

Agentenwerkzeuge

– Tools zur Agentenentwicklung –

Ein Vortrag im Rahmen des Seminars

Agenten

an der Humboldt Universität zu Berlin

Inhalt

- Entwicklungsmodell
- Werkzeugarten
 - Programmiersprachen
 - Komponenten
 - Entwicklungsumgebungen (IDEs)
 - Experimentierumgebungen

Entwicklungsmodell

- verteiltes System
- eigenständige Einheiten: Agenten
- prinzipiell Objektorientiertes Vorgehen
 - jeder Agent kann als Objekt aufgefasst werden, das ein bestimmtes Verhalten nach außen hin zeigt
- aber: Agenten sind komplexer als „normale“ Objekte

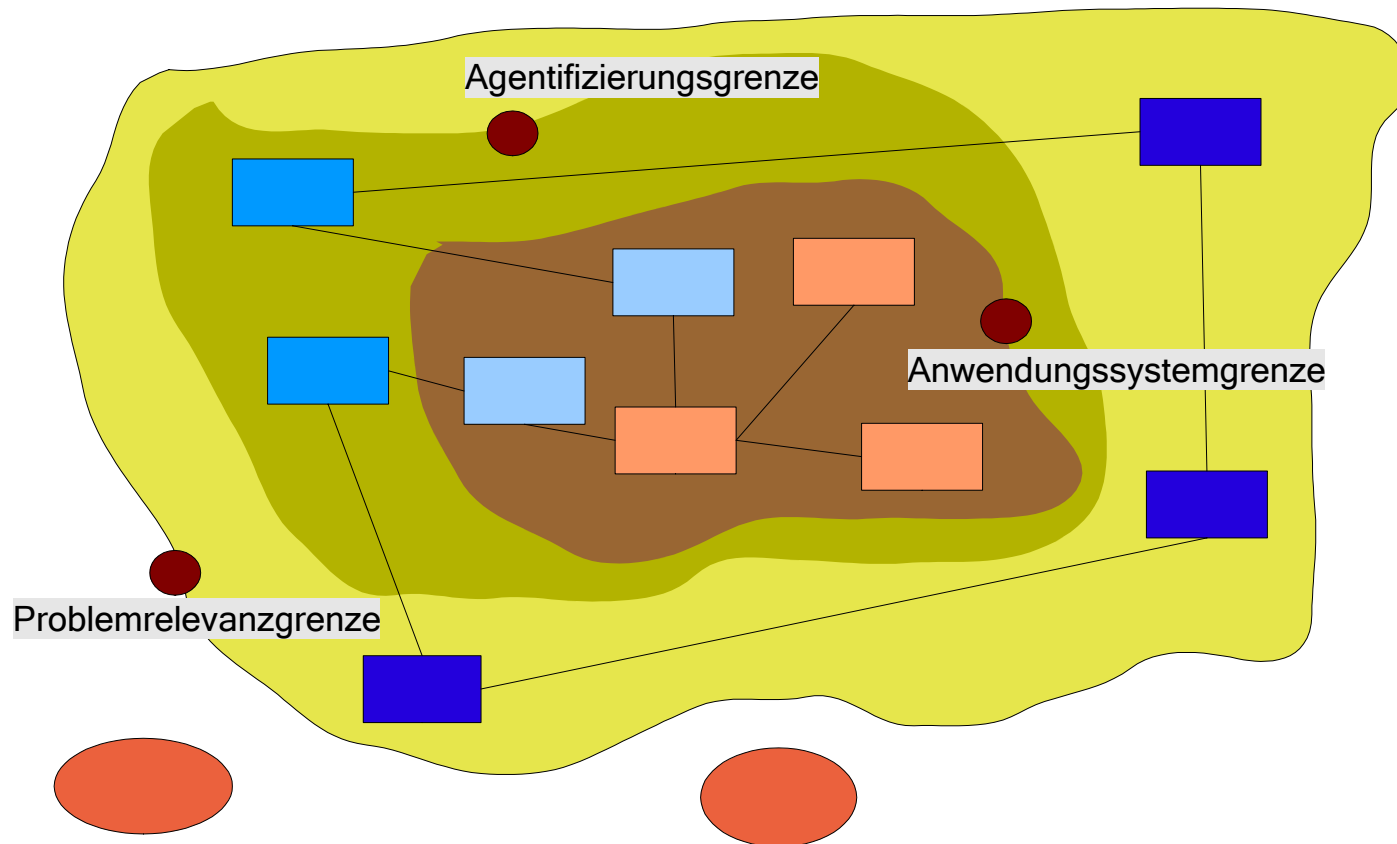
Agenten-OOP

- Agenten ähneln im inneren einem ganzen Objektsystem
- haben Methoden, Attribute, mentale Zustände, Konzepte, Pläne und Ziele
- aktiv
- größeres Kommunikationsspektrum
- kaum Vererbung

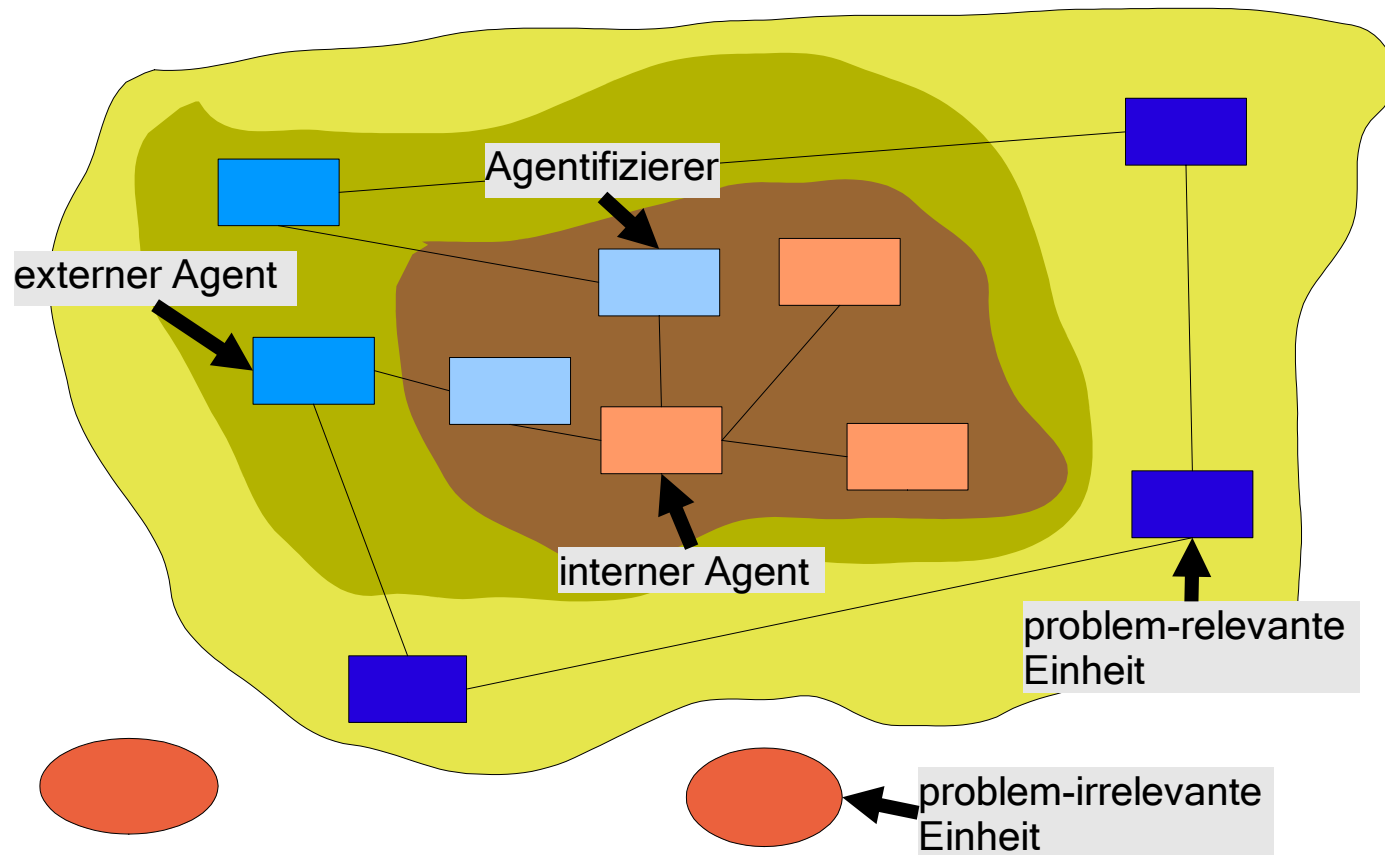
Entwicklungsmodell 2

- Analyse
Spezifikation erstellen
- Design
Struktur, Funktionalität und Verhalten der Agenten bestimmen
- Implementierung
Architektur realisieren
- Test
Fehler entfernen

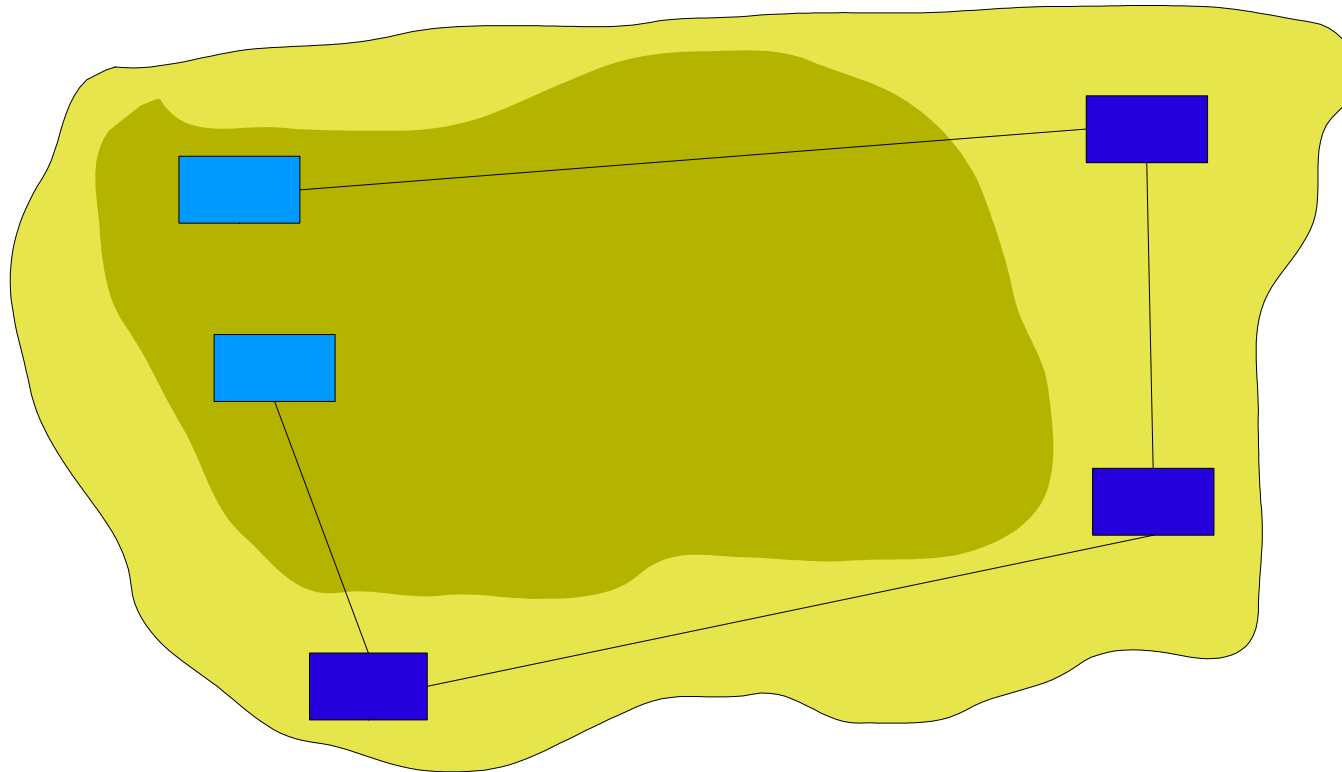
Agentifizierung



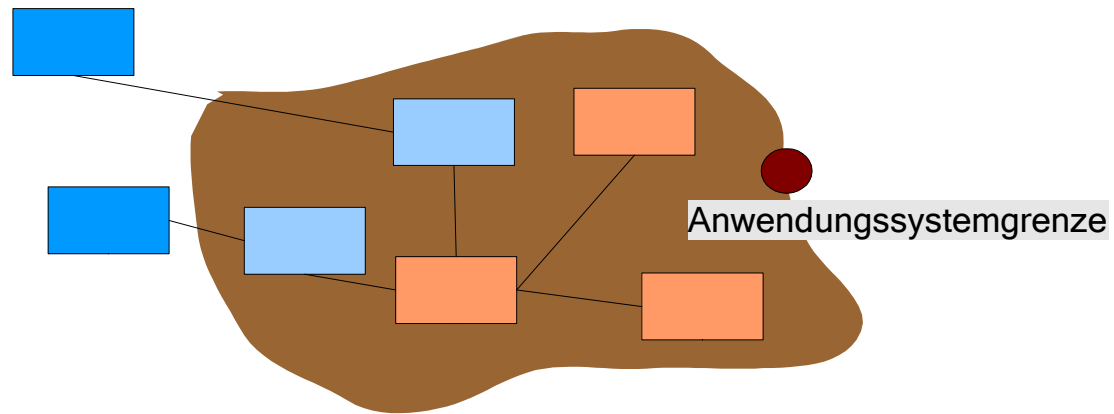
Agentifizierung



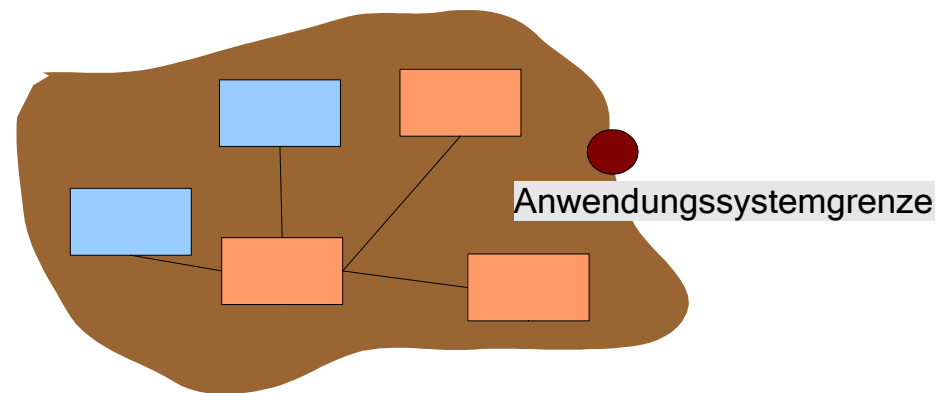
Analyse



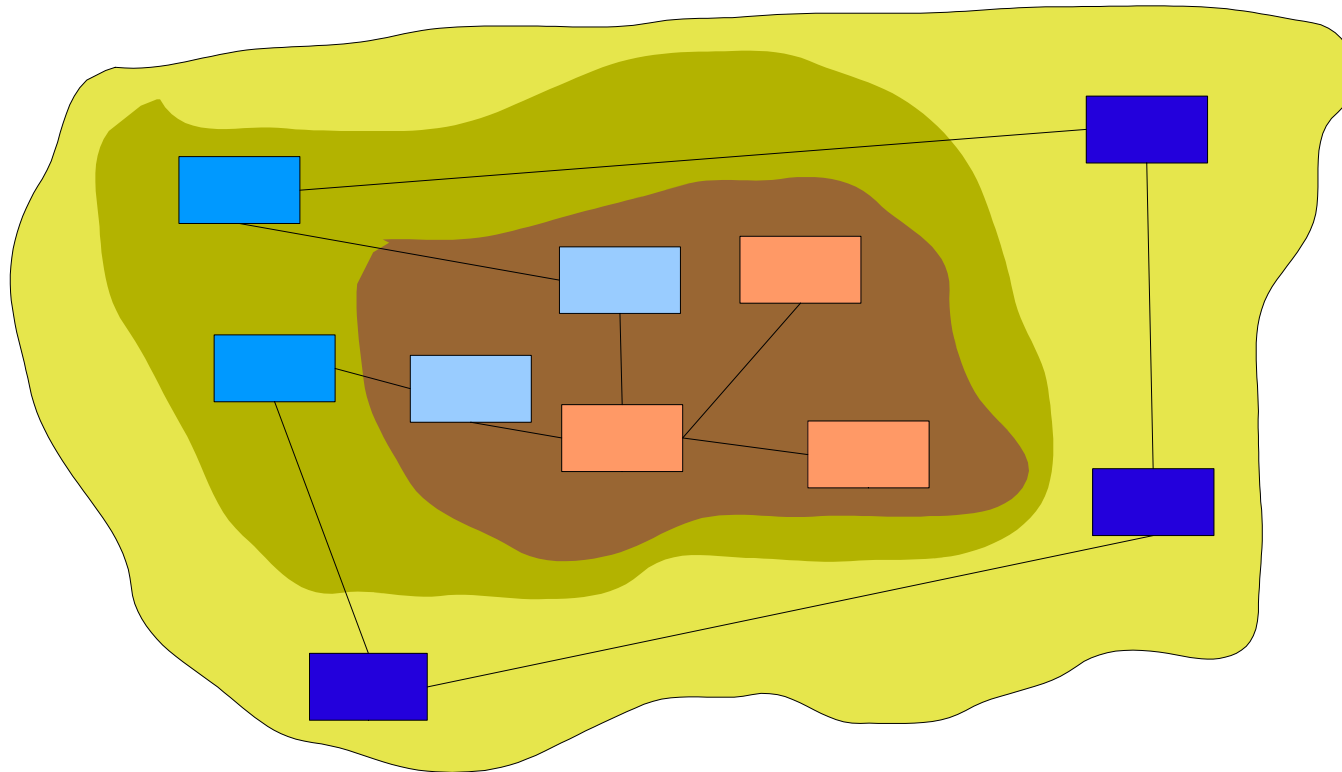
Design



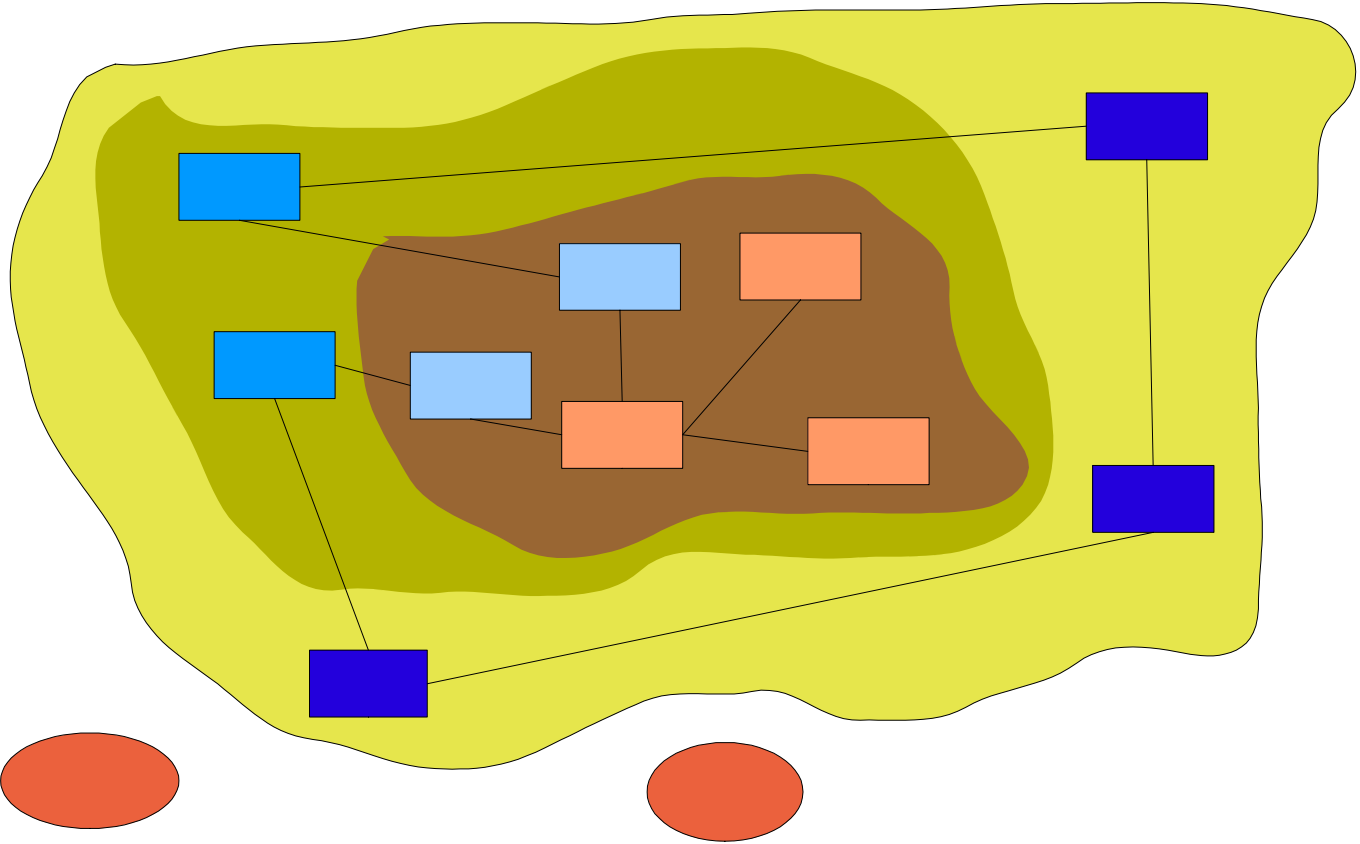
Implementierung



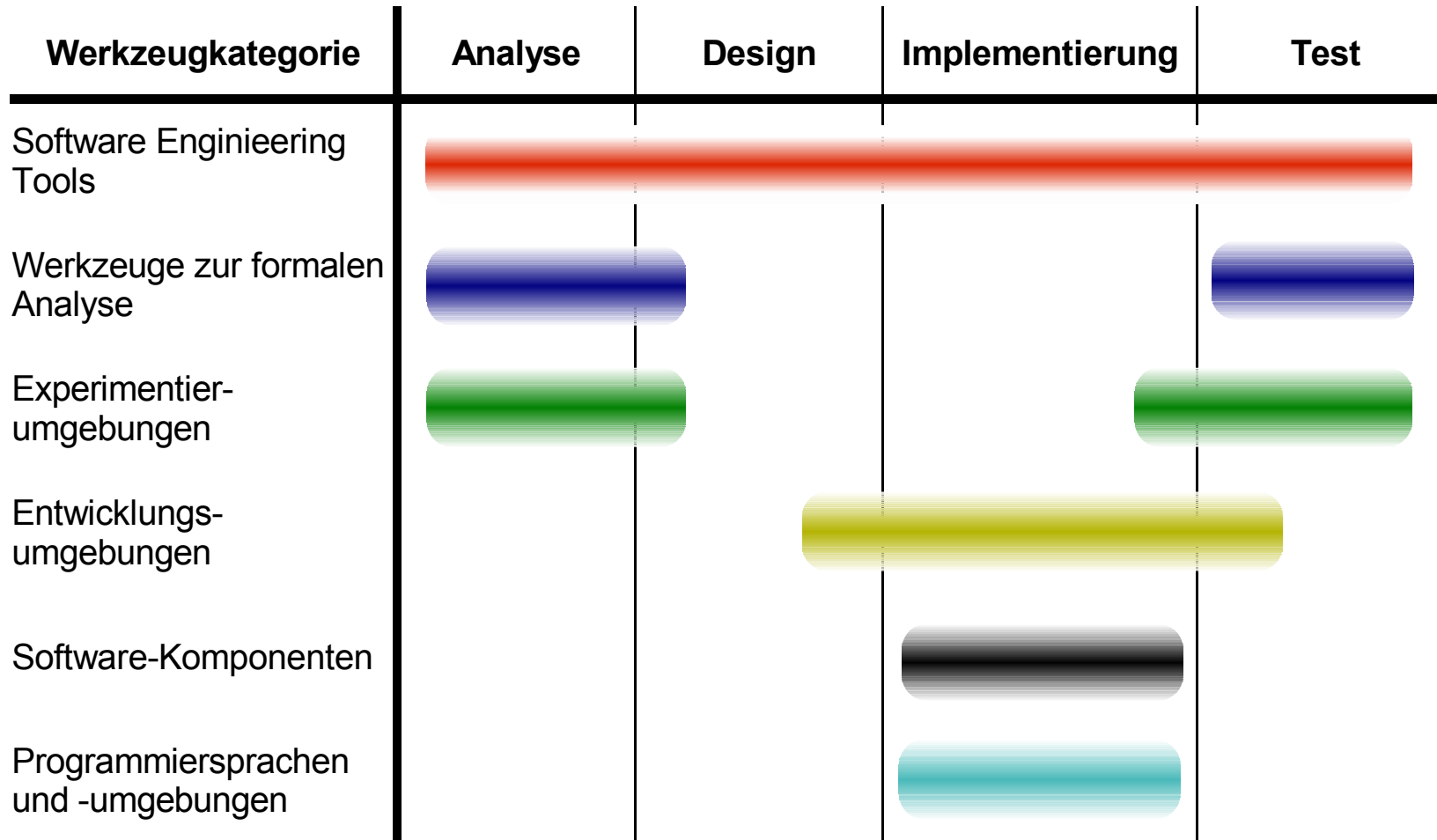
Test



Einsatz



Werkzeuge



Programmiersprachen

- Anforderungen
 - objektorientiert
 - Objekte als Prozesse
 - Verteiltheit der Objekte
 - gute Objektkommunikation
 - Wissensrepräsentation
 - Plattformunabhängig
 - Sicherheit

Programmiersprachen 2

- Programmierumgebungen, Debugfähigkeit
- grafische Oberflächen
- Interaktion mit vorhandenen Systemem

Übersicht über Programmiersprachen

- **Aktoren-Sprachen**
 - Act1, Act2, Act3, Actors
 - ABCL/1, ABCDL/1
 - Agent-0
- **Objektorientiert**
 - Smalltalk
 - Java
- **Prolog und Derivate**
- **Lisp und Derivate**

Übersicht über Programmiersprachen 2

- Prozedurale Sprachen
 - C
 - Agent-Tcl

Java

- Vorteile

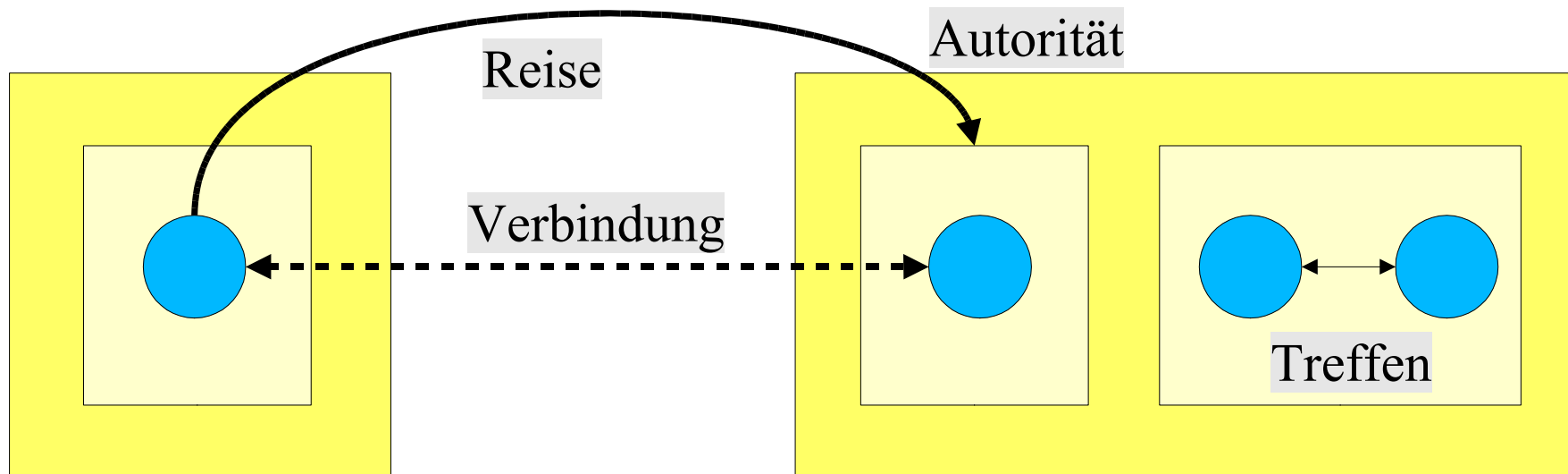
- objektorientiert
- betriebssystemunabhängig
- Threads zur Modellierung paralleler Agenten
- Verteiltheit durch gute Netzwerkfähigkeit
- Sicherheitsmodell

- Nachteile

- keine direkte Unterstützung für Kommunikation auf höherem Level
- keine Wissensrepräsentation, Planung etc.

Telescript

- System zur Implementation mobiler Agenten
- Agenten, Plätze, Treffen, Verbindungen, Autoritäten und Berechtigungen



Telescript 2

- Agenten
 - befindet sich immer an einem Platz
 - kann den Platz wechseln, um eine andere Dienstleistung zu bekommen
- Platz
 - ein oder mehrere pro Rechner
 - bietet eine Dienstleistung an
 - Aufenthaltsort für die Agenten

Telescript 3

- Reise (*go*)
 - Ticket
- Treffen (*meet*)
 - zwischen Agenten am selben Platz
 - Petition
- Verbindung (*connect*)
 - zwischen Agenten an unterschiedlichen Plätzen

Telescript 4

- **Autoritäten (*name*)**
 - definiert die Identität eines jeden Platzes und Agenten
 - repräsentiert durch das Individuum/Organisation die der Agent/Platz vertritt
 - durch Signatur/Zertifikat nachgewiesen
 - kann jederzeit festgestellt werden
 - wird geprüft, wenn ein Agent die Region wechselt
- **Berechtigungen (*permit*)**
 - erlauben den Agenten die Ausführung von Aktionen
 - werden durch die Autorität definiert

Telescript 5

- Telescript-Engine auf jedem beteiligten Rechner
- Speicher-, Transport- und Anwendungs-API
- eigene Programmiersprache
- Sicherheitsaspekte

Agent-Tcl

Software-Komponenten

- Implementierung spezieller Konzepte
- oft als Bibliothek realisiert
- innere Abgeschlossenheit
- Beispiele
 - KIF, KQML
 - Begründungsverwaltung (DTMS, DATMS, DARMS)

Entwicklungsumgebungen

- Anforderungen
 - Framework für Agenten
 - Wissensrepräsentation
 - Benutzerschnittstellen
 - Kommunikationsschnittstellen
 - Betriebssystemschnittstellen
 - Testsystem
 - Offenheit
 - Verteiltheit der Agenten ermöglichen
 - Unterstützung der Wartung

Beispiele

- *Mozart*
- GroupKit
- MAKILA
- NetGBB
- RTworks

Mozart und Oz

[to do]

Experimentierumgebungen

- Anforderungen
 - Modellierung einer Umgebung ermöglichen
 - Modellbeschreibungssprache
 - Sensoren und Aktoren für die Agentifizierer
 - Modelldatenbanken
 - Experimentsteuerung
 - Starten, Beenden, Unterbrechen
 - Speichern und Laden von Zuständen
 - Fortsetzen der geladenen Zustände
 - Aufzeichnung, Abspielen
 - Wiederholbarkeit

Experimentierumgebungen 2

- Anforderungen
 - Datenaufzeichnung, Analysemethoden, Präsentationsmethoden
 - Experimentbeschreibungssprache
- anwendungsunabhängiger Kern,
anwendungsabhängiges Simulationsmodell

Beispiele

- MACE
- DASEDIS
- Swarm

Literatur

Gerhard K. Kraetzschmar, Rolf Reinema. VKI Tools und Experimentierumgebungen. In: Jürgen Müller. Verteilte künstliche Intelligenz. Bi Wissenschaftsverlag, Mannheim 1993

Brenner, Walter. Intelligente Softwareagenten: Grundlagen und Anwendungen. Springer 1998

James E. White. Mobile Agents. In: Jeffrey M. Bradshaw. Software Agents. AAAI Press/The MIT Press 1997

Ralf Kühnel. Java and Agent Oriented Programming.
<http://www.ki.informatik.hu-berlin.de/alumni/kuehnel/papers/ki96.ps>

Robert S. Gray. Agent Tcl. In: Dr. Dobb's Journal March 1997.
<http://www.ddj.com/documents/s=943/ddj9703a/9703a.htm>