



1. Einführung
2. Eine Roadmap für CAgents
3. Testprojekt für CAgents

Simulation von Wissen -

Konstruktion eines kognitiven Agenten (1)

>> Achtung : Skript gibt den mündlichen Vortrag nur unvollständig wieder !!! <<<

Einführung

Nachdem die Frage nach der generellen Lernfähigkeit von universellen Turingmaschinen und dazu äquivalenten Formalismen grundsätzlich positiv beantwortet werden konnte, wurde für den weiteren Fortgang die kognitive Psychologie als ein interessanter Partner für die Informatik geortet. Es wurde das Paradigma der Psychoinformatik skizziert, unter dem die Informatik methodisch gezielt die Ergebnisse der kognitiven Psychologie nutzt, genauso wie umgekehrt, die kognitive Psychologie die Methoden der Informatik nutzt, um ihre eigene Theorenbildung zu verbessern.

In der ersten Überlegung nach einem möglichen Anwendungsgebiet war die Diagnose- und Therapiesituation des Arztes als mögliches interessantes Anwendungsszenarium identifiziert worden, dies abgewandelt und transformiert als die Situation eines Kindes, das lernen muss, wie man über die verschiedenen Aspekte dieser Welt spricht.

Im weiteren Verlauf der Analyse ergab sich aber noch ein weiteres Szenario, was sich für die weitere Arbeit anbietet. In diesem anderen Szenarium wird ein *kognitiver Agent* schrittweise aufgebaut, der mit einer definierten *Testwelt* kommuniziert.

Als Kontext für diese Überlegung sei hier das *planeearthsimulator-Projekt* angenommen (siehe <http://www.planeteearthsimulator.org>) zusammen mit dem Konzept des *smart ambient computing* des kommenden gleichnamigen Masterstudienganges des Fachbereichs 2 zusammen mit Fachbereich 4.

In diesem Projekt wird angenommen (siehe nachfolgendes Schaubild), dass von beliebigen Aspekten der *realen Welt* ein *logisches Modell (LModell)* erstellt werden kann. Diese LModelle werden in einem öffentlichen Pool von Modellen vorgehalten. Über einen Server, der als *Simulation Control (SC)* fungiert, können vorhandene LModelle für eine Simulation ausgewählt werden. Wurde ein LModell für eine Simulation ausgewählt, dann wird ein *Simulationsauftrag* erstellt, der dann einem Server übermittle wird, der als *Cluster Manager (CM)* fungiert. Der Clustermanager kann sovielen *Simulationsprozesse* starten, wie für den Simulationsauftrag notwendig sind. Zusätzlich zu den LModellen, die Weltprozesse repräsentieren, können auch noch spezielle *kognitive Agenten (Cognitive Agents, CAgents (CAs))* konstruiert werden. Auch diese

werden in einem Pool vorgehalten. Zu jedem LModell kann mindestens ein CAgent gestartet werden.

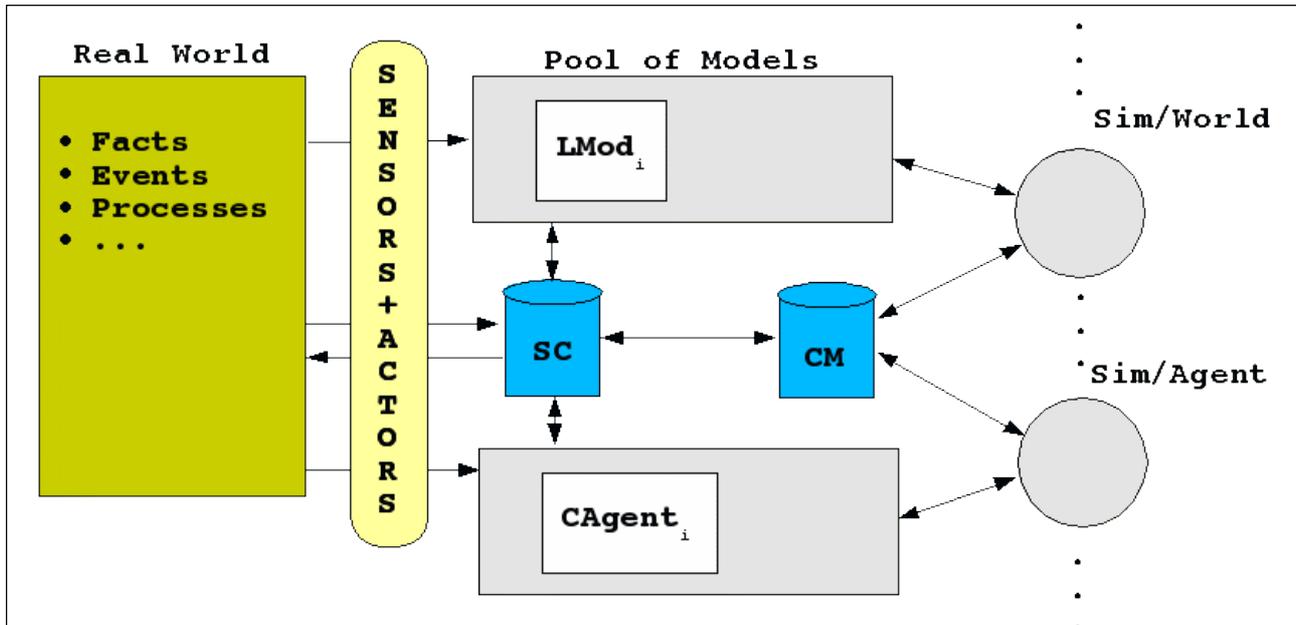


Bild: Erweitertes Planet Earth Simulator Projekt

Aus Sicht des Clustermanagers sind Simulatoren, die LModelle simulieren und Simulatoren, die CAgenten simulieren, gleichwertig; intern haben sie aber eine unterschiedliche Struktur; auch ist ihre Aufgabenstellung verschieden. Ein CAgent versucht, *menschenähnliche Kommunikation* so realistisch wie möglich anzunähern. Zu diesem Zweck werden alle Strukturen zur Verfügung gestellt, die dazu notwendig sind. Ein LModell dagegen repräsentiert im allgemeinen nur generische Veränderungsprozesse.

Die Verbindung zum *Paradigma der Psychoinformatik* ist wie folgt zu sehen: die Beziehung zwischen kognitiver Psychologie und Informatik ist nicht nur uni-direktional, sondern bi-direktional. Die Informatik wartet nicht nur

<i>Phase</i>	<i>Experiment</i>	<i>Theorie</i>
1		Kognitionspsychologisches Modell vorher
2	Kognitionspsychologisches Experiment mit menschlichen VPNS	
3		Kognitionspsychologisches Modell nachher
4		Informatik Modell vorher



Phase	Experiment	Theorie
5	Kognitionspsychologisches Experiment mit künstlichen VPNs	
6		Informatik Modell nachher
7		Kognitionspsychologisches Modell nachher

'passiv' auf Modellierungsanforderungen seitens der kognitiven Psychologie --obgleich hier natürlich eine Überfülle an Daten und partiellen Modellen vorliegt--, sondern sie kann auch --ganz analog der Rolle der Mathematik im Falle der Physik-- aktiv Modelle entwickeln, die dann die kognitive Psychologie nutzen kann, um ihre experimentellen Situationen unterstützend zu modellieren.

In diesem Sinne versteht sich das zuvor skizzierte Szenario als ein *Meta-Modell* für eine ganze Fülle möglicher konkreter Modelle für die kognitive Psychologie.

Eine Roadmap für CAgents

Das Ziel des weiteren Vorgehens besteht nun also darin, im Sinne des so modifizierten Paradigmas der Psychoinformatik unter Berücksichtigung des Konzepts des *PlanetEarthsimulator*-Projektes und des Paradigmas des *smart ambient computing* Schritt für Schritt sowohl einen CAgenten aufzubauen wie auch einen dazu passenden Pool von LModellen. Das Ganze kann man als eine Art *Roadmap* auffassen, die festlegt, welche Eigenschaften eines CAgenten wann erreicht sein sollen.

Erster Entwurf einer Roadmap für CAgenten:

Phase	Sim/Welt	Sim/CAgent	Memory von Sim/CAgent
1	Einfaches Echo	Kann Nachrichten senden und empfangen	Leer
2	Die Welt besteht aus <i>Objekten</i> , die <i>Positionen</i> in einer <i>Fläche</i> einnehmen; jedes Objekt kann <i>Eigenschaften</i> haben	<i>Bewegungen</i> sind <i>zufällig</i> ; Objekte für <i>Nahrungsaufnahme</i> können <i>erkannt</i> und <i>konsumiert</i> werden; Energieverlust unter einem Schwellwert führt zum <i>Tod</i>	Leer
		Bewegungen können <i>Gedächtnisinformtionen</i> nutzen	<i>Imitierendes passives</i> Gedächtnis für Positionen und Objekte an diesen Positionen
			<i>Imitierendes und aktiv expandierendes</i> Gedächtnis; Unterscheidung von ' <i>registriert</i> ' und <i>konstruiert</i>

Phase	Sim/Welt	Sim/CAgent	Memory von Sim/CAgent
3	Eine Welt als ' <i>Stockwerk</i> ' : Flure und Zimmer mit Türen, Fenstern, Möbeln, Personen, Tieren, Pflanzen.	Forts.	Zusätzlich <i>Konzeptbildung</i> unter Voraussetzung von Objekten mit Eigenschaften, die Positionen in einer Fläche einnehmen können; Konzepte sind <i>abstrakt</i> und können <i>rekursiv</i> sein. Objekte sind konkret und nicht rekursiv
4	Zusätzlich <i>sprachliche Ereignisse</i>	Zusätzlich <i>Sprechakte</i>	<i>Verknüpfung</i> von <i>sprachlichen Ereignissen</i> mit <i>Konzepten</i> durch Lernen

Dieser Entwurf einer Roadmap ist noch sehr grob. Der Aspekt der *Planung*, die Rolle möglicher *Emotionen* und die *Wahrnehmung* sind kaum bis garnicht spezifiziert.



Ein erstes Testprojekt für CognitiveAgents^{*)}

Um die Roadmap umzusetzen, wird jetzt ein erstes Testprojekt definiert, das als Teilprojekt zum aktuellen Planet Earth Simulator (PES) Projekt organisiert wird.

Dieses Testprojekt wird *SimWorld + SimCognitiveAgent* genannt. Der gemeinsame Rahmen für *SimWorld + SimCognitiveAgent* ist eine Umgebung im Rahmen des MPI-Protokolls (MPI := Message Passing Interface) in der Implementierung von LAM (:= Local Area Multicomputer)(URL: <http://www.lam-mpi.org/>) oder MPICH (URL: <http://www-unix.mcs.anl.gov/mpi/mpich/indexold.html>)(Beide Versionen sind Teil der SuSe-Linux Distribution version 9.0).

Der MPI-Standard Version 1.1 wurde im Juni 1995 veröffentlicht. Die Erweiterungen MPI-1.2 and MPI-2 enthalten "process creation and management, one-sided communications, extended collective operations, external interfaces, I/O, and additional language bindings." Es handelt sich um eine Softwareumgebung für paralleles Rechnen auf dezidierten Parallelrechner, heterogenen Netzen wie auch speziellen Grid-Netzen. Die meisten gängigen Plattformen werden unterstützt.

Im einfachen Fall wird ein *MPI-Frame* geladen. Innerhalb des MPI-Frames kann man dann nahezu beliebig viele Prozesse dadurch starten, dass man eine *MPI-Startdatei* angibt, die eine Liste von Nodes enthält, auf der Prozesse gestartet werden sollen. Dann gibt man an welche Programme auf welchen Nodes in wievielfacher Ausfertigung gestartet werden sollen. Dann werden diese Programme vom MPI-Frame gestartet.

Jedes dieser *Programme (C, C++ oder Fortran)*, die auf diese Weise gestartet werden sollen, muss zuvor beim Linken mit der *mpi-Bibliothek* gelinkt werden. Dadurch ist es möglich, in diesen Programmen alle die speziellen mpi-Funktionen für die Kommunikation und für das Rechnen im Verbund zu benutzen.

Für die ersten Testversionen eines Programms reicht es aus, den MPI-Frame auf einem lokalen Rechner (127.0.0.1) zu starten und alle Nodes auf diesem lokalen Rechner zu halten. Später kann man über die MPI-Startdatei externe Rechner mittels Namen (Auflösung über DNS-Server) angeben.

Für das Testprojekt *SimWorld + SimCognitiveAgent* nehmen wir an, dass ein geeigneter MPI-Frame gegeben ist und dass sowohl verschiedene Versionen von Testwelten --realisiert als *SimWorld*-- wie auch verschiedene Versionen von CAgents --realisiert als *SimCognitiveAgent*-- entwickelt werden sollen. Von daher wird das Testprojekt *SimWorld + SimCognitiveAgent* in zwei Teilprojekte aufgespalten:

^{*)} Da der Ausdruck CAgent und verwandte Begriffe schon besetzt ist --z.T. mit ganz anderen Bedeutungen als hier angezielt-- wird hier der Ausdruck *CognitiveAgent* eingeführt, um Verwechslungen zu vermeiden (siehe auch Anhang)

1. Teilprojekt *SimWorld*
2. Teilprojekt *SimCognitiveAgent*

Teilprojekt *SimWorld*

Das Teilprojekt *SimWorld* wird von einer Gruppe ausserhalb des Seminars 'Simulation von Wissen' bearbeitet (KW21- KW25 2004). Hier nur der Vollständigkeit halber eine kurze Beschreibung.

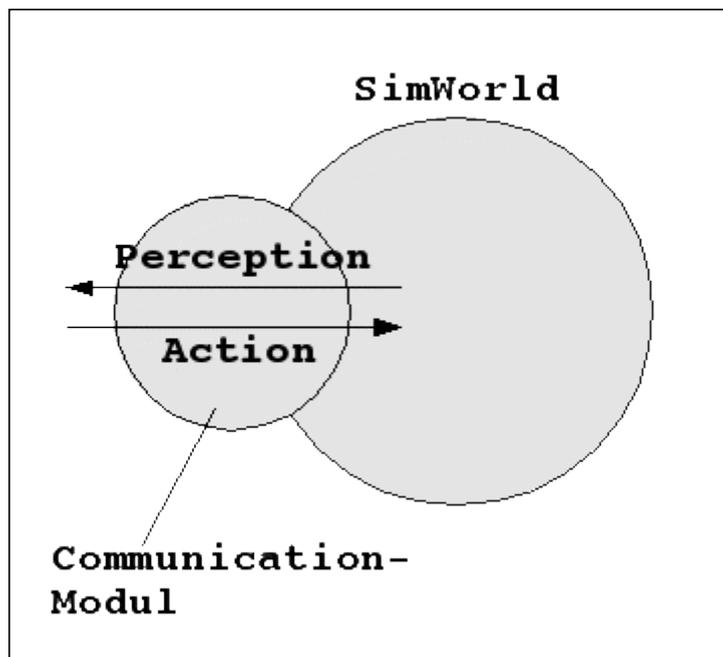


Bild: SimWorld1

Simworld ist ein Prozess im Rahmen eines MPI-Frames. Aufgrund der MPI-Bibliothek besitzt der Prozess Simworld Mechanismen, eine *Kommunikation mit anderen Prozessen* z.B. über *TCP/IP* zu realisieren. Diese Kommunikation dient dazu, sowohl die *Wahrnehmung (Perception)* von assoziierten kognitiven Agenten zu simulieren wie auch deren *Aktivitäten (Actions)*. Sowohl Wahrnehmung wie auch Aktionen werden über *Nachrichten (Messages)* realisiert.

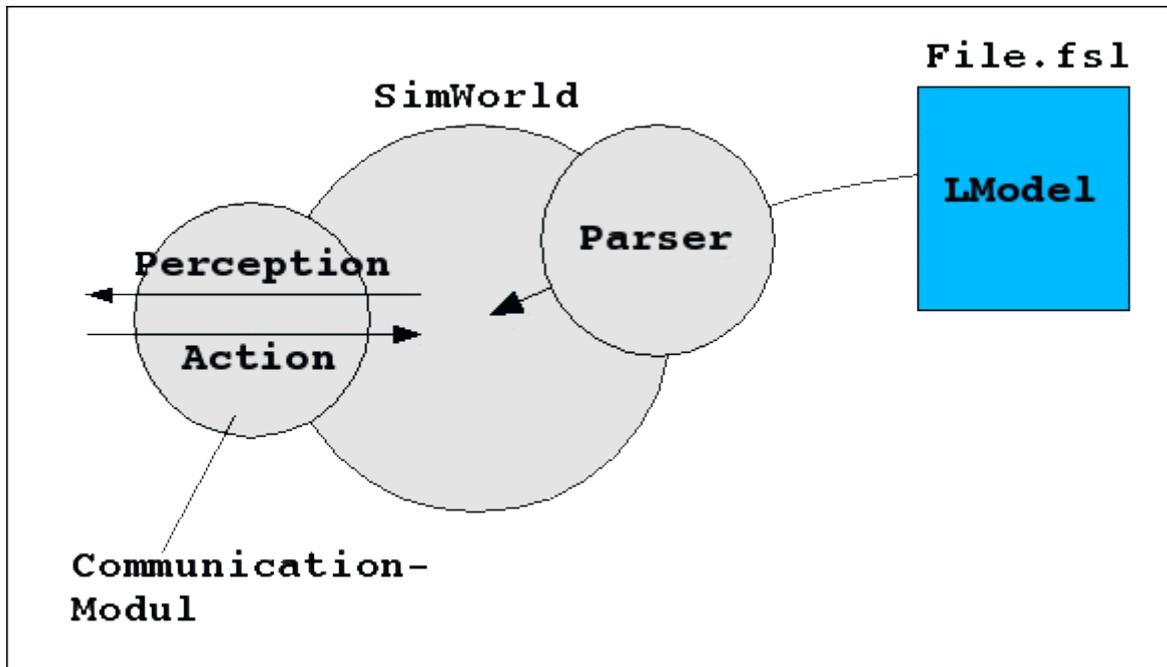


Bild: SimWorld 2

Bevor eine SimWorld in Aktion treten kann, muss aus einer Datei im *FSL-Format* (Endung *.fsl*) die Beschreibung einer zu simulierenden Welt eingelesen werden. Dies geschieht mit Hilfe eines Parsers, der mit dem flex+bison-Parsergenerator erzeugt worden ist. Dieser Parser wurde gebaut, um die Sprache *FSL-v1.1* zu definieren (*FSL := Formal Systems Language*). Diese Sprache wurde im Rahmen des PES-Projektes definiert, um Prozessbeschreibungen leicht implementieren zu können, die auf dem Systembegriff basieren. Der Begriff *LModell* (*:= Logical Model*) rührt daher, dass jede Beschreibung von Weltprozessen als eine *Konstruktion eines logischen Modells* des realen Prozesses aufgefasst wird, eine abstrahierende Konstruktion, so, wie es im Rahmen des Softwareengineerings und von Theoriebildungen generell üblich ist. So gesehen ist der Parser hier eine Abbildung von FSL-Daten in Welt-Daten:

SimWorld (1): SimWorld-Parser: FSL-Daten ---> World-Daten

Die Welt-Daten (World-Daten) sind eine *interne Datenstruktur* des SimWorld-Prozesses. Diese können von dem SimWorld-Prozess beliebig geändert werden.

SimWorld (2): SimWorld-Compute: World-Daten x Com-Daten ---> World-Daten x Com-Daten

Die Veränderung der Welt-Daten soll allerdings in Abhängigkeit von möglichen Kommunikations-Daten (Com-Daten) erfolgen. Damit ist gemeint, dass z.B. ein kognitiver Agent eine Aktion gesendet



hat, deren Ausführung zu überprüfen und gegebenenfalls umzusetzen ist.

Die Kommunikation mit externen kognitiven Agenten wird allerdings durch eine eigene Funktion realisiert, die Evaluationsfunktion:

SimWorld (3): SimWorld-Eval: Com-Data ---> Com-Data

Dazu gehört die Verwaltung der Benutzer. Liste möglicher Benutzer, Login/Logout, Time-Outs, Rechte usw.

Wichtig ist, dass jeder angemeldete Benutzer in den Welt-Daten (World-Data) *repräsentiert* ist. Die SimWorld-Compute-Funktion muss wissen, welche *Art von Wahrnehmung (Perception, 'View')* zu einem Agenten gehört, um berechnen zu können, welche Daten der simulierten Welt in einem bestimmten Augenblick dem Agenten zugänglich sind. Ferner muss der Funktion SimWorld-Compute bekannt sein, welche *Art von Aktionen* ein angemeldeter Agent ausführen kann. Nur auf der Basis dieser Informationen kann die Funktion SimWorld-Compute eine *Interaktion zwischen Welt und Agent* realisieren.

Die *Interaktion* zwischen möglichen kognitiven Agenten und einer SimWorld läuft in *Zyklen* ab: jeder Zyklus besteht (i) aus einer Information der SimWorld an alle Agenten, was sie gerade *wahrnehmen* können. Dann (ii) wertet die SimWorld aus, ob die kognitiven Agenten irgendwelche *Aktionen ausführen wollen*. Diese (iii) werden *überprüft*. Falls nicht ausführbar, gibt es eine *Fehlermeldung* zurück, andernfalls eine *Bestätigung*. Danach (iv) werden die Aktionen ausgeführt.

Zahlreiche weitere *Hilfs- und Verwaltungsfunktionen* sind im Kontext des SimWorld-Prozesses möglich (MPI unterstützt u.a. auch eine grafische Ausgabe).

Soweit die Kurzbeschreibung des Teilprojektes SimWorld.



Anhang

Hier ein kleiner Überblick über die Verwendung der Begriffe *CAgent* und ähnlicher Terme im Netz:

CAgents:

<http://www.igi.tugraz.at/ril-toolbox/reference/index.html>

<http://www.codeproject.com/shell/MSAgentClass.asp>

CBOT := Abkürzung für Chicago Board of Trade

CogBot:

<http://act-r.psy.cmu.edu/workshops/workshop-2003/proceedings/24.pdf>

<http://www.cs.drexel.edu/~fjl/research.html>

CogniBot:

http://vereda.saber.ula.ve/proyecto_vereda/invena/situacion.htm

http://vereda.saber.ula.ve/proyecto/proyecto_vereda.htm

Knowbots

<http://www.cnri.reston.va.us/home/koe/>

CogniAgent:

CognitiveAgent:

http://virtual.cvut.cz/kifb/en/concepts/_cognitive_agent.html

Neu reserviert:

www.cognitiveagent.org