

USARSIM – Game-Engines in der Robotik-Lehre

Joachim Hertzberg, Kai Lingemann, Andreas Nüchter

Universität Osnabrück
Institut für Informatik, AG Wissensbasierte Systeme
Albrechtstraße 28, 49069 Osnabrück
hertzberg@informatik.uni-osnabrueck.de

Abstract: In der Lehre zum Thema Wissensbasierte Robotik verwenden wir seit Kurzem den Robotersimulator USARSIM, der weltweit im Kontext der *RoboCup Rescue Real Robot* Liga eingesetzt wird. Wir stellen den Lehr-Kontext vor, in dem wir arbeiten, skizzieren den Simulator und beschreiben seine Einbindung in unsere Lehre. Unsere Erfahrungen bezüglich der Motivation der Studierenden und ihrer Leistungen der Verwendung des Simulators sind sehr positiv.

Zum WS 2004/05 wurde am Osnabrücker Institut für Informatik ein Lehrstuhl für Wissensbasierte Systeme erstmals besetzt. Als erster Arbeitsschwerpunkt wird das Thema Wissensbasierte Robotik bearbeitet. In der Lehre ist es regelmäßig mit Vorlesungen, Übungen, Praktika, Seminaren und Arbeitsgemeinschaften vertreten.

Für unsere Forschung wie für praktische Elemente (Übungen, Praktika, Examensarbeiten) im Rahmen dieser Lehre setzen wir Roboter des Typs KURT2 bzw. Kurt-3D ein. Durch unsere eigene Teilnahme an Wettbewerben der *RoboCup Rescue Real Robot* Liga motiviert, nutzen wir den Simulator USARSIM, für den von den Koordinatoren dieser Liga eine Simulationsumgebung zur Verfügung gestellt wird. Es hat sich schnell herausgestellt, dass dieser Simulator für unsere Lehre in vielerlei Hinsicht sehr hilfreich und gut geeignet ist. Dieser Text skizziert den Simulator und unsere Lehr-Erfahrungen damit. Zuvor gehen wir kurz auf die Situation der Informatik-Lehre in Osnabrück ein, die naturgemäß sowohl unsere Auswahl von Lehrinhalten wie die damit erzielten Ergebnisse prägt.

1 Kontext: Osnabrücker Informatik-Lehre

Als Lehrgebiet war die Informatik in Osnabrück bis vor kurzem nur als Nebenfach vertreten und wurde damit im wesentlichen von Diplomstudierenden der Fächer Mathematik, Physik, Systemwissenschaft sowie von Bachelor- und Master-Studierenden aus der Kognitionswissenschaft nachgefragt. Seit Kurzem gibt es einen Bachelorstudiengang Mathematik/Informatik, der bei entsprechender Wahl von Wahlpflichtmodulen ähnlich einem grundständigen Informatik-Bachelor studierbar ist. Den Kern der Lehre bildet ein jährlich angebotener, viersemestriger Zyklus Informatik A-D von Einführungsvorlesungen. Darauf aufsetzend können oder müssen je nach Studiengang Vertiefungen in Bereichen wie

Bioinformatik, Computergrafik, Datenbanken, KI, Kombinatorische Optimierung, Kryptographie und Neuroinformatik gewählt werden; zwei Professuren (Technische Informatik, Software-Engineering) befinden sich in Besetzung. Unsere Robotik-Veranstaltungen dienen in diesem Kontext dazu, die *Informatik der Systeme* an einem Thema zu lehren, das technische wie theoretische Informatik-Grundlagen, KI und Kognitionswissenschaft sowie unsere eigenen Forschungsarbeiten verbindet. Die Arbeitsgruppe Neuroinformatik (M. Riedmiller) setzt mit anderen Inhalten in Forschung und Lehre ebenfalls Roboter ein. Für die Lehre in Wissensbasierter Robotik bedeutet das:

- Die inhaltlichen Voraussetzungen der Studierenden sind äußerst heterogen. Der größte gemeinsame Nenner sind die Einführungsveranstaltungen Informatik A-C; einführende Kenntnisse in KI werden empfohlen, konnten aber bislang noch nicht als verbindlich vorausgesetzt werden.
- Die Interessen der Studierenden zielen weit überwiegend auf Robotersteuerung auf höheren, einschließlich „kognitiven“ Ebenen. Systemprogrammierung, Mikroprozessorprogrammierung oder gar Mechatronik-Arbeiten stoßen auf wenig Interesse und auf wenig Vorkenntnis. Die Informatik-Einführungen programmieren derzeit in Java; entsprechend haben einige Studierende keine C-Praxis.

2 RoboCup Rescue und USARSIM

Neben den Ligen für fußballspielende Roboter im RoboCup gibt es die Liga der *Real Rescue Robots*. Die Aufgaben sind, Opfer zu suchen und eine Karte zu erstellen, nach der die Opfer geborgen werden können. Dabei gibt es drei verschiedene Schwierigkeitsstufen (*gelbe, orange* und *rote Arena*, vgl. Abb. 1), die nach jeder Runde in der „Earthquake-Phase“ neu errichtet werden. Anschließend platzieren Mitglieder der Jury an beliebigen Stellen Opferpuppen. Rescue-Roboter werden derzeit noch teleoperiert durch einen Operator, der während der Mission abseits der Arena in einem abgeschotteten Raum sitzt.

USARSIM ist eine Simulation für Roboter und Szenarien der Rettungs- und Katastrophenrobotik. Sie wurde parallel zu den vom amerikanischen *National Institute of Standards* (NIST) entwickelten physischen Arenen konzipiert. Im Fokus der Entwicklung stehen Forschungsarbeiten zur Untersuchung und Entwicklung von Benutzerschnittstellen für die Mensch-Roboter-Interaktion sowie zur Erforschung kooperierender Roboter.

USARSIM ist die erste hochentwickelte Robotersimulation auf Basis einer *game engine*,



Abbildung 1: Reale und simulierte Rescue-Arenen. Von links: Orange Arena real und simuliert, rote Arena real und simuliert [Wan05].



Abbildung 2: Links: AVZ-Gebäude Osnabrück und KURT2 real. Rechts: Gebäude und KURT2 simuliert. Mehr Material unter <http://kos.informatik.uos.de/download/UOSSim/index.html>

die hier aus dem Computerspiel *Unreal Tournament 2003* bzw. *2004* stammt. Damit nutzt der Simulator die fortschrittliche Grafik und physikalische Modellierung eines kommerziellen Programms. Da Computerspiele für einen Massenmarkt produziert werden, bleiben jedoch die Kosten sehr niedrig: Eine Lizenz kostet derzeit 13,99 Euro.

Unreal Tournament ist ein Mehrspieler Ego-Shooter für Windows, Linux und MacOS. Die Grafik ist exzellent, wie man es von kommerzieller Spielsoftware erwartet. Die *Unreal*-Umgebung bietet eine Skriptsprache, die es Entwicklern erlaubt, Objekte zu kreieren und deren Verhalten zu kontrollieren. Der *Unreal Editor*, der mit *Unreal Tournament* ausgeliefert wird, ermöglicht, eigene Umgebungskarten zu erstellen und eigene Modelle von Roboterplattformen zu entwickeln.

Mehrspieler Ego-Shooter nutzen Client-Server Architekturen, wobei jeder Spieler einen Client darstellt. Das schnelle Rendering leistet der Client, während der Server die Spieler koordiniert und für die Umgebungsinteraktion zuständig ist. Das Kommunikationsprotokoll ist proprietär. Das Projekt Gamebots [Gam] modifiziert *Unreal Tournament* so, dass Agenten über eine herkömmliche TCP/IP Schnittstelle gesteuert werden können. Umgekehrt versorgt das Spiel die Agenten über diese Schnittstelle mit Sensordaten.

Die Physik der Agenten wird mit Hilfe der „Karma Physics Engine“ simuliert. Karma verarbeitet Starrkörperbewegungen und erlaubt, Motoren, Räder, Federn, Scharniere und Gelenke zu simulieren. Aus diesen Basismodulen lassen sich leicht komplizierte Objekte zusammensetzen, deren Verhalten die Gesetze der Physik befolgt.

USARSIM stellt für alle drei Robocup-Arenen *Unreal*-Umgebungen zur Verfügung. Abb. 1 zeigt je ein Foto und eine gerenderte *Unreal*-Darstellung der orangen und roten Arena. Mit Hilfe des *Unreal*-Editors lassen sich beliebige Szenen erzeugen. Abb. 2 (außen) zeigt ein Foto unseres Institutsflurs und die entsprechende *Unreal*-Szene.

Eine Robotersimulation wird in zwei Schritten realisiert. Zunächst benötigt man ein Gittermodell der Hardware, das sich im *Unreal*-Editor erstellen lässt. Abb. 2 (Mitte) zeigt den realen und den simulierten Roboter. Im zweiten Schritt wird die Roboterkontrollsoftware entwickelt. Hierzu war lediglich eine Modifikation der für den realen Roboter entwickelten Software notwendig [NLH⁺ed]. Derzeit werden die folgenden Komponenten simuliert:

- Motoren treiben den Roboter an. Hierbei werden PWM-Signale eingestellt.
- Odometrie bestimmt die Radumdrehungen in Ticks.
- Laserscanner liefert 181 Abstandswerte in einer Ebene vor dem Roboter.
- Kamera liefert ein Bild der Umgebung.

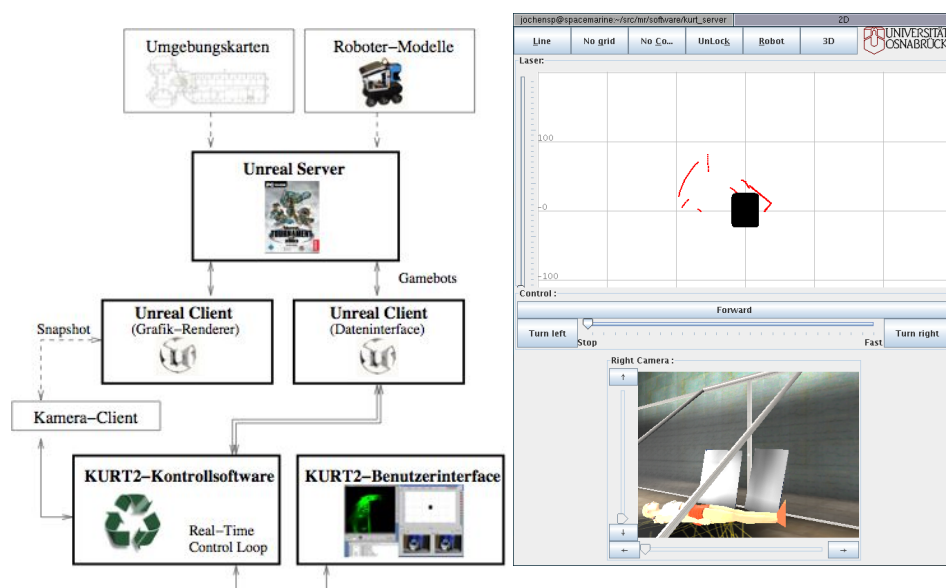


Abbildung 3: Links: Die Softwarestruktur. Pfeile zeigen den Datenfluss, durchgezogene Linien stellen TCP/IP-Verbindungen dar (vgl. 2), doppelte Linien entstehen durch Linken und gestrichelte durch Einlesen von Daten. Rechts: KURT2-Benutzerinterface.

Die vorhandenen Gerätetreiber wurden so verändert, dass sowohl auf die physischen Komponenten als auch auf über den *Unreal*-Client auf simulierte Werte zugegriffen werden kann. Die Gerätetreiber für die Kameras auf KURT2 beziehen nun ihre Daten von einem Kameraserver, der Umgebungsbilder mittels eines Snapshots aus *Unreal* erzeugt. Abb. 3 skizziert die Softwarestruktur.

3 *Unreal*-Erfahrungen in der Lehre

Es scheint allgemein Einigkeit zu herrschen, dass Simulatoren F&E in mobiler Robotik unterstützen, die code-identische Steuerungsprogramme simuliert wie real mit ähnlichen Ergebnissen nutzen lassen. Unsere kurze Erfahrung, USARSIM in der Lehre einzusetzen, setzt diesen Nutzen fort. Wir haben USARSIM im Robotik-Praktikum (WS 04, SS 05) und teilweise bei den Übungen zur Vorlesung Wissensbasierte Robotik (SS 05) verwendet. In beiden Fällen wurden parallel (andere Praktikumsgruppen, andere Übungsstunden) die realen Roboter verwendet, sodass Abbildung und Abbildbarkeit zwischen Roboter und Simulator evident waren. Sinn und Strategie der Simulatorverwendung für unsere Forschung und für potenzielle Examensarbeiten wurden erläutert.

Wir hatten die Hypothese, der Einsatz eines Simulators könnte mechatronik-fernen Osnabrücker Studierenden zusätzliche oder komplementäre Motivation bieten, Robotik-Themen zu bearbeiten: Es ist möglich, sinnvolle Themen im Robotik-Praktikum als reine Software-Arbeiten zu konzipieren. Diese Hypothese hat gestimmt. Insgesamt zeigte sich Folgendes:

- Der Ursprung des Simulators in *RoboCup Rescue* macht nicht nur Ergebnisse der Studierenden einbindbar in unsere Forschungsarbeiten. Er motiviert auch zusätzlich: Die Arbeiten stehen erkennbar in einem Kontext. Besonders gelungene Arbeiten bieten eine Chance, mit Stipendium an einem RoboCup-Wettbewerb teilzunehmen.
- Der Simulator auf Basis einer „echten“ Spiele-Engine hat weitere Motivation dadurch erzeugt, dass er über die Robotik hinaus ein reales Anwendungsgebiet der Informatik einbezieht. Einige Studierende kannten das Spiel bereits als *black box* und waren motiviert, seine Umgebung nun auch als Werkzeug zu nutzen.
- Professionelle 3D-Grafik hilft Frust bei mühsamen Modellierungsaufgaben überwinden. Man muss zwar nach wie vor tagelang Maße aufnehmen, um einen kompletten Gebäudeflur mit Möbeln zu modellieren; das Ergebnis sieht dann aber auch überzeugend aus und motiviert, auch noch letzte Unschönheiten auszubügeln.
- Die minimalen Kosten von *Unreal* haben bewirkt, dass Studierende es privat gekauft und intensiver als erwartet an ihren Themen gearbeitet haben.
- Der Robotersimulator wurde von den Studierenden als vollwertiger Ersatz zum physischen Roboter angenommen; zwischen Ansteuerung des Simulators und Ansteuerung des Roboters wurde problemlos hin und her gewechselt. Das hilft, wenn physische Roboter zwischenzeitlich nicht für die Lehre verfügbar sind.
- Die Dokumentation von *Unreal* ist verbesserungsfähig, nicht alles Beschriebene liegt implementiert vor. Über öffentliche FAQ-Listen und Emails an die Entwickler konnten unsere Studierenden die auftretenden technischen Probleme jeweils klären. Nicht in allen Fällen erlaubt *Unreal* elegante Modelle, aber wir können damit leben.

Die Ergebnisse des Robotikpraktikums im WS 04 (Blockpraktikum im Februar/März 2005) haben unsere Erwartungen an die Ergebnisse gemessen an der teilweise schwierigen Vorkualifikation der 10 Teilnehmenden bei Weitem übertroffen. Zwei von ihnen können mit Stipendium im Rahmen der Teilnahme der AG Wissensbasierte Systeme am *Robocup Rescue* Wettbewerb im Juli 2005 in Osaka teilnehmen. Ein Aspekt dieser Teilnahme ist die Demonstration der entsprechenden Arbeiten auf dem USARSIM-Demonstrationsworkshop, der in diesem Jahr erstmalig durchgeführt wird.

Fazit: Dass Simulatoren ihren Platz in der Robotik-F&E haben, wussten wir. Ihren entsprechenden Platz in der Lehre hatten wir erwartet. Der Erfolg unserer Verwendung von Simulation auf Basis von USARSIM in der Lehre hat uns selber überrascht.

Literatur

- [Gam] Gamebots. <http://www.planetunreal.com/gamebots/>.
- [NLH⁺ed] A. Nüchter, K. Lingemann, J. Hertzberg, H. Surmann, K. Pervölz, M. Hennig, K. R. Tiruchinapalli, R. Worst und T. Christaller. Mapping of Rescue Environments with Kurt3D. In *Proceedings of the IEEE International Workshop on Rescue Robotics (SSRR '05)*, Kobe, Japan, June (accepted).
- [Wan05] Jijun Wang. USARSim - A Game-based Simulation of the NIST Reference Arenas <http://usl.sis.pitt.edu/ulab/usarsim/download.page.htm>, 2005.