

Verteilte Künstliche Intelligenz

(Kurze Einführung)

Abstract zum Vortrag

Ilia Revout 06.2005

Inhaltsverzeichnis

1	Was ist VKI?	2
2	Agenten	4
2.1	Softwareagenten	5
2.2	Deliberative und Reaktive Agenten	6
2.2.1	Deliberative Agenten	6
2.2.2	Reaktive Agenten	7
2.3	Stationäre und Mobile Agenten	8
2.3.1	Stationäre Agenten	8
2.3.2	Mobile Agenten	8
2.3.2.1	Konzept der mobilen Agenten	9
2.3.2.2	Migration	9
2.3.2.3	Anforderungen an das System	10
2.3.2.4	Kommunikation	10
3	Ferienclub	12

Kapitel 1

Was ist VKI?

Die Verteilte Künstliche Intelligenz (VKI) versucht die Defizite der klassischen KI im Bezug auf die Entwicklung intelligenten Agenten auszugleichen. Die Schwerpunkte liegen dabei bei:

- Entwicklung von Organisationsstrukturen
- Problemlösungsstrategien
- Kooperation- und Koordinationmechanismen

In der Abbildung 1.1 ist die seit 1988 verbreitete Klassifikation der VKI von Bond und Gasser [Bond/Gasser 1988] dargestellt.

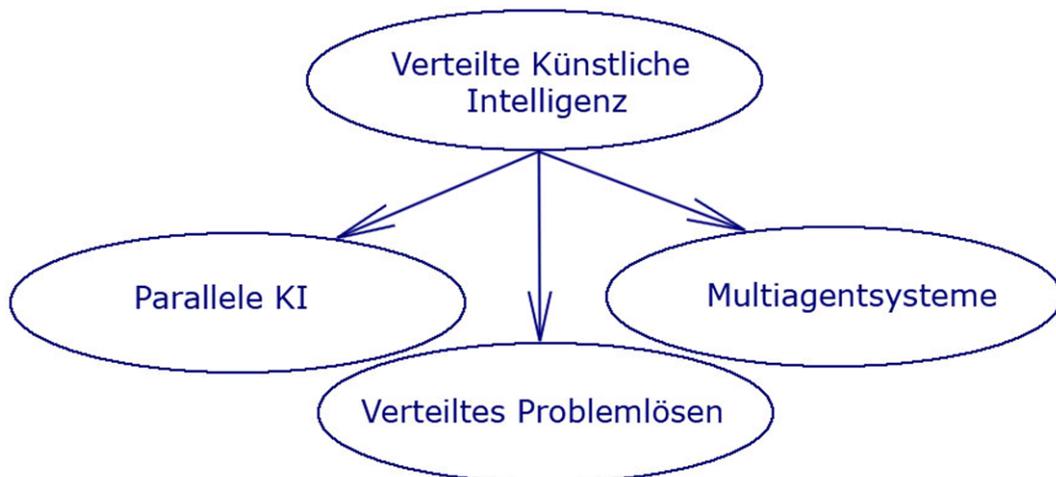


Abbildung 1.1: Die Klassifikation der VKI

Die Schwerpunkte diesen Unterklassen sind folgendermaßen verteilt:

-
- Parallele KI untersucht inwieweit sich die komplexen Problemstellungen allein durch deren Verteilung auf eine größere Zahl von Ressourcen beschleunigen lässt. Dabei spielen die verteilte Problemlösungsstrategien oder Kooperation- und Koordinationmechanismen eine untergeordnete Rolle.
 - Beim Verteilten Problemlösen steht der Kooperationsaspekt im Vordergrund. Es wird untersucht, inwieweit ein gemeinsames Problem mit Hilfe von mehreren unabhängigen Modulen gelöst werden kann. Schwerpunkt liegt also in der Entwicklung von Algorithmen, die Probleme in Teilprobleme zerlegen können und in der Lage sind die Teilergebnisse zu einem Gesamtergebnis zu verbinden. Dabei handelt es sich um Top-Down Verfahren, da das zu lösende Problem von Anfang an bekannt werden muss. Die häufigste, in diesem Bereich gestellte Frage ist, wie findet die Kommunikation zwischen einzelnen Modulen statt. Die drei Verfahren, die am verbreitetsten angewendet werden, sind:
 - Blackboard - Prinzip
 - Kontraktnetz System
 - Partial Global Planning
 - Multiagenten-Systeme setzen voraus, dass die Agenten auch über eigene Handlungsfähigkeit verfügen und sich somit selbst koordinieren. Anders als beim Verteilten Problemlösen handelt es sich hier um einen Bottom-Up Prozess. Der Vorteil liegt in der Tatsache, dass die Agenten die Probleme lösen können, die während der Implementierung nicht vorgesehen waren.

Kapitel 2

Agenten

„Agent: Eine Person oder Sache, die in der Lage oder ermächtigt ist, im Auftrag dritter zu handeln.“ [Caglayan 1998]

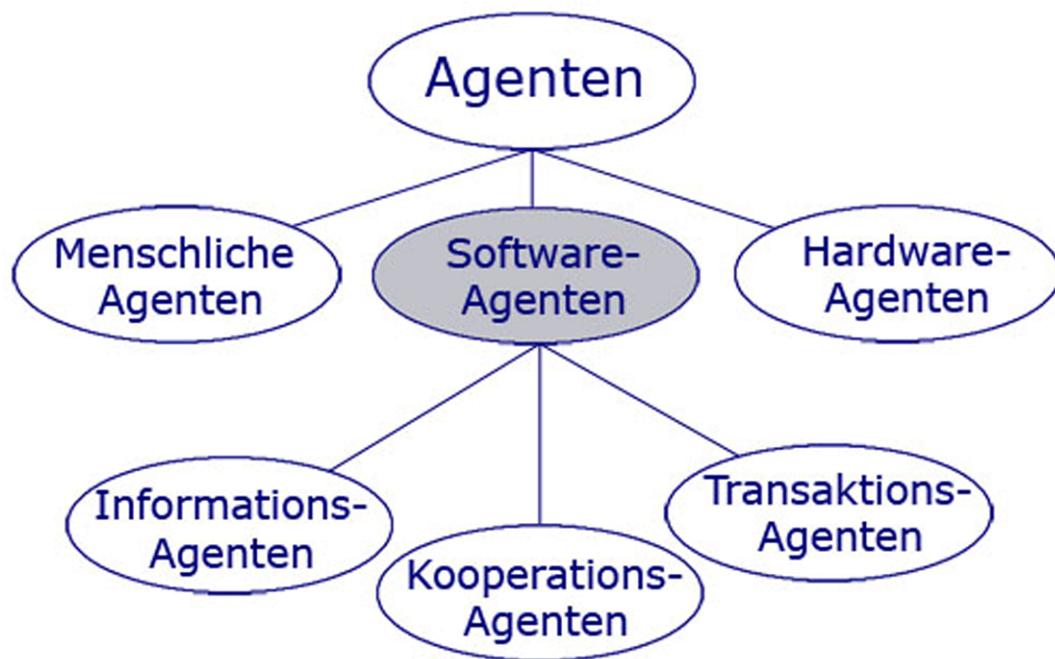


Abbildung 2.1: Kategorien von Agenten

2.1 Softwareagenten

Agenten werden oft nach Kategorien unterteilt. Die Abbildung 2.1 zeigt typische Kategorien der Agenten. Der Schwerpunkt dieses Vortrages liegt bei Softwareagenten, die wiederum weitere Unterteilung haben:

- Informationsagenten
 - Unterstützen den Benutzer bei der Suche nach Informationen in verteilten Systemen bzw. Netzwerken.
 - Informationsquelle finden
 - Informationen daraus extrahieren
 - Die Gesamtmenge abhängig von dem Benutzer filtern
 - Ergebnisse in einer anschaulichen Form präsentieren
- Kooperationsagenten
 - Hauptaufgabe besteht darin, komplexe Problemstellungen durch die Kommunikation und Kooperation mit anderen Objekten wie Agenten, Menschen oder externen Ressourcen zu lösen.
 - Kommen dann zum Ansatz, wenn das Problem von einem einzigen Agent nicht gelöst werden kann
 - Viel komplexer als Informationsagent
- Transaktionsagenten
 - Aufgabe besteht darin, die Transaktionen durchzuführen oder zu überwachen
 - Kauf von Produkten mit Hilfe einer Kreditkarte des Benutzers
 - Sicherheit, Datenschutz, Robustheit, Vertrauenswürdigkeit

„Ein Agent ist ein Computersystem, welches in einer Umgebung arbeitet, in der das System autonom Aktionen ausführen kann, um die ihm gestellte Aufgabe zu erfüllen.“ [Jennings 1998]

Obwohl es unterschiedliche Definitionen von Softwareagenten gibt, ist es am vielversprechendsten, Agenten über ihre Charakteristika zu beschreiben. Es müssen mindestens sechs Charakteristika erfüllt werden:

- Autonomie : Agenten können ohne direktes Eingreifen eines Anwenders agieren und ihre Aufgaben erfüllen

- Soziale Fähigkeiten : Besonders wenn Agenten mit dem Anwender kommunizieren, müssen sie soziale Kompetenz besitzen
- Reaktives Verhalten : Agenten beobachten ständig ihre Umwelt und reagieren zeitnah auf Veränderungen
- Proaktives Verhalten: Agenten können ihre Umwelt auch selbst beeinflussen, um den gewünschten Zustand herzustellen.
- Kontinuität : Agenten bleiben über sehr lange Zeit hinweg aktiviert
- Zielorientiertheit: Agenten brauchen ein klares Ziel, welches sie erfüllen sollen.

2.2 Deliberative und Reaktive Agenten

Aus den Eigenschaften können zwei unterschiedliche Gruppen von Agenten gebildet werden. Diese Gruppen unterscheiden sich wesentlich in der Art der Wahrnehmung der Umwelt, in der die Agenten agieren.

2.2.1 Deliberative Agenten

Deliberative Agenten zeichnen sich dadurch aus, dass sie die gesamte Umwelt kennen. Dadurch sind sie auch in der Lage, bestimmte Schlussfolgerungen zu ziehen. Problematisch wird es dann, wenn die reale Welt sich ändert. Die dabei entstehenden Probleme liegen im Bereich von 'Erkennung, dass die reale Welt sich geändert hat' bis zu 'Änderung der abgespeicherten Welt'. Eine sehr verbreitete Architektur der deliberativen Agenten ist in der Abbildung 2.2 dargestellt.

- Überzeugungen sind die Ansichten bezüglich der Umwelt.
- Wünsche sind alle möglichen Aktivitäten, die der Agent machen könnte. Diese Aktivitäten können zueinander im Konflikt stehen oder sogar nicht realisierbar sein.
- Ziele sind die Wünsche, die auf die Realisierbarkeit geprüft sind. Die Ziele dürfen nicht mehr zueinander im Konflikt stehen.
- Intentionen sind die Ziele, die tatsächlich von dem Agent ausgeführt werden.
- Pläne sind die Intentionen, die für eine bestimmte komplexe Aufgabe notwendig sind.

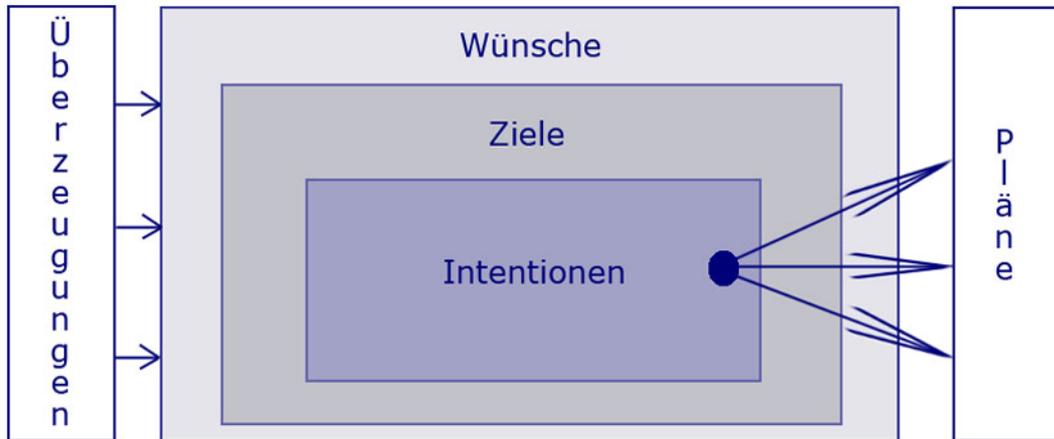


Abbildung 2.2: BDI - Architektur

2.2.2 Reaktive Agenten

Anders sieht es bei den reaktiven Agenten aus. Sie besitzen keine Information über die Umwelt. Mit Hilfe von Sensoren bekommen sie immer die Information über die reale Welt und sind in der Lage, auf diese Signale sofort zu reagieren. Die logische Schlussfolgerung ist dabei nicht möglich. In

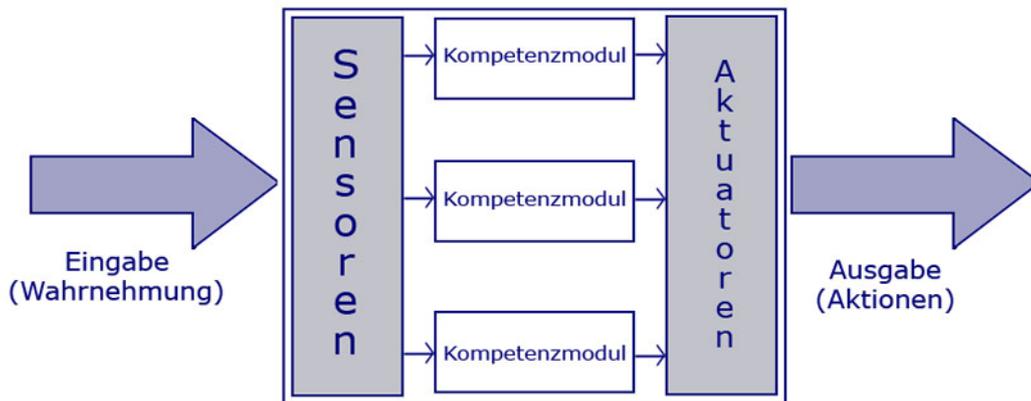


Abbildung 2.3: Reaktive - Architektur

der Abbildung 2.3 ist eine mögliche Architektur für die reaktiven Agenten dargestellt. Zu beachten sind die Kompetenzmodule. Jeder Kompetenzmodul ist komplett unabhängig von anderen Kompetenzmodulen. Damit wird gewährleistet, dass der Ausfall eines Modules nicht zum Absturz des gesam-

ten System führen kann, was typischer Fall bei den deliberativen Agenten ist.

Oft werden die deliberativen und reaktiven Agenten kombiniert, so dass die Vorteile beiden Gruppen ausgenutzt werden.

2.3 Stationäre und Mobile Agenten

Abhängig davon, ob die Agenten nur auf einem Rechner aktiv sind oder in der Lage sind, selbst zu entscheiden, wo sie ausgeführt werden, spricht man von stationären oder mobilen Agenten.

2.3.1 Stationäre Agenten

Stationäre Agenten verlassen nicht mehr den Rechner, wo sie gestartet wurden. Diese Agenten sind am meisten verbreitet, da für sie keine weitere Infrastruktur notwendig ist. Die typischen Kommunikationsprobleme sind an der Stelle auch für die stationären Agenten relevant. Sobald die Information nicht lokal vorhanden ist, muss der Agent nicht nur den Ort kennen, wo sie zu finden ist, sondern auch die Schnittstelle, über die man sie abfragen kann. Da die Schnittstellen meistens allgemein gehalten werden, bekommt der Agent unnötige Daten, die dann lokal ausgewertet werden müssen. Dieses Prozess verursacht nicht nur überflüssige Datentransfer, sondern auch den zusätzlichen Ressourcenverbrauch des lokalen Rechners.

2.3.2 Mobile Agenten

Mobile Agenten verhalten sich prinzipiell anders. Sie entscheiden selbständig, wo sie ihre Arbeit verrichten sollen. Falls eine entfernte Information gebraucht wird, geht der Agent selbst zu dem Rechner. Dieses Verhalten hat bestimmte Vorteile. Da die Schnittstelle nach Außen nicht mehr gebraucht wird, hat der Agent die Möglichkeit, die Abfrage für sich optimiert durchzuführen. Die Daten werden sofort auf dem entfernten Rechner selektiert, so dass zurück nur die gesuchte Information transportiert werden muss. Dabei reduziert sich die Menge der Daten, die durch den Netzwerk geschickt werden. Die Ressourcen des lokalen Rechner werden auch nicht mehr für unnötiges Selektieren verbraucht. Ein Hacken gibt es leider auch. Die Voraussetzung für ein freies Bewegen der Agenten zwischen Systemen ist, dass diese Systeme die dafür notwendige Umgebung zur Verfügung stellen.

2.3.2.1 Konzept der mobilen Agenten

Mobile Agenten sind nicht dem mobilen Code gleich zu stellen! In der Tabelle 2.4 sind die möglichen Paradigmen der Verteilung dargestellt. Mobile

Paradigma	Rechner 1	Rechner 2
Client - Server		Code Daten Prozessor
Remote Evaluation	Code	Daten Prozessor
Code on Demand	Daten Prozessor	Code
Mobile Agenten	Code Daten Prozessor	Code Daten* Prozessor

Abbildung 2.4: Mobiler Code

Agenten unterscheiden sich vom mobilen Code dadurch, dass nicht nur der Code oder die Daten allein versendet werden, sondern der Agent sich selbst (Programmcode) und die eigenen Daten (Zustand) von einem Rechner zu einem anderen überträgt. Dabei ist er in der Lage, auf jedem Rechner die Arbeit durchzuführen.

2.3.2.2 Migration

Die Fähigkeit, von einem Rechner zu einem anderen zu gehen, wird als 'Migration' bezeichnet. Wenn der Agent in der Lage ist, seine Arbeit exakt an der Stelle fortzusetzen, wo sie abgebrochen wurde, beherrscht er die starke Migration. Wenn die Ausführung immer an der gleichen Stelle fortgesetzt wird, ist die schwache Migration gemeint. Inzwischen ist die schwache Migration dominierend, da die Programmiersprachen dieses Konzept wesentlich stärker unterstützen.

2.3.2.3 Anforderungen an das System

Wie es oben schon erwähnt wurde, ist die Grundvoraussetzung für mobile Agenten eine spezielle Umgebung. Dieses System muss nicht nur in der Lage sein, die Agenten zu empfangen und zu starten, es soll auch:

- Muss Dienste und Betriebsmittel bereitstellen
- Monopolisten und Randalierer erkennen und sperren
- Festgelegtes Fehlerverhalten haben
- Kommunikation ermöglichen
- Sicherheit garantieren

Die Sicherheit spielt eine zentrale Rolle in diesem System. Dabei muss nicht nur die Sicherheit für das System selbst garantiert werden, sondern auch die Sicherheit für die Agenten. Die hier unter anderem gestellte Frage lautet: was passiert mit den Agenten, wenn das System abstürzen wird.

2.3.2.4 Kommunikation

Auch die Kommunikation ist bei mobilen Agenten nicht trivial. Wie kann erkannt werden, wo sich der Agent zur Zeit befindet, damit die Kommunikation mit ihm möglich wird. In der Abbildung 2.5 sind drei mögliche Verfahren dargestellt.

- Logging: Dabei hinterlässt der Agent eine Spur auf dem System, welches er gerade verlässt, mit der Information, wohin er weiter gegangen ist.
- Brute - Force : Dabei werden gleichzeitig alle bekannten Systeme abgefragt, ob der Agent sich gerade auf dem System befindet.
- Registrierung : Jeder Agent muss sich bei einer zentralen Stelle zuerst registrieren, bevor er mit der Arbeit auf einem bestimmten System anfangen kann.

Obwohl alle drei Verfahren auf den ersten Blick funktionieren sollen, tritt bei mobilen Agenten ein weiteres Problem auf. Die Agenten bewegen sich durch das Netz genau so schnell, wie die Nachrichten selbst.

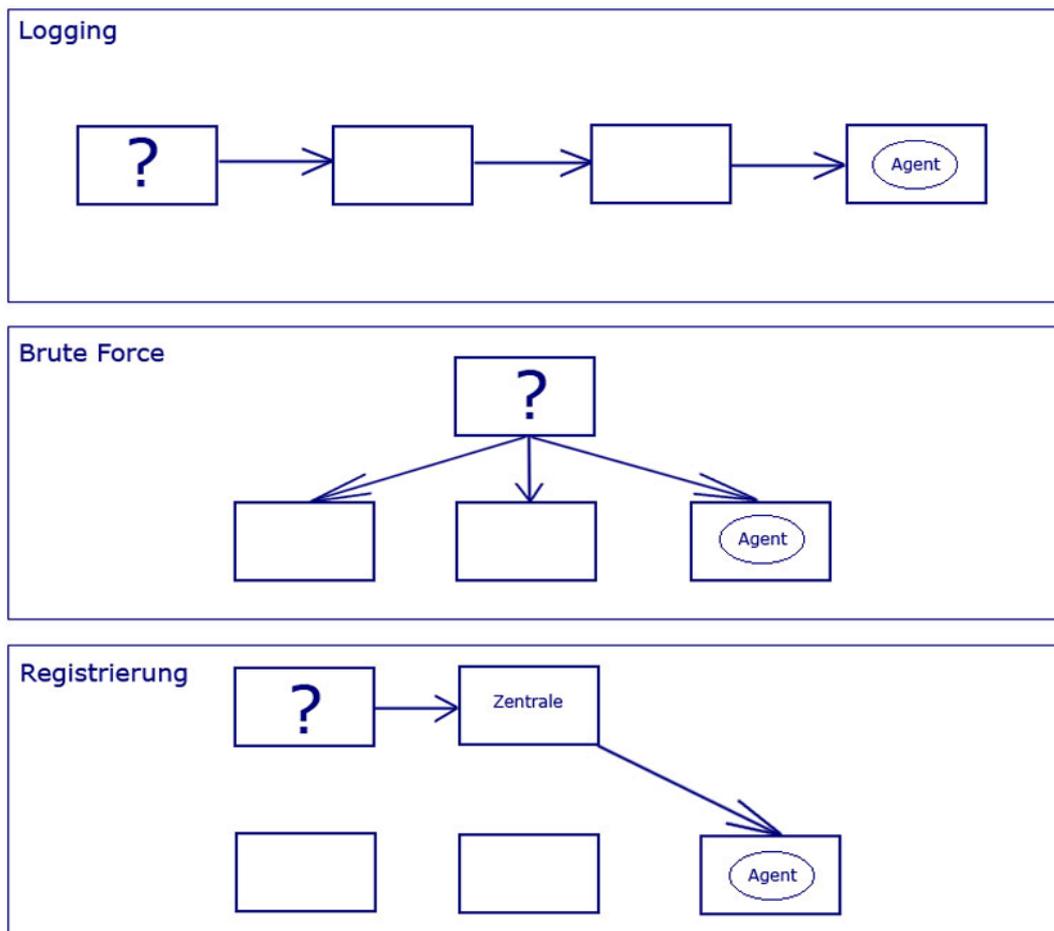


Abbildung 2.5: Lokalisierung

Kapitel 3

Ferienclub

Da ich mobile Agenten sehr spannend finde, werde ich im Projekt versuchen, selbst eine Plattform für mobile Agenten zu implementieren. Dabei soll ein Framework entstehen, mit dessen Hilfe mobile Agenten leicht implementiert werden sollen. Mit dem entstehenden Gesamtsystem soll eine Art der Programmierung mit Hilfe von mobilen Agenten getestet werden. Der Schwerpunkt soll dabei bei den Kommunikationsagenten (Programm - Mensch Kommunikation) liegen.

Literaturverzeichnis

- [Bond/Gasser 1988] Bond, A., Gasser, L. (Hrsg), Reading in Distributed Artificial Intelligence, Morgan Kaufman Publishers, San Mateo 1988
- [Caglayan 1998] Caglayan, Alper K.; Harrison, Colin G.: Intelligente Software-Agenten.Grundlagen, Technik und praktische Anwendung im Unternehmen;Hanser Verlag München Wien, 1998.
- [Jennings 1998] Jennings, Nicholas R.; Wooldridge, Michael J.: Applications of Intelligent Agents. In: Jennings, Nicholas R.; Wooldridge, Michael J. (Hrsg.): Agent Technology. Foundations, Applications, and Markets; Springer Verlag Berlin Heidelberg New York, 1998.
- [Rao/Georgeff 1995] Rao, A. S., Georgeff, M. P., BDI Agents: From Theory to Practice, in: Proceedings of the First International Conference on Multi-Agent-Systems (ICMAS), San Francisco 1995
- [Brooks 1986] Brooks, R.A., A robust layered control system for a mobile robot, in: IEEE Journal of Robotics and Automation, RA-2(1986)1, S, 14-23
- [iicm.edu] <http://www.iicm.edu/greif/images/node1.html>
- [Img1] <http://www.cnn.com/interactive/specials/9911/future.gallery/content/01.html>
- [wiki1] http://de.wikipedia.org/wiki/Deep_Blue
- [seti1] <http://seti-team-nickles.de>
- [iicm1] <http://www.iicm.edu/greif/node12.html>
- [mit1] <http://agents.www.media.mit.edu/groups/agents/projects/>