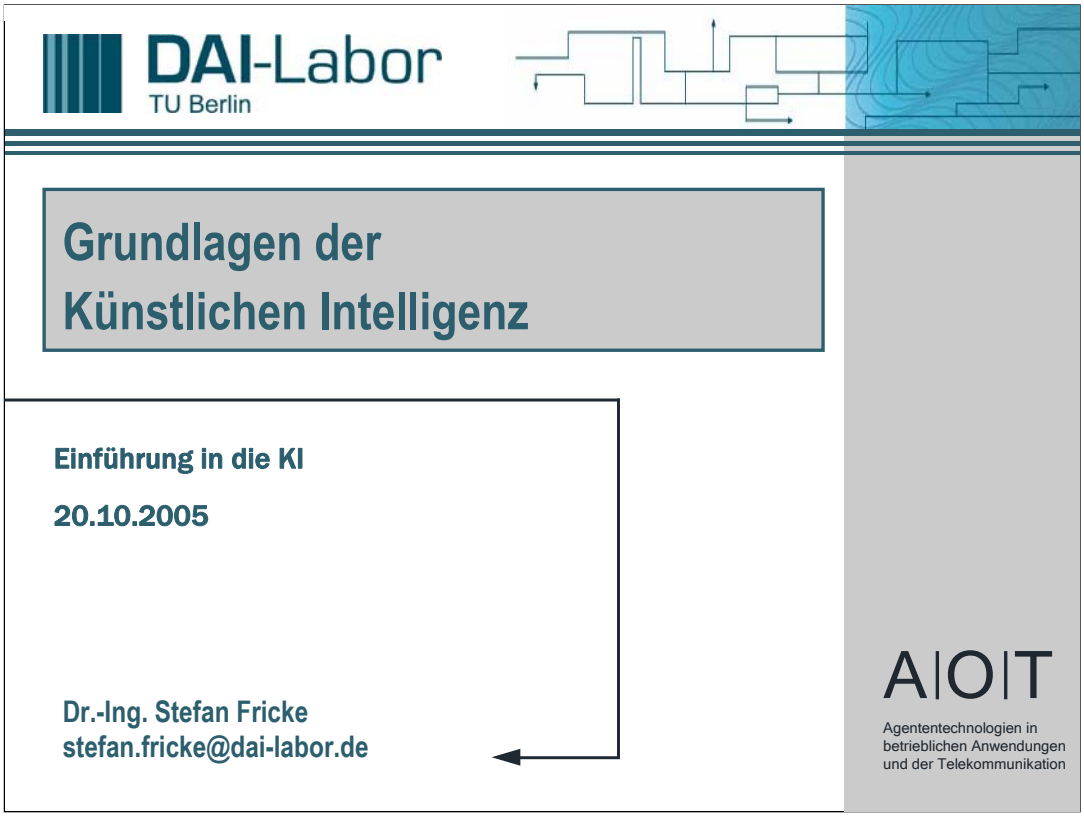


Lernziele:

Verstehen, worum es in der KI geht, was ihre Wurzeln sind und wie sie sich entwickelt hat.

Den Begriff des Agenten kennen lernen und die damit verknüpften Konzepte der Rationalität, Umgebung und Agentenarchitektur.

Einen Überblick über die inhaltliche und organisatorische Struktur dieser LV bekommen.

Gliederung

- ⇨ **Die Lehrveranstaltung GKI im Überblick**
- ⇨ **Was ist Künstliche Intelligenz?**
- ⇨ **Geschichtlicher Überblick**
- ⇨ **Intelligente Softwareagenten**
- ⇨ **Zusammenfassung und Ausblick**

Inhalte der LV GKI

- **Problemlösen (3 Termine)**
  - Suchalgorithmen, Informed Search, Constrainterfüllung, Spiele, Planen
- **Logik &Wissensrepräsentation (3 Termine)**
  - Logisches Schließen, First Order Logic, Wissensbasen, Inferenzen
- **Intelligente Agenten (2 Termine)**
  - Agenten und Environment, reaktive Agenten, kooperatives Problemlösen, BDI-Agenten, modale Logiken, nichtmonotone Logiken
- **Maschinelles Lernen (7 Termine)**

Lernziele

- **Theoretisches Methodenwissen**
  - Methoden der Wissensrepräsenation, Suchalgorithmen und Lernverfahren erlernen
- **Praktisches Anwendungswissen**
  - Einsatzmöglichkeiten, Stärken und Schwächen von KI-Methoden kennen
- **Einordnung, Überblick und Motivation**
  - Weiter führende Lehrveranstaltungen, Forschungsthemen für Diplomarbeiten, ...

Organisatorisches

- **Vorlesung:**
  - Prof. Albayrak und Prof. Obermayer
- **Übungen:**
  - 2 Termine zur Auswahl: Mittwoch 10-12 u. 12-14 Uhr im EMH 225
  - 5+x Übungsaufgaben, zu lösen von Arbeitsgruppen
  - Klausur in der letzten Semesterwoche
- **Ansprechpartner:**
  - Brijnesh Johannes Jain (brijnesh-johannes.jain@dai-labor.de)

Gliederung

- ⇨ Die Lehrveranstaltung GKI im Überblick
- ⇨ Was ist Künstliche Intelligenz?
- ⇨ Geschichtlicher Überblick
- ⇨ Intelligente Softwareagenten
- ⇨ Zusammenfassung und Ausblick

Was ist Künstliche Intelligenz? – Eine Annäherung

- **Intelligentes Verhalten verstehen vs. intelligente Systeme bauen**
- **Merkmale natürlicher Intelligenz**
  - Sehen und erkennen --> Computer Vision, Wissensrepräs.
  - Texte verstehen --> ?
  - Sprachfähigkeit --> Spracherkennung /-synthese
  - Probleme lösen --> Suchalgorithmen, Reasoning
  - Zielgerichtet handeln --> Belief-Desire-Intention
  - Planen --> Planer
  - Lernen --> Maschinelles Lernen

Intelligentes Verhalten verstehen: Kognitionswissenschaft (auch: Cognitive Science) beschäftigt sich mit mentalen Prozessen: Probleme lösen, lernen, aber auch Emotionen, Meinungen, Einstellungen, Wünsche, Absichten. Es geht um die Beschreibung und Nachbildung (der Prozesse) des Gehirns.

Intelligente Systeme bauen: Kann mittels kognitiver Methoden geschehen (sofern diese verstanden sind und umsetzbar sind – auf viele Prozesse (Lesen, Kommunizieren, Verstehen) trifft dies nicht zu). Alternativ aber auch durch andere Ansätze, z.B. Emergent Intelligence / Selbstorganisation in Insektenstaaten, Schachcomputer mit Brute-Force-Algorithmen, Eliza als einfaches Pattern-Matching-Programm).

Merkmale von Intelligenz orientieren sich klarmam Menschen. Intelligenz ist schließlich ein vom Menschen geprägter Begriff (möglicherweise zur Unterscheidung von allen anderen Lebewesen).

Texte verstehen: Lesen können, Texte zusammenfassen können, Fragen beantworten können.

Sprachfähigkeit: Natürlichsprachlich kommunizieren, d.h. Sprache sprechen und verstehen.

Zielgerichtet handeln: Begriff der Rationalität (s.u.), Handlungen und Ziele wichten und beurteilen können.

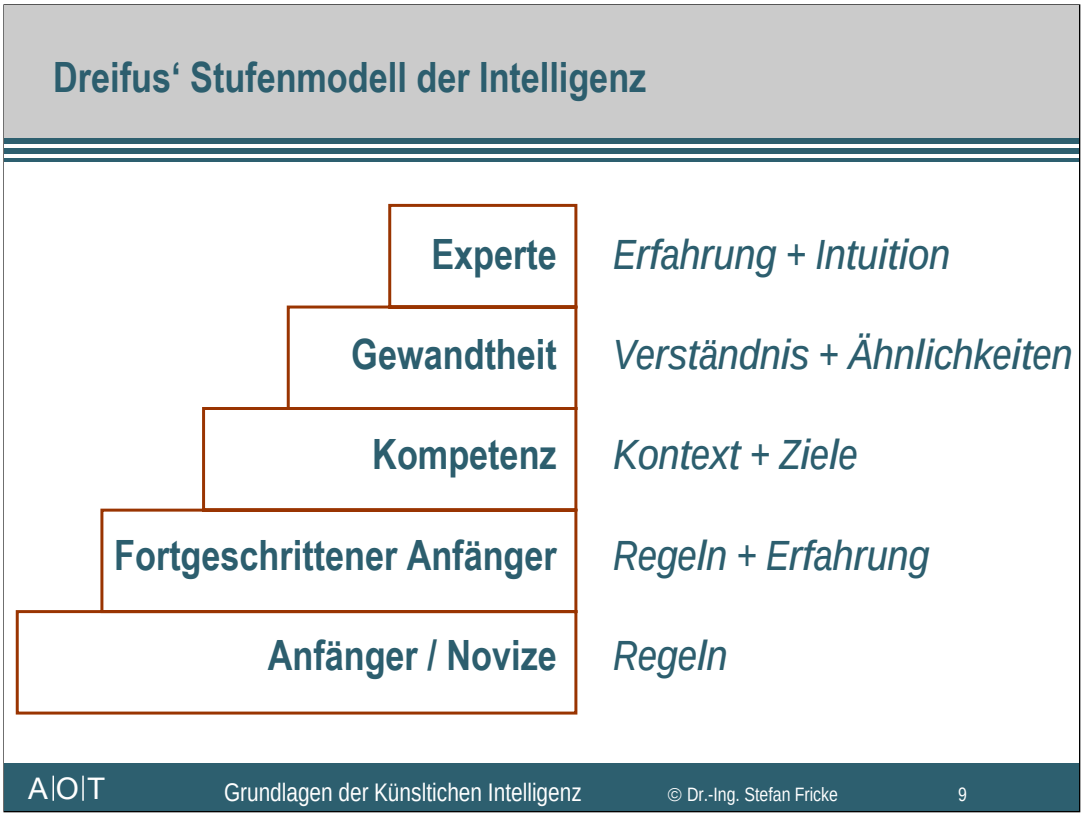
Planen und Lernen gehören zum Handwerkszeug menschlichen iuntelligenten Handelns. Planen, um große, langfristige Vorhaben mit einem relativ beschränkten Vorrat an Handlungsalternativen durchzuführen und Lernen, um flexibel in unbekannten Situationen handeln zu können.

Was ist Künstliche Intelligenz?

- **Maschinen entwickeln, die sich verhalten, als verfügten sie über Intelligenz (John McCarthy, 1955)**
- **Starke KI**
  - eine Intelligenz erschaffen, die wie der Mensch nachdenken und Probleme lösen kann
- **Schwache KI**
  - Anwendungen bauen, zu deren Lösung nach allgemeinem Verständnis "Intelligenz" notwendig ist

Starke KI: Bewusstsein und Emotionalität gehören dazu.

Schwache KI: pragmatischer Ansatz, der auch wenig intelligente (nicht kognitive) Methoden ausdrücklich mit einbezieht. Vergleiche die Vorlesungen zu Suchproblemen, insbes. Contraints.



Dreyfus/Dreyfus (1987) - These des Fünf-Stufen-Modells des Fertigkeitenerwerbs. Die fünf Stufen sind Neuling (Novice), Fortgeschrittener Anfänger (Advanced Beginner), Kompetenz (Competent), Gewandtheit (Proficient) und Experte (Expert). Die These besagt, dass es einen prinzipiellen Unterschied zwischen Wissen und Können gibt. Da mit typischen Expertensystemen nur *Wissen* verarbeitet werden kann, sind sie prinzipiell nicht in der Lage, volle Problemlösefähigkeit von Experten zu simulieren. Für den Fall, dass die Dreyfus-Thesen zutreffen, könnte man mit Expertensystemen nicht über das Kompetenz-Niveau hinaus gelangen.

Novizen verfügen über Regeln (und eventuell grundlegende Fakten). Sie wenden diese Regeln unhinterfragt (losgelöst vom Kontext, d.h. kontextfrei) an, weil sie eine Situation nicht beurteilen können. Dieses unreflektierte Handeln ist unflexibel und kann in Störungssituationen zu falschem Verhalten oder Chaos führen.

Fortgeschrittener Anfänger nutzt praktische Handlungserfahrungen (Fälle/Situationen) als Kontext zur Anwendung von Regeln. Erinnerungen an ähnliche Fälle ermöglichen den Transfer dieser Erfahrungen auf neue Situationen. Anfänger können noch nicht selbständig handeln.

Kompetente Leute haben Problemlösekompetenz, sie können auf einem Gebiet selbständig handeln. Sie kennen alle relevanten Fakten und Regeln und deren Anwendbarkeit bezogen auf den Kontext. Sie könne Situationen vollständig analytisch erfassen, korrekt einschätzen, zielorientiert vorgehen, Lösungswege planen, Schlüsse ziehen usw. Jedoch ist jede Problemlösung immer noch ein (möglicherweise lange dauernder) Prozess vieler kleiner Schritte, eingebettet in ein Lösungsverfahren.

Gewandte nehmen eine Situation ganzheitlich wahr: die Situation offenbart bereits ihre Lösung. Viel Erfahrung ist dazu notwendig, um Ähnlichkeiten in unterschiedlichen Situationen zu erkennen.

Experten schließlich handeln intuitiv, ohne Anstrengung und jeder Situation angemessen. Auch die komplexesten Situationen erscheinen als bekannte "Fälle" bereits vertraut. Unübersichtliche Situationen kann ein Experte auf "Fälle" zurückführen, die ihre eigene Lösung bereits mit einschließen.

Quellen: http://beat.doebe.li/bibliothek/w01259.html und andere

Dreyfus, Hubert .L.; Dreyfus, Steward .E., 1987: Künstliche Intelligenz – Von den Grenzen der Denkmaschine und dem Wert der Intuition. Reinbek bei Hamburg: Rowohlt.

Gliederung

- ⇨ Die Lehrveranstaltung GKI im Überblick
- ⇨ Was ist Künstliche Intelligenz?
- ⇨ Geschichtlicher Überblick
- ⇨ Intelligente Softwareagenten
- ⇨ Zusammenfassung und Ausblick

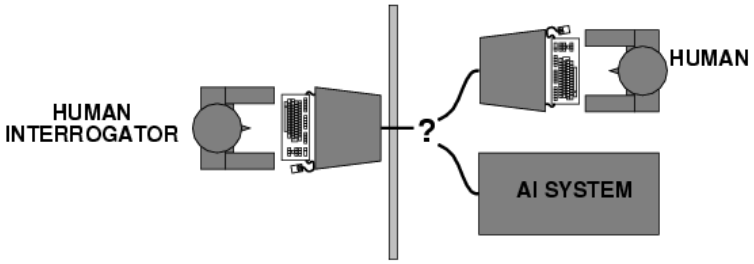
Wurzeln der Künstlichen Intelligenz

- ⇨ **Aristoteles (384-322 AC): Intelligenz durch Anwendung von Regeln, bestehend aus Wissen, Zielen und Handlungen**
- ⇨ **Thomas Hobbes (1588-1679): Denken ist Numerik**
- ⇨ **John S. Mill (1863): Utilitarismus bestimmt das Handeln**
- ⇨ **Wittgenstein (1889-1951): Logik**
- ⇨ **Léon Walras (1834-1910): Nutzentheorie**
- ⇨ **Cook / Karp (1971): NP-Vollständigkeit**
- ⇨ **Chomsky, Austin: Linguistik**

Aristoteles Gedanken zum zielgerichteten Handeln mittels „means-ends-analysis“ (Mittel-Zweck-Analyse) wurde später von Simon/Newell bei der Implementierung des General Problem Solvers aufgegriffen.

Turing Test – Imitationsspiel (Turing 1950)

- **Ein Mensch führt einen Dialog mit einem Computer oder Menschen – per Bildschirm und Tastatur.**
  - Kann der Mensch beurteilen, ob sein Dialogpartner ein Mensch und welcher ein Computer ist?



Alan Turing in "Computing machinery and intelligence":

"Can machines think?" → "Can machines behave intelligently?"

Ausführbarer Test für intelligentes Verhalten: “Imitation Game”

Turing sagte für das Jahr 2000 voraus, dass Computer eine 30%ige Chance haben würden, den Turing-Test gegen einen Laien mindestens 5 Minuten lang zu bestehen.

Nebenbei schlug Turing folgende wichtige KI-Komponenten vor (die im Zuge der Entwicklung auch tatsächlich relevant wurden): knowledge, reasoning, language understanding, learning.

Eine kurze Geschichte der KI
Phase 1: Enthusiasmus

- ⇨ **Vision: Maschinen die denken, lernen und kreieren**
  - → KI ist machbar! [Simon 1957]
- ⇨ **1957 General Problem Solver [Simon, Newell]**
- ⇨ **1958 LISP [McCarthy]**
- ⇨ **1965 Resolusionsprinzip [Robinson]**
- ⇨ **1966 ELIZA [Weizenbaum]**
- ⇨ **1972 SHRDLU [Winograd]**

Als Tafelbild motivieren (siehe Folie „Entwicklungsphasen“ im Anhang): Die Entwicklungsphasen bei innovativen Produkten (z.B. auch: Electronic Commerce) verlaufen üblicherweise zunächst sehr steil (gemessen am Hype, an Publikationen, an Forschungsgeldern, an Börsenkursen, etc.), beginnend mit einem Technologieauslöser (hier: u.a. LISP). Viele springen auf den Zug, erkennen die Potenziale und kommunizieren diese. Unterlegt werden diese Visionen durch Prototypen, die die Machbarkeit aufzeigen (GPS, ELIZA, SHRDLU, KI-Planer, Constraint-Solver). Die sehr steile Entwicklung gipfelt in der Euphorie, um danach ebenso steil abzustürzen, durch offensichtlich überzogene Erwartungen ins Tal der Desillusion. Diese Phase der Ernüchterung dient gleichzeitig auch als Konsolidierung, die Spreu wird vom Weizen getrennt. Danach steigt die Erfolgskurve langsam wieder an (Phase der Etablierung/Aufklärung): tatsächliche Potenziale werden erkannt und ausgenutzt. In der abschließenden Phase der Profitabilität/Produktivität steigt die Kurve noch schwächer an bzw. stagniert.

GPS beruht auf Problemreduktion und –transformation (Termersetzung) und wurde erfolgreich zum Beweisen einfacher logischer und geometrischer Theoreme und Wortpuzzles eingesetzt. Das Scheitern des GPS an komplizierteren und allgemeineren Problemen führte schließlich zur Entwicklung von Expertensystemen, die auf einem engeren Wissensgebiet zu besseren Ergebnissen gelangen sollten.

LISP ist eine symbolische Programmiersprache. Bis dahin waren nur numerische Sprachen bekannt, sodass Intelligenz immer auf Zahlenrechnerei abgebildet werden musste. Dieses Postulat fiel mit LISP (und später auch mit Prolog). Symbolverarbeitung, so formulierte Newell später in seiner „physical symbol hypothesis“, bietet notwendige und hinreichende Voraussetzung für intelligentes Verhalten.

Resolutionsprinzip als Grundlage des Theorembeweisens: Eine Inferenzregel (AND(OR(a,b), OR(NOT(b),c)) → OR(a,c) solange auf eine Menge von Disjunktionen anwenden, bis die leere Klausel abgeleitet werden kann. (Die zu beweisende Klausel wird zuvor negiert, damit der bewiesene Widerspruch äquivalent zur Wahrheit der zu testenden Klausel ist).

ELIZA ist der erste Chatbot. Er startete mit „How do you do. Please state your problem.” und nutzte Textfragmente des menschlichen Dialogpartners, um Fragen oder Aussagen zu bilden und so den Dialog fortzuführen.

SHRDLU ist ein sehender einarmiger Roboter, der geschriebene Sprache versteht und Befehle ausführen und Fragen beantworten kann – innerhalb der Blocks-World (eine Mikrowelt, bestehend aus einfachen geometrischen Figuren, die farblich und nach Größe unterschieden werden).

ELIZA (Weizenbaum 1966)

- ELIZA simuliert einen Psychotherapeuten.
- Durch den geschickten Einsatz von Phrasen, wie „Erzählen sie mir mehr von X“ oder „Denken sie oft an X“ kann ELIZA Testpersonen längere Zeit täuschen.

```
>> I spend too much time with computers.
Why do you mention computers ?
>> You are one.
Does it please you to believe I am one ?
>> Yes.
You seem to be quite positive.
>> I suppose I am.
How long have you been ?
```


Quelle: Wikipedia.de

Der Erschaffer von ELIZA, Joseph Weizenbaum, wollte mit diesem Programm gar keine Intelligenz demonstrieren. Als er sah, dass sein triviales Programm von vielen Menschen als “intelligent” betrachtete wurde, wendete sich von der KI-Forschung ab. Er wurde zu einem Kritiker der KI.

Eine kurze Geschichte der KI
Phase 2: Desillusion (1966 - 1972)

- ⇨ **Mikrowelten sind trivial. Ergebnisse lassen sich nicht auf reale Probleme übertragen.**
- ⇨ **Die Komplexität der schwachen Methoden liegt zumeist in NP.**
- ⇨ **Reine Symbolmanipulation ist offensichtlich nicht genug**
  - ➙ Genetische Algorithmen, Neuronale Netzwerke arbeiten auf subsymbolischen Strukturen, repräsentieren aber kein Wissen.

Erfolge in den Mikrowelten wie die Blocks World, Wort-Puzzel, Zahlenspiele, einfache Strategiespiele lassen sich nicht einfach auf komplexe, reale Welten übertragen. Die Komplexität fast aller KI-Verfahren, insbesondere der generellen (so genannten schwachen Methoden, z.B. Planung, Suche, Deduktion, ...) ist NP-vollständig. Die bis dahin entwickelten Methoden (auch Genetische Algorithmen und neuronale Verfahren) arbeiten nur mittels sehr einfacher Strukturen und ohne Wissen.

Weiterlesen: Hubert Dreyfus: *Was Computer nicht können. Die Grenzen künstlicher Intelligenz*. 1989

Eine kurze Geschichte der KI
Phase 3: Konsolidierung (ab 1971)

- ⇨ **Expertensysteme**
  - → Wissensrepräsentation, knowledge engineering, starke Methoden
- ⇨ **1971 STRIPS Planer [Fikes]**
- ⇨ **1965-75 DENDRAL: Molekurlarchemie-XPS [Feigenbaum]**
- ⇨ **1972 Prolog [Colmerauer, Roussel]**
- ⇨ **1972-80 MYCIN: Medizinisches Expertensystem**
- ⇨ **1977 OPS: Regelinterpreter [Forgy]**
- ⇨ **1978-82 R1/XCON: Computerkonfiguration [McDermott, Drew]**

Expertensysteme sind wissensbasierte Systeme, mit einer Wissensbasis und einer Inferenzmaschine. Der Inferenzmechanismus ist typischerweise keine reine schache Methode, sondern nutzt Heuristiken, um den ansonsten sehr langen Suchprozess zu verkürzen. Eine Erklärungskomponente ist ebenfalls eine typische Komponente eines Expertensystems.

DENDRAL ist ein regelbasiertes Expertensystem im Bereich der organischen Molekularchemie. Wahrscheinliche Molekularstrukturen wurden aufgrund von Spektrometrie und anderen chemischen Analysemethoden vorhergesagt. Alles wesentliche Wissen wurde in Form von Regeln gegossen.

STRIPS beruht auf dem 1969 von [McCarthy, Hayes] eingeführten Situationskalkül: Konjunktionen positiver Fakten mit Closed World Assumption treten in Regeln auf, die durch eine Anwendbarkeitsbedingung und einen Effekt (bestehend aus hinzunehmenden und zu entfernenden Fakten) bestehen.

1982 Temporallogik [McDermott]

1984 Zeitlogik mit Intervallen [Allen]

MYCIN trennt Regeln von Inferenzmaschine. Wahrscheinlichkeitsbasiertes Schließen. Entwickelt in Stanford diagnostiziert MYCIN Infektionskrankheiten des Blutes. > 450 Regeln, in LISP implementiert.

XCON (for eXpert Configurer) (John P. McDermott (1982). *R1: a rule-based configurer of computer systems*. Artificial Intelligence, 19, 39-88.) ist ein XPS zur Konfiguration von Großrechnersystemen (VAX-11/780 der Firma DEC) mit Layout-Planung. Mit ca. 750 Regeln wurden >400 Komponenten zusammengestellt. Später (1987) hatte XCON bereits >10000 Regeln und es wurde erkannt, dass derart große Regelmengen nicht mehr handhabbar waren.

DARPA's funding of research on understanding speech has been extremely important. First, it pushed the research frontiers of speech recognition and AI more generally. HEARSAY-II is particularly notable for the way it parsed information into independent knowledge sources, which in turn interacted with each other through a common database that CMU researchers labeled a "blackboard" (Englemore et al., 1988).

Charles Forgy, John P. McDermott: OPS, A Domain-Independent Production System Language. IJCAI 1977: 933-939

Frames, Constraints

Eine kurze Geschichte der KI
Phasen 4 & 5: Aufklärung und Produktivität (seit 1982)

- ⇨ Von Werkzeugen zu industriellen Anwendungen
- ⇨ ab 1985 Expertensystemschalen
- ⇨ seit 1986: Neuronale Netze zur Mustererkennung
- ⇨ 1987: Erste Anwendung von Fuzzy Logic in Japan
- ⇨ Entwicklung heuristischer und effizienter Suchstrategien
- ⇨ fachspezifisches und alltägliches Wissen in Anwendungen
- ⇨ 1995: Intelligente Agenten...

Eine Shell enthält problemunabhängige Komponenten, typischerweise eine Wissensrepräsentationssprache, Inferenzmechanismen, Werkzeuge, u.U. eine Erklärungskomponente. Z.B. enthält E-MYCIN die Inferenzmaschine aus Mycin.

Beispiele: E-MYCIN, SOAR, KEE (1986): (Knowledge Engineering Environment): LISP, Frames. BABYLON (1985): LISP, Frames und Logik.

1992: KADS (Task Domains)

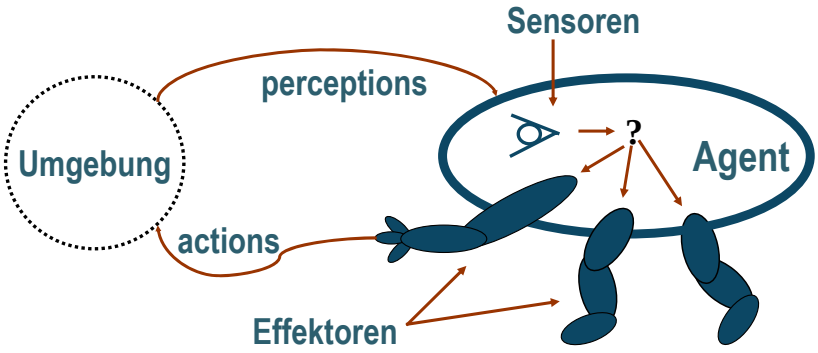
1. Viele Probleme sind Suchprobleme. Die nachfolgenden Vorlesungstermine werden Algorithmen zur Suche behandeln.
2. z.B. in Spielen (Simulationen, Rollenspielen, Strategiespielen)
3. Fuzzy Logic in Kameras, Computer Vision in Industrierobotern, Muster- und Spracherkennung in Avataren, Lernen und Planen in Non-Player-Characters. Die Fuzzy Logic beruht auf Fuzzy Sets und wurde bereits in den 60er Jahren von L.A. Zadeh (Fuzzy sets. *Information and Control*, 8, 338-353) begründet.

Gliederung

- ⇨ **Die Lehrveranstaltung GKI im Überblick**
- ⇨ **Was ist Künstliche Intelligenz?**
- ⇨ **Geschichtlicher Überblick**
- ⇨ **Intelligente Softwareagenten**
  - → Rationales Handeln
  - → Umgebung des Agenten
  - → Agentenarchitekturen
- ⇨ **Zusammenfassung und Ausblick**

Agenten nach Russell & Norvig ...

- ... nehmen ihre **Umwelt** durch **Sensoren** wahr (perceptions),
- ... manipulieren ihre Umwelt mit Hilfe von **Effektoren** (actions).



Verhaltensbasierte Charakterisierung: "An agent perceives its environment through sensors and acts upon that environment through effectors."

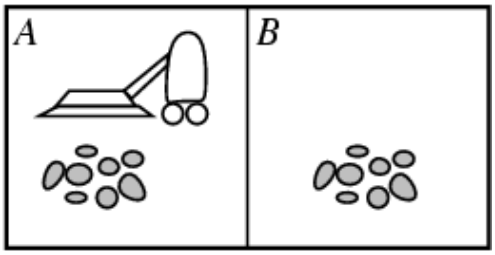
Beispiele: Menschen und Tiere, Roboter und Software-Agenten (Softbots), technische Geräte wie Thermostaten.

Im Folgenden werden folgende synonyme Begriffe für Wahrnehmung und Manipulation der Umgebung verwendet:

Wahrnehmung: Sensoren, Sensors, Signale, Perception.

Manipulation: Aktoren, Actuators, Aktorik, Action, Aktion, Effectors.

Beispiel: Staubsauger-Agent



- **Wahrnehmungen:**
  - Ort und Zustand, z.B. [A, schmutzig] oder [B, sauber]
- **Aktionen:**
  - {nach_links, nach_rechts, saugen, warten}
- **Welche Wahrnehmungen soll der Agent mit welchen Aktionen verknüpfen? ...**

Gliederung

- Die Lehrveranstaltung GKI im Überblick
- Was ist Künstliche Intelligenz?
- Geschichtlicher Überblick
- Intelligente Softwareagenten
  - Rationales Handeln
  - Umgebung des Agenten
  - Agentenarchitekturen
- Zusammenfassung und Ausblick

Zum Begriff der Rationalität

- **Rationalität ist optimales Verhalten [Anderson 91]**
  - "The cognitive system *optimizes* the adaptation of the behavior...„
- **Rationalität is ressourcenbedingt begrenzt [Simon 57]**
  - "property of an agent that behaves in a manner that is nearly optimal with respect to its *goals* as its *resources* will allow".
- **Rationalität ist zielgerichtetes Verhalten [Newell 82]**
  - "If an agent has *knowledge* that one of its actions will lead to one of its *goals*, then the agent will select that action"

Andersons „Principle of Rationality“ betrachtet Optimalität als Notwendige Voraussetzung für Rationalität.

Simons „Bounded Rationality“ geht von einem konsistenten Verhalten aus, das einerseits zielgerichtet in Richtung Optimalität geht, aber andererseits im Angesicht begrenzter Ressourcen (Methoden, Rechnerleistung, Wissen über die Umwelt) doch zumeist nur suboptimal sein kann.

Newells „Maximum Rationality Hypothesis“ geht nicht von der bestmöglichen Aktion aus, sondern drückt aus, dass eine Beziehung zwischen den Zielen und dem Verhalten existiert.

Rationalität in Agentensystemen

- **Ein *allwissender* Agent kennt den tatsächlichen Weltzustand und damit die tatsächlichen Effekte seiner Aktionen.**
  - → Optimales Handeln ist jedoch in offenen Systemen mit unvollständigem Wissen und begrenzten Ressourcen nicht möglich.

- **Ein *rationaler* Agent handelt dagegen auf Grund seiner Wahrnehmungen und seines (beschränkten) Wissens.**

Der ideale rationale Agent existiert nicht (auch nicht in der Natur)

Ein allwissender Agent kennt den tatsächlichen Weltzustand und damit die tatsächlichen Effekte seiner Aktionen.

Ein offenes System zeichnet sich dadurch aus, dass seine Grenzen fließend sind, dass Änderungen auftreten können, die der Agent selbst nicht kontrolliert und dass das Wissen über das System daher notwendigerweise unvollständig ist und teilweise auch falsch (nicht mehr aktuell) sein kann. Typischerweise ist der Agent nicht der einzige Handelnde im System, sondern weitere Agenten sind zugegen, betreten und verlassen das System.

Ein rationaler Agent handelt dagegen auf Grund seiner Wahrnehmungen (Perzepte) und seines (beschränkten) Wissens und versucht, die erwartete Leistung zu maximieren.
Dieser Umstand wird durch die Bounded Rationality ausgedrückt.

Verständnis der Rationalität in Agentensystemen

- **Starke Definition: Maximum Expected Utility (MEU)**
  - MEU drückt aus, dass ein rationaler Agent diejenige Aktion wählen soll, die seinen erwarteten Nutzen maximiert.
  - Der Agent benötigt also eine Nutzenfunktion.
- **Schwache Definition: Zielgerichtetes Verhalten**
- → Auswahl einer Aktion, die zur Erfüllung eines Zieles dient.
  - Der Agent benötigt Repräsentationen von Zielen.
  - Noch schwächer: Der Beobachter billigt dem Agenten Ziele zu.

MEU: Der Agent benötigt also eine Nutzenfunktion.

Konsistenz: Mindestens wird konsistentes Verhalten erreicht. Generally, if an agent would perform two different actions with the same knowledge in two identical (environmental) situations, it is not said to be fully rational.

Der Beobachter billigt dem Agenten Ziele zu: Beispiel Thermostat (hat das Ziel, die Raumtemperatur konstant zu halten). Diese Beschreibung von technischen Systemen mittels so genannter „mentalistischer Notationen“ (Ziele, Absichten, Glauben, Emotionen...) wurde vom Philosophen D. Dennet als „intentional stance“ eingeführt. Wenn Systeme hinreichend komplex sind, ist es adäquat sie über den intentional stance zu beschreiben. Beispielsweise ein Computerschachprogramm „will gewinnen“.

Anwendungsfälle für Rationalität

- **Handel auf elektronischen Märkten:**
  - für welches Produkt entscheiden? Wie viel dafür bieten?
- **Strategische Spiele:**
  - welches ist der beste Zug (bei begrenzter Bedenkzeit)?
- **Verhalten in unbekannten Situationen:**
  - Erfahrungswissen, Faustregeln, Normen, etc. anwenden
- **Entscheidung über Teilnahme an Lotterie:**
  - Wahrscheinlichkeiten, Erwartungswerte, Risikobereitschaft

Bezüglich des Bietverhaltens auf Auktionen wird meine Entscheidung u.U. auch durch das Bietverhalten der Anderen beeinflusst.

welches ist der beste Zug? – Begrenzte Rationalität im Schach: Bedenkzeit reicht nicht aus, um sämtliche Kombinationen zu berechnen.

Verhalten in unbekannten Situationen: z.B. Krankheitsfall in einem fremden Land.

Teilnahme an Lotterie: Entscheidung hängt sowohl von den Gewinnchancen als auch von meiner individuellen Risikobereitschaft ab.

Zusammenfassung Rationalität: „do the right thing“

- **Rational handeln bedeutet, basierend auf Wissen, Wahrnehmungen und Handlungsalternativen die beste Aktion auszuführen.**

- **Fragestellungen und Probleme dabei:**
  - Wie lässt sich der Nutzen messen und bewerten?
  - Kurzfristigen oder langfristigen Nutzen optimieren?
  - Problem der unvollständigen und inkorreken Wahrnehmung...

Wissen kann dem Agenten vorgegeben sein oder auch durch vergangene Wahrnehmungen (durch Speicherung / Caching) erlernt worden sein.

Rationales Handeln setzt nicht unbedingt Denken voraus – beispielsweise können (antrainierte) Reflexe rational sein (z.B. das Erschlagen einer Mücke nach einem Stich; Ducken oder Fortrennen bei Gefahr).

Was ist die beste Aktion? Diejenige, die den Nutzen des Agenten optimiert. Zur Beurteilung der Rationalität wird eine Nutzen- oder Performance-Funktion benötigt.

Nutzenfunktion wird bei Russel/Norvig mit dem besseren Begriff „performance measure“ bezeichnet, der ausdrückt, dass das Ergebnis eines Agenten hinsichtlich seiner Performance messbar sein muss. Wie im Beispiel des reflexartigen Handelns muss der Agent diese Nutzenfunktion nicht selbst besitzen und anwenden; es reicht aus, dass dies ein Beobachter tut. Die Nutzenfunktion ist dann nur implizit vorhanden.

Beispiel einer Performance-Funktion eines Taxifahrers: sicher, schnell, regelgerecht, komfortabel, Profitmaximierung

Problem der unvollständigen und inkorreken Wahrnehmung... führt zum nächsten Thema, dem Environment (Umgebung)...

Gliederung

- Die Lehrveranstaltung GKI im Überblick
- Was ist Künstliche Intelligenz?
- Geschichtlicher Überblick
- Intelligente Softwareagenten
  - Rationales Handeln
  - Umgebung des Agenten
  - Agentenarchitekturen
- Zusammenfassung und Ausblick

Typen von Umgebungen

- **Vollständig vs. partiell beobachtbar**
  - kann Umgebung jederzeit vollständig wahrgenommen werden?
- **Deterministisch vs. stochastisch**
  - hängt der Folgezustand ausschließlich von der Aktion ab?
- **Episodisch vs. sequentiell**
  - beeinflusst eine Aktion nur eine begrenzte (zeitliche) Episode oder möglicherweise die gesamte Zukunft?

Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.

Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)

Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself. Episodische Umgebungen sind sehr viel einfacher, und es kann der Nutzen (-> Rationalität) sehr viel schneller beurteilt werden als in sequentiellen Umgebungen (z.B. im Schach: wie gut ist ein Eröffnungszug?)

Typen von Umgebungen

- **Statisch vs. dynamisch**
  - ändert sich die Umgebung, obwohl der Agent nicht handelt?
- **Diskret vs. kontinuierlich**
  - sind Zustände, Wahrnehmungen, Aktionen und Zeit endlich abzählbar?
- **Ein Agent vs. viele Agenten**
  - Komplexität entsteht bei mehreren Agenten: Konkurrenz, Kooperation, Kommunikation, Koordination

Static (vs. dynamic): Z.B. wenn der Agent gerade über die nächste Aktion nachdenkt (Beispiel Taxifahrer). The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does

Discrete (vs. continuous): Zeit ist kontinuierlich, kann aber oftmals in diskrete Intervalle zerlegt werden.

Umgebungstypen anhand von Beispielen

	Kreuzwort-rätsel	Schach m. Uhr	Taxi fahren	Bild-analyse
Beobachtbar	√	√	–	√
Deterministísch	√	strategisch	stochastisch	√
Episodisch	–	–	–	√
Statisch	√	semi	–	–
Diskret	√	√	–	–
Ein Agent	√	–	–	√

Die reale Welt ist partiell beobachtbar, nicht deterministisch (stochastisch), nicht episodisch (sequentiell), dynamisch, kontinuierlich und bevölkert mit vielen Agenten.

Gliederung

- Die Lehrveranstaltung GKI im Überblick
- Was ist Künstliche Intelligenz?
- Geschichtlicher Überblick
- Intelligente Softwareagenten
  - Rationales Handeln
  - Umgebung des Agenten
  - Agentenarchitekturen
- Zusammenfassung und Ausblick

Vom Agentenverhalten zur Agentenarchitektur

⇨ **Ein Agent führt Handlungen aufgrund von Wahrnehmungen aus:**

$$[f : \mathcal{P}^* \rightarrow \mathcal{A}]$$

- Eigenschaften der Umgebung beeinflussen die Wahrnehmung
- Die Handlungen sollen rational sein.

⇨ **Agent = Architektur + Programm**

- Das Agentenprogramm läuft auf der Architektur und realisiert *f*
- Welche Architektur setzt die Anforderungen optimal um? ...

Bisher wurde über das Verhalten von Agenten geredet, nun geht es um Strukturen.

Wahrnehmung ist nie 100%ig vollständig und korrekt, mehr zu diesen Problemen in VL #7 und 8.

Als Funktion bildet der Agent eine Menge von Perceptions (Wahrnehmungen) auf Handlungen ab.

An agent can perceive its own actions, but not always it effects.

Ein sehr einfaches Agentenprogramm

- **Der Agent besitzt eine Tabelle, in der sämtliche möglichen Wahrnehmungsseqenzen auf Aktionen abgebildet werden**
  - Beispiel Staubsaugeragent: Position x Zustand → Aktion
  - Beispiel Schach: Jede Stellung kennt einen optimalen Zug
  - Problem: Die Tabelle hätte ca. 10^{150} Einträge
- **Die Struktur (Architektur) eines Agenten beeinflusst offensichtlich dessen Performance...**
  - Strukturen & Algorithmen zur Repräsentation & Verarbeitung großer Zustandsräume werden später (→ Suchprobleme) behandelt.

Der Zustandsraum des Staubsaugeragenten wäre erheblich kleiner als beim Schach, denn Position-Zustands-Tupel wären auf eine der 4 Aktionen abzubilden. Jedoch in echten Räumen (die kontinuierlich sind und praktisch unendlich viele Positionen kennen) würde die Tabelle wiederum stark (quadratisch) anwachsen.

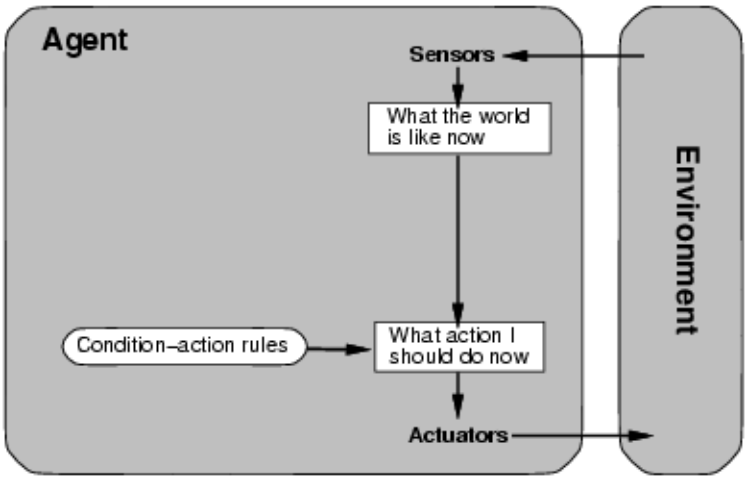
4 Agententypen im Überblick

- ⇨ **Einfache reaktive Agenten**
- ⇨ **Zustandsbasierte reaktive Agenten**
- ⇨ **Zielorientierte Agenten**
- ⇨ **Nutzenorientierte Agenten**

... mit wachsender Komplexität (werden im Folgenden vorgestellt).
Reaktive Agenten werden bei Russell/Norvig als Reflexagenten bezeichnet.

Einfache reaktive Agenten

... wählen eine Aktion aufgrund der aktuellen Wahrnehmung durch Anwendung einer passenden Regel.



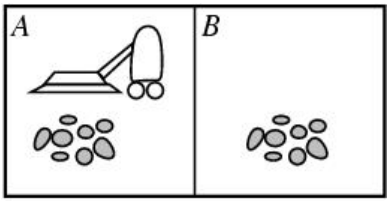
... funktionieren nur in beobachtbaren Umgebungen

Das Verhalten des Agenten ist direkt an die Umgebung gebunden.

state ← INTERPRET-INPUT(*percept*)
rule ← RULE-MATCH(*state*, *rule*)
action ← RULE-ACTION[*rule*]
return *action*

Will only work if the environment is *fully observable* otherwise infinite loops may occur.

Einfache reaktive Agenten, Beispiel Staubsauger



function REFLEX-VACUUM-AGENT ([*location, status***])**

if *status* == *Dirty* **then return** *Suck*

else if *location* == *A* **then return** *Right*

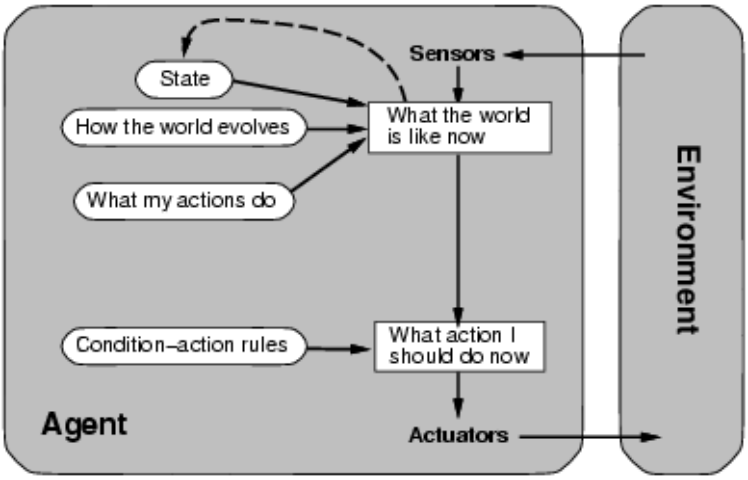
else if *location* == *B* **then return** *Left*

Regeln sind viel effizienter aufschreibbar als eine zustandsorientierte Tabelle (siehe oben), weil nicht in jeder Regel ein gesamter Zustandsvektor beschrieben werden muss und auch nicht für jedes mögliche Tupel eine Regel existieren muss. Z.B. die erste Regel fragt nur den Status, nicht aber den Ort ab.

Zustandsbasierte reaktive Agenten

... verwalten einen internen Zustand.

Dieses Weltmodell wird zyklisch aktualisiert auf Basis der Aktionen und Wahrnehmungen.



... funktionieren in partiell beobachtbaren Umgebungen

Zustandsbasierte Agenten werden auch als modellbasiert bezeichnet, wenn sie ein Welt*modell* aufbauen und pflegen.

Die Wahrnehmungen beeinflussen über eine Zustandsaktualisierungsfunktion den internen Zustand des Agenten. Im einfachsten Fall werden die Wahrnehmungen einfach in den Zustandsspeicher hinzugefügt. Aber auch komplexere Aktualisierungsfunktionen sind möglich, die veraltetes Wissen löschen oder Widersprüche erkennen und behandeln.

Das Handeln erfolgt zustandsabhängig. Auch hier kommen typischerweise Regeln zum Einsatz, nur dass diese nicht durch die Perceptions gefeuert werden, sondern über Eigenschaften des internen Zustands. Insbesondere im realistischen Fall, dass nicht in jedem „Verarbeitungszyklus“ eine vollständige Wahrnehmung stattfinden kann (z.B. ein Roboter mit einem 180%-Blick nimmt eben nur maximal die Hälfte seiner Umgebung wahr) ist es von Vorteil, wenn vorher gesammeltes Wissen noch abrufbar ist (z.B. wenn ein Roboter sich dreht, kennt er die Objekte / den Zustand der Umgebung in seinem Rücken weiterhin).

Zustandsbasierte reaktive Agenten

function REFLEX-AGENT-WITH-STATE(*percept***)**

static: *rules*, **a set of condition-action rules**

state, **a description of the current world state**

action, **the most recent action.**

state ← **UPDATE-STATE(***state*, *action*, *percept***)**

rule ← **RULE-MATCH(***state*, *rule***)**

action ← **RULE-ACTION[***rule***]**

return *action*

rot gefärbt sind Ergänzungen zum einfachen reaktiven Agenten.

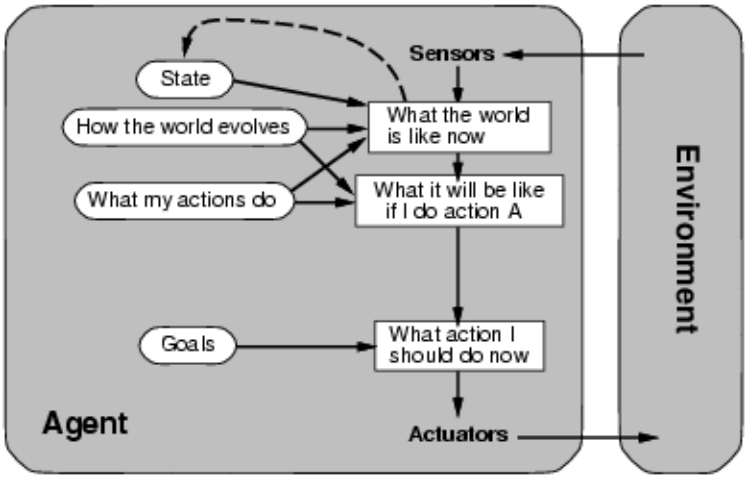
Der Folgezustand ist abhängig vom aktuellen Zustand, der gerade ausgeführten Aktion und den aktuellen Wahrnehmungen. Von der Aktion deshalb, weil der Agent in vielen Fällen die Effekte der Aktion auf die Umgebung kennt (z.B. ein Buch kaufen → weniger Geld im Portemonaie und irgendwann das Buch besitzen).

Das Matchen der Regeln bezieht entsprechend den Zustand mit ein. Dies bedeutet aber auch erhöhten Aufwand, denn je umfangreicher der Zustand (= das Wissen) des Agenten ist, desto komplexer (vom Rechenaufwand her gesehen) ist das Regel-Matching.

Zielorientierte Agenten

... verwalten Ziele, um *wünschens-werte* Situationen zu erreichen.

... beziehen die Zukunft mit ein.



... sind flexibler aufgrund des wissensbasierten Ansatzes

auch: zielbasierte Agenten.

Ein Ziel ist ein Zustand, den der Agent noch nicht erreicht hat (z.B. Buch besitzen). Ein zielorientierter Agent wird immer eine solche Aktion wählen, die entweder ein Ziel erreicht (z.B. Buch kaufen) oder zumindest ihn dem Ziel näher bringt (z.B. nach Buchläden suchen).

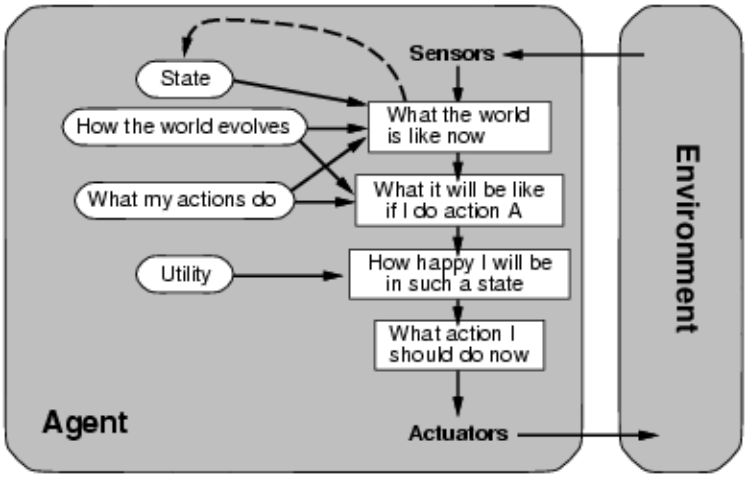
Derartige Agenten handeln also, anders als die bisher behandelten, nicht nur in der Gegenwart (und Vergangenheit) sondern versuchen, die Zukunft (ausgedrückt durch Ziele) „aktiv zu gestalten“. Ein weiterer Vorteil von Zielen im Agenten ist, dass der Agent seine Handlungen rechtfertigen/begründen kann bzw. das Verhalten des Agenten prognostiziert werden kann (sofern der Agent nicht einander widersprechende Ziele verfolgt).

Mehr zu diesem Thema an den Vorlesungsterminen zu intelligenten Agenten.

Nutzenorientierte Agenten

... gewichten Zustände mit Werten.

... können Ziele und Aktionen entsprechend priorisieren.



... eignen sich für Konfliktsituationen und Konkurrenz

auch: nutzenbasierte Agenten.

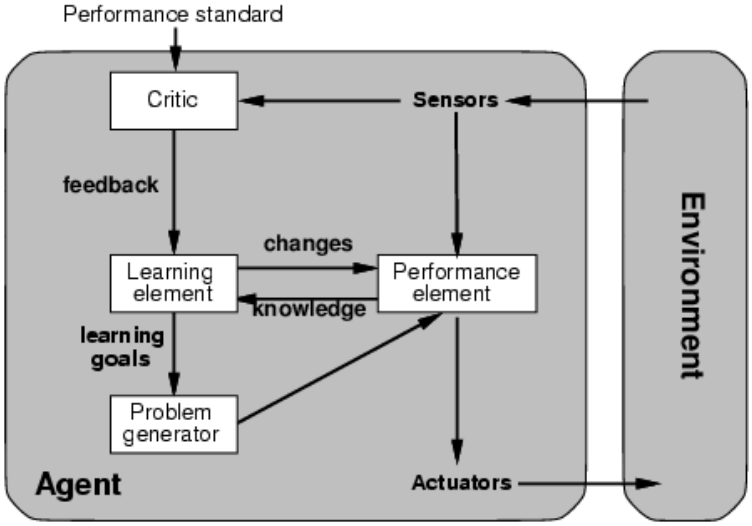
Während zielorientierte Agenten gut in kooperativen Situationen handeln, verhalten sie sich in nicht kooperativen Szenarien oftmals nicht intelligent. Denn blindes Verfolgen eines Ziels gelingt nur unter gewissen Annahmen (z.B. dass keine so genannten antagonistische Agenten vorhanden sind, die gegensätzliche Ziele haben). In Konkurrenzsituationen (z.B. 2 Agenten, die gegeneinander Schach spielen) würde ein rein zielorientierter Agent versuchen, mit möglichst wenigen Zügen den gegnerischen König zu schlagen, jedoch dabei nicht berücksichtigen, dass der Gegner ebenfalls dieses Ziel hat und sich gegen die Züge und Bedrohungen auch verteidigen kann.

Erst wenn der Agent seine möglichen Aktionen unter Berücksichtigung seiner Ziele und des aktuellen Zustands bewerten kann (einen reellen Nutzenwert zuweisen kann), ist er in der Lage, auch Konflikte / Konkurrenzsituationen intelligent/adäquat zu bewältigen.

Lernende Agenten

... erlernen die Auwahl der besten Aktion.

Jede der vorher beschriebenen Architekturen lässt sich um Lernverfahren erweitern.



... stärken die Performance und Robustheit.

Critic provides feedback on agents performance based on fixed performance standard.

Learning element: introduce improvements in performance element

Performance element: selecting actions based on percepts – entspricht einer der vorher beschriebenen Agentenarchitekturen.

Problem generator: suggests actions that will lead to new and informative experiences.

Lernen (auch: Adaption/Adaptivität) bedeutet, seine Leistung über die Zeit zu verbessern. Viele Lernverfahren existieren und werden im 2. Teil dieser Lehrveranstaltung behandelt. Agenten, die in eine Umbegung eingebettet sind, lernen typischerweise auf Basis von Feedback (dieser Umgebung).

Gliederung

- ⇨ Die Lehrveranstaltung GKI im Überblick
- ⇨ Was ist Künstliche Intelligenz?
- ⇨ Geschichtlicher Überblick
- ⇨ Intelligente Softwareagenten
- ⇨ Zusammenfassung und Ausblick

Zusammenfassung

- **Starke KI zielt auf generelles intelligentes Verhalten, schwache KI auf intelligente Anwendungen**
  - Geschichtliche Entwicklung von generischen, *schwachen* Methoden zu spezialisierten, *starken* Methoden
- **Ein Agent führt Handlungen aufgrund von Wahrnehmungen aus**
  - Rationale Agenten handeln zielgerichtet / nutzenoptimierend
  - Die Umgebung beeinflusst das Verhalten des Agenten
  - Die Agentenarchitektur definiert die Struktur eines Agenten

Wo stehen wir heute?

- **Turing-Test als Benchmark**
  - teilweise auch von nicht intelligenten Systemen bestanden
- **Schachcomputer schlagen Großmeister**
- **Einsatz von Expertensystemen in bestimmten Bereichen**
- **Aber: allgemeine Intelligenz nicht erreicht**
  - Grundschulfertigkeiten im Lesen, Schreiben und Zusammenfassen

Schachcomputer arbeiten aber anders als Großmeister (nicht intuitiv) sondern hauptsächlich rechnend (sehr viele Stellungen berechnen und beurteilen).

In relativ geschlossenen Welten sind Expertensysteme als Einagentensysteme gut geeignet, um das Wissen menschlicher Experten abzubilden. Expertensysteme finden Einsatz z.B. in der Diagnose von Anlagen/Maschinen und in der Medizin.

DAI-Labor
TU Berlin

Grundlagen der Künstlichen Intelligenz

20.10.2005: Einführung in die KI

Nächster Termin: Problemlösen 1

Dr.-Ing. Stefan Fricke
stefan.fricke@dai-labor.de

A|O|T
Agententechnologien in betrieblichen Anwendungen und der Telekommunikation

Referenzen

- S. Russell, P. Norvig Artificial Intelligence: A Modern Approach Prentice Hall, 2003, Second Edition

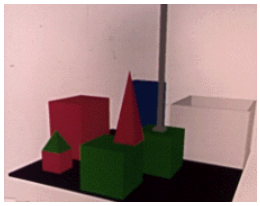
Anhänge

⇨ **Nicht verwendete Folien**

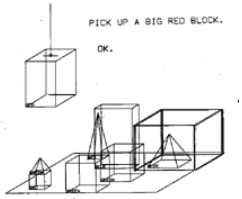
SHRDLU

Winograd (M.I.T., 1968-70)

⇨einfacher Dialog in natürlicher Sprache über Klötzchenwelt

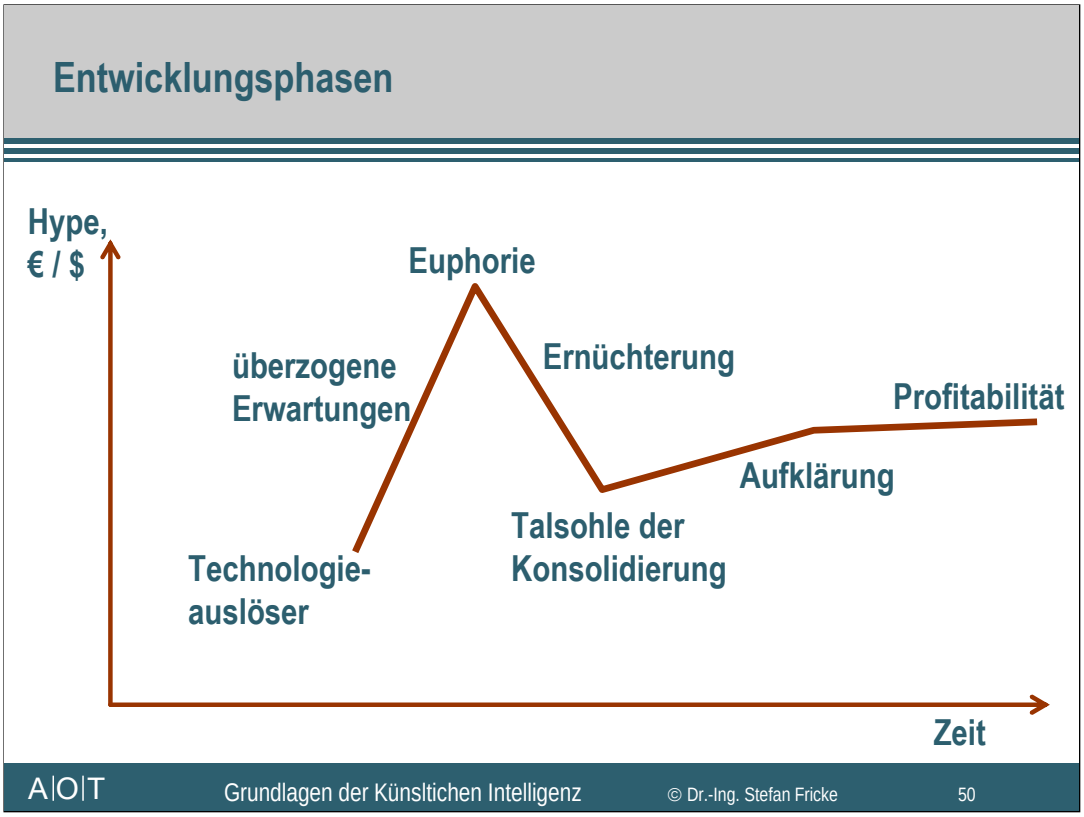


Person: PICK UP A BIG RED BLOCK.
Computer: OK. (does it)
Person: GRASP THE PYRAMID.
Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
Person: FIND A BLOCK WHICH IS TALLER THAN THE ONE YOU ARE HOLDING AND PUT IT INTO THE BOX.
Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.
Computer: OK. (does it)



Natürliche Intelligenz vs. Künstliche Intelligenz

- ⇨ Nicht jedes intelligente Verhalten beruht auf (logischem) Denken
- ⇨ Worin liegt der Nutzen des Denkens? Welche Gedanken sollte man haben?
- ⇨ Symbol grounding problem bei reinen Software-Anwendungen



Die Entwicklungsphasen bei innovativen Produkten (z.B. auch: Electronic Commerce, UMTS) verlaufen üblicherweise zunächst sehr steil (gemessen am Hype, an Publikationen, an Forschungsgeldern, an Börsenkursen, etc.), beginnend mit einem Technologieauslöser (hier: u.a. LISP). Viele springen auf den Zug, erkennen die Potenziale und kommunizieren diese. Unterlegt werden diese Visionen durch Prototypen, die die Machbarkeit aufzeigen (GPS, ELIZA, SHRDLU, KI-Planer, Constraint-Solver). Die sehr steile Entwicklung gipfelt in der Euphorie, um danach ebenso steil abzustürzen, durch offensichtlich überzogene Erwartungen ins Tal der Desillusion. Diese Phase der Ernüchterung dient gleichzeitig auch als Konsolidierung, die Spreu wird vom Weizen getrennt. Danach steigt die Erfolgskurve langsam wieder an (Phase der Etablierung/Aufklärung): tatsächliche Potenziale werden erkannt und ausgenutzt. In der abschließenden Phase der Profitabilität/Produktivität steigt die Kurve noch schwächer an bzw. stagniert.