

# Wissensbasierte Konzepte und visuelle interaktive Systeme

M. Dücker, S. Flake, C. Geiger\*, W. Müller, V. Paelke, C. Reimann,  
W. Rosenbach, D. Zimmermann, S. Zelder

C-LAB

Fürstenallee 11

33102 Paderborn, Germany

<http://www.c-lab.de/vis>

(Visual Interactive Systems)

## Zusammenfassung

Dieser Beitrag stellt eine Auswahl aktueller Arbeiten des C-LAB Instituts in Paderborn vor, die sich im wesentlichen in der Schnittmenge der Anwendung wissensbasierter Techniken im Systementwurf und dem Design interaktiver visueller Systeme ansiedeln. Der allgemeinen Ausrichtung des C-LABs folgend, haben die meisten Projekte einen stark anwendungsbezogenen Charakter. Der Beitrag gibt einen kurzen Überblick der wichtigsten aktuellen Arbeiten in diesem Bereich („Breite vor Tiefe“) und skizziert die zukünftigen F&E-Aktivitäten, die sich auf die Anwendung wissensbasierter Konzepte beim Entwurf interaktiver visueller Systeme konzentrieren.

## 1 Einblick

Das C-LAB (<http://www.c-lab.de>) ist eine Kooperation der Siemens AG und der Universität Paderborn. In diesem Institut wird seit 1985 Forschungs- und Entwicklungsarbeit im Bereich Informations- und Kommunikationstechnologie betrieben. Seit 1994 haben sich die Kernbereiche Multimedia-Based Systems, Cooperative Systems und Optical Interconnection Technology entwickelt, an denen zur Zeit ca. 80 MitarbeiterInnen der Siemens AG und der Universität Paderborn arbeiten. Neben der Durchführung von Förderprojekten (auch in Zusammenarbeit mit anderen Institutionen) arbeitet das C-LAB sowohl in rein kommerziellen Projekten als auch in ausschließlich universitären Bereichen wie Forschung und Lehre.

Dabei beschäftigt sich der Arbeitsbereich „Multimedia-Based Systems“ (M2S) mit der Entwicklung und Anwendung multimedialer Systeme in den verschiedensten Bereichen (meist jedoch im Ingenieursumfeld). Die M2S-Arbeitsgruppe „Visuelle Interaktive Systeme“ (VIS, <http://www.c-lab.de/vis>) konzentriert sich dabei unter anderem auf die Anwendung wissensbasierter Konzepte beim strukturierten Entwurf interaktiver grafischer Anwendungen. Stellvertretend sollen hier einige Projekte aus den Bereichen „Interaktive MM-Illustration“, „Prototyping von 3D-Animationen“, „3D-GUIs“ und „agentenbasierte Modellierung“ kurz skizziert werden. Die Projekte sind sowohl im Rahmen geförderter Vorhaben, in Zusammenarbeit mit industriellen Partnern als auch im Rahmen von Lehrveranstaltungen / Abschlußarbeiten entstanden. Neben existierenden Arbeiten werden auch geplante Vorhaben skizziert, die auf die bereits erzielten (Teil)-Ergebnisse aufbauen bzw. diese miteinander kombinieren.

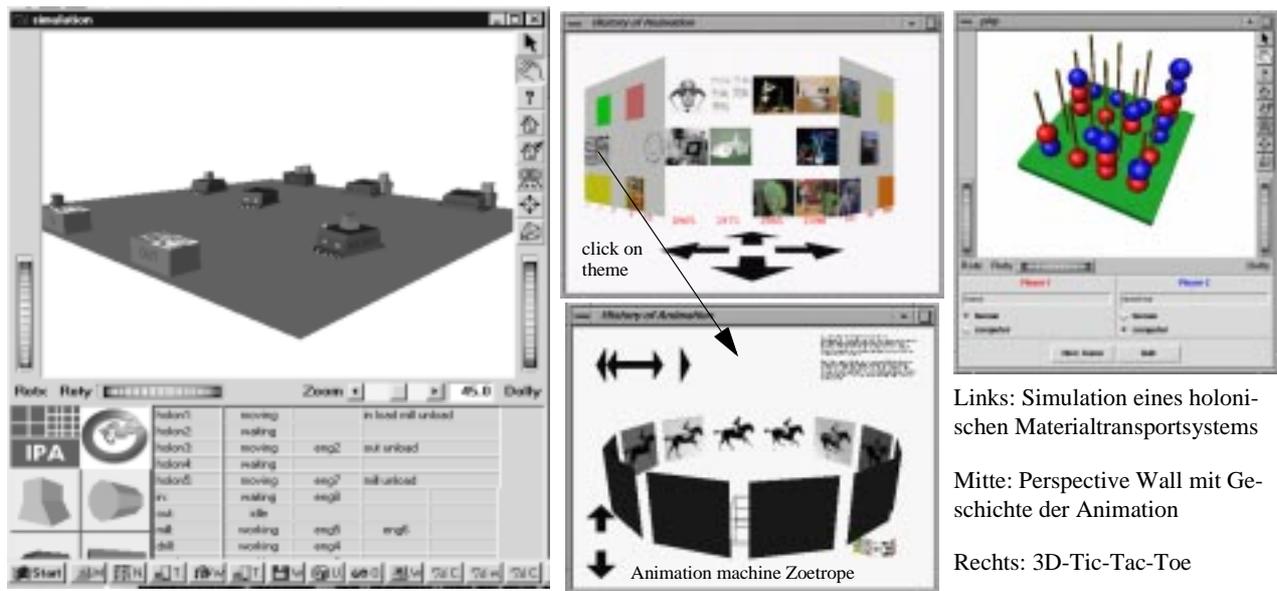
## 2 Werkzeuge zur schnellen Erzeugung interaktiver 3D Animationen

Basierend auf den Ergebnissen bisheriger Arbeiten in Bereich 3D-Visualisierung wurde insbesondere der Bedarf für eine schnelle prototypische Erzeugung interaktiver 3D-Anwendungen identifiziert. Daher wurde die 3D-Animationsbibliothek AAL konzipiert, die auf verschiedenen Abstraktionsebenen vorgefertigte Bausteine zur einfachen Kombination anbietet [8]. Auf unterster Ebene wird dabei die 3D-Grafikbibliothek OpenInventor verwendet. Darauf aufbauend folgen Ebenen für abstrakte Animationen / strukturierte Objekte (A2), animierte Objekte (A3) und einfache animierte Agenten (A4). Letztere erlauben zustandsabhängige Berechnungen, das Ausführen listenbasierter Animationen auf Grundlage wahrgenommener Ereignisse und die Kommunikation mit anderen Agenten und animierten Objekten. Um den schnellen Entwurf interaktiver 3D-Anwendungen durch Skripting zu unterstützen, wurde ein System zur Integration von 3D-Grafikfunktionalität in die Skriptsprache Tcl/Tk entwickelt [12]. Mit dem System Ishell lassen sich beliebige, mit OpenInventor realisierte Klassenhierarchien als Befehlserweiterung einer Tcl/Tk-Shell nutzen. Dazu müssen die relevanten Klassen und Methoden nur textuell

\* Ansprechpartner: Christian Geiger, eMail: [chris@c-lab.de](mailto:chris@c-lab.de)

spezifiziert werden und Ishell erzeugt automatisch alle benötigten Dateien, startet den Übersetzungsvorgang und generiert eine Bibliothek, die sich dynamisch in eine Tcl/Tk-Shell laden läßt. Neben ca 85% der OpenInventor Bibliotheksfunktionen wurden zusätzlich ein Prototyp von AAL, sowie externe OI-Funktionen wie animierte Texturen und Metaballs, integriert. Zudem erlaubt dieser Ansatz die quellcodekompatible Transformation von Anwendungen zwischen WinNT und Unix.

Mit Ishell erweiterte Tcl/Tk-Shells wurden bereits in einer Anzahl von Arbeiten verwendet. In kurzer Zeit konnte ein Demonstrator für ein flexibles Transportsystem realisiert werden, bei dem mobile Serviceroboter (Holons) Werkstücke auf unterschiedlichen Routen zu den entsprechenden Arbeitsstationen bringen (Kooperation mit dem IPA der FhG Stuttgart). Ein anderes Beispiel illustriert interaktiv die historische Entwicklung der Animation beginnend bei den Funktionsprinzipien von Animationsmaschinen des vorigen Jahrhunderts bis zur Generierung von 3D-Computergrafik für Film und Fernsehen.



Links: Simulation eines holonischen Materialtransportsystems

Mitte: Perspective Wall mit Geschichte der Animation

Rechts: 3D-Tic-Tac-Toe

**ABBILDUNG 1. Ishell Anwendungen**

Über eine einfache TCP/IP Schnittstelle sollen in zukünftigen Arbeiten Kontrollstrategien, die in anderen Systemen modelliert wurden (z. B. dem Multi-Agentensystem CASA aus Abschnitt 3) mit diesem Demonstrator (oder anderen Ishell-Anwendungen) verbunden werden. Einige mit Ishell erzeugte Bibliotheken sind auf der VIS Webseite verfügbar.

### 3 Agentenbasierte Modellierung

Aufbauend auf früheren Arbeiten im Bereich paralleler logischer Programmiersprachen (Concurrent Prolog, Pictorial Janus, PCN, DFKI Oz) konzentrieren sich die derzeitigen Arbeiten im Bereich Systemmodellierung auf agentenbasierte Ansätze, teilweise kombiniert mit unscharfen Kontrollstrategien (Fuzzy Logik) [5]. Basierend auf dem allgemein anerkannten Ansatz der BDI-Architektur [17] und insbesondere der formalen Spezifikation eines Agentensystems durch AgentSpeak [18] wurde die Mikrosicht des Multi-Agentensystems CASA spezifiziert, das folgende Eigenschaften besitzt:

- Agentenverhalten wird durch eine Variante bewachter Hornklauseln (Guarded Horn Clauses) spezifiziert, die zusätzlich ereignisgetrieben, unterbrechbar und priorisiert sind. In der praktischen Implementierung sind diese als mehrstufige Produktionsregeln realisiert, bei denen Vorbedingungen und Aktionen teilweise parallel ausgewertet werden können.
- Vorbedingungen können beliebig komplex sein, d.h. sie können die Abarbeitung weiterer Regeln nach sich ziehen. Dadurch sind komplexe spekulative Berechnungen möglich, bevor über das Feuerm einer Regel entschieden wird.
- Es existieren verschiedene Typen von Regeln, die bei der Auswahl unterschiedlich gewichtet werden. Reaktive Regeln besitzen keine spekulativen Berechnungen, deliberative Regeln erlauben dies, jedoch ohne die Möglichkeit zur Kommunikation mit anderen Agenten. Kommunikative Strategien können auch Nachrichten mit anderen Agenten austauschen.
- Ein Prototyp einer Fuzzy-Bibliothek wurde realisiert, der die Entwicklung einfacher Fuzzy-Regler ermöglicht.

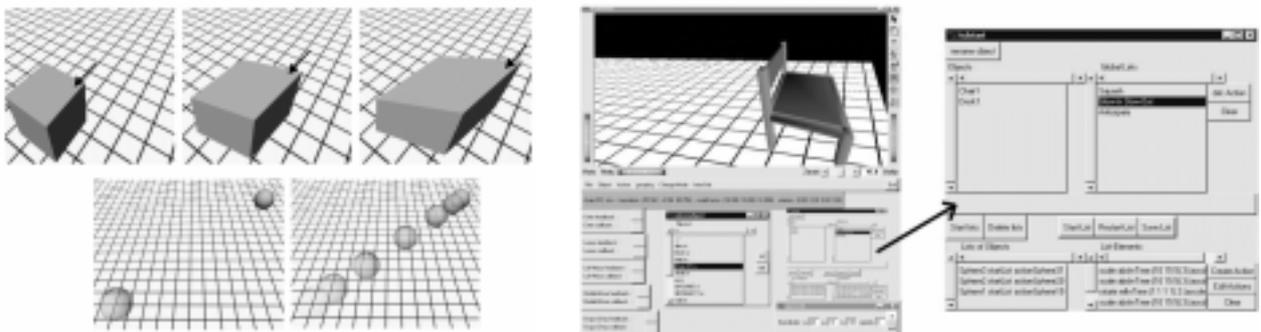
Der Ablauf in einem CASA-Agenten wurde operational durch einen abstrakten Interpreter spezifiziert und in JAVA und unter Verwendung der Bibliothek JAM [11] zu einem lauffähigen Prototypen weiterentwickelt [4]. CASA Agenten lassen sich in einer einfachen Sprache textuell spezifizieren. Anschließend wird diese vom CASA System geparkt und auf der Java Virtual Machine ausgeführt. Für die Makrosicht des MAS, d.h. die Kommunikation und Verwaltung zwischen Agenten, wurde das Agent-Management System Mecca der Siemens AG [1] verwendet, das plattformübergreifend die Verteilung von (mobilen) Agenten auf heterogenen Netzwerken erlaubt. Durch die Unterstützung des FIPA/ACL Standards für die Kommunikation sind auch komplexe Protokolle wie das Contract-Net Protocol in Mecca (und dadurch in CASA) möglich. Durch die Entwicklung einer TCP/IP Schnittstelle kann CASA zudem mit anderen Systemen (z.B. Animationsumgebungen) kommunizieren. Als Beispiel wurden bereits erste Ansätze eines flexiblen Fertigungssystem in CASA modelliert (Farbsortierspeicher für Autokarossern).

## 4 Entwurf neuer Benutzungsschnittstellen

Als weiteres Anwendungsgebiet wissensbasierter Computergrafik wird insbesondere der Entwurf neuartiger GUIs betrachtet. Aktuelle Arbeiten konzentrieren sich dabei auf die Entwicklung animierter 3D-Widgets, visuelle Programmierung und die Gestaltung virtueller Charaktere als Bestandteil von 3D-Applikationen.

### 4.1 Entwurf von animierten 3D Widgets

Beim Entwurf von dreidimensionalen grafischen Benutzungsschnittstellen müssen andere Techniken als in konventionellen 2D-Ansätzen von UIs berücksichtigt werden damit die neuartigen Ansätze nützlich sind. So ist neben einer wirksamen Darstellung der Schnittstellenelemente auch die Navigation des Benutzers und die Manipulation der 3D-Objekte auf ihre intuitive Benutzbarkeit hin zu untersuchen.



**ABBILDUNG 2. Animierte 3D-Widgets + Widget Editor**

Während bisher technische Betrachtungen wie Rendereffizienz im Vordergrund der Entwicklung von interaktiven 3D-Applikationen standen, bekommen mittlerweile Usability-Aspekte eine zentrale Bedeutung beim Entwurf virtueller Welten [15]. Für die Darstellung der Schnittstellenelemente betrachten wir zur Zeit den Einsatz von Prinzipien der traditionellen Animation und die Verwendung eines agentenbasierten Ansatzes zur Kapselung der UI-Funktionalität [9]. Cartoontechniken wurden bereits in den 30er Jahren von Disney verwendet [19], um die Trickfilmdarstellung lebens echter und intuitiv verständlich zu machen - ein Vorsatz, der auch beim Entwurf von virtuellen Welten oft propagiert wird. Der gezielte Einsatz von Animation, z.B. als Feedback bzw. zur Aufmerksamkeitserregung, kann ein erster Schritt in diese Richtung sein [2]. Die Verwendung eines agentenbasierten Ansatzes als Basis soll es ermöglichen, aktive wiederverwendbare Schnittstellenelemente zu entwickeln, die in Abhängigkeit ihrer Umgebung und der Kommunikation mit dem Benutzer entscheiden, welche Aktionen ausgeführt werden und in welcher Art und Weise.

In einer laufenden Diplomarbeit wurde auf Basis von Ishell und AAL ein 3D-Editor entwickelt, der die Erzeugung wiederverwendbarer animierter 3D-Widgets unterstützt (Abb. 2, rechts). Neben der direkt manipulativen Gestaltung von Geometrie und Interaktionsereignissen können auch einfach Keyframeanimationen als resultierende Aktionen ausgeführt werden. Somit lassen sich traditionelle Animationsprinzipien wie Squash&Stretch, Staging, Secondary Action, Anticipation, etc. für Widgets realisieren.

## 4.2 Visuelle Programmierung

Als Weiterführung der Idee dreidimensionaler grafischer Oberflächen wurde exemplarisch ein Ansatz zur visuellen Programmierung in 3D entwickelt [6]. Die Arbeit an visuellen Programmiersprachen hat eine langjährige Tradition im C-LAB. So wurde bereits 1994 die am Xerox PARC von K. Kahn und V. Saraswat entwickelte logische visuelle Programmiersprache Pictorial Janus um zeitliche Aspekte erweitert, durch Abstract State Machines formal spezifiziert und die Entwicklung der visuellen Programme durch eine interaktive Entwurfsumgebung mit Editor, Laufzeitumgebung und visuellem Debugger unterstützt [16]. Diese animierte 2D-visuelle Sprache wurde zum 3D-Ansatz SAM (Solid Agents in Motion) weiterentwickelt. SAM ist eine synchron parallele Programmiersprache, bei der die Programmkonstrukte durch kommunizierende 3D-Objekte dargestellt werden und deren Verhalten durch Regeln definiert ist, die ebenfalls eine dreidimensionale Repräsentation besitzen. Neben der Modellierung verschiedener Kommunikationstypen (send, multicast, broadcast, reply) sind auch lokale Aktionen eines Agenten möglich (z. B. einfache Animationen, Statusänderung, etc.) Die Ausführung eines SAM-Programms (Kommunikation, Regelevaluierung, Ausführen lokaler Aktionen) wird durch entsprechende 3D-Animationen visualisiert. Neben einer vordefinierten abstrakten Darstellung ist auch eine anwendungsspezifische 3D-Repräsentationen möglich, wobei zwischen diesen während des laufenden Programms gewechselt werden kann. Das Beispiel zeigt ein einfaches SAM Programm mit drei parallel laufenden Objekten Produce/Transducer/Consumer, die Nachrichten erzeugen, weiterleiten und konsumieren. SAM wurde inklusive eines 3D-Editors mittels der AAL-Bibliothek realisiert.

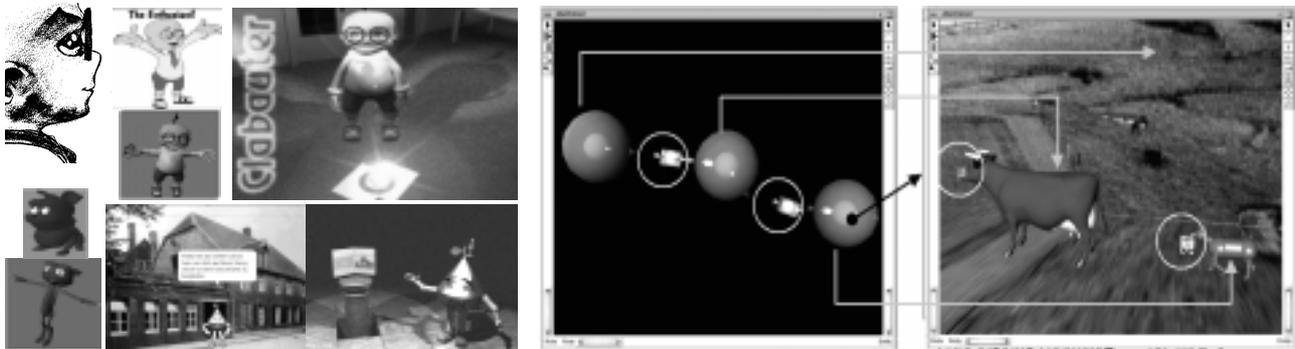


ABBILDUNG 3. Verschiedene 3D-Charaktäre (links) und Sichten in SAM-Programmen (rechts)

## 4.3 Virtuelle Charaktere

Als eher praktisch orientierter Aspekt beim Entwurf von neuartigen UI's bzw. interaktiven 3D-Applikationen werden am Rande auch virtuelle Charaktere betrachtet. Bereits für mehrere Anwendungen (meist für Multimedia-CDs, spezielle Präsentationen, 3D-Filmanimationen) wurden entsprechende Figuren entworfen (vgl. Abbildung 4, links).

Die obere Figur „CLABauter“ wurde als Maskottchen des Instituts entworfen, als Cartoonfigur in 2D gezeichnet und in einem 3D-Programm modelliert. Neben der Entwicklung reiner Filmanimationen wurde auch eine „interaktive“ Version des Clabauters auf VRML-Basis umgesetzt. Mittels eines Keyframe-Editors (ebenfalls in VRML realisiert) lassen sich unterschiedliche Animationen einer zweibeinigen Figur direkt manipulativ generieren und als VRML-Datei abspeichern. Zur Zeit existiert eine erste Version eines Java3D-AnimationViewers, der diese Datei einliest und konvertiert, so daß sich die verschiedenen Animationen in Java3D-Anwendungen ausführen lassen.

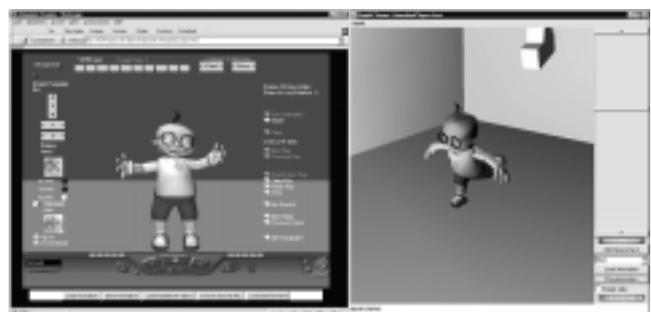


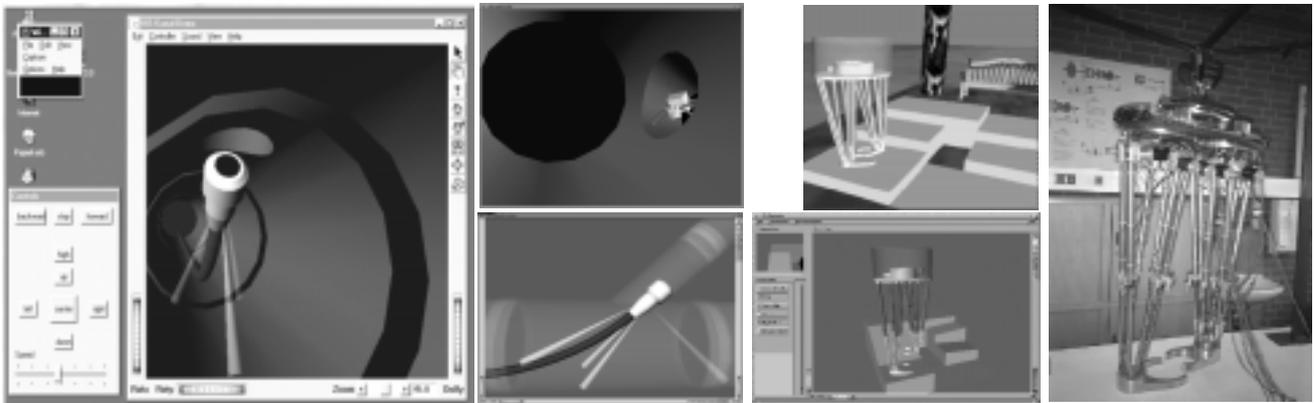
ABBILDUNG 4. Keyframe-Editor und Player

## 5 Interaktive MM-Illustrationen

Illustrationen sind in vielen Bereichen für die Dokumentation, Repräsentation und Kommunikation zwingend notwendig (z.B. bei Schulungen, Messepräsentationen, Analysen), um komplexe Zusammenhänge wirksam darzustellen und dem Betrachter effizient zu verdeutlichen. Dabei kann die gezielte Interaktion des Benutzers mit der Illustration (im Sinne von „Ausprobieren“) zu einem deutlich besseren Verständnis der illustrierten Sachverhalte führen. Im Gegensatz zu möglichst realistischen Simulationen machen Illustrationen gezielt Gebrauch von Abstraktionstechniken, Übertreibungen oder Auslas-

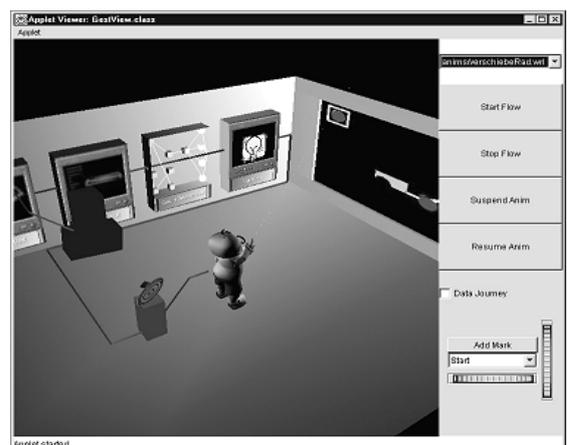
sungen, wenn dies dem schnellen Erreichen des Kommunikationsziels dient. Insbesondere für den Bereich interaktiver Illustration von technischen Systemen (z. B. mobile Roboter, Fertigungssysteme, etc.) lassen sich Techniken der interaktiven 3D-Animation anwenden, wenn es Werkzeuge gibt, die den schnellen Entwurf und die Kombination mit weiteren Darstellungstechniken (2D-Animation, Sound, Video, Bilder, Text) ermöglichen.

Für die Präsentation auf der Industriemesse Hannover (DIM) wurden zwei interaktive 3D-Illustrationen entwickelt. Im HIS-Projekt wurde der Prototyp eines vom Bediener zu steuernden Kanal-Inspektionsroboters animiert, der durch Wasserstrahlen angetrieben Abwasserkanäle mittels einer montierten Videokamera inspiziert, über Hindernisse hinwegschwebt und sogar in die Seitenkanäle abbiegen kann. Diese Funktionalität konnte der Messebesucher am virtuellen Prototyp selbst „erleben“, was mit dem ebenfalls ausgestellten physikalischen Prototyp schwierig war. Sowohl zur Illustration der realisierten Funktionalität als auch zum visuellen „Debugging“ der entwickelten Steuerstrategien wurde im Rahmen einer interdisziplinären Lehrveranstaltung (Projektgruppe „Zweibein“) 1998 eine interaktive 3D-Simulation eines 1.70 großen zweibeinigen Schreitroboters entwickelt. Die Simulation verfügt über die gleiche Schnittstelle zur Steuerungssoftware wie das reale System, so daß die zu entwickelnden Strategien zunächst intensiv am virtuellen Modell getestet werden konnten, bevor der reale Prototyp gesteuert wurde.



**ABBILDUNG 5. Illustration mobiler Roboter**

Die interaktive Illustration von konkreten Darstellungen (z. B. Körpergesten) kombiniert mit abstrakten Informationen (z. B. Datenfluß in SW-Moduln) wird in einem aktuellen Projekt (im Rahmen des MWF Forschungsverbund „Virtuelle Wissensfabrik“) betrachtet. Dabei dient die zuvor beschriebene Figur des „Clabauter“ dazu, die Funktionalität eines Systems zur sprachbegleiteten Körpergestik in multimedialen Umgebungen, das an der technischen Fakultät der Universität Bielefeld entworfen wurde [14], darzustellen. Diese Illustration soll sowohl die Funktionalität einzelner Systemkomponenten als auch die des Gesamtsystems interaktiv illustrieren. Entsprechende Körpergesten können in dem Editor einfach erzeugt werden, in der Illustration ausgewählt und ausgeführt werden. Die Funktionsweise der einzelnen Module läßt sich auf unterschiedlichen Detaillierungsebenen visualisieren. Verschiedene Illustrationstechniken (z. B. Modifikation der Szenenobjekte, Metaobjekte, Kamerafahrten) werden zur detaillierten Erläuterung verwendet und können bei Bedarf vom Benutzer während des Ablaufs einer Gestererkennung ausgewählt werden.



**ABBILDUNG 6. Interaktive Illustration eines Sprach/Gestenerkennungssystems**

## 6 Benutzerzentrierte Softwareentwicklung

Usability Engineering ist ein weiterer wichtiger Schwerpunkt im C-LAB, daher wurde 1997 eine entsprechende M2S-Arbeitsgruppe gebildet, die sich insbesondere mit Fragestellungen der Benutzbarkeit von Softwareanwendungen befaßt [13]. Zur Evaluierung existierender Applikationen wurde ein eigenes Usability-Labor aufgebaut, in dem das Verhalten von Testpersonen mit den dafür üblichen Analyseverfahren evaluiert werden kann. Dabei konzentrieren sich die Anwendungsgebiete

zur Zeit auf Web Usability, Navigation in 3D-Welten und die Unterstützung der Multimediagegestaltung durch sogenannten Kritiksyste me [3].

Kritiksyste me stellen eine spezielle Form wissensbasierter Systeme dar. Sie können als persönliche Assistenten bei der Erstellung eines Designs, beispielweise von Multimediapräsentationen, angesehen werden. Über Kritiksyste me werden Endbenutzern von Autorensystemen Wissen und Fachkompetenz von Experten zugänglich gemacht, so daß eine Verbesserung des ursprünglichen Designs erreicht wird. Hierbei stellt die Wissensbasis des Kritiksyste ms die Grundlage für die Beurteilung eines Designs dar.

Bislang ist die Erstellung eines Kritiksyste ms, insbesondere der Aufbau der Wissensbasis, eine sehr zeitintensive Aufgabe. Die Eingabe von Wissen erfordert oftmals Programmierkenntnisse, die bei Experten aus den verschiedensten Fachrichtungen nicht unbedingt vorausgesetzt werden können. Doch nur die Berücksichtigung verschiedenster Fachrichtungen, wie z. B. der Psychologie, der Gestaltungslehre, dem HCI- und Usability-Bereich, gewährleistet eine möglichst genaue Abbildung eines Designbereichs durch die Wissensbasis. Erschwerend kommt hinzu, daß Experten oftmals räumlich und zeitlich voneinander getrennt sind.

Ziel ist es, eine Entwicklungsumgebung zur kollaborativen Erstellung von Kritiksyste men zu konzipieren, um die schnelle Entwicklung und die Wartung von Kritiksyste men zu ermöglichen [3]. Besonders berücksichtigt werden hierbei die Formulierung, Evaluierung, Verhandlung und Verifikation von Wissen. Wissen über die unterschiedlichsten Designbereiche muß von verschiedensten Experten standardisiert formulierbar sein. Die Evaluierung des formulierten Wissens in Form einer Konsistenzüberprüfung mit bereits existierenden Formulierungen garantiert die Konsistenz der Wissensbasis gegenüber den Endbenutzern. Bei der Konsistenzprüfung entdeckte Inkonsistenzen werden visualisiert und somit beteiligten Experten zugänglich gemacht. Experten werden dahingehend unterstützt, aufgetretene Inkonsistenzen, aber auch unterschiedliche Ansichten, verhandeln zu können.

Die Simulationen der Auswirkungen vorgenommener Formulierungsänderungen und eine methodische Unterstützung bei der Fehlersuche helfen dabei, das formulierte Wissen zu verifizieren. Hierbei nimmt jeweils der kollaborative Aspekt einen hohen Stellenwert ein. Existierende CSCW Systeme und Konzepte wie Text-/Audioconferencing und Application Sharing sollen eingesetzt werden, um gemeinsame Sichten auf Wissen und die gemeinsame Manipulation von Wissen zu realisieren.

## **7 Zukünftige Arbeiten.**

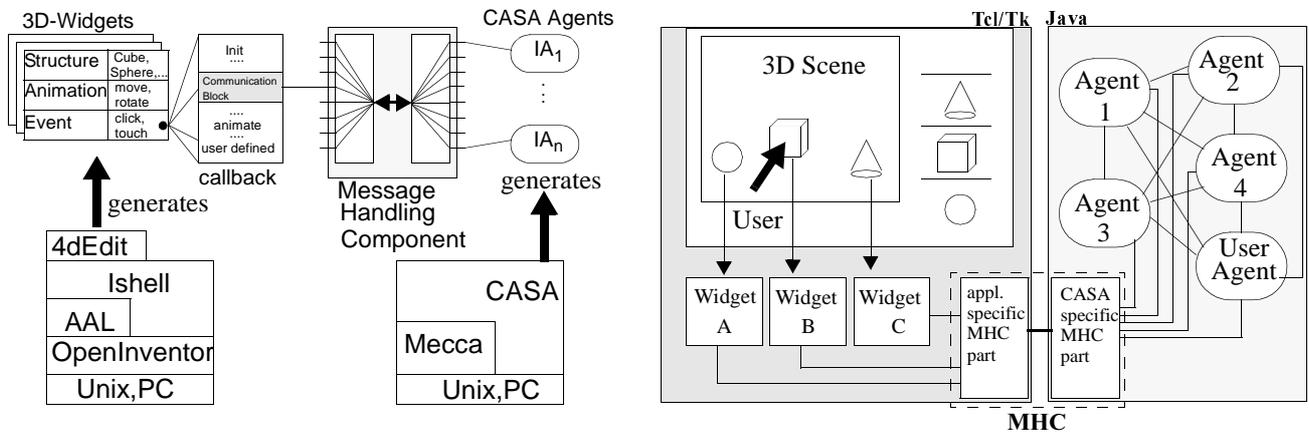
Neben der Weiterführung derzeitiger Aktivitäten steht insbesondere die Verflechtung der einzelnen Projekte im Vordergrund. Als Beispiel einer Verknüpfung der unterschiedlichen Bereiche soll exemplarisch die agentenbasierte Unterstützung von Usabilitytest im Bereich Multi-User 3D-Welten skizziert werden.

### **7.1 Usability von 3D-Welten**

Die Benutzbarkeit von 3D/VR-Schnittstellen ist ein Bereich, in dem zur Zeit noch wenig Aktivitäten existieren, da hauptsächlich auf die effiziente Modellierung und Darstellung fokussiert wird. Bisherige Arbeiten zeigen jedoch, daß die Anwendung dreidimensionaler Grafik in virtuellen Welten nicht automatisch zu einer einfach benutzbaren Schnittstelle führt [15]. Existierende Ansätze zu Usability-Evaluierungen aus dem konventionellen WIMP-Bereich (WIMP=Windows, Icons, Menüs, Pointer) können nur bedingt zur Überprüfung von 3D/VR-Schnittstellen angewandt werden. So bleibt die empirische Untersuchung durch Benutzertests als einzige derzeit verfügbare Technik. Für die Feedback-Repräsentation in Multi-User Welten besteht bei der Darstellung von Aktionen „anderer Benutzer“ das Problem, diese angemessen darzustellen. In solchen Fällen kann es oft vorkommen, daß der Benutzer wichtige Aktionen, die von anderen initiiert wurden, nicht mitbekommt, da er sich zur Zeit mit einem anderen Aspekt der Multi-User-Umgebung befaßt. Hier bietet der Einsatz von Animation und insbesondere von Cartoon-Animationstechniken einen möglichen Ansatz, Aktionen hinreichend wirksam darzustellen. Obwohl diese Idee für 2D-WIMP-Schnittstellen bereits verwendet wurde [10] und auch erste Ansätze im „single user“ 3D-Bereich existieren [12], gibt es zur Zeit keine belegbare Benutzbarkeitsstudie zu dieser Thematik [10].

Dabei besteht insbesondere bei Multi-User-Welten zusätzlich die Notwendigkeit, für jeden Testfall eine ähnliche Situation zu garantieren, ohne den zu testenden Benutzer in seinen Interaktionen zu sehr einzuschränken. Ein offensichtliches Problem besteht in der Simulation der „anderen“ Benutzer, da für empirische Untersuchungen üblicherweise aus Kostengründen keine realen Benutzer (außer der Testperson) zur Verfügung stehen. Einerseits müssen die zu testenden Szenarios in allen Testfällen auftreten, andererseits hängen diese oft von den Aktionen der Testperson ab und können daher zu unterschiedlichen Zeiten in verschiedenen Ausprägungen auftreten. Die Verwendung eines Agentensystems zur Simulation ist hier ein naheliegender Ansatz. Unsere Idee ist es, das entwickelte CASA-System zur Simulation von Aktionen „fremder“ Benutzer zu verwenden. Animierbare Widgets können in kurzer Zeit mit dem Ishell-System erzeugt werden. Jedes Widget besitzt in

der Call-Back Routine einen Kommunikationsblock, der über eine Message Handling Komponente (via TCP/IP) mit dem CASA-System kommunizieren kann.



**ABBILDUNG 7. Kommunikation von 3D-Widgets und CASA-Agenten bzw. 3D-Anbindung**

Abbildung 8 skizziert den Ablauf des Systems: Interagiert der Benutzer mit einem Widget, werden entsprechende Informationen über die Message Handling Komponente an das Agentensystem weitergegeben. Je nach Zielen der einzelnen Agenten können nun unterschiedliche Aktionen der simulierten Benutzer ausgeführt werden. Dabei simuliert in der aktuellen Version ein CASA-Agent eine „fremde“ Benutzertask und nicht einen fremden Benutzer. Ein spezieller CASA Agent protokolliert die Aktionen des Benutzers. Dies dient während des Tests als Informationsquelle für andere Agenten und nach dem Test als Basis für die Auswertung.

Für einen ersten Vortest wurde mit Ishell ein einfaches Szenario entworfen, in dem eine simulierte Multi-User Umgebung realisiert wurde, bei der die Benutzer gemeinsam ein Büro einrichten. Die auftretenden Konflikte (Benutzer A bewegt ein Element von Benutzer B, Selektion eines bereits selektierten Elementes, etc) werden durch 3D-Animationen auf Basis traditioneller Animationsregeln visualisiert. Im Test illustrieren verschiedenen Alternativen (ohne Animation, einfache Animation, CartoonAnimation) dem Benutzer die Aktionen der simulierten „anderen“ Benutzer. Gemessen werden subjektive Kriterien (Intuitivität der Darstellung, Eindruck) als auch quantitative Größen (Schnelligkeit / Genauigkeit beim Lösen einer Aufgabe).

## 7.2 Sonstige Vorhaben

Aus Platzgründen werden im weiteren andere Vorhaben in diesem Bereich nur kurz skizziert:

- „Believable Characters“. Die dargestellten 3D-Figuren (und weitere) sollen mit dem Multi-Agentensystem CASA/Mecca verknüpft werden. Dadurch soll der schnelle Entwurf und Test komplexen (ggf. adaptiven) Verhaltens möglich sein. Als Demonstrator wird beabsichtigt, zusätzlich Arbeiten im Bereich Chat-Roboter zu integrieren und eine Erweiterung in Richtung mobiler Agenten (auf Basis von Mecca) zu untersuchen.
- Der Entwurf neuartiger GUIs basierend auf animierten 3D Widgets soll durch ein strukturiertes Vorgehen ergänzt werden. Neben einem iterativen Prozeßmodell basierend auf explorativem Rapid Prototyping soll die Integration wissensbasierter Techniken als Grundlage betrachtet werden. Des weiteren sollen Techniken aus dem Bereich Usability Engineering in alle Phasen des Entwurfs integriert werden.
- Die bisherigen Ansätze für den Entwurf interaktiver Multimedia-Illustrationen und animierte 3D-Grafik werden kombiniert. In einer laufenden Promotionsarbeit wird an einem szenariobasierten Entwurf gearbeitet, der mittels agentenbasierter Techniken die Gestaltung interaktiver animierter 3D-Illustrationen in frühen Phasen unterstützt. Komplementär wird an der Integration dieser Komponenten in existierende Multimedia-Autorensysteme gearbeitet.



**ABBILDUNG 6. Simulierte Multi-User Umgebung für Benutzbarkeitstest**

- Als Entwurfsunterstützung soll Designwissen, soweit dieses externalisierbar und formalisierbar ist, in Form einer Kritikkomponente ebenfalls in das oben skizzierten Szenario integriert werden.
- Die bisherigen Anwendungsbereiche mobile Roboter, flexible Fertigungssysteme und der Bereich eingebetteter Systeme (z. B. im Automobilbau) sind auch weiterhin die erste Wahl bei der Auswahl aussagefähiger Anwendungsbeispiele.

## 8 Zusammenfassung

Dieser Beitrag gibt einen kurzen Einblick in einige aktuelle Arbeiten des C-LAB, Paderborn. Betrachtet wurden Projekte die sich mit dem Entwurf interaktiver visueller Systeme (z. B. 3D-Welten) befassen und dabei Gebrauch wissensbasierter Techniken machen.

An den beschriebenen Arbeiten waren außer den Autoren weitere Kollegen des C-LAB / der Universität Paderborn sowie studentische Hilfskräfte und Diplomanden beteiligt. Insbesondere die anwendungsorientierten Projekte profitierten dabei von der Kooperation mit anderen Instituten (IPA FhG Stuttgart, AG Wissensbasierte Systeme der Universität Bielefeld, Laboratorium für Konstruktionstechnik Universität Paderborn, etc.).

## 9 Literatur

- [1] B. Bauer, D. Steiner. *Mecca Schulungsunterlagen*, Siemens AG.
- [2] R. Baecker, I. Small. *Animation at the Interface*, B. Laurel. *The Art of Human-Computer Interface Design*, Laurel, Addison Wesley, 1990,
- [3] M. Duecker, B. Gutkauf, and S. Thies. *Negotiation Support for Compiling Knowledge*. Proc. of the GROUP'99 International Conference on Supporting Group Work, November 14-17, 1999, Phoenix, Arizona, USA (to appear).
- [4] S. Flake. *CASA - Entwurf und prototypische Realisierung einer Spezifikationsprache für intelligente Software-Agenten*, Universität-GH Paderborn, 1999
- [5] S. Flake, C. Geiger, G. Lehrenfeld, W. Mueller, V. Paelke. *Agent Based Design of Holonic Manufacturing Systems Using Fuzzy Control*. NAFIPS Workshop on Fuzzy Logic. New York, USA, June 1999
- [6] C. Geiger, W. Mueller, W. Rosenbach. *SAM - An Animated 3D Programming Language*. IEEE Symposium on Visual Languages, Halifax, Canada, Sep. 1998.
- [7] C. Geiger, G. Lehrenfeld, W. Mueller. *Prototyping einer Robotersteuerung durch interaktive 3D-Simulation, Simulation und Visualisierung 99*, Magdeburg, Germany, 1999
- [8] C. Geiger. *Schneller Entwurf interaktiver dreidimensionaler Computeranimationen - Beiträge zur wissensbasierten Modellierung und Illustration komplexer technischer Systeme*. Dissertation. Uni Paderborn, September 1998
- [9] C. Geiger, V. Paelke. *Enhancing 3D User Interfaces with Animation Principles Encapsulated in Agents*, in: ECAI Workshop "Combining AI and Graphics for the Interface of the Future", Brighton, UK, Aug. 1998.
- [10] C. Gonzales. *Does Animation in User Interfaces Improve Decision Making?*. Proc. ACM CHI, ACM Press, April 1996, pp.27-34
- [11] J. M. Huber. *JAM - A BDI-theoretic Mobile Agent Architecture*. Proc. of the 3rd Int. Conf. on Autonomous Agents, Seattle, USA, 1999.
- [12] <http://www.c-lab.de/software/ishell>
- [13] <http://www.c-lab.de/usability>
- [14] M. Latoschik, M. Fröhlich, B. Jung, I. Wachsmuth: *Utilize Speech and Gestures to Realize Natural Interaction in a Virtual Environment*. IECON'98 - Proceedings of the 24th IEEE Conference of the Industrial Electronics Society, Vol. 4, IEEE, 1998,
- [15] K. Kaur. *Designing Virtual Environments for Usability*, Ph.D. thesis, Centre for HCI Design, City University, London, June 1998
- [16] W. Mueller, *Executable Graphics for VHDL-Based Systems Design*, Dissertation, Paderborn University, Paderborn, November 1996.
- [17] A.S. Rao, M. Georgeff. *Modeling Agents Within a BDI-Architecture*. In R. Fikes, E. Sandewall (Ed.): Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning, 1991
- [18] A.S. Rao. *AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language*. in W. Van de Velde, J. P. Perram. *Agents Breaking Away*. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'96, Eindhoven, The Netherlands, Springer 1996.
- [19] F. Thomas; O. Johnston. *Disney Animation - The Illusion of Life*, Bevelled Press, New York, 1981