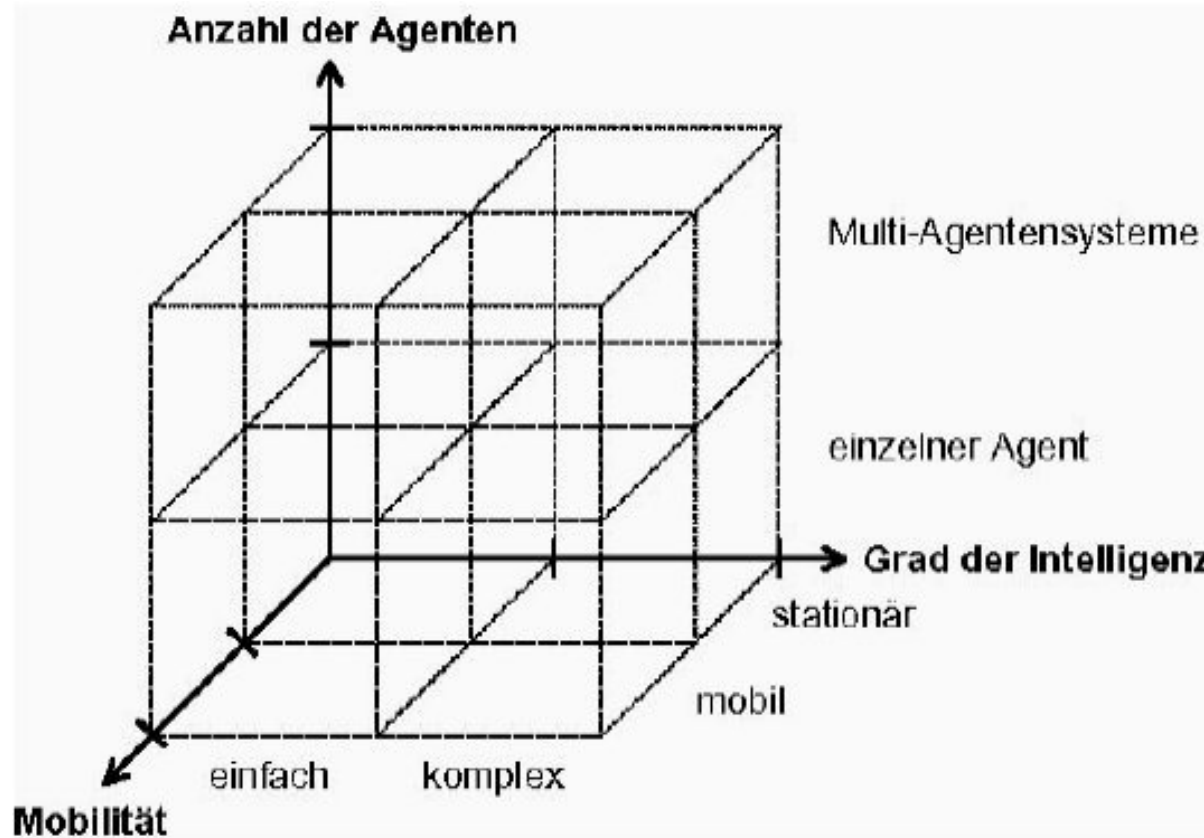
Quelle: W. Brenner et al., Intelligente Softwareagenten, 1998

## Charakteristika nach Brenner



Quelle: W. Brenner et al., Intelligente Softwareagenten, 1998

# Klassifikation nach Brenner



Quelle: W. Brenner et al., Intelligente Softwareagenten, 1998

## Definition: Was ist ein Multiagentensystem???

- *Multiagent Systems are computational systems in which several **semi-autonomous** agents interact or work together to perform some set of tasks or satisfy some set of goals.*
- *Agents that are homogenous or heterogeneous*
- *Agents having in common goals or goals that are distinct*
- *Agents that are humans or intelligent computational systems*
- *Not concerned with low-level parallelization or synchronization issues that are more the focus of distributed computing*



Zusammenarbeit zur Erreichung der eigenen Ziele!

*Quelle: Lesser 2003*

## Wichtige Unterscheidung von Agenten

- wohltätige Agenten
  - Das gesamte System mit allen Agenten gehört uns
  - Alle Agenten handeln somit in unserem Interesse
  - Problemlösen in Systemen mit ausschließlich wohltätigen Agenten heißt: Cooperative Distributed Problem Solving (CDPS)
  - Vereinfacht den Systementwurf
  
- egoistische Agenten
  - Agenten, die in eigenem Interesse agieren
  - Unsere eigenen Agenten machen nur einen Teil des Systems aus
  - Fremde und unbekannte Agenten im System vorhanden
  - Hohes Konfliktpotential
  - Erschwert den Systementwurf

## 2. Verteiltes Problemlösen (*Distributed Problem Solving*)

## Einführung ins Verteilte Problemlösen:

Unterkategorie der Künstlichen Intelligenz

Ziel: Zusammenarbeit zw. Agenten zum Lösen v. Verteilten Problemen

Vorbedingungen:

Kohärenz (Agenten wollen zusammenarbeiten)

und

Kompetenz (Agenten sind fähig zusammen zu arbeiten)

Annahme: Kohärenz ist gegeben, da Agenten für die Zusammenarbeit entwickelt wurden



**Konzentration auf Kompetenz!!!**

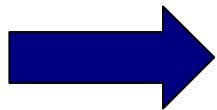
# Einführung I

Distributed Problem Solving vs. Cooperative Distributed Problem Solving

egoistische Agenten vs. wohltätige Agenten

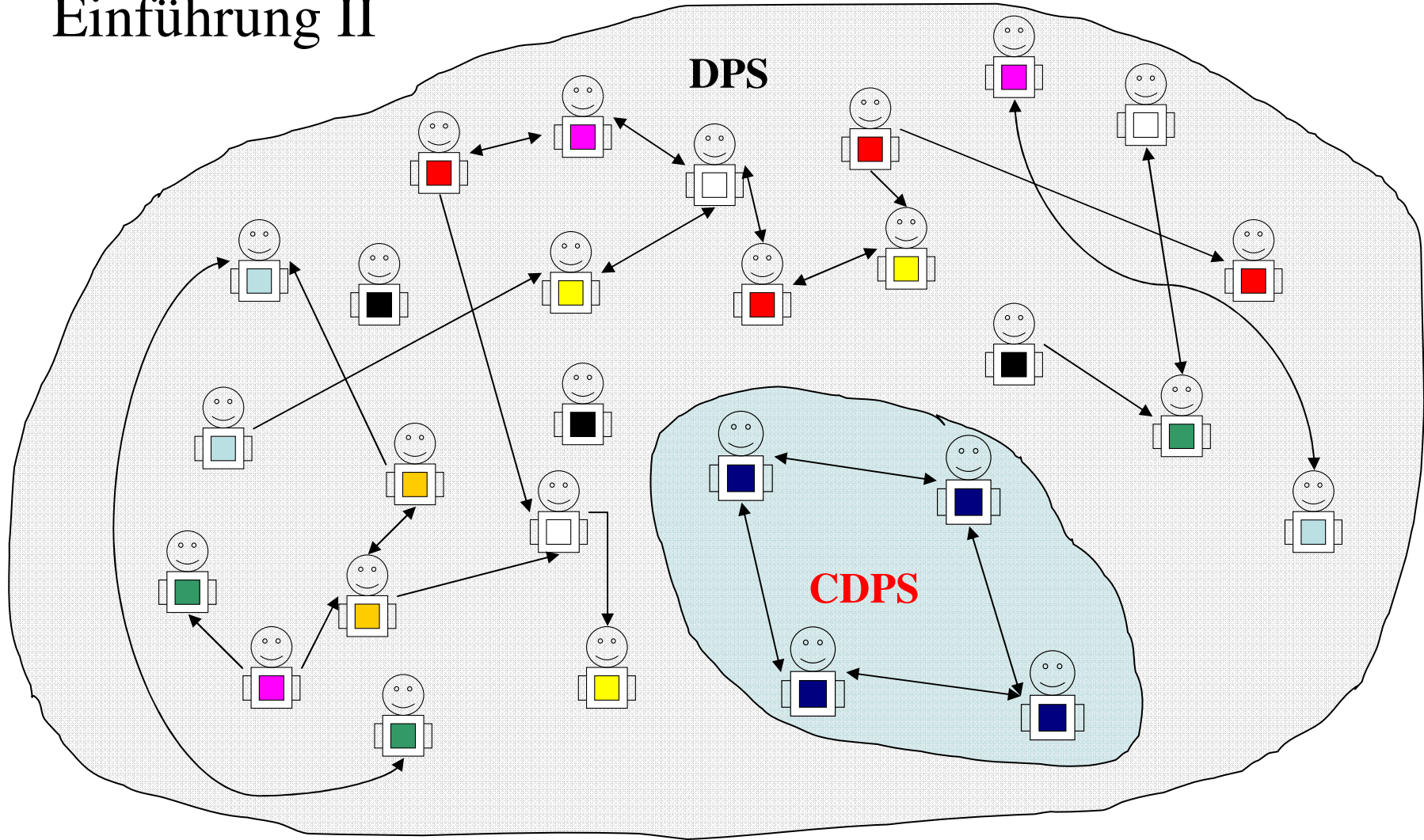
fremde / unbekannte Entwicklung vs. eigene Entwicklung

Offenes System vs. Eigenes System



Zielsetzung des Systems ist entscheidend!

# Einführung II



## Distributed Problem Solving

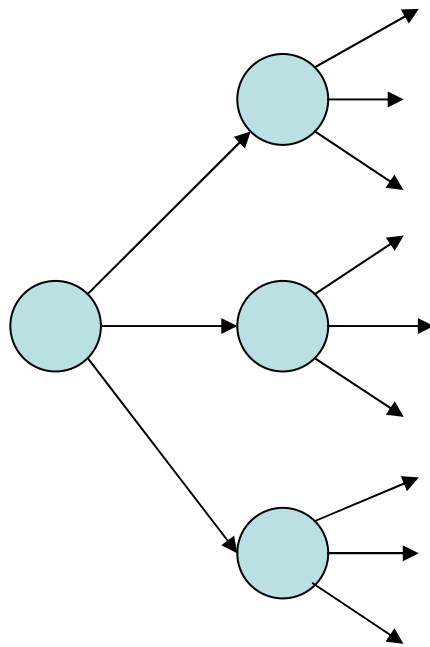
- Kriterien zur Bewertung von MAS
  - Kohärenz (Wie gut arbeiten die Agenten zusammen?)
    - Ergebnisqualität
    - Effizienz des Ressourceneinsatzes
    - System Performance und Fehlerminimierung
  - Koordination (Grad der Vermeidung von zusätzlicher Aktivität)
    - Konflikte zwischen Agenten kosten Zeit und Ressourcen
    - Behinderung der Agenten untereinander
    - Keine übertriebene Kommunikation nötig
    - Agenten können das Verhalten anderer vorhersehen
    - Evtl. besitzen Agenten Informationen über Skills anderer Agenten.

## (Cooperative) Distributed Problem Solving

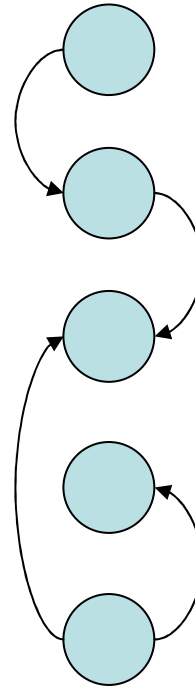
### Kernpunkte:

- Problemzerlegung und -verteilung an Agenten
- Effektive Generierung einer Problemlösung aus diversen Teilergebnissen
- Optimierung des Kohärenzverhaltens der Agenten
- Koordination der Agenten zur Vermeidung von überflüssiger Interaktion und Maximierung der Effektivität

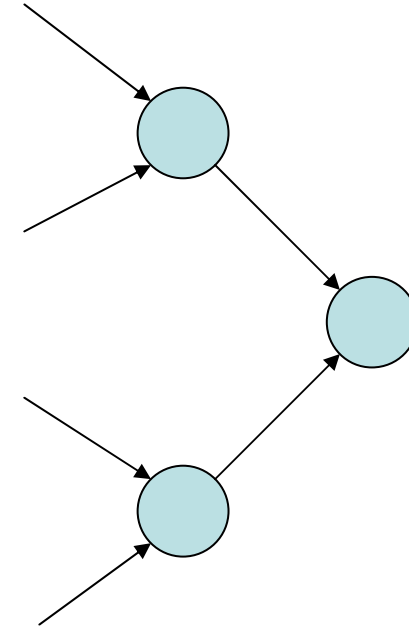
# Ablauf des (Cooperative) Distributed Problem Solving



1. Problemzerlegung



2. Lösung von Teilproblemen



3. Ergebnissynthese

Quelle: M. Wooldridge, An Introduction to Multiagentsystems, 2002

## Wie können Agenten interagieren?

Aufgaben (-ver-) teilung (*Task Sharing*)

und

Ergebnispartizipation (*Result Sharing*)

Ablauf:

1. Aufgabenzerlegung (*Task decomposition*)
2. Aufgabenzuweisung (*Task allocation*)
3. Aufgabenvollendung (*Task accomplishment*)
4. Ergebnissynthese (*Result synthesis*)

## Aufgabenzerlegung:

- Aufgaben werden in kleinere (leichtere) Teilaufgaben zerlegt
- Primäres Ziel: Unabhängigkeit von Teilaufgaben
- Sekundär: Minimierung von...
  - Kommunikation
  - Koordination
  - Daten
  - Ressourcen
- Probleme:
  - Hohe Komplexität → Zerlegung durch Entwickler
  - Aktuell kein bekannter Algorithmus vorhanden

## Exkurs: Systemunterscheidung

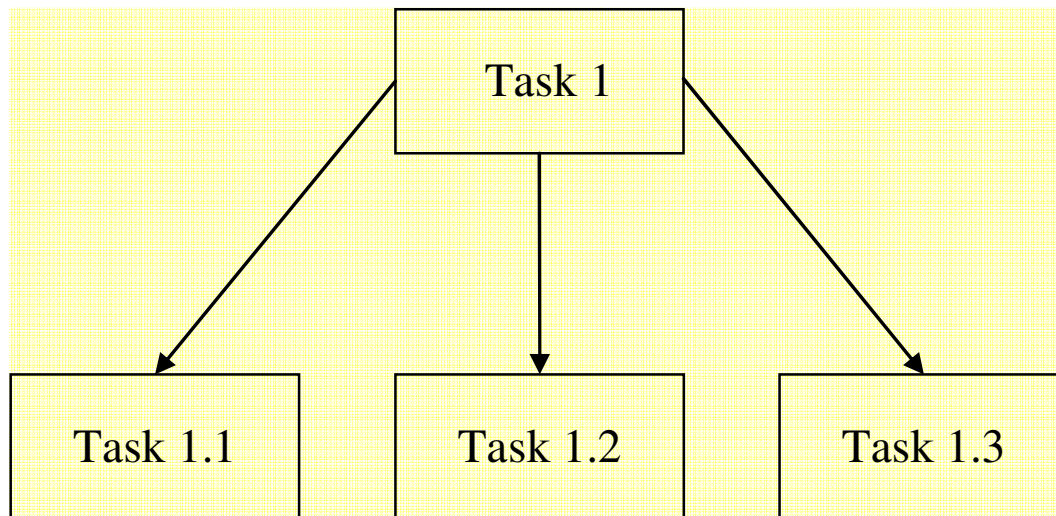
- Homogene Systeme
  - Agenten sind gleich
  - Besitzen identische Fähigkeiten und Ressourcen
  - Einfache Aufgabenverteilung
- Heterogene Systeme
  - Agenten sind verschieden
  - Besitzen unterschiedliche Fähigkeiten und Ressourcen
  - Aufgabenverteilung muss den Agenten angepasst sein
  - Teilweise hohe Komplexität



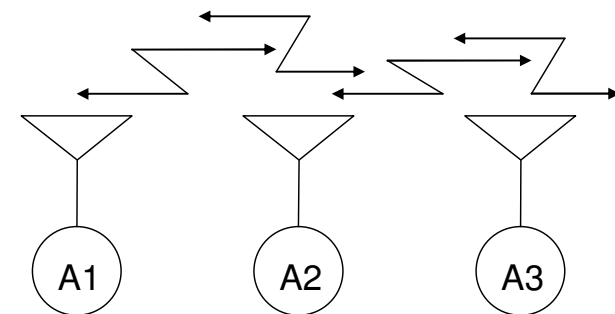
Konzentration auf heterogene Systeme!

## Aufgabenverteilung und Ergebnispartizipation:

*Task Sharing:*

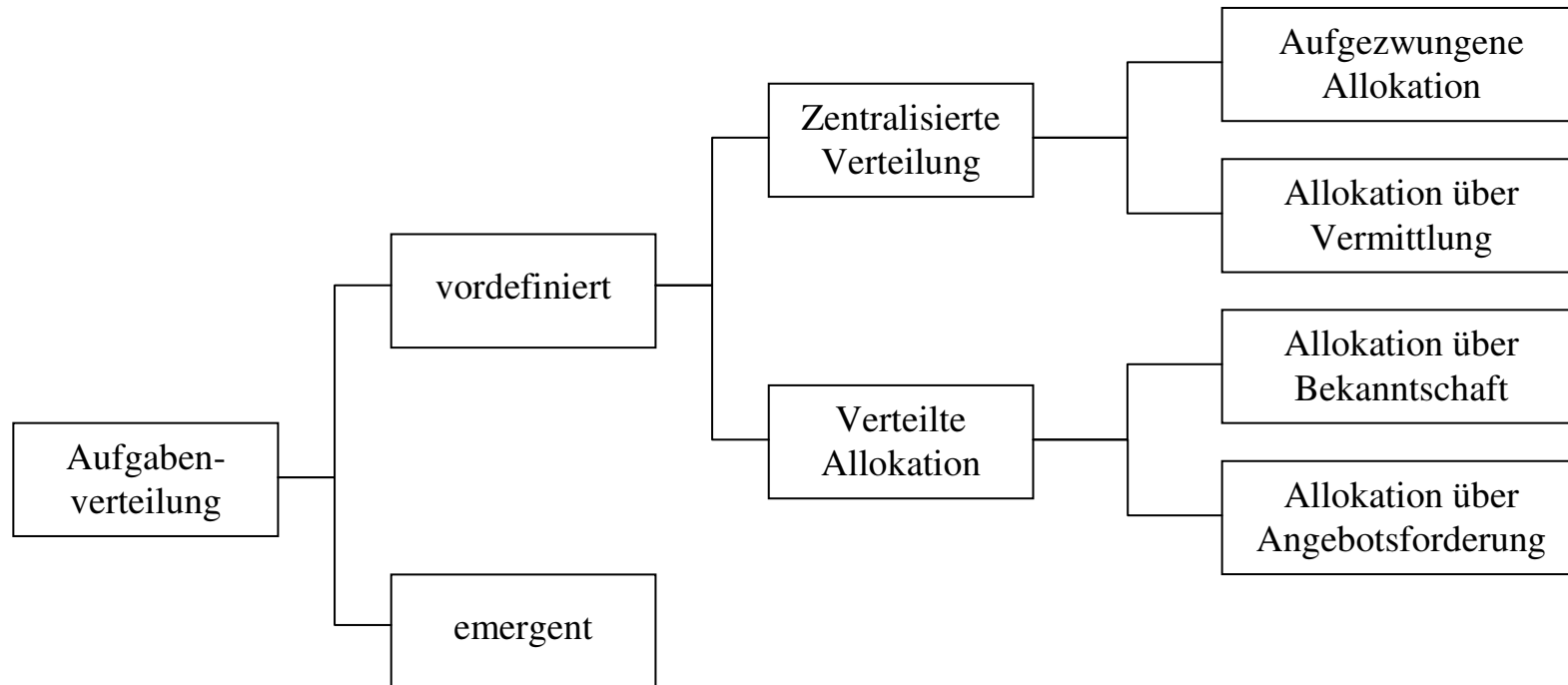


*Result Sharing:*



Quelle: M. Wooldridge, An Introduction to Multiagentsystems, 2002

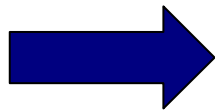
# Aufgabenverteilung im Überblick:



Quelle: J. Ferber, Multiagentsysteme, 2001

## Agentenzustände bei der Aufgabenverteilung:

- Zwei mögliche Zustände
  - Client (Auftraggeber)
  - Server (Dienstanbieter)
- Agenten können ihren Zustand wechseln
- Trader / Händler sind spezielle Clients
- Agenten können sowohl Client als auch Server gleichzeitig sein
  - Beispiel: Delegation oder Subkontraktion (später)
- Ziel der Aufgabenverteilung (*Task Sharing*):

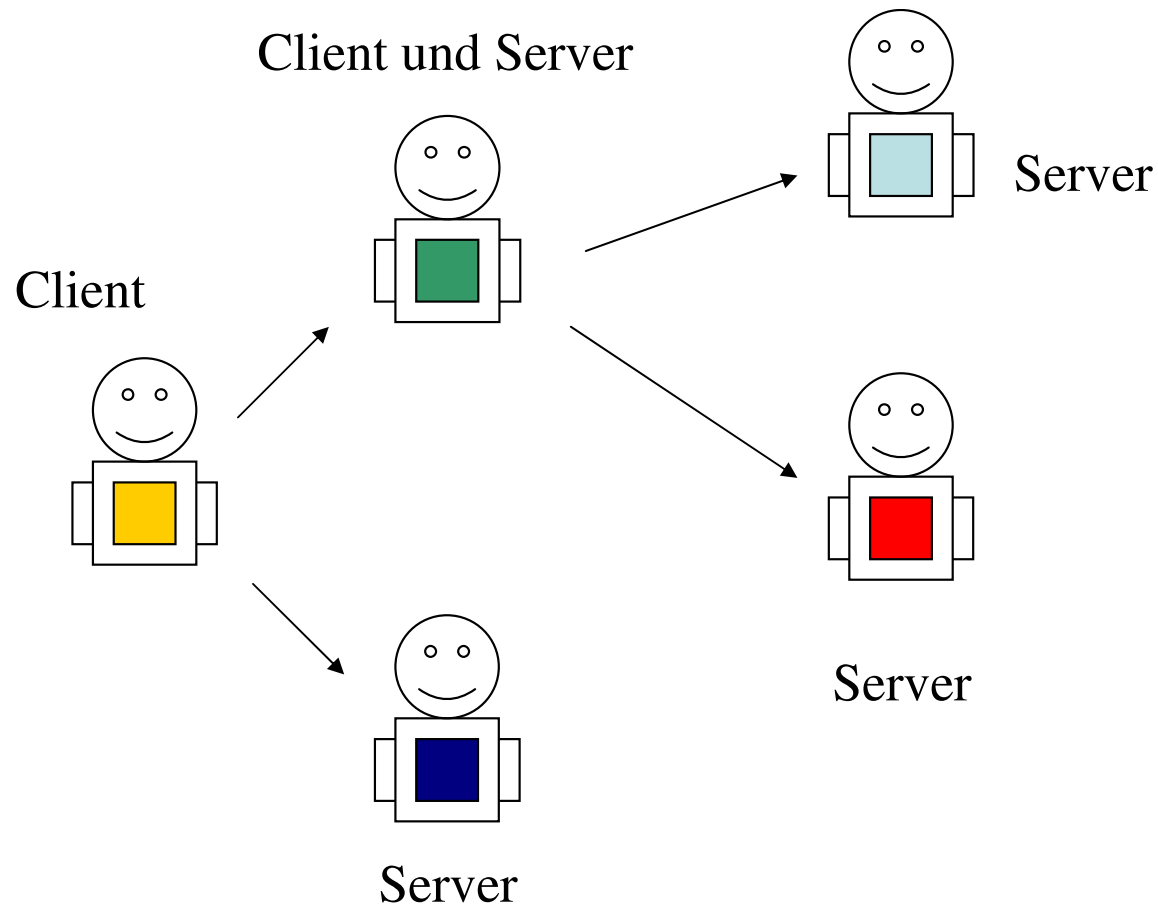


Die Zusammenarbeit von Clients und Server zu ermöglichen!!!

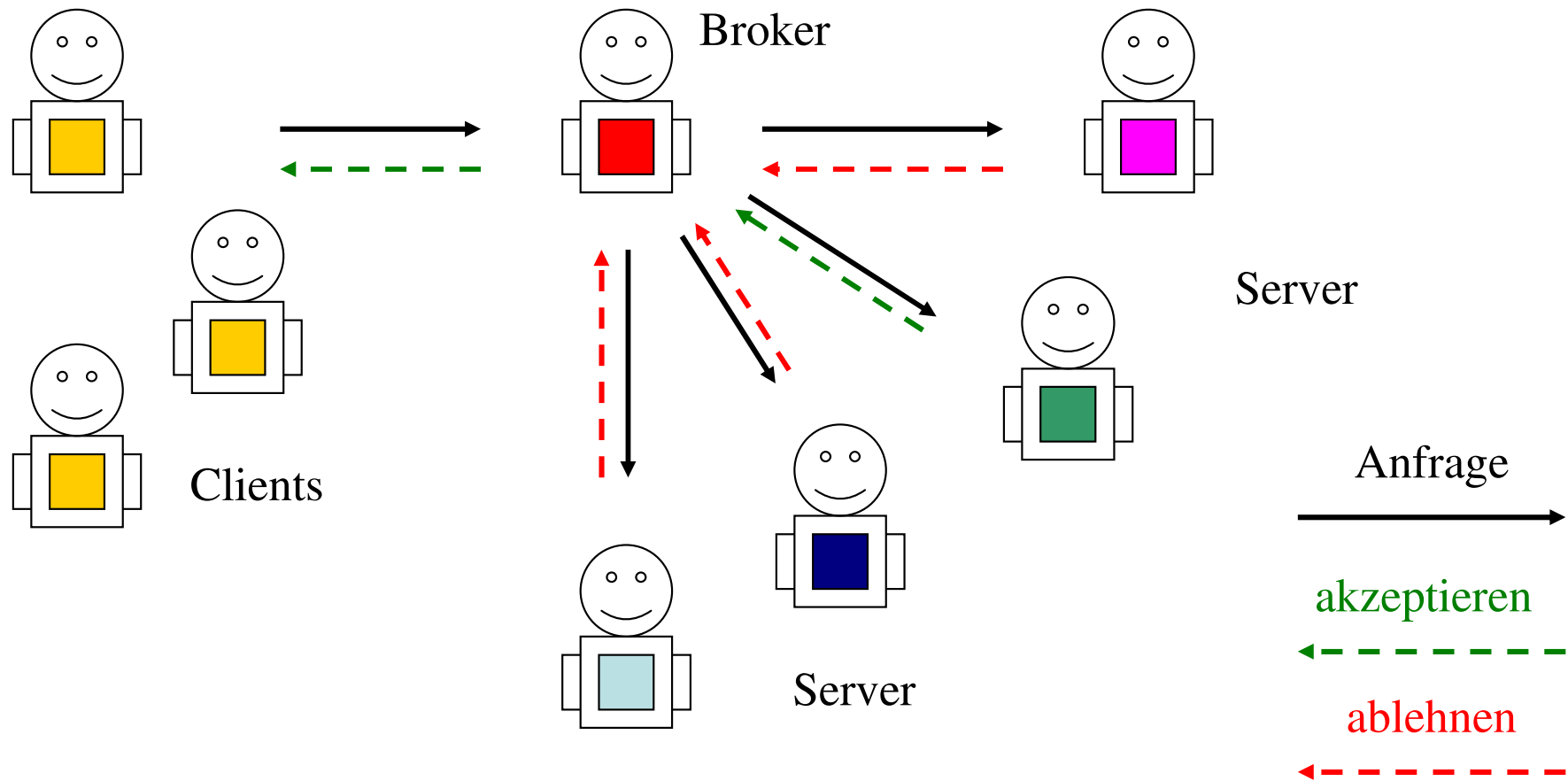
## Zentrale Aufgabenverteilung:

- Hierarchisch
  - Übergeordneter Agent vergibt Aufgabe an untergeordneten
    - feste bzw. (vor-) definierte Allokation
  - Verwendung nur in starren Organisationsformen
- Egalitär
  - Allokation über Vermittlung
    - Spezielle Agenten fungieren als Broker
    - Broker führen Anbieter und Nachfrager zusammen
  - Zentrale Aufgabenverteilung auch in variablen Organisationen möglich

# Beispiel: Hierarchische Aufgabenverteilung



# Beispiel: Allokation über Vermittlung



## Verteilte Aufgabenallokation

- Bekanntschaftsnetzwerke
  - Vorbedingung: Kohärenz
    - Direkte Allokation
    - Allokation durch Delegation
- Kontraktnetz-Ansatz
  - Markt-Mechanismus

## Aufgabenverteilung über Bekanntschaftsnetzwerke

- Annahme: Jeder Agent besitzt eine Tabelle mit den Fähigkeiten aller ihm bekannten Agenten:

$$\{C_1:[A_{11}, \dots, A_{1k}], \dots, C_n:[A_{n1}, \dots, A_{nk}]\}$$

Wobei:  $C_i$  = Fähigkeiten  
 $A_{ij}$  = Agenten  
( $T_i$  = Aufgaben)

Beispieltabelle:

|       | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|-------|-------|-------|-------|-------|
| $C_1$ | 0     | 1     | 1     | 0     |
| $C_2$ | 0     | 0     | 1     | 0     |
| $C_3$ | 1     | 0     | 0     | 1     |

Agent  $A_2$  besitzt hier also die Fähigkeit  $C_1$

Quelle: J. Ferber, Multiagentensysteme, 2001

## Aufgabenverteilung über Bekanntschaftsnetzwerke

- Annahmen:
  - Ein Agent kennt nicht alle Fähigkeiten aller anderen Agenten
  - Korrektheit der Bekanntschaftstabellen (kohärent)
  - Vollständige Vernetzung
  - Tabellen sind a priori gegeben
- Darstellung als Graphen
  - Agenten sind Knoten
  - Skills werden als Pfeile dargestellt
- Verfahren der Allokation
  - Direkte Allokation (Aufgabendelegation verboten)
  - Allokation durch Delegation

# Beispiel: Bekanntnetzwerk

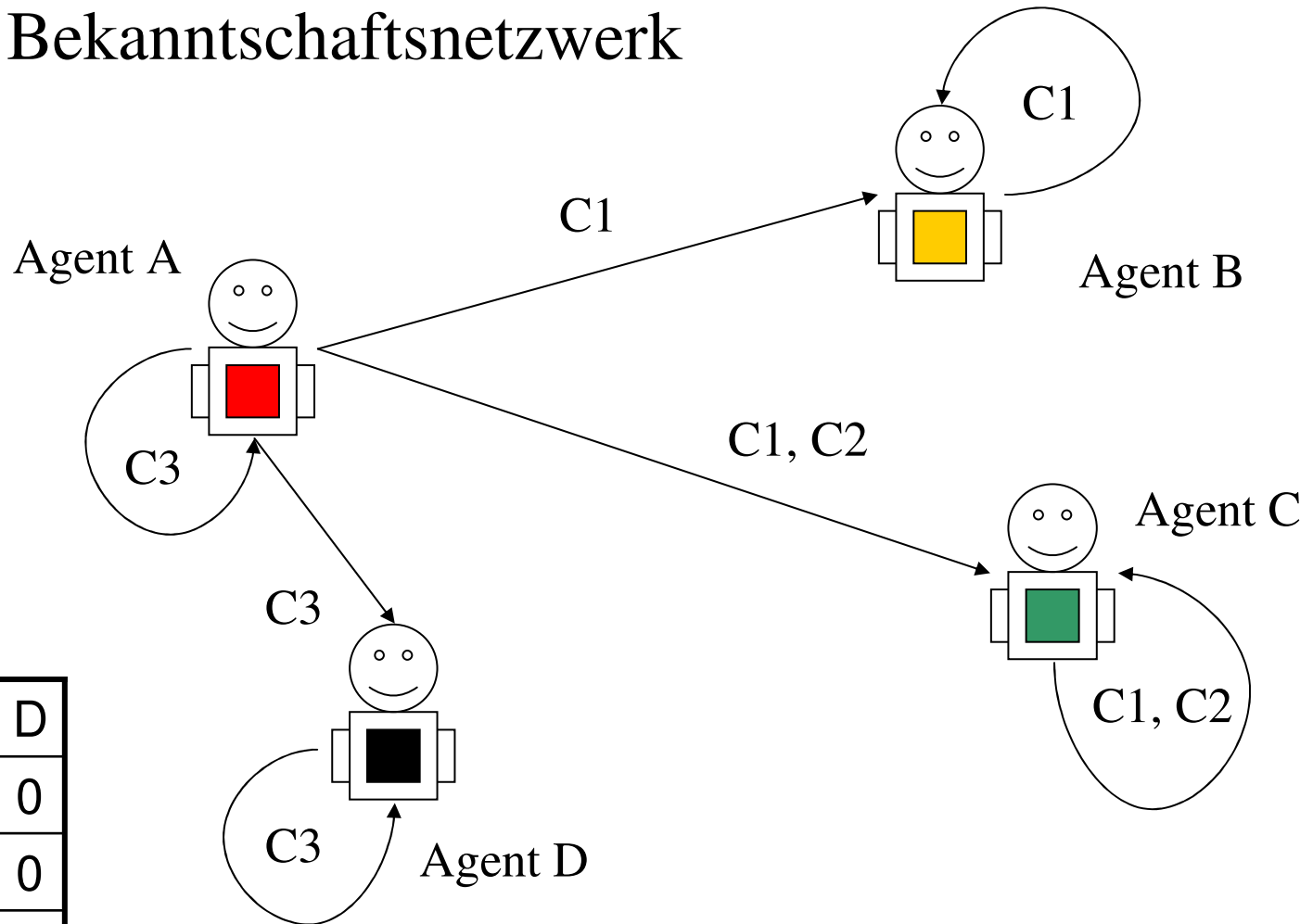
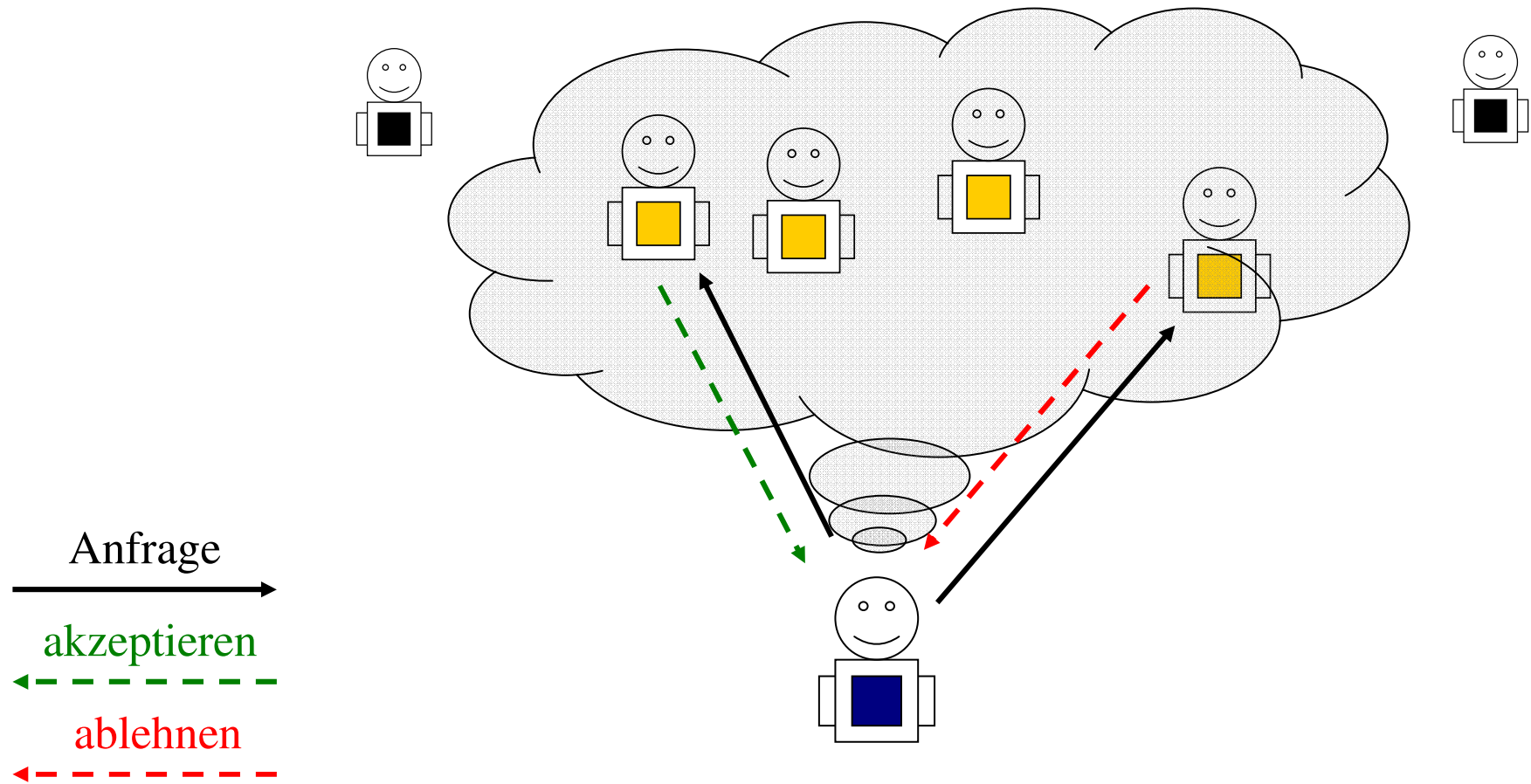


Tabelle von Agent A:

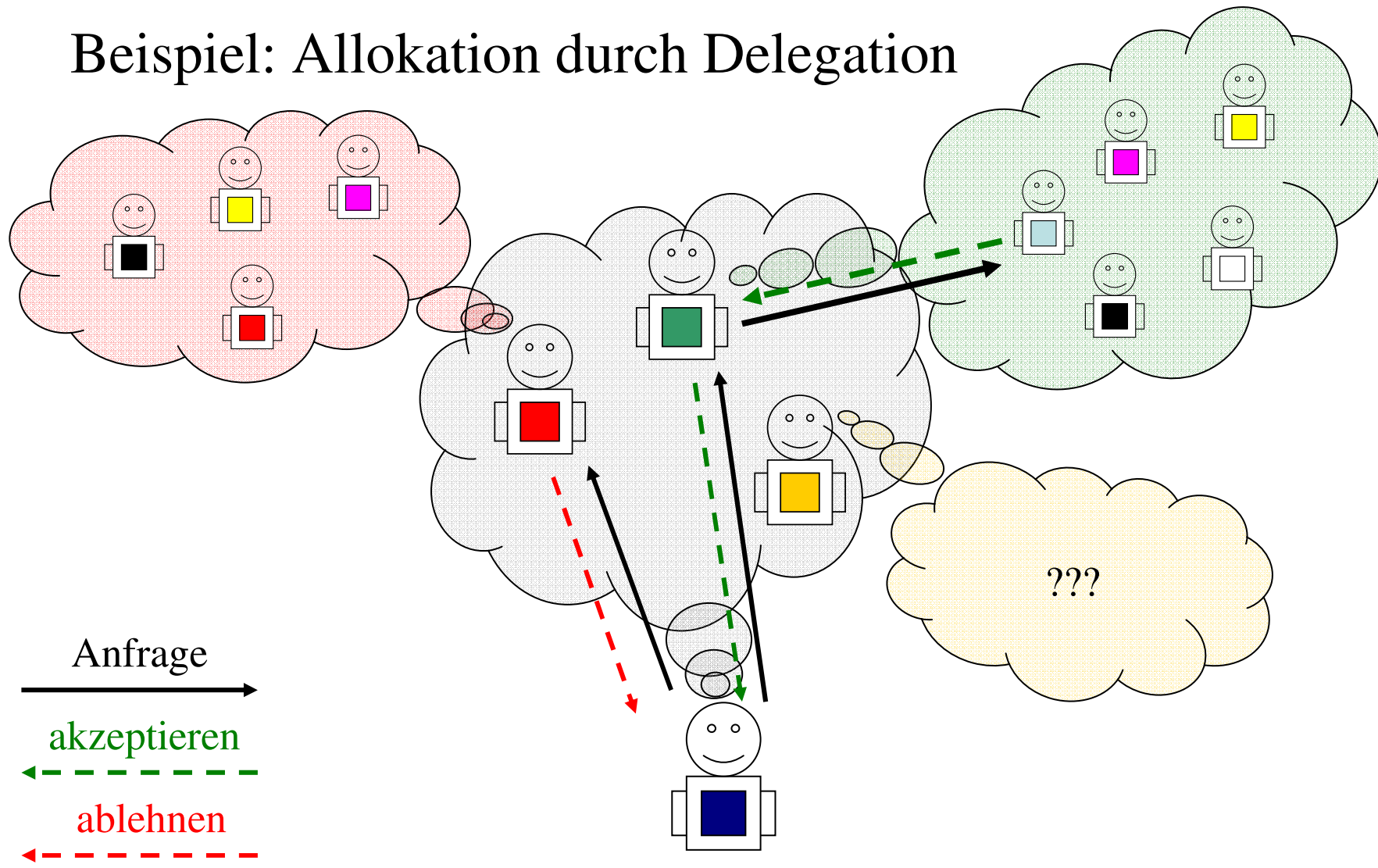
|                | A | B | C | D |
|----------------|---|---|---|---|
| C <sub>1</sub> | 0 | 1 | 1 | 0 |
| C <sub>2</sub> | 0 | 0 | 1 | 0 |
| C <sub>3</sub> | 1 | 0 | 0 | 1 |

Quelle: J. Ferber, Multiagentensysteme, 2001

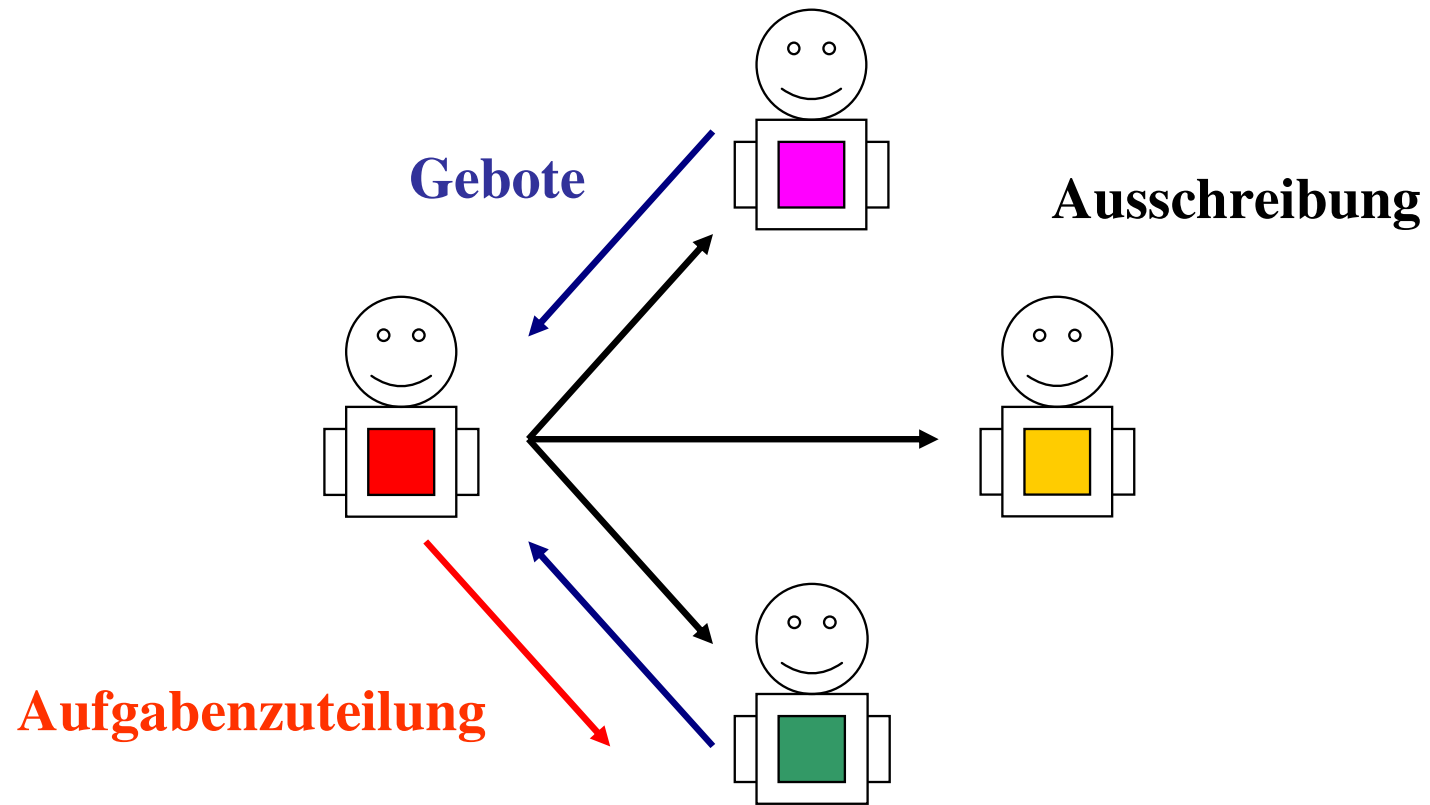
# Beispiel: Direkte Allokation



# Beispiel: Allokation durch Delegation



# Beispiel: Contract Net

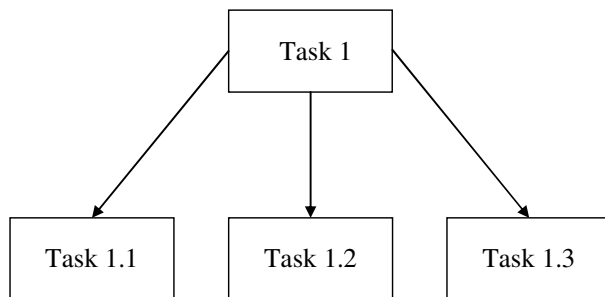


## Aufgabenverteilung im Überblick

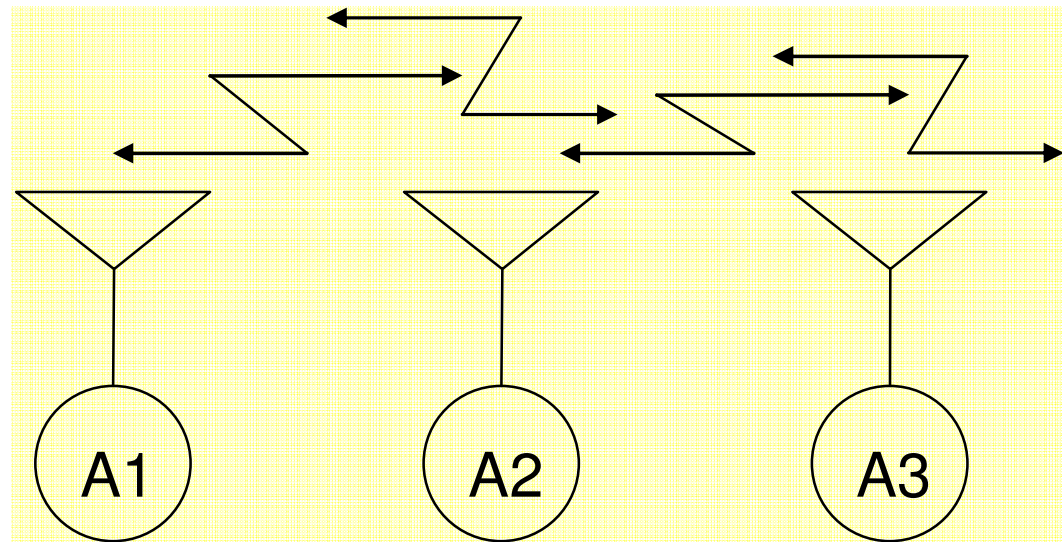
- Zentralisierte Verteilung
  - Einsatz eines Traders
    - Vorteile: Kohärenz gegeben
    - Nachteile: Fehleranfällig, Ressourcen- und Kapazitätsgrenzen
  
- Verteilte Allokation
  - Bekanntschaftsnetzwerke
    - Vorteile: Geringe Fehlerquote, hohe Performance
    - Nachteile: Kohärenz, Skalierbarkeit
  
  - Contract Net
    - Vorteile: Relativ einfach zu implementieren, Flexibilität
    - Nachteile: Hohes Kommunikationsaufkommen

# Aufgabenverteilung und Ergebnispartizipation:

*Task Sharing:*



*Result Sharing:*



Quelle: M. Wooldridge, An Introduction to Multiagentsystems, 2002

## Merkmale beim Result Sharing

- Vertrauen (*confidence*)
  - Eigene Ergebnisse werden mit fremden Ergebnissen der gleichen Aufgabe verglichen
- Vollständigkeit (*completeness*)
  - Kombination von Teilergebnissen führt zu größeren (Teil-)Ergebnissen
- Geschwindigkeit (*timeliness*)
  - Paralleles Problemlösen
- Genauigkeit (*precision*)
  - Verbesserung der eigenen Ergebnisse durch Informationsaustausch

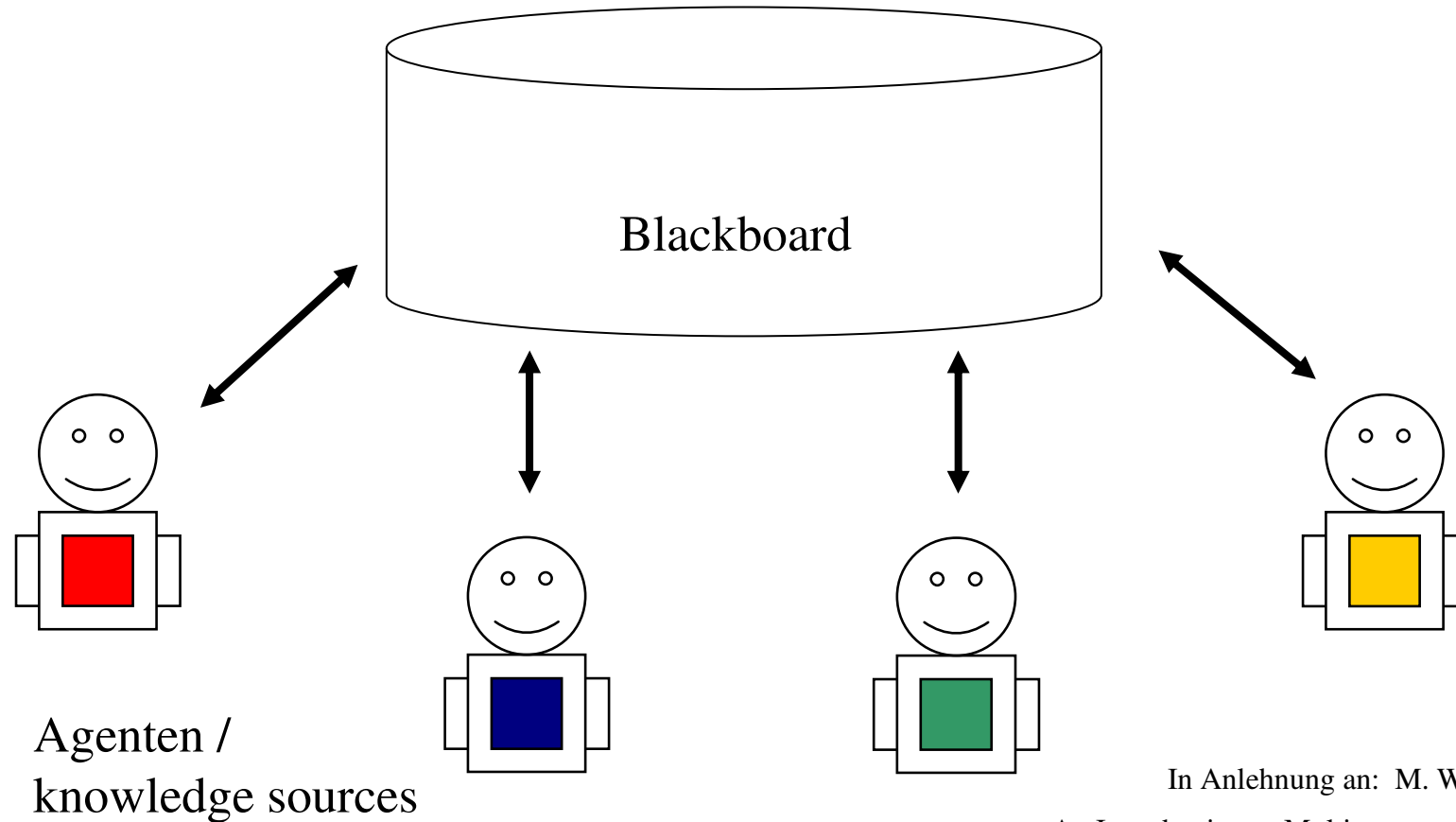
## Functionally Accurate Cooperation

- Kollektives Problemlösen
- Agenten müssen miteinander kooperieren um eigene Teilaufgaben zu lösen
- Iterativer Informationsaustausch
- Fehlinformationen
- Teilergebnisse sind nur vorläufig als Ergebnisse zu betrachten
- Beeinflusst werden:
  - Vertrauen, Genauigkeit und Vollständigkeit
- Problem
  - Problemlösung durch einen Agenten theoretisch möglich
    - Verschwendung von Ressourcen
  - Distraction

## Shared Repositories

- Ziel: Reduzierung des Kommunikationsbedarf und Minimierung überflüssiger Nachrichten
- Konzentration von vorläufigen Teilergebnisse in Shared Repositories
- Blackboard-Ansatz:
  - Verteilte Datenstruktur,
  - Agenten stellen Teilergebnisse im Blackboard zur Verfügung
  - Verschiedene Agenten können Daten im BB lesen und schreiben
  - Agenten lösen (Teil-) Probleme (auf dem BB), wenn sie dazu in der Lage sind
  - Blackboard kann hierarchisch strukturiert sein
  - Keine Parallelität → u.U. mäßige Performance

# Blackboard-Ansatz

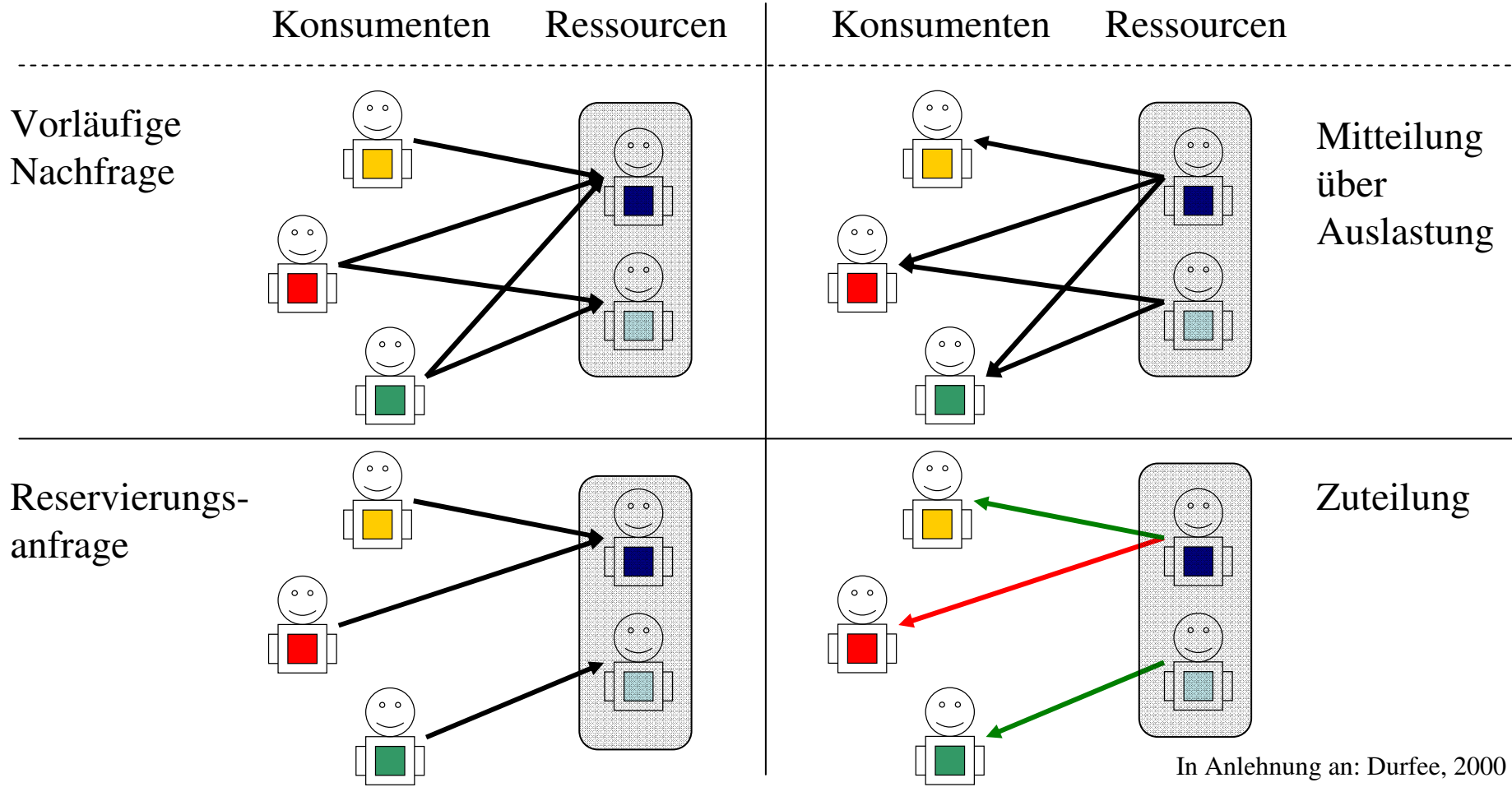


In Anlehnung an: M. Wooldridge,  
An Introduction to Multiagentsystems, 2002

## Distributed Constrained Heuristic Search

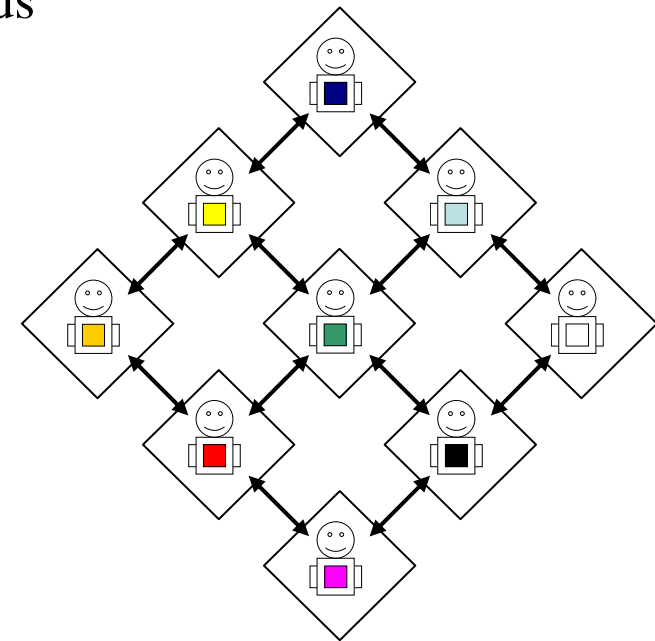
- Einsatz in *Distributed Resource Allocation Problems*
- Ziel: Zusammenbringen von Agenten und Ressourcen
- Agenten repräsentieren Konsumenten
- Ressourcen sind Agenten die Kapazitäten / Wissen zur Verfügung stellen
- Konsumenten „suchen“ Ressourcen für ihre (Teil-) Probleme
- Ablauf:
  1. Konsumenten: Vorläufige Nachfrage
  2. Ressourcen: Berechnung der eigenen Nachfrage (Nachfrageprofile) und Mitteilung über Auslastung
  3. Konsumenten: Reservierungsanfrage
  4. Ressourcen: Zuteilung und Update der Nachfrageprofile

# Distributed Constrained Heuristic Search



## Organizational Structuring

- Vorbedingung: Kohärenz
- Ziel: Reduzierung des Kommunikationsaufwands
- Jeder Agent hat seine ganz spezielle Aufgabe
- Agenten kennen die Aufgaben(-felder) ihrer Nachbarn
- Organisationsstruktur entscheidet, welche Agenten über was informiert werden



Voneinander getrennte und geografisch verteilte Zellen

## Result Sharing im Überblick

- Functionally Accurate Cooperation
  - Vorteile: Genauigkeit, Vollständigkeit
  - Nachteile: Distraction
- Shared Repositories
  - Vorteile: Begrenzte Kommunikation
  - Nachteile: Fehleranfällig, Ressourcen- und Kapazitätsgrenzen
- Distributed Constrained Heuristic Search
  - Vorteile: Sehr effiziente Suche
  - Nachteile: konträrer Ansatz
- Organizational Structuring
  - Vorteile: Reduziertes Kommunikationsaufkommen,
  - Nachteile: Ressourcenentwicklung u. U. problematisch

## Inkonsistenz

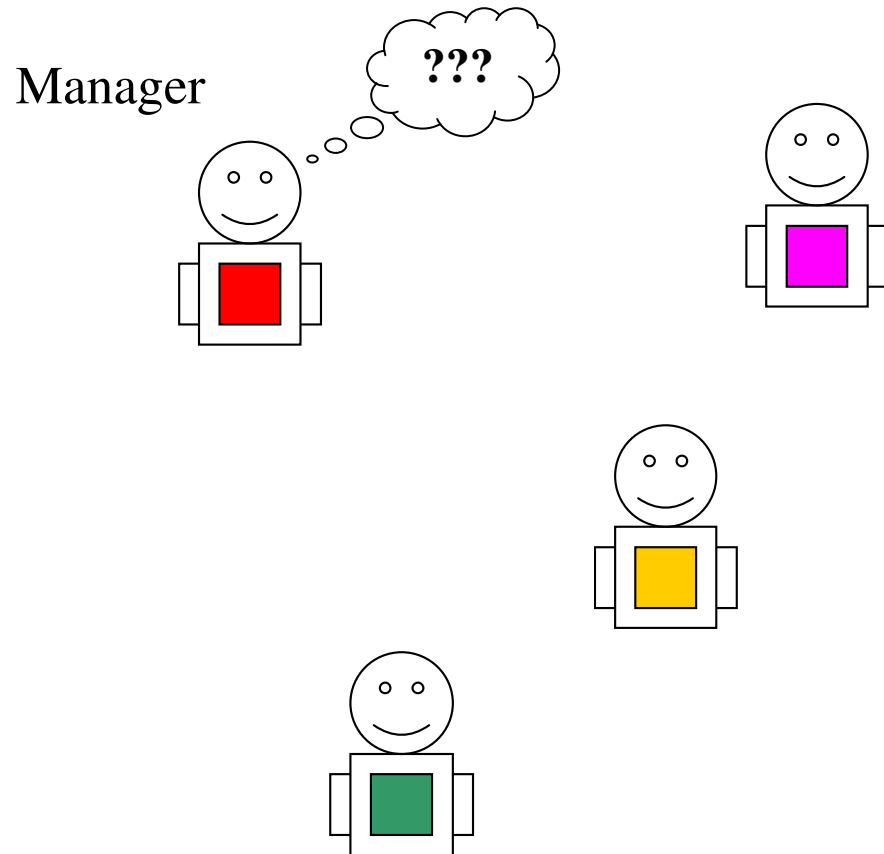
- ...in Bezug auf Informationen (Annahmen), Interessen und Ziele
- Autonomie der Agenten führt zu inkonsistenten Zielen und Interessen
- Inkonsistente Informationen (Annahmen):
  - Versch. „Sichtweisen“ der Agenten (Blickwinkel)
  - Umwelteigenschaften (Bild der Umwelt...)
  - Informationsquellen
  - Sensoren
- Umgang mit Inkonsistenz:
  - Ignorierung (Contract Net)
  - Verhandlungen
    - Nachteil: Enormer Kommunikationsaufwand, theoretisch möglich
      - praktisch kaum realisierbar
  - Minimierung während des Systementwurfs

# 3. Contract Net-Protokoll im Detail

## Einführung

- Erster Entwurf von Reid G. Smith und Randall Davis, Stanford (1978)
  - Protokoll zur verteilten (Aufgaben-) Allokation
  - Ablauf nach Wooldridge:
    - *Recognition*
    - *Announcement*
    - *Bidding*
    - *Awarding*
- Elementare Schritte
- Dynamisches Protokoll
  - Sehr weit verbreitet
  - Einfacher „Ansatz“

# Das Contract Net-Protokoll im Detail I

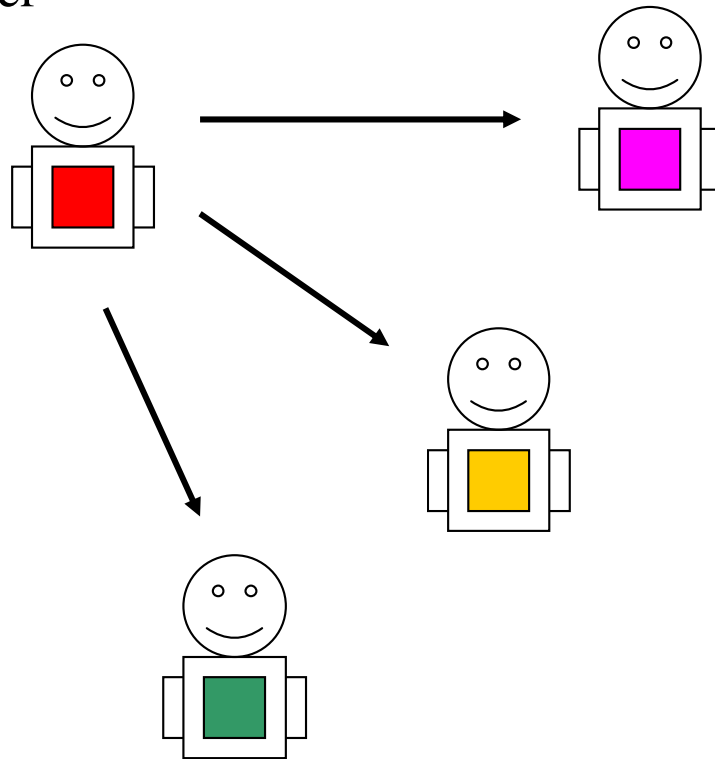


Agent „rot“ hat ein Problem...

In Anlehnung an: M. Wooldridge,  
An Introduction to Multiagentsystems, 2002

# Das Contract Net-Protokoll im Detail II

Manager



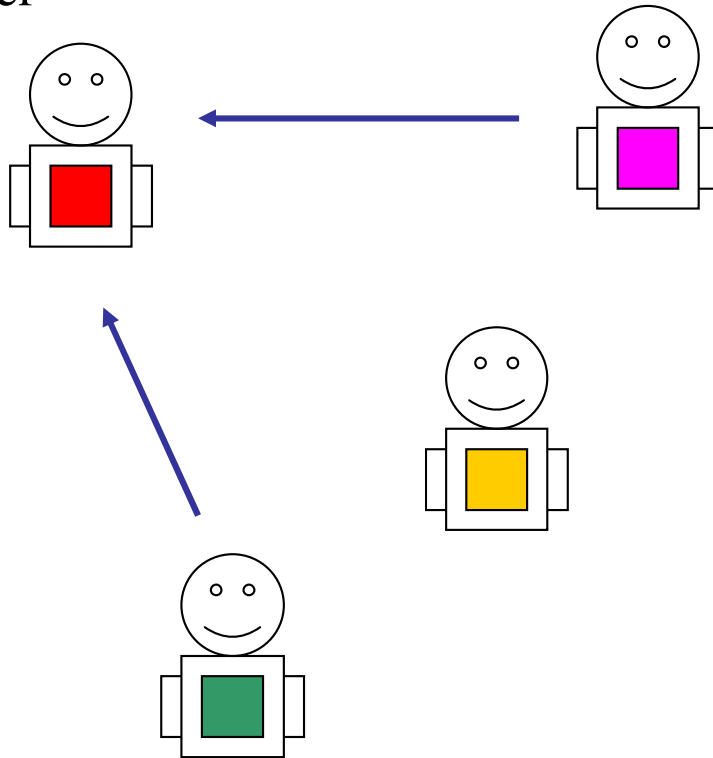
Das Problem / die Aufgabe wird  
ausgeschrieben;  
Eine Aufgabenspezifikation wird  
verschickt

In Anlehnung an: M. Wooldridge,  
An Introduction to Multiagentsystems, 2002

Versandte Nachrichten: **3**

# Das Contract Net-Protokoll im Detail III

Manager

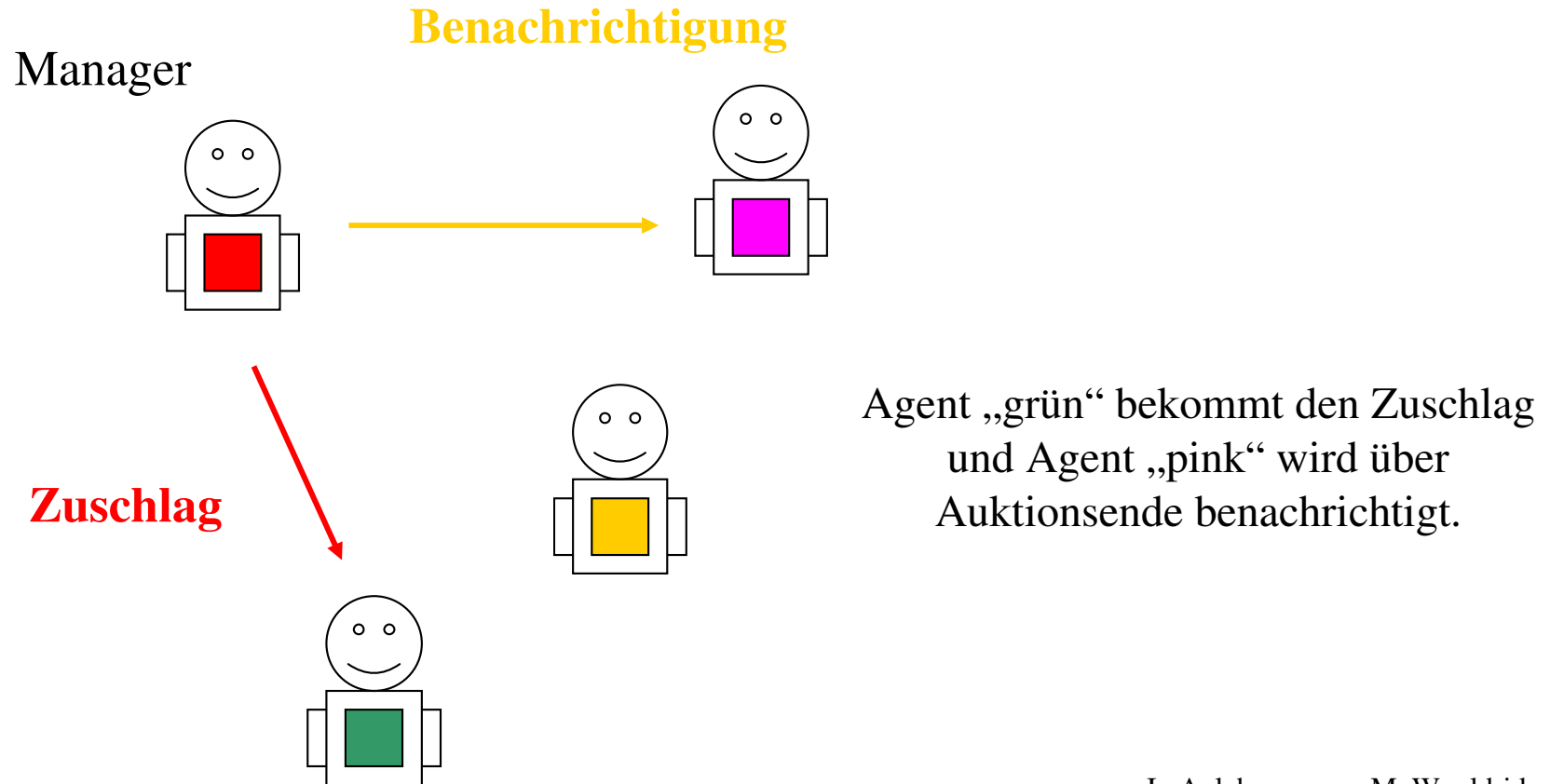


Die Agenten bieten, falls sie diese Aufgabe lösen können / möchten

Versandte Nachrichten: 3 + 2 = 5

In Anlehnung an: M. Wooldridge,  
An Introduction to Multiagentsystems, 2002

# Das Contract Net-Protokoll im Detail IV

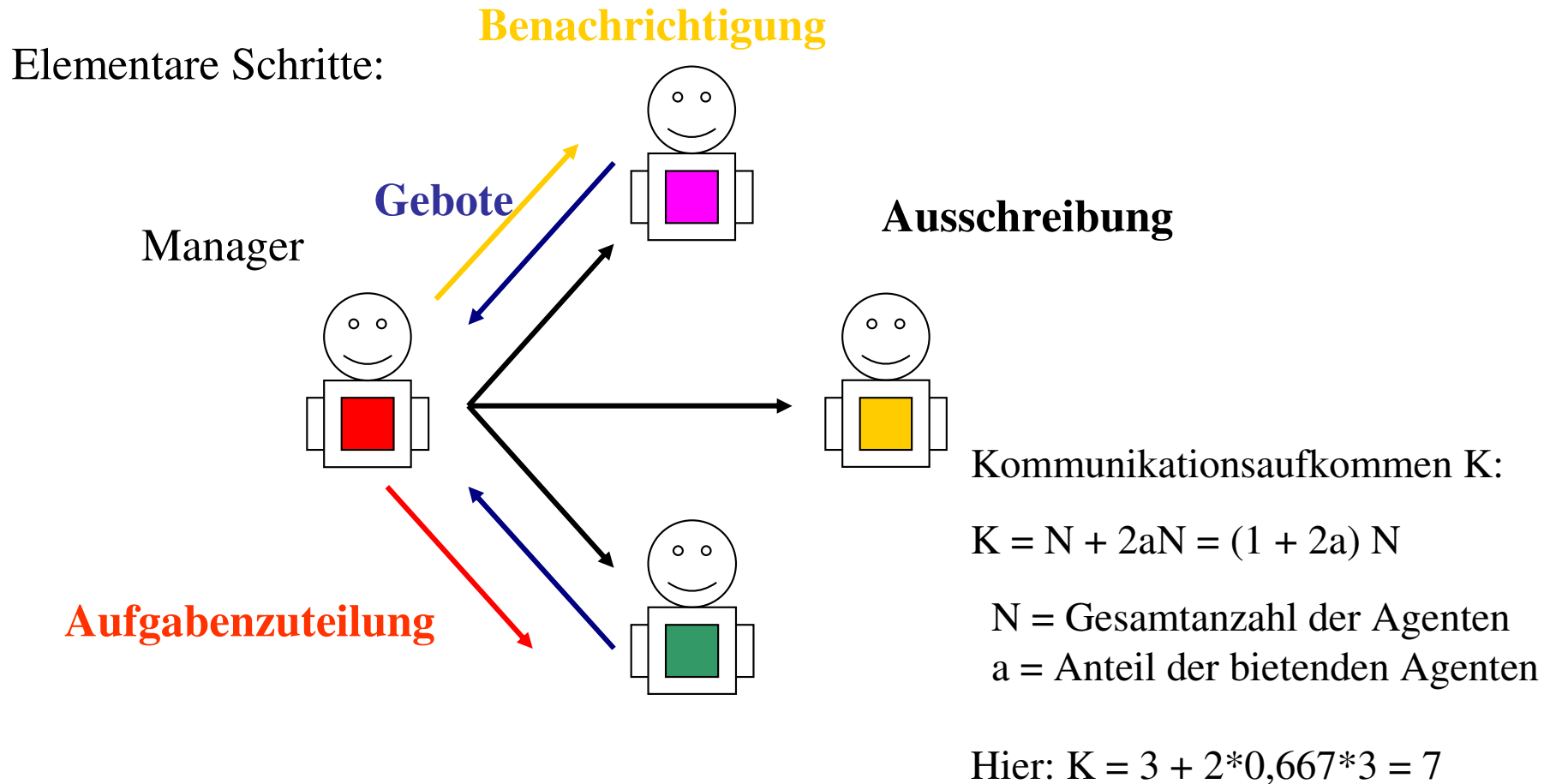


Agent „grün“ bekommt den Zuschlag  
und Agent „pink“ wird über  
Auktionsende benachrichtigt.

In Anlehnung an: M. Wooldridge,  
An Introduction to Multiagentsystems, 2002

Versandte Nachrichten: 3 + 2 + 2 = 7

# Das Contract Net-Protokoll im Überblick



## Kritik:

- Implementierungsspezifisch:
  - Inhalt der Spezifikation
    - Detaillierte Aufgabenbeschreibung
    - Limits (z.B. Zeitlimit, Qualität etc.)
    - Meta-Task Informations (z.B. nur bestimmte Agenten dürfen bieten)
  - Broadcast als Mittel der Ausschreibung
    - Hohes Kommunikationsaufkommen
    - Ressourcenhungrig
  - Evtl. Sub-Kontraktoren
    - frühes Binden vs. spätes Binden
    - Vervielfachung des Ressourcenbedarfs
    - Gefahr durch mögl. Deadlocks

## Mögliche Probleme:

- Mehrere Manager
  - Zeitbezogene Unwissenheit
  - Räumliche Unwissenheit
  - Ressourcenunkenntnis
- Parallelismus
  - Verpflichtungen
  - Reservierungen
  - Vertragsbruch / Abweisung

## Zusammenfassung:

- Vorteile
  - Dynamisch
  - Flexibilität
  - Einfache Implementierung (des Ansatzes)
  - Weite Verbreitung
  
- Nachteile
  - Verpflichtungen / Sub-Kontraktoren
  - Performance und Kommunikationsaufwand
  - Implementierungsabhängigkeit
  
- Anm.: Kein Modell zur Problemlösung  
Protokoll für die Organisation und dynamischen Allokation

## Erweiterungen:

- Iteratives Contract Net-Protokoll
  - Mehrere Gebotsrunden
    - „explodierender“ Kommunikationsbedarf
    - evtl. lukrativere / bessere Angebote
- Kombination aus CNET und Bekanntschaftsnetzwerken
  - Kleine Aufgaben → Bekanntschaftsnetze
  - Große Aufgaben → CNET
  - Dynamische Bekanntschaftsnetze
- Bekanntschaftsnetze als Zwischenspeicher von CNET

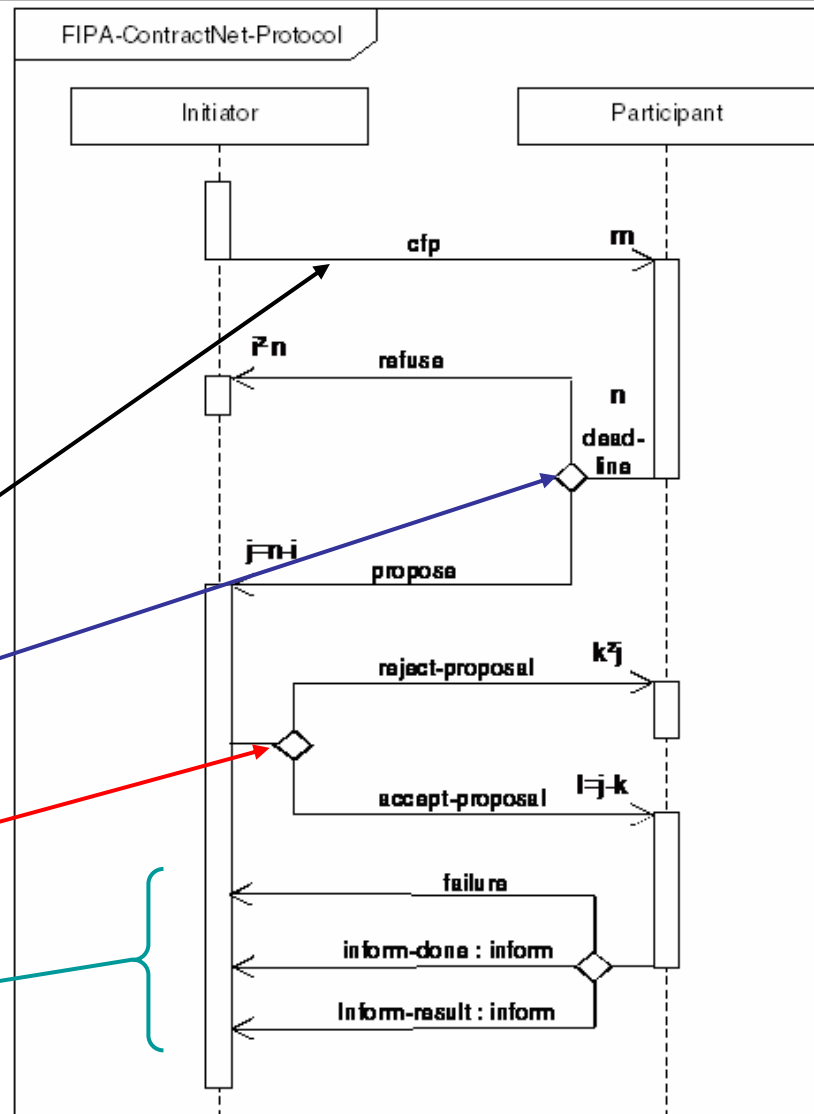
## 4. FIPA

## FIPA

- Foundation for Intelligent Physical Agents
- Gründung: 1996
- Sitz: Genf
- Ziel: Entwicklung von Spezifikationen und Standardisierung im Bereich der Agententechnologie
- Non-profit Organisation
- Mitglieder aus Wissenschaft und Industrie
- 4 Meetings pro Jahr
- [www.fipa.org](http://www.fipa.org)

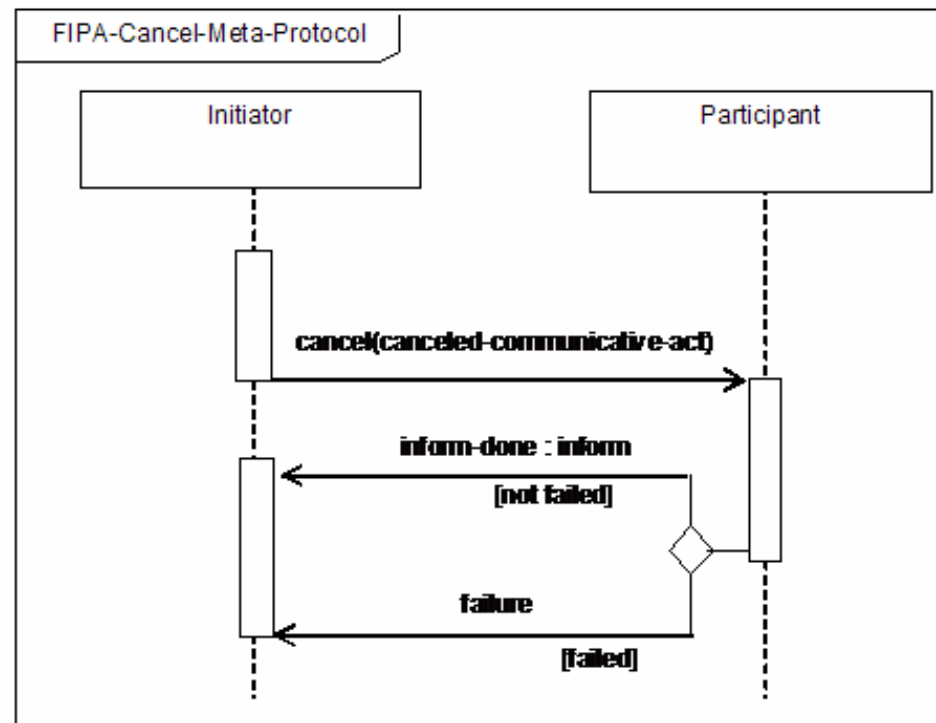
# Beispiel: FIPA Contract Net Interaction Protocol Specification

- Standard seit 06.12.2002
- **1. Ausschreibung**
- **2. Gebote**
- **3. Zuteilung**
- **4. Ergebnis**



# FIPA Contract Net Interaction Protocol Specification

- Exceptions to Interaction Protocol Flow
- Standard seit 06.12.2002



# 5. Zusammenfassung und Ausblick

## Zusammenfassung

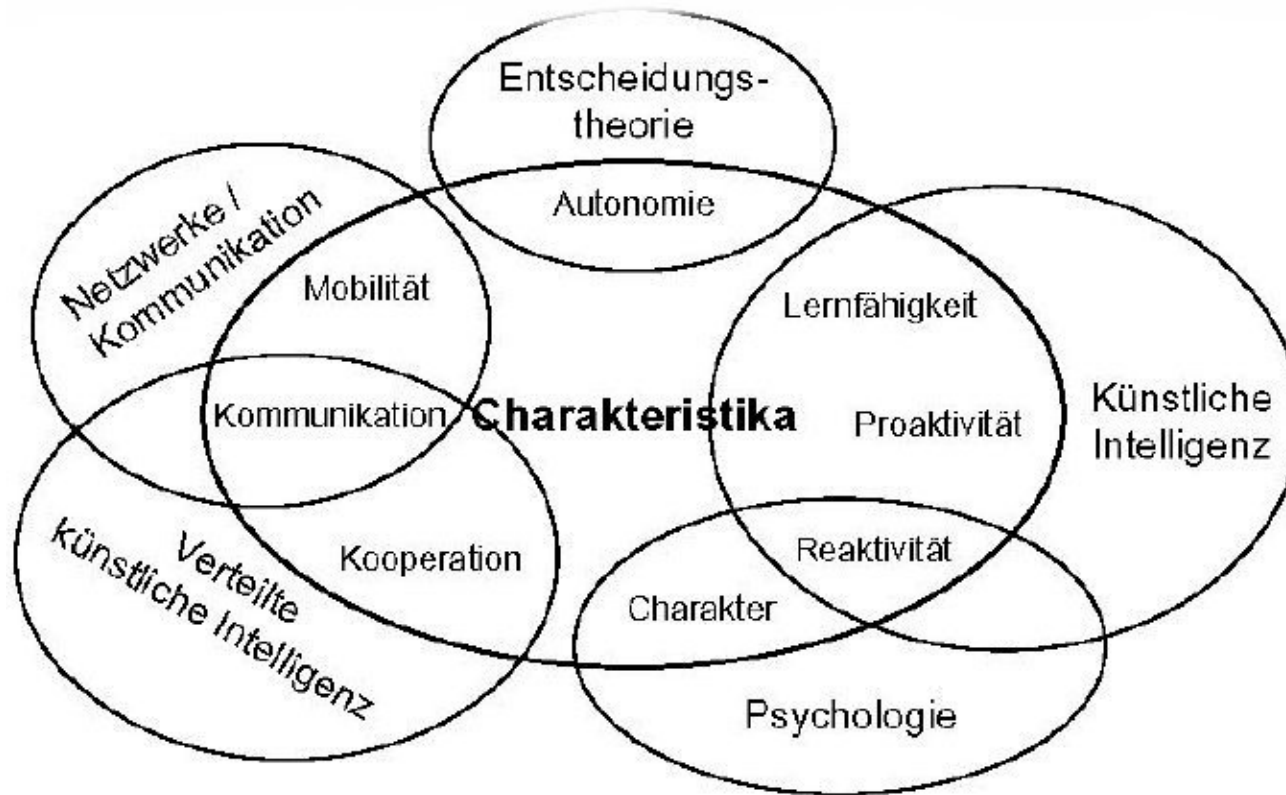
- Was sind Agenten und Multiagentensysteme?
- Es gibt auch unter den Agenten „Gute und Böse“
- Klassifikationen
- Verteiltes Problemlösen und CDPS
  - Aufgabenverteilung (*Task Sharing*)
    - Zentrale Allokation vs. Dezentrale Allokation
  - Ergebnispartizipation (*Result Sharing*)
    - Blackboardansatz
- Contract Net –Ansatz und –Protokoll
- FIPA

## Ausblick

- Wie werden sich Agenten u. Multiagentensysteme entwickeln???
- Probleme:
  - Abgrenzung zur anderen Forschungsgebieten
  - Entwicklung generell respektierter Standards und Spezifikationen
  - Uneinigkeit der „Forschungsgemeinde“
- Adaption vieler Ansätze in anderen [Disziplinen](#)
- Unklar: Wird die Agentenforschung (DAI) eine eigene Disziplin (bleiben)?
- Potenzial für neue Perspektive der Informatik
  - Fast grenzenlose Möglichkeiten (Mobilität, Flexibilität, Grid Computing, Knowledge resources etc.)



# Einflußgebiete



Quelle: W. Brenner et al., Intelligente Softwareagenten, 1998

*Vielen Dank für Ihre  
Aufmerksamkeit!*

