

# Offene 3D-Umgebungen als Framework für rollenspielbasierte Lernszenarien

Nils Malzahn, Hanno Buhmes, Sabrina Ziebarth, H. Ulrich Hoppe

Informatik und angewandte Kognitionswissenschaft  
Universität Duisburg-Essen  
Lotharstr. 63  
47057 Duisburg  
hanno.buhmes@gmail.com  
{malzahn,ziebarth,hoppe}@collide.info

**Abstract:** Das vorliegende Papier stellt ein Framework für rollenspielbasierte „Serious Games“ in offenen 3D-Welten vor. Als Anwendung wird die Vorbereitung auf ein Bewerbungsgespräch für einen Ausbildungsplatz demonstriert. Dabei werden Phasen der Immersion im 3D-Rollenspiel mit Phasen der distanzierten Reflexion verknüpft, um lernförderliche Perspektivwechsel zu unterstützen.

## 1 Einleitung und Motivation

Im Rahmen aktueller Informatik-Anwendungen bilden Computerspiele eine der Sparten mit der größten Zahl von Anwendern. Der Ansatz des „Serious Gaming“ versucht die Attraktivität dieses Mediums (im Sinne des „Spaßfaktors“) nicht nur zu Unterhaltungszwecken einzusetzen, sondern zugleich weiterführende „seriöse“ Inhalte zu vermitteln. Als Anwendungsbereiche für Serious Gaming werden u.a. Bildung und Ausbildung, Gesundheitsfürsorge, politische Meinungsbildung und Sondierung diskutiert [Zy05].

Obwohl es nicht unmittelbar als Spiel konzipiert wurde, bietet Second Life eine 3D-Welt mit virtualisierter sozialer Interaktion und der Möglichkeit von Identitätswechseln. In der Hype-Phase von „Second Life“ haben verschiedene Bildungsinstitutionen (z.B. die Universität Edinburgh) versucht dort Bildungsangebote zu platzieren. Es haben sich daraus jedoch keine Erfolgsgeschichten entwickelt, da sich bei der Verlegung klassischer inhaltsorientierter Bildungsangebote in eine virtuelle Parallelwelt in Bezug auf das Lernen keine klaren Mehrwerte zeigen [Ro99]. Weder 3D-Umgebungen noch Identitätswechsel sind hier von nachvollziehbarem Vorteil.

Anders ist dies bei Rollenspielen in virtuellen Umgebungen [To05]; [RB01], ggf. mit Verkürzungen in Zeit und Raum gegenüber einem realen „Enactment“. Klare Vorteile liegen hier in der Verfügbarkeit (insbesondere bei virtuellen Ko-Akteuren) wie auch der Möglichkeit des strukturierten Mitschnitts auf der Basis der im System explizit repräsentierten Dialogbeiträgen und Aktionen. Die strukturierte Transkription unterstützt insbesondere auch das Auf- und Wiederfinden interessanter Situationen [Ro99] und damit die Analyse sowie individuelle und gruppenorientierte (Selbst-) Reflexion.

Die genannten Vorteile bilden unsere Motivation für eine Anwendung im Bereich des Trainings für Vorstellungsgespräche. Der Nutzen von Rollenspielen ist offensichtlich und eine gängige Lehrmethode in Bewerbertrainings. Allerdings erfordern nicht-virtuelle Szenarien erheblichen Koordinationsaufwand und die Auswertung von Video-Mitschnitten ist aufwändig. In unserem Ansatz wird daher das Rollenspiel virtualisiert und die Rolle des Personalmanagers wird durch einen Bot ausgefüllt.

## **2 Die Lernumgebung**

Die Grundlage für die 3D-Umgebung ist der freie Second Life-Klon OpenSimulator (OpenSim). Der Server ist in C# geschrieben und erlaubt, innerhalb der Spielwelt Skripte sowohl in C# als auch in der „Linden Scripting Language“ (LSL), einer JavaScript ähnlichen Interpretersprache, auszuführen. Mittels LSL-Skripten ist es möglich, das in der Welt beobachtete Handeln und die dazugehörigen Dialoge zu protokollieren und den Spielverlauf zu kontrollieren, indem notwendige Interventionen automatisiert veranlasst werden.

### **2.1 Basiskomponenten**

Für die Modellierung und Spezifikation der Dialoge wird die Artificial Intelligence Markup Language (AIML) verwendet. Diese XML-basierte Modellierungssprache wurde ursprünglich entwickelt, um Dialoge für Eliza-artige Chatbots wie A.L.I.C.E. zu „skripten“. Es gibt Hinweise, dass sich mit solchen Chatbots Lernerfolge erzielen lassen [KHB07]. AIML beschränkt sich in der ursprünglichen Fassung auf Textein- und ausgabe. Für viele Szenarien sind jedoch Gesten zur Unterstützung des Ausdrucks hilfreich. Gerade in Szenarien wie das hier beschriebene Vorstellungsgespräch sind auch der Austausch von Gesten und andere formalisierte Handlungsabläufe (z.B. Hände schütteln, Platz nehmen) relevant für den praktischen Erfolg. Aus diesem Grund sollten die Gesten zusammen mit dem Dialog modelliert werden können. Dazu wurde die AIML-Grammatik um Gestik-Elemente erweitert. Gesten können durch den Nutzer wie in Chats üblich durch Emoticons von den Lernenden eingebunden werden.

Dies ermöglicht eine einfache Eingabe und wird bei entsprechender Konfiguration von den gängigen OpenSim-Clients in Gestenanimationen umgesetzt. Als Client für die Lernenden wird der quellcodefreie Meerkat-Viewer eingesetzt.

Die Anbindung der AIML-Dialoge an die OpenSim-Umgebung erfolgt durch einen OpenSim-Bot ("Marvin"), der ursprünglich von der Pixelpark AG implementiert und von uns weiterentwickelt wurde. Marvin nimmt Nachrichten aus der 3D-Welt entgegen, um sie weiter zu verarbeiten. Der Dialogtext wird an eine erweiterte AIML-Bibliothek propagiert und die entsprechenden Antworten an die Lernenden zurückgesendet. Dabei werden kleine Antwort-Verzögerungen genutzt, um die Illusion eines menschlichen Gegenübers zu erzeugen. Gleichzeitig protokolliert Marvin den Gesprächsverlauf und das durch AIML aktuell identifizierte Thema, in dem sich das Gespräch bewegt. Für die Erkennung von Gesten des Spielers, die nicht über die Eingabe von Emoticons erfolgen und für die Verfolgung der Bewegungen des Spielers werden LSL-Skripte eingesetzt, die dieses Verhalten ebenfalls protokollieren.

Die Analyse und Reflexion des Rollenspiels erfolgt anhand eines Videomitschnitts. Dieses Video wird, während die Lernenden mit der 3D-Welt interagieren, durch „frame grabbing“ mitgeschnitten und im Matroska-Container als XVID-Video abgelegt. Der Matroska-Container erlaubt zusätzlich die Definition von Untertiteln und Sprungmarken. Diese werden in der Reflexionsphase genutzt, um Lernenden und Tutoren eine einfache Navigation im Lernvideo (vgl. Abbildung 1) zu ermöglichen und bemerkenswerte Stellen hervorzuheben. Die Untertitel können ferner genutzt werden, um Hinweise in einem Selbstlern-Szenario zu transportieren. Grundsätzlich erzeugt jedes in AIML definierte und durchlaufende Thema ein Kapitel im Video.



Abbildung 1: Screenshot des Reflexionsvideos

## 2.2 Architektur

Der Inhalt der Untertitel, sowie optionaler weiterer Dokumente zur Unterstützung der Reflexionsphase wird durch ein Agentensystem automatisiert erstellt. Zu diesem Zweck arbeiten mehrere Agenten gemeinsam an der Auswertung der unterschiedlichen Protokolleinträge (vgl. Abbildung 2). Zum Austausch der Protokolldaten, Analyseergebnisse und Feedbackinträge dienen die SQLSpaces [WGH07]. Die SQLSpaces sind eine Implementierung des „Tupel Space“-Konzepts. Dieses Konzept basiert auf der Idee einer Blackboardarchitektur, die schon sehr lange erfolgreich als Basisarchitektur für Sprachverarbeitung eingesetzt wird [EL80].

Die verwendete Implementierung bietet native Clients für verschiedene Programmiersprachen (u.a. C#, Java u. Prolog). Zusätzlich existiert ein generischer Webservice, der den Zugriff über LSL-Skripte erlaubt. Dies lässt die Verarbeitung der verschiedenen Analyseprobleme (Textanalyse, Gestenanalyse, regelbasierte Feedbackgenerierung) sowie der Exportfunktionen in dem jeweils angemessenen Programmierparadigma zu. Abbildung 2 skizziert einen Ausschnitt des Inhalts des Blackboards im laufenden Betrieb. Die Agenten zur Gesten- und Textanalyse analysieren die im TupelSpace vorhandenen Daten aus den unterschiedlichen Protokollierungsquellen. Sie schreiben ihre jeweiligen Analyseergebnisse wieder auf den TupelSpace. Diese werden anschließend von Feedback-Agenten genutzt, um Untertitel für das Video und Übersichtsdocuments für Tutoren zu generieren.

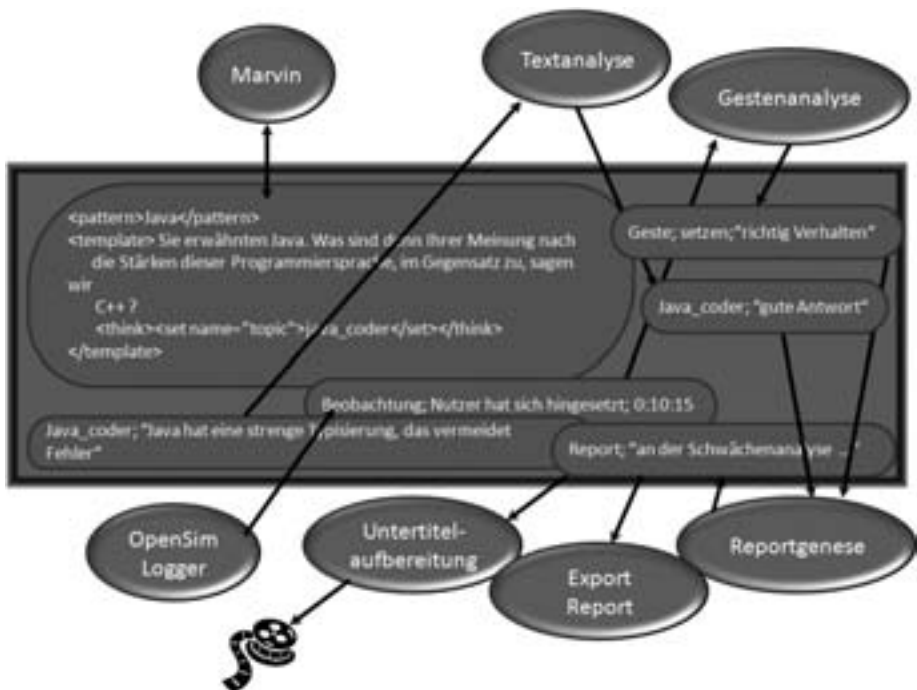


Abbildung 2: Schematische Darstellung der TupelSpace-basierten Architektur mit Beispieldaten und Agentenzugriff

## 2.3 Analyseagenten

Die Agenten handeln unabhängig voneinander. Der Prozess wird datengetrieben gesteuert. Das bedeutet, dass nur die zur Verfügung stehenden Daten den nächsten Schritt im Prozess bestimmen. Dies führt zu einer großen Ausfallsicherheit (Robustheit) und Flexibilität bezüglich der eingesetzten Analysemethoden. Diese können je nach Bedarf und Domäne ausgetauscht oder verfeinert werden.

Das Szenario eines Bewerbungsgesprächs für künftige Azubis hat einen großen Freiheitsgrad bezüglich der möglichen Antworten seitens der Bewerber. Daher wird nicht versucht zu beurteilen, ob die Antworten semantisch vollständig richtig sind, da dies eine zu komplexe Aufgabe wäre, selbst wenn die Domäne z.B. auf IT-nahe Berufe eingeschränkt würde. Trotzdem gibt es einige markante Stellen in jedem Bewerbungsgespräch, in denen die Antwort evaluiert und hilfreiche, allgemeine Hinweise zu möglichen Problemen gegeben werden können. Die Gestenanalyse kann vor allem „versäumte Gelegenheiten“ identifizieren. So können z.B. übliche Rituale bei der Begrüßung bewertet werden (wer grüßt wen, Hände schütteln, Platz nehmen). Aufgrund dieser Analysen werden Hinweise oder lobende Worte für das korrekte Verhalten (vor allem in einem Selbstlernszenario) unter dem Video (s. Abbildung 1) eingeblendet.

## 3 Diskussion & Fazit

Häufig wird in Anwendungen von 3D-basierten Serious Games [RB01] [BEJ06] versucht den gesamten Lernprozess in der 3D-Welt stattfinden zu lassen. Wir denken jedoch, dass die mit der 3D-Welt verbundene Immersion in das Geschehen [Os08] den für den Lernprozess wichtigen Schritt der kritischen (Selbst-)Reflexion eher behindert. Wir halten es für plausibel, dass Immersion durch die starke Bindung von „reaktiver Aufmerksamkeit“ die für die Selbstreflexion förderliche Rollendistanz behindert. Hinzu kommt die kognitive Überlast durch die nicht-triviale Steuerung in 3D-Umgebungen [Os08]. Aus diesem Grund schlagen wir für die Reflexionsphase einen Medienwechsel hin zu einer einfacher zu bedienenden 2D-Umgebung, wie sie eine konventionelle Desktop-Anwendung bietet, vor. Diese Umgebung bietet auch einen vereinfachten Zugriff auf weitere Wissensquellen (Dokumente, Internetseiten) und ermöglicht das Bilden multipler Foci und Handlungsstränge.

Abbildung 3 zeigt den von uns vorgeschlagenen phasierten Prozess im Detail. Das Rollenspiel findet in einer 3D-Umgebung statt. In Szenarien mit einem menschlichen Tutor kann durch das gezielte Navigieren zu Auffälligkeiten im Spiel oder auch das wiederholte Abspielen interessanter Situationen u.E. der Reflexionsprozess gezielter und schneller in Gang gesetzt werden. Einzelne Kriterien für Auffälligkeit oder Interessantheit können als Suchfunktionen auf der Transkription automatisiert werden. Dies wie auch die Planung und Durchführung experimenteller Studien sind Gegenstand laufender Arbeiten.

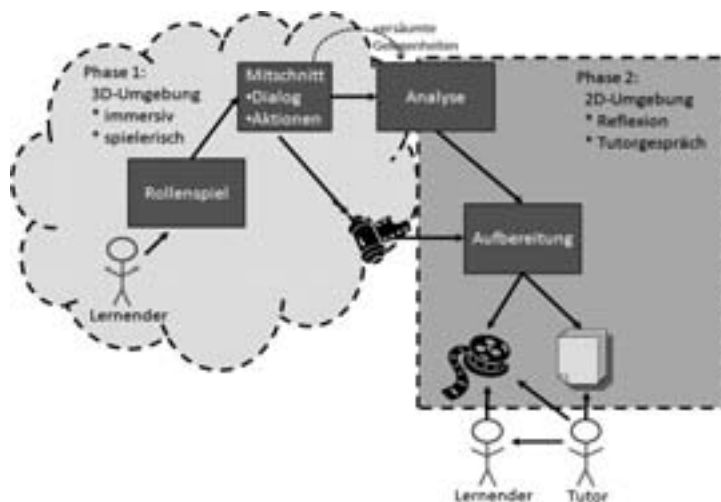


Abbildung 3: Interaktion von Lernenden und Tutor mit der Lernumgebung in den unterschiedlichen Phasen des Lernens.

## Literaturverzeichnis

- [BEJ06] Backlund, P.; Engström, H.; Johannesson, M.: Computer Gaming and Driving Education. In: Proceedings of the workshop Pedagogical Design of Educational Games affiliated to ICCE 2006, 2006.
- [EL80] Erman, L.; Lesser, V.R.: The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. In: Computing Surveys, 12, 1980; S. 213-253.
- [KHB07] Kerly, A.; Hall, P.; Bull, S.: Bringing chatbots into education: Towards natural language negotiation of open learner models. Knowledge.-Based Systems. 20 (2), 2007.
- [Ro99] Roussos, M. et al.: Learning and Building Together in an Immersive Virtual World. In: Presence 8 (3), MIT Press, 1999; S. 247-263.
- [Os08] Ojstersek, N.: Gestaltung und Betreuung virtueller Lernszenarien in Second Life. In: Selbstorganisiertes Lernen im Internet - Einblick in die Landschaft der webbasierten Bildungsinnovationen. Innsbruck, Wien, Bozen, 2008; S. 296 – 300.
- [RB01] Romano, D.; Brna P.: Presence and Reflection in Training: Support for Learning to Improve Quality Decision-Making Skills under Time Limitations. In: CyberPsychology & Behavior 4 (2). 2001; Mary Ann Libert, Inc; S. 265-277.
- [To05] Totty, M.: Better training through gaming. In: Wall Street Journal - Eastern Edition, 245 (80), R6-0, 2005.
- [WGH07] Weinbrenner, S.; Giemza, A.; Hoppe, H.U.: Engineering heterogenous distributed learning environments using TupleSpaces as an architectural platform. In: Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies (ICALT 2007). IEEE Computer Society, Los Alamitos (USA), 2007.
- [Zy05] Zyda, M.: From visual simulation to virtual reality to games. In: IEEE Computer, 2005; S. 25–32.