Multimodale Interaktion mit einem System zur Virtuellen Konstruktion

Marc Erich Latoschik, Bernhard Jung, Ipke Wachsmuth

AG Wissensbasierte Systeme
Technische Fakultät, Universität Bielefeld
Postfach 100131, D-33501 Bielefeld, Deutschland
e-mail: {marcl, jung, ipke}@TechFak.Uni-Bielefeld.DE

Zusammenfassung Dieser Beitrag stellt ein System für die sprachlich-gestische Interaktion zur Steuerung eines Systems zur Virtuellen Konstruktion vor. Eine Übersicht über verschiedene Manipulationsaufgaben in dieser Domäne dient als Grundlage, um Interaktionsbeispiele zu erläutern. Neben deiktischen Gesten des Benutzers werden mimetische Gesten, die gewünschte Veränderungen "vormachen", betrachtet. Diese werden durch sprachliche oder gestische Trigger eingeleitet und bewirken eine Anpassung in den Funktionsmodi der Auswertung, wobei zwischen diskreten und kontinuierlichen Interaktionen unterschieden wird. Um kontinuierliche Modifikationen in der virtuellen Szene umzusetzen, werden neben dem Konzept der Manipulatoren sogenannte Aktuatoren als Repräsentanten für Benutzermodalitäten sowie Motion-Modifikatoren zur Korrektur unscharfer Sensor-Eingaben eingeführt.

1 Sprach- und Gesten-Interfaces für Multimedia-Systeme

Rahmenthema der dargestellten Arbeiten sind Sprach- und Gesten-Interfaces für Multimedia-Anwendungen. Ziel ist die Entwicklung von Techniken, die dem Benutzer den Einsatz grober, auf Körper-, Arm-, Hand- und Fingerstellung basierender gestischer Kommunikation ermöglichen. Damit sollen Begrenzungen von üblichen Bildschirm-Displays überwunden werden und durch sprachlich-gestische Interaktionstechniken für den Einsatz mit Groß-Displays (Wandprojektionen, Workbenches, Caves) ersetzt werden, die ein freistehendes, komfortables Agieren, und dadurch eine möglichst natürliche Form der Mensch-Maschine-Kommunikation (MMK) erlauben. Die Benutzung auf bildschirmorientierte Arbeitsplätze bezogener Eingabegeräte und Interaktionsmetaphern ist mit diesen neuen Ausgabegeräten nicht mehr adäquat. Versuche, die Eingabemetaphern dieser bisherigen WIMP (Windows, Icons, Mouse, Pointer) Interfaces in die dritte Dimension zu transportieren, führten zu der Entwicklung diverser Pointingdevices wie dem Stylus oder der 3D-Space-Mouse. Gerade eine herausragende Qualität VR-gestützter Anwendungen macht jedoch alternative Interaktionsmöglichkeiten wünschenswert: Durch die Art der Simulation steht nicht mehr der Computer, bzw. Desktop-orientierte Metaphern als Werkzeug, im Zentrum der Interaktion, sondern die Anwendung selbst. Ziel ist damit der Verzicht auf eine vermittelnde Schicht zwischen Benutzer und Benutztem. Tastatur und Maus weichen den natürlichen Modalitäten Gestik und Sprache.

Erste Bestrebungen zur Verwendung der Modalitäten Sprache und Gestik in der Mensch-Maschine-Kommunikation reichen bis in die 80er Jahre zurück. Das Put-That-There System [3] war ein früher Versuch, Gestik und Sprache als Eingabemodalitäten auszuwerten. Als „Gestenerkennung“ wurde hier die Zeigerichtung einer Extremität (eines Armes) auf eine zweidimensionale Projektionsfläche mit statischen Objekten ausgewertet; unberücksichtigt blieben zusätzliche Informationen über Körper-, Kopf,

Hand- und Fingerstellung. Der Zeigevektor wurde mit den Ergebnissen eines Wort-basierten Spracherkenners integriert; dabei wurden Plätze verbal unterspezifizierter Referenzen („...this...“, „...there...“) durch die Auswertung der Position eines ständig präsenten, per Armstellung gesteuerten Cursors ausgefüllt. Die Umsetzung der Benutzerinstruktionen nach der Eingabeanalyse erfolgte ausschließlich als diskrete Zustandsänderungen. Viele der in den 90er Jahren entstandenen Arbeiten konzentrieren sich ganz speziell auf die multimodale Integration. Bei gleichzeitiger graphischer Repräsentation von Objekten steht hier vor allem die Benutzerdeixis, also gestisches Zeigen auf Objekte, deren verbale Benennung oder Blickrichtung im Interesse [5][8][13]. Andere Arbeiten konzentrieren sich zwar konkret auf den Einsatzzweck in VR-Umgebungen [1][2], betrachten aber nur eingeschränkte Gestentypen, zum Beispiel symbolische Gesten, und bilden diese auf Systemkommandos ab (vgl. Übersicht in [9]). Diese Einschränkung wird auch in [16] kritisch bemerkt, wobei der Aspekt der Multimodalität jedoch nicht weiter verfolgt wird. Ein Ansatz, ikonische (formbeschreibende) Gesten zu berücksichtigen, findet sich in [15]. Hier dient eine Hand dazu, Kurven im Raum zu beschreiben und zu verändern. Einige der genannten Arbeiten befassen sich zwar als Einzelaspekt mit der Dynamik der jeweils betriebenen Gestenerkennung, legen aber keine Lösungsansätze für den umfassenderen Aspekt der Interaktionsdynamik vor.

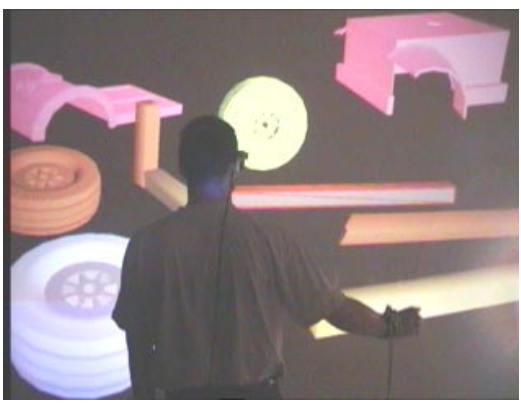

Abbildung 1: Virtuelles Konstruieren an einer interaktiven Wand.

Die im folgenden dargestellten Arbeiten erweitern bisherige sprachlich-gestische Interfaces mit dem Ziel einer möglichst natürlichen Mensch-Maschine-Kommunikation. Dazu werden neben deiktischen auch mimetische („vormachende“) Gesten zugelassen; neben diskreten Interaktionen sind auch kontinuierlich ausgewertete Manipulationen möglich; die Interaktionssemantik wird dabei durch den jeweiligen sprachlichen Kontext moduliert. Die Arbeiten sind eingebettet in das SGIM-Projekt (Speech and Gesture Interfaces for MultiMedia), einem Teilprojekt des Multimedia-NRW Verbundprojekts „Virtuelle Wissensfabrik“[1]. Die technische Realisierung der Spracherkennungskomponente ist Teil eines Partnerprojektes innerhalb des Verbundes und ist anderer Stelle erläutert [4]. In diesem Beitrag wird die Konzeption des SGIM-Systems für die multimodale Integration erläutert. Im Hinblick auf die im nächsten Schritt vorgesehene Zusammenführung von Gestik und Sprache sind die Beispiele in Abschnitt 3 und 4 auf koverbale Gesten bezogen.

[1] Die Forschungsarbeiten in der Virtuellen Wissensfabrik werden unterstützt vom MSWWF des Landes Nordrhein-Westfalen unter OZ IV A3 -107 032 96

2 Manipulationsaufgaben in der Virtuellen Konstruktion

Als Anwendungsszenario für die dargestellten Forschungsarbeiten zur sprachlich-gestischen MMK dient die Steuerung eines Systems zur interaktiven Montagesimulation in virtuellen Umgebungen, des „Virtuellen Konstrukteurs“ [7]. Im diesem werden CAD-basierte Grundbausteine dreidimensional auf einer virtuellen Montagefläche präsentiert; die Aufgabe des Systems liegt in der wissensbasierten Unterstützung des Benutzers beim Zusammenbau dieser Bauteile. Im Virtuellen Konstrukteur bisher verfügbare Interaktionstechniken, sprachliche Instruktionen und (Maus-basierte) direkte Manipulation, beziehen sich auf konventionelle Bildschirm-orientierte Arbeitsplätze. Die hier beschriebenen Arbeiten zielen auf eine sprachlich-gestische Steuerung an Großbild-Displays. Neben der Benutzer-Navigation stellt die Manipulation von Objekten eine zentrale Klasse von Interaktionsaufgaben in virtuellen Umgebungen dar. Im allgemeinen betreffen Manipulationen die folgenden Veränderungen visueller Objektattribute:

- Positionsänderung (Translation)
- Ausrichtungsänderung (Rotation)
- Größenänderung (Skalierung)
- Formänderung (Deformation)
- Erscheinung (Färbung, Texturierung)

Im Zusammenhang der Virtuellen Konstruktion sind insbesondere die ersten drei Manipulationsaufgaben von Interesse. Sie gehören zu den Standardoperationen in CAD- und anderen graphischen Modellierungs Systemen. Im Virtuellen Konstrukteur, der speziell die interaktive Montagesimulation in virtuellen Umgebungen unterstützt, werden zusätzlich folgende Manipulationsaufgaben betrachtet:

- Verbinden von Bauteilen
- Trennen von Bauteilen
- Modifikation von Aggregaten durch Relativbewegung von Komponenten entlang zulässiger Freiheitsgrade von Objektverbindungen

Die Umsetzung dieser Manipulationen basiert im Virtuellen Konstrukteur auf einer wissensbasierten Modellierung der Verbindungsstellen („Ports“) und Verbindungsarten zwischen diesen Ports. Abbildung 2 zeigt Beispiele der bisher modellierten Taxonomie von Port-Typen. Eine weitere Taxonomie klassifiziert mögliche Verbindungsarten bezüglich der zulässigen Freiheitsgrade bei eingegangenen Verbindungen [9]; z.B. sind bei Steckverbindungen Translation und Rotation ungekoppelt, während sie bei Schraubverbindungen gekoppelt sind, wodurch Hinein- und Hinausbewegung von Schrauben mit entsprechender Drehung erfolgt.

Abbildung 2: Typen von Verbindungsports beim Virtuellen Konstruieren: Extrusion ports (links), plane ports (mitte) und point ports (rechts). Bei Verbindungen zwischen den jeweiligen Ports besteht jeweils ein Freiheitsgrad bzgl. Rotation sowie bis zu zwei Freiheitsgrade bzgl. Translation (Abbildungen nach [7]).

Bei den Manipulationsaufgaben der Virtuellen Konstruktion, wie dem Verbinden oder Trennen von Bauteilen, bestehen somit im Vergleich zu den allgemeinen Manipulationsaufgaben – wie Translation und Rotation von Bauteilen – zusätzliche Randbedingungen die im Virtuellen Konstrukteur explizit modelliert und bei der Auswertung von Benutzerinteraktionen zugänglich sind. So kann z.B. das Verbinden von Bauteilen als Spezialfall der Transformation (Translation und Rotation) eines Bauteils betrachtet werden, bei welcher die Zielposition des transformierten Objekts durch die Verbindungsports beider Bauteile eingeschränkt ist. Auf ähnliche Weise kann die Rotation von Teilaggregaten als Spezialfall der Rotation freier, d.h. unverbundener Bauteile betrachtet werden, wobei jedoch die Rotationsachse durch den Typ der Verbindung festgelegt ist. Diese Interaktionsaufgaben sind zunächst Eingabe-unabhängig. Sie können jeweils durch verschiedene Modalitäten wie Sprache oder Gestik, oder mit Hilfe spezieller Eingabegeräte, beispielsweise Maus-basierter Manipulationen, erfolgen. Bei Verarbeitung gestischer Benutzerinteraktionen werden die geschilderten Konstruktionsrandbedingungen ausgenutzt in dem Ungenauigkeiten bei der Gestenerkennung durch systemseitiges Wissen über den Anwendungsbereich ausgeglichen werden, bzw. ungenau erfolgende gestische Benutzerinteraktionen wissensgestützt justiert werden.

3 Diskrete und kontinuierliche sprachlich-gestische Benutzerinteraktionen

In den bisherigen Abschnitten wurden Manipulationsaufgaben in Virtual Reality Systemen im allgemeinen sowie bei der Virtuellen Konstruktion im speziellen betrachtet. Diese Manipulationsaufgaben stellen die Interaktionsziele für die in unserem System behandelten sprachlich-gestischen Benutzereingaben dar. Zur Durchführung einer Manipulationsaufgabe werden, je nach Art der Manipulation, unterschiedliche Informationen benötigt. Soll zum Beispiel ein Objekt rotiert werden, so müssen das Objekt, die Rotationsachse und die Rotationsweite bestimmt werden. Die Art und Weise, wie diese Informationen kommuniziert werden, bzw. wie Benutzereingaben in Änderungen des Systemzustands umgesetzt werden, legt eine Unterscheidung in zwei unterschiedliche Interaktionsmodi nahe: *diskrete* und *kontinuierliche* Interaktionen. Die Abbildungen in diesem Abschnitt zeigen Beispielinteraktionen mit dem SGIM-Demonstrator.

3.1 Diskrete Interaktion

Bei diskreten Interaktionen werden Änderungswünsche des Benutzers als instantane Zustandsänderungen der virtuellen Umgebung umgesetzt. Diskrete Interaktionen können unimodal geäußert werden, z.B. *„Drehe das gelbe Rad um 45 Grad nach hinten“*, oder multimodal, z.B. *„Stecke <Zeigegeste> dieses Rohr <Zeigegeste> da dran“*. Gestische Interaktionen sind dabei zumeist auf Zeigegesten beschränkt, welche Hinweise auf die auszuwählenden Objekte liefern ([10]; vgl. [3][8]). In multimodalen Konstruktionsdialogen sind die (diskreten und kontinuierlichen) Interaktionen des Benutzers oft unterspezifiziert, so daß eine Ergänzung der Eingaben um Kontextwissen über den Anwendungsbereich – bei der Virtuellen Konstruktion etwa Wissen über die Verbindungsmöglichkeiten der Bauteile – und Vorannahmen notwendig ist. Dies bedingt, daß bei der systemseitigen Interpretation von unterspezifizierten Benutzereingaben Systemzustände erzeugt werden können, die nicht den ursprünglichen Intentionen des Benutzers entsprechen. Bei diskreten Interaktionen, deren Auswirkungen sofort in der virtuellen Szene angezeigt werden, sind Korrekturen nur in folgenden Interaktionsschritten möglich. Der Benutzer hat jedoch keine Möglichkeit, die Interpretation einer Anweisung noch während deren Auswertung zu beeinflussen. Der Einsatz von VR-Techniken zielt jedoch oft gerade darauf, den Benutzer in die Szene zu integrieren und ihm unmittelbare Kontrolle der Manipulationen zu ermöglichen. In vielen Bildschirm-orientierten Anwendungen hat sich dafür der Einsatz direkter Manipulationen mittels Maus-Steuerung

als nützlich erwiesen. Im folgenden werden dazu analog kontinuierliche Interaktionen im Kontext der natürlichen Modalitäten Gestik und Sprache betrachtet.

3.2 Kontinuierliche Interaktion

In der menschlichen Kommunikation kommen neben den schon oben behandelten deiktischen Gesten u.a. auch mimetische Gesten vor, die dem „Vormachen“ einer beabsichtigten Änderung dienen (vgl. funktionale Klassifikation von Gestentypen in [10]). Dabei werden die Extremitäten (vor allem die Hände) als Platzhalter gebraucht, um dem Kommunikationspartner die Art und Weise der gewünschten Manipulation vorzumachen. Auf virtuelle Umgebungen bezogen legen mimetische Gesten – im Gegensatz zu diskreten Interaktionen – eine über die Dauer des „Vormachens“ folgende kontinuierliche Veränderung der virtuellen Umgebung nahe (Abb. 3 u. 4).

Abbildung 3: Auswahl und Drag & Drop - Der Benutzer selektiert ein Objekt mittels einer Zeigegeste und führt es durch kontinuierliche Interaktion an einen neuen Platz.

In kontinuierlichen multi-modalen Interaktionen ist mimetische Benutzer-Gestik oft begleitet durch spezifische Schlüsselworte in der sprachlichen Äußerung, z.B. „*so*“ wie in „*Drehe das Rad <Beginn Rotation> so herum <Ende Rotation>*“ (Abb. 4).

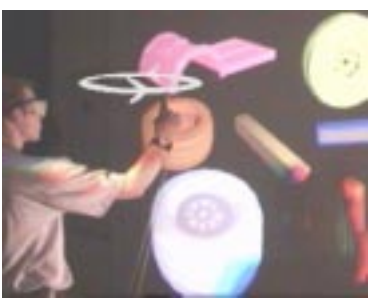

Abbildung 4: Kontinuierliche Interaktion mit mimetischer Gestik zur Beschreibung einer Rotation (Trajektorie zur Veranschaulichung hinzugefügt).

Auswertung, Interpretation und Umsetzung kontinuierlicher Interaktionen erfolgen im SGIM-Demonstrator schritthaltend, wobei Benutzereingriffe zur unmittelbaren Korrektur möglich sind. Bei der Analyse dynamischer Gesten muß i.a. eine zeitbezogene Filterung körperbezogenen Daten erfolgen [11]. Die technische Realisierung diskreter und kontinuierlicher Interaktionen ist im folgenden Abschnitt beschrieben.

4 Interaktionsformen in der Systemmodellierung

Die beiden Interaktionsmodi, diskret und kontinuierlich, erfordern konzeptionelle Unterschiede in ihrer technischen Realisierung. Ihre gemeinsame Auswertung in einem realen System zur Virtuellen Konstruktion bedingt ebenfalls zwei Ausführungsmodi. Bei beliebigen Eingaben, also gesprochenen und klassifizierten Worten oder einzelnen erkannten Gesten, wird während der multimodalen Integration versucht, benötigte Integrationsschemata zu füllen. Der Informationsfluß wird von einzelnen parallelen Erkennermodulen getrieben, welche ihre Resultate als singuläre Events an eine Integrationskomponente weiterleiten (s. Abb. 5).

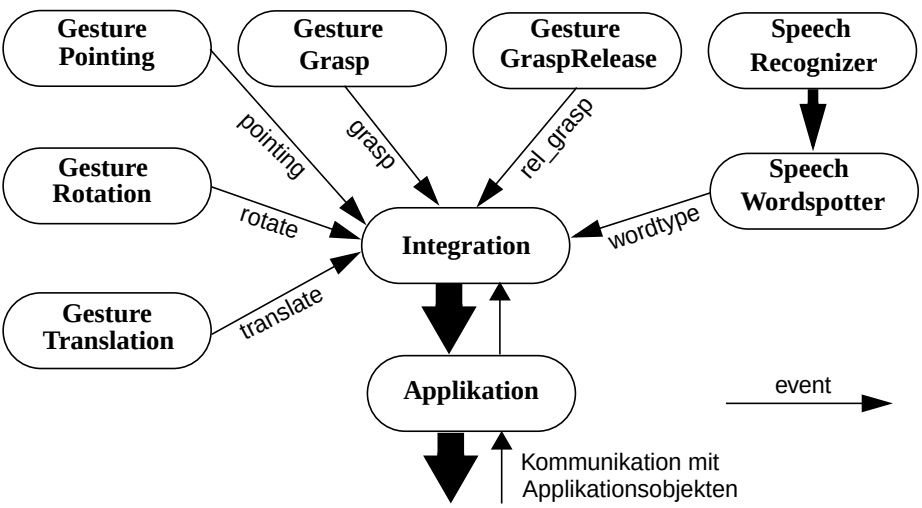

Abbildung 5: Event-getriebene Erkenner und Integrationsstruktur

Die einzelnen Ergebnisse der Erkennermodule werden in der Integration in eine gemeinsame Struktur gebracht. Signifikante Gesten, zum Beispiel ein Zeigen, oder spezielle Aktionswörter, wie „...drehe...“, „...schiebe...“ oder „...verbinde...“, aktivieren jeweils einen speziellen Integrationsframe. Jeder dieser Frames hat spezifische Slots, welche durch die einkommenden Events gefüllt werden. Ein Objekt-Referenzframe benötigt Objektspezifikationen. Verbal können dieses neben Benennungen auch die visuellen Objektattribute Farbe, Lage oder Form sein. Gestisch wird das bedeutete Objekt durch die Richtung während einer Zeigegeste ermittelt. Ziel dieses Referenzframes ist es eine eindeutige Objekt-Instanz zu ermitteln. Dagegen benötigt ein Rotationsframe den Rotationsmittelpunkt, eine Rotationsachse und den Winkel der Änderung. Ist die Integration abgeschlossen, so wird eine mit dem Frametyp assoziierte Funktion ausgeführt. Ein Referenz-Frame aktiviert eine Selektion des referenzierten Objektes, ein Rotations-Frame führt zu einer entsprechenden Objektlage oder -positionsänderung.

4.1 Umsetzung diskreter vs. kontinuierlicher Interaktionen

Der Frame-Abschluß kann durch drei verschiedene Ereignisse ausgelöst werden: Im einfachsten Fall ist der Frame vollständig spezifiziert und die assoziierte Aktion kann im diskreten Ausführungsmodus umgesetzt werden. Ein unterspezifizierter Frame kann durch zwei Arten von Ereignissen in verschiedene Ausführungsmodi gesetzt werden. Detektiert die Spracherkennung das Ende einer Äußerung, und liegt kein Ergebnis eines Gestenerkenners vor, so wird - unter Ergänzung der Eingabe durch Vorannahmen - die Benutzereingabe ebenfalls diskret umgesetzt. Wird ein Triggerwort („...*so*...“) erkannt und eine entsprechende mimetische Geste ausgeführt, so wird in den kontinuierlichen

Modus umgeschaltet und versucht, die unterspezifizierten Werte aus der Gestik zu ermitteln. Kann dieses nicht erfolgen, wird die Interaktion abgebrochen.

4.2 Umsetzung diskreter Interaktionen über Manipulatoren

Die Modellierung graphischer Szenen erfolgt i.a. durch die hierarchische Anordnung der Objekte als Knoten in einem Szenengraphen. Veränderungen werden durch sogenannte *Manipulatoren* ausgeführt. Diese können je nach Art bestimmte Attribute dieser Objekt-Knoten verändern. Für die technische Realisierung einer Selektion und Drehung benötigen wir mindestens zwei Manipulatoren, einen zur Suche des entsprechenden Knotens und anschließender Hervorhebung (Selektion und Highlighting), einen für die Manipulation der Knoten-Transformationsmatrix (Rotation). Abbildung 6 illustriert ein Beispiel für die diskrete Umsetzung einer Rotationsanweisung.

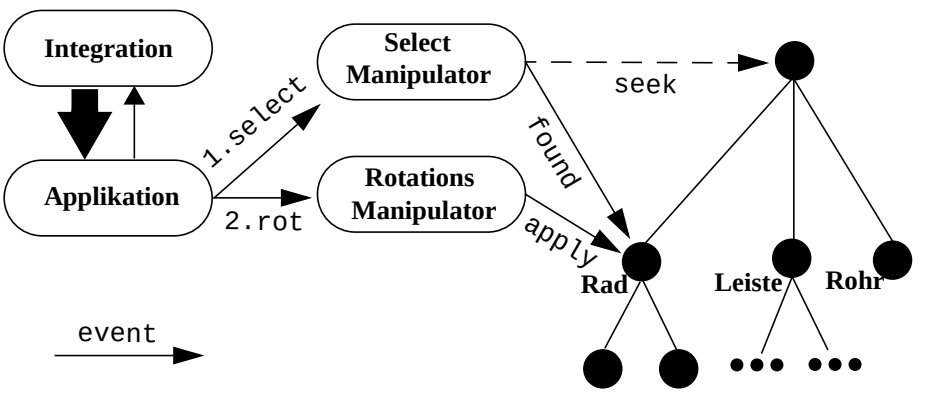

Abbildung 6: Einfache Manipulator-Szenengraphmodifikation bei instantaner Ausführung („Dreh das Rad“)

4.3 Umsetzung kontinuierlicher Interaktionen über Aktuatoren und Motion-Modifikatoren

Auf die Funktion bestimmter Trigger als Einleitung einer mimetischen Beschreibung wurde bereits in Abschnitt 3.2 hingewiesen. Auf der Anwendungsebene bewirkt ein Trigger einen Moduswechsel. Nach einem solchen Modustrigger wird die Interaktion nicht in einem kompletten Schritt mittels eines Manipulators umgesetzt; stattdessen wird die gewünschte Manipulation kontinuierlich aus den Bewegungsänderungen des Benutzers ermittelt. Für diesen Vorgang wird ein mehrstufiges Konzept benutzt.

Datenfluß zwischen den Komponenten

Die multimodale Integration arbeitet auf Event-Basis. Das bedeutet, daß die Kommunikation zwischen Integration, Applikation und den Manipulatoren auf dem Vorhandensein von Nachrichten als diskreten Signalen beruht. Im Gegensatz dazu arbeitet die Visualisierung der virtuellen Szene in einer Schleife, der sogenannten Rendering-Loop. Diese ist treibende Kraft und impliziter Taktgeber, um eine stetige Framerate (Anzahl der gerenderten Bilder/Zeiteinheit) zu gewährleisten. Kommunikation mit der Anwendung geschieht in den Zeiträumen zwischen den Berechnungen der einzelnen Bilder. Events übertragen nur den Wechsel zwischen verschiedenen Zuständen und treten vergleichsweise selten auf, die Rendering-Loop wird kaum belastet. Würde auch die Umsetzung einer kontinuierlichen Interaktion und die Auswertung der Benutzergestik vor der Integration erfolgen, so müßte jede erkannte Geste über die Integrationskomponente und die Applikation als Event weitergegeben werden.

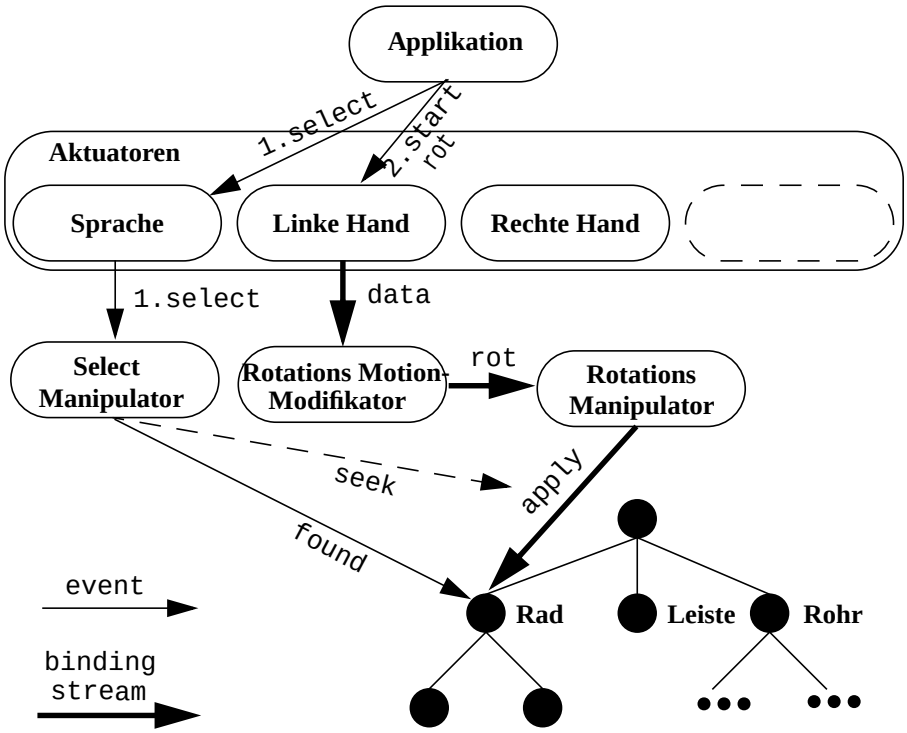

Abbildung 7: Kontinuierliche Interaktion mittels Aktuatoren und Modifikatoren („Dreh das Rad so herum“)

Dieser Ansatz würde in einem Daten-Streamingkonzept resultieren und widersprüche grundlegend der Struktur einer Eventauswertung. Weiterhin läuft die Gestenerkennung parallel ab, zu einem beliebigen Zeitpunkt können für die gleiche Extremität gültige, unterschiedlich gewichtete Resultate vorliegen. Diese würden übermittelt, obwohl sie für eine stattfindende kontinuierliche Manipulation nicht relevant wären. Sie erforderten eine vorgeschaltete Filterung der Erkennerresultate, selbst wenn die entsprechende Extremität gerade eine Manipulation ausführen würde. Die asynchrone Kommunikation fände immer statt. Ein derartiges Event-basiertes Modell ist aus Performanzgründen für flüssige kontinuierliche Interaktion in einem virtuellen Szenario nicht akzeptabel. Wir setzen daher einen in Abbildung 7 illustrierten speziellen Anwendungsmodus ein. Die speziellen Trigger-Events veranlassen die Umschaltung in den kontinuierlichen Modus. Kontrolliert und umgesetzt wird die Benutzer-Interaktion von dann aktivierten Modulen, deren Datenfluß synchron und parallel dem der virtuellen Umgebung ist. Gegenseitige Bindungen zwischen diesen Modulen etablieren über den Zeitraum mehrerer Frames hinweg die kontinuierliche Manipulation.

Die Interaktionen des Benutzers werden durch *Aktuatoren* in der Repräsentation der virtuellen Umgebung vermittelt. Für eine Hand sind dies zum Beispiel die aktuelle Lage und Position des Handgelenkmittelpunktes in Weltkoordinaten. *Motion-Modifikatoren* binden an diese Daten und testen, ob der jeweilige Aktuator zwischen jedem Renderschritt die Modifikator-eigenen Bedingungen erfüllt, zum Beispiel weiterhin eine Drehung ausführt [11]. Ist dieses der Fall, so versorgen sie entsprechende Manipulatoren über den Zeitraum der Bindung mit kontinuierlichen, durch Template-Intervallvergleich geglättete Manipulationsanweisungen. Sind die Bedingungen nicht mehr erfüllt,

signalisieren die Modifikatoren den Abbau der Bindungen, die Interaktion wird insgesamt beendet. Aktuatoren und Motion-Modifikatoren sind, im Gegensatz zu der Event-getriebenen Erkennung, synchronisiert mit dem Datenfluß der virtuellen Umgebung. Für jedes neue zu berechnende Bild wird ein Update der eingebetteten Objekte durchgeführt. Solange Bindungen zwischen Aktuatoren, Motion-Modifikatoren und Manipulatoren bestehen, werden bei jedem neuen Frame die internen Aktionen der gebundenen Objekte durchgeführt; gebundene Aktuatoren können keine weitere Aktion als die gerade aktuelle ausführen. Da die einzelnen Erkenner ihre Ergebnisse in Form von gewichteten Hypothesen an das System weitergeben, und da gewisse Formanteile einer Geste denen einer anderen zu erkennenden Geste entsprechen können (die Merkmalsvektoren der einzelnen Gesten sind nicht vollständig orthogonal), kann es bei den Erkennern zu Überschneidungen kommen. Ist aber der durch die Erkenner referenzierte Aktuator bereits in einer kontinuierlichen Manipulation gebunden, so werden alle anderen diesen Aktuator möglicherweise betreffenden Erkenner-Events ignoriert. Inkonsistenzen im Interaktionsfluß werden so vollständig vermieden.

Die strikte Unterscheidung zwischen diskreter und kontinuierlicher Manipulation bietet einen weiteren Vorteil für zukünftige Arbeiten. Beispielinteraktionen wie „Dreh die Leiste *so* um diese Rad“ offenbaren weitere Herausforderungen. Offensichtlich folgen nach der Einleitung der kontinuierlichen Interaktion (Trigger „so“) weitere Informationen über die Manipulation. Die Modalität würde hier *mehrfach* gewechselt, um die Interaktion zu beschreiben. Hier könnte ein Korrekturansatz (wie in [12] erarbeitet) nützlich sein. Die nach einem Trigger schon erfolgende kontinuierliche Manipulation könnte abgebrochen, und die Erkennerergebnisse des Modifikators (z.B. über die erkannte Rotationsachse) an die Applikation und die Integration zurückgegeben werden, um eine vollständige Spezifikation für eine diskrete Manipulation in der Szene zu generieren.

5 Zusammenfassung / Stand der Realisierung

In diesem Beitrag haben wir ein System zur sprachlich-gestischen Steuerung einer Anwendung der Virtuellen Konstruktion vorgestellt. Im Gegensatz zu anderen und als Erweiterung unserer bisherigen Arbeiten, werden dabei, neben deiktischen und symbolischen Gesten, insbesondere auch Gesten mit mimetischem Charakter verarbeitet. Erste Experimente bestätigen die Nützlichkeit dieser Gestentypen in Fällen, wo ausschließlich sprachliche Äußerungen zu komplex, unpräzise oder unnatürlich sind.

Zur Verarbeitung von sowohl diskreten wie auch kontinuierlichen multimodalen Interaktionen wurde ein gemischt Event/Binding-basiertes Architekturkonzept vorgestellt, das unterschiedlich getriebene und synchronisierte Programmkomponenten umfaßt. Die Umsetzung kontinuierlicher Interaktionen erfolgt dabei über Aktuatoren, Motion-Modifikatoren und Manipulatoren, welche durch sprachlich oder gestisch getriggerte Events in den Kontext der VR-spezifischen Rendering-Loop gesetzt und für die Dauer einer Interaktion aneinander gebunden werden. Das vorgeschlagene Architekturkonzept leistet über die Manipulatoren auch die Integration in das Anwendungssystem zur Virtuellen Konstruktion.

Im gegenwärtigen Demonstratorsystem des SGIM-Projekts sind Interaktionen zur Deixis-Auswertung und zur Führung von Objekten vollständig implementiert. Die Auswertung von Rotationen befindet sich im Experimentalstadium. Derzeitige Arbeiten beinhalten u.a. die Portierung des Systems von Performer[2] auf Avocado, einer Plattform zum Rapid Prototyping von VR-Anwendungen [14].

[2] Echtzeit- und multiprozessorfähige 3D-Graphikbibliothek der Firma Silicon Graphics.

6 Literatur

1. K. Böhm, W. Hübner & K. Väänänen: *GIVEN: Gesture Driven Interactions in Virtual Environments, A Toolkit Approach to 3D Interactions*. In Interfaces to Real and Virtual Worlds, Montpellier, France, 1992.
2. K. Böhm, W. Broll & M. Sokolewicz: *Dynamic Gesture Recognition Using Neural Networks; A Fundament for Advanced Interaction Construction*. In SPIE Conference Electronic Imaging Science & Technology, San Jose California, USA, 1994.
3. R. A. Bolt: *„Put-That-There": Voice and Gesture at the Graphics Interface*, Computer Graphics 14(3), S. 262-270, 1980.
4. G. A. Fink, C. Schillo, F. Kummert & G. Sagerer: *Incremental speech recognition for multimodal interfaces.* In IECON'98: Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, Vol. 4, IEEE, 1998.
5. A. G. Hauptmann & P. McAvinney: *Gestures with speech for graphic manipulation*. In International Journal of Man-Machine Studies, Vol. 38, S. 231-249, 1993.
6. C. Huls, E. Bos & W. Claassen: Automatic Referent Resolution of Deictic and Anaphoric Expressions. In Computational Linguistics, Vol. 21, No 1, S.59-79, 1995.
7. B. Jung, M. Latoschik & I. Wachsmuth: Knowledge-Based Assembly Simulation for Virtual Prototype Modeling. *IECON'98 -Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, Vol. 4*, IEEE, 1998, 2152-2157.
8. D. B. Koons, C. J. Sparrell & K. R. Thorisson: *Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures*. In M. Maybury (Ed.): Intelligent Multimedia Interfaces, AAAI Press, S. 257-276, 1993.
9. S. Kopp: *Ein wissensbasierter Ansatz zur Modellierung von Verbindungen zur virtuellen Montage*, Diplomarbeit an der Technischen Fakultät der Universität Bielefeld, 1998.
10. M. Latoschik & I. Wachsmuth: *Exploiting Distant Pointing Gestures for Object Selection in a Virtual Environment.* In I. Wachsmuth & M. Fröhlich (Eds.): Gesture and Sign Language in Human-Computer Interaction, (pp. 185-196), Lecture Notes in Artificial Intelligence, Volume 1371, Springer-Verlag, 1998.
11. 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 Annual Conference of the IEEE Industrial Electronics Society, Vol. 4, IEEE, S. 2028-2033, 1998.
12. B. Lenzmann: *Benutzeradaptive und Multimodale Interface-Agenten*. Dissertation an der Technischen Fakultät der Universität Bielefeld, Infix Verlag, DISKI 184, 1998.
13. M. T. Maybury: *Research in Multimedia and Multimodal Parsing and Generation.* In P. McKevitt (Eds.): Journal of Artificial Intelligence Review: Special Issue on the Integration of natural Language and Vision Processing, Vol. 9, Kluwer, 1995.
14. H. Tramberend: *Avocado: A Distributed Virtual Reality Framework.* In L. Rosenblum, P. Astheimer & D. Teichmann (Eds.): Proceedings of the Virtual Relity'99 IEEE Conference, Houston, USA, S. 14-21, 1999.
15. D. Weimer & S. K. Ganapathy: Interaction Techniques using Hand Tracking and Speech Recognition. In M. M. Blattner & R. B. Dannenberg (Eds.): Multimedia Interface Design, ACM Press, S. 109-126, 1992.
16. A. D. Wexelblat: *An Approach to Natural Gesture in Virtual Environments*. In ACM Transactions on Computer-Human Interaction, Special Issue on Virtual Reality Software and Technology, Vol. 2 #3, 1995.