

# **Virtuelle Werkstatt – Multimodale Interaktion für intelligente virtuelle Konstruktion**

***Dipl. Inform. Christian Fröhlich***

*Arbeitsgruppe Wissensbasierte Systeme – Technische Fakultät*

*Universität Bielefeld*

*Universitätsstraße 25, 33615 Bielefeld*

*0521 – 106 2922*

*0521 – 106 2962*

*cfroehli@techfak.uni-bielefeld.de*

***Prof. Dr. Ipke Wachsmuth***

*Arbeitsgruppe Wissensbasierte Systeme – Technische Fakultät*

*Universität Bielefeld*

*Universitätsstraße 25, 33615 Bielefeld*

*0521 – 106 2924*

*0521 – 106 2962*

*ipke@techfak.uni-bielefeld.de*

***Prof. Dr. Marc Erich Latoschik***

*Arbeitsgruppe Multimedia und Visualisierung*

*Universität Bayreuth*

*Universitätsstraße 30, 95447 Bayreuth*

*0921 – 55 7601*

*marc.latoschik@uni-bayreuth.de*

## **Zusammenfassung**

Das von der Deutschen Forschungsgemeinschaft geförderte Projekt „Virtuelle Werkstatt“ wurde im November 2001 mit dem Ziel begonnen, computergrafisch visualisierte dreidimensionale Modelle realer Konstruktionsteile in der Virtuellen Realität erprobbar zu machen.

Das Projekt wurde in zwei Phasen gefördert, wobei die erste von November 2001 bis Dezember 2004 und das anschließende Folgeprojekt von Januar 2006 bis Dezember 2007 durchgeführt wurde. Während dieser zwei Förderungsphasen wurden viele technische Neuerungen auf dem Gebiet der Virtuellen Konstruktion erarbeitet. Zu diesen Neuerungen gehören unter anderem Arbeiten in den Bereichen der Künstlichen Intelligenz, des Virtuellen Prototypings, der multimodalen Mensch-Maschine-Interaktion, sowie der Softwareentwicklung für Simulationssysteme der Virtuellen Realität.

In diesem Artikel wird ein Überblick über die durchgeführten Arbeiten gegeben. Der Beitrag geht dabei weniger auf gängige methodische Vorgehensweisen aus dem Bereich der CAD-basierten Produktentwicklung ein, sondern konzentriert sich auf innovative Ansätze des Projektes „Virtuelle Werkstatt“. Kapitel 1 beschäftigt sich mit einer Einleitung in das Thema „Virtuelles Konstruieren“ und erläutert verwandte Arbeiten auf diesem Gebiet. Hierbei wird die Virtuelle Werkstatt im Kontext dieser Arbeiten eingeordnet. Kapitel 2 behandelt im Detail die Arbeiten auf dem Gebiet der intelligenten Computergrafik, bevor in Kapitel 3 auf Neuerungen in der multimodalen Mensch-Maschine-Interaktion eingegangen wird. Abschnitt 4 erläutert die durchgeführten Arbeiten in Bezug auf Software Engineering für intelligente virtuelle Umgebungen. Kapitel 5 zieht abschließend ein Fazit über das Projekt.

### **Schlüsselwörter**

Virtuelles Prototyping, multimodale Mensch-Maschine-Kommunikation, Künstliche Intelligenz, Virtuelle Realität

*Ersch. in: Gausemeier, J. & Grafe, M. (eds.) 8. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, p. 241–255. Paderborn: HNI (2009)*

# **1 Einleitung: Virtuelle Konstruktion am Beispiel eines Citymobils**

Virtuelle Konstruktion nimmt in heutigen Produktionssystemen bereits einen hohen Stellenwert ein. Das Verwenden virtueller Prototypen erspart den Entwicklern viel Zeit im Designprozess, wie auch enorme Kosten, die bei der Konstruktion realer Prototypen entstehen. Das virtuelle Prototyping beschäftigt sich im Allgemeinen mit der Konstruktion von nicht-realen Prototypen und wird häufig in verschiedenen Bereichen des Maschinenbaus eingesetzt. In der Automobilindustrie wird innerhalb des virtuellen Designprozesses die Handhabbarkeit neuer Einzelteile und Modelle, sowie auch die Ergonomie neuer Fahrzeugcockpits erforscht und verbessert. Der Fokus der virtuellen Werkstatt in Bezug auf das virtuelle Prototyping liegt auf der Verbesserung der Interaktion mit den Konstruktionssystemen, sowie auf dem intelligenten Zusammenspiel einzelner Bauteile und Komponenten.

## **1.1 Verwandte Arbeiten**

Auf dem Gebiet der Virtuellen Konstruktion werden in der Regel zwei unterschiedliche Arten des Konstruierens unterschieden. Eine Art beschäftigt sich mit der realitätsgetreuen externen Modellierung der virtuellen Bauteile, beispielsweise mit Hilfe von CAD-Modellierungswerkzeugen, wie zum Beispiel bei (Loock & Schömer, 2001) oder (Zachmann & Rettig, 2001). Diese Bauteile werden dann anschließend in der virtuellen Realität zusammengebaut, um sie im Zusammenspiel zu erproben. Der Nachteil an dieser Methode ist, dass die Teile meist statisch sind und nicht in der Anwendung direkt verändert werden können. Stellt man also ein Problem bei der Konstruktion des Gesamtsystems fest, muss das Bauteil erst wieder in den CAD-Werkzeugen überarbeitet werden, und der Designprozess beginnt von vorne. Eine sogenannte Variantenkonstruktion in Echtzeit ist mit dieser Art der Konstruktion nur sehr selten möglich.

Bei der zweiten Art der Konstruktion werden die virtuellen Objekte mit semantischem Wissen angereichert, siehe auch (Soto & Allongue, 2002) und (Peters & Shrobe, 2003). Die Verbindung von Methoden der Künstlichen Intelligenz und der Virtuellen Realität (siehe (Luck & Aylett, 2000)) hat in den letzten Jahren immer mehr an Bedeutung gewonnen und wurde in verschiedenen Richtungen erforscht. (Cavazza & Palmer, 2000) schlagen für die Integration des semantischen Wissens eine gemeinsam genutzte Schicht vor, um die Wiederverwendbarkeit und den Grad der Adaptivität zu verbessern. Dieses Wissen umfasst sowohl intelligente Verbindungsstellen, sowie Informationen über mögliche parametrische Veränderungen der Teile. Die bei diesem Ansatz verwendeten Bauteile sind jedoch anders als die bei der ersten Methode gebrauchten CAD-Modelle nicht so detailliert und originalgetreu und erinnern eher an ein Baukastensystem. Allerdings bieten sie dem Benutzer deutlich mehr Interaktionsmöglichkeiten.

## 1.2 Die Virtuelle Werkstatt

Die virtuelle Werkstatt verfolgt den zweiten der oben genannten Ansätze, wobei ein starker Fokus auf der Anreicherung der virtuellen Bauteile mit semantischem Wissen durch die sogenannten *Semantic Entities* (Latoschik, Biermann, & Wachsmuth, 2005) gelegt wird. Diese semantischen Informationen enthalten sowohl Wissen über mögliche intelligente Verbindungsstellen der Teile, als auch über Parameter, die verwendet werden, um Teile innerhalb der Anwendung in Echtzeit zu verändern. So sind zum Beispiel Skalierungen der Teile bei Erhalt ihrer Verbindungsstellen möglich. Detaillierte Informationen zum Thema *Semantic Entities* und intelligente Computergrafik folgen in Kapitel 2. Ein weiterer Aspekt, der die Echtzeitfähigkeit der virtuellen Werkstatt fördert, ist die multimodale Mensch-Maschine-Interaktion. Diese ermöglicht durch eine Integration von gesprochener Sprache und verschiedenen Arten der Gestik eine natürliche Interaktion mit dem System (siehe Kapitel 3).

### 1.2.1 Das technische Setup

Der Arbeitsplatz und Interaktionsraum der Virtuellen Werkstatt wird mittels einer CAVE-artigen dreiseitigen Mehrseitenprojektions-Umgebung realisiert (siehe Abbildung 1). Auf jeder der Seiten werden zwei Bilder projiziert, eines für das rechte Auge und eines für das linke. Die beiden Bilder werden jeweils durch zirkuläre Polarisationsfilter vor den Projektoren wie auch an der zu tragenden Brille des Benutzers getrennt, um einen orientierungs-unabhängigen Stereoeffekt zu erzeugen. Jedes der insgesamt sechs Bilder wird von einem einzelnen Rechner erzeugt, sodass allein für die Darstellung der Szene sieben Computer benötigt werden, wobei einer davon als Server der Anwendung fungiert und sowohl die Applikationslogik verwaltet, wie auch die Verteilung der Szene auf die sogenannten Renderclients übernimmt. Die einzelnen Rechner sind untereinander über ein Hochgeschwindigkeits-Netzwerk (Infiniband) verbunden, welches einen sehr hohen Datendurchsatz (bis zu 20Gbit/s) bei einer sehr geringen Latenzzeit im niedrigen Millisekundenbereich erlaubt.

Für die Interaktion ist ein Trackingsystem der Firma A.R.T. im Einsatz, welches in unserem Setup mit neun Kameras betrieben wird. Diese Kameras senden infrarotes Licht aus, welches von kleinen Markern reflektiert wird, die der Benutzer in der CAVE am Körper trägt. In einer typischen Anwendung befinden sich dedizierte Markertargets an den Händen, um die Gestik des Benutzer erfassen zu können, sowie an einer Brille, damit die Perspektive immer korrekt für den Anwender berechnet werden kann, um stets einen realistischen immersiven Eindruck von der Szene vermitteln zu können. Um verschiedene Arten der Gestik verarbeiten zu können, ist ein einfaches optisches Tracking der Hände natürlich nicht ausreichend, da diese nur Trajektorien der Hände erfassen können. Will man auch beispielsweise Greif- oder Zeigegesten erkennen können, muss zusätzlich zu den Trajektorien die Fingerstellung des Benutzers erkannt werden. Dies wurde in der virtuellen Werkstatt durch Datenhandschuhe (Cybergloves) realisiert, wel-

che über den Widerstand der eingearbeiteten Bimetall-Streifen in der Lage sind die Gelenkwinkel der Finger zu bestimmen. Gegen Ende des Projektes wurde auch ein optisches Trackingverfahren der Finger von der Firma A.R.T. erfolgreich exploriert, welches ein haptisches Feedback an den Fingerspitzen enthält.



Abbildung 1: Eine beispielhafte Szene der virtuellen Werkstatt beim Zusammenbau eines "Citymobils".

## 2 Wissensbasierte Computergrafik

Auf dem Gebiet der Wissensbasierten Computergrafik hat die Virtuelle Werkstatt einige Neuerungen erbracht. Die virtuellen Bauteile werden mit semantischem Wissen angereichert, welches intelligente Verbindungen, sogenannte Ports, wie auch parametrische Veränderungen der Teile, beispielsweise Skalierungen oder Formveränderungen, ermöglicht. Dieses Wissen, wird über *Semantic Entities* an den Teilen verankert.

*Semantic Entities* reichern die virtuellen Bauteile mit semantischem Wissen an, indem sie eine Wissensrepräsentationsschicht zu der Simulation hinzufügen. Mit Hilfe der *Semantic Entities* ist es dem Applikationsentwickler auf sehr leichte Art und Weise möglich, Objekten bestimmte Attribute zuzuweisen, die in einer grafischen Repräsentation allein nicht zu realisieren wären. So kann in einem *Entity* beispielsweise annotiert werden, ob ein Objekt auswählbar oder skalierbar ist, oder aber auch, ob es mit anderen Teilen kollidieren oder mit diesen verbunden werden kann.

## 2.1 Intelligente Bauteile

Bauteile sind mit Wissen darüber angereichert, mit welchen „Partnern“ sie sich verbinden können und wie sie ihre Form verändern können. Am Erprobungsbeispiel eines virtuellen Citymobils „weiß“ ein Rad, dass es nur mit den vier entsprechenden „Ports“ verbunden werden kann, welche sich am Fahrwerk des Fahrzeuges befinden. Bei jeder detektierten Kollision mit einem anderen Bauteil wird geprüft, ob der entsprechende Gegenport vorhanden ist. Falls dieser nicht gefunden wird, wird keine Verbindung hergestellt. Wird allerdings eine Kollision mit dem Fahrwerk erkannt, wird zunächst geprüft, ob noch ein Port für ein Rad frei ist. Ist dieses nicht der Fall, bleibt die Kollision erneut ohne Ergebnis. Befindet sich aber genau eine freie kompatible Verbindungsstelle an dem Fahrwerk, wird der Reifen mit dieser verbunden. Für den Fall, dass mehrere freie Ports zur Verfügung stehen, werden diese durch eine Metrik qualifiziert, welche die relative Rotation des Reifens und des Ports, sowie auch den Abstand der beiden einbezieht. Diese Metrik liefert für jedes mögliche Paar von Verbindungen einen qualitativen Wert, sodass zwischen dem Paar mit dem besten Wert anschließend eine intelligente Verbindung aufgebaut werden kann. Die hergestellten Verbindungen werden von sogenannten *Constraint Mediatoren* überwacht.



Abbildung 2: Zweihändige Skalierung eines Bauteiles in der Virtuellen Werkstatt.

Die intelligenten Bauteile verfügen außerdem über Wissen darüber, ob sie sich in ihrer Form verändern können, und wenn ja, in welcher Form und Dimension. So sind in der virtuellen Werkstatt parametrische Skalierungen möglich. Parametrische Skalierungen bewirken, dass zum Beispiel ein Reifen, bei Erhalt der Dimension und relativen Position seiner Verbindungsstelle am Bauteil, skaliert werden kann. Bei diesem konkreten Beispiel werden also nur die „Felge“ und der „Gummireifen“ in ihrer Größe verändert (siehe Abbildung 2). Das Loch in der Mitte, welches den Verbindungsport repräsentiert, bleibt in seiner Ausdehnung unverändert. Diese Skalierungen werden durch hierarchisch verknüpfte Constraints ermöglicht, welche anschließend in einen Propagationsgraphen umgesetzt werden. Diese benötigten Strukturen werden in dem deklarativen XML-Format *VPML* (Variant Part Markup Language (Biermann & Jung, 2004)) spezifiziert. Diese Spezifikation enthält sowohl den relativen Aufbau der Einzelteile zueinander, wie auch Definitionen der *Semantic Entities* und sogenannten *Constraint Mediatoren*. Con-

*straint Mediatoren* dienen zum Beispiel dazu, hergestellte Verbindungen zwischen zwei Bauteilen zu überwachen. Sie sorgen dafür, dass die Bauteile sich miteinander bewegen, sodass der realistische Eindruck einer „echten“ Verbindung der Teile entsteht.

## 2.2 Parametrisierung von gekrümmten Formen

Erweiterungen des VPML-Formates über Templates (Biermann, Fröhlich, Latoschik, & Wachsmuth, 2007) erlauben eine Simulation krümmbarer Extrusionsformen in Echtzeit, durch Formaflösung im Krümmungsbereich. Die Kombination variabler Segmentanzahlen und sukzessiver Elementtransformationen ermöglicht Krümmungen unter Erhalt des korrekten optischen Eindrucks. Abbildung 3 zeigt Bauteile, die mittels Template-Definitionen mit verschiedenen Ausgangsformen und Krümmungsparametern erzeugt wurden. Die Definition von Templates über VPML-Beschreibungen erlaubt das Instanzieren komplexer Bauteile durch eine Meta-Beschreibung. Sie gestattet den Aufbau beliebiger Segmentanzahlen, die sich nach geometrischen Constraints gleichförmig bewegen und Krümmungen approximieren. Abbildung 4 zeigt Definition und schematischen Aufbau eines krümmbaren Rohres. Die Kopplung der Transformationen der Segmente über die in dem VPML-Template definierten geometrischen Constraints erlaubt eine einfache Parametrisierung von Krümmungen der Bauteilvarianten.

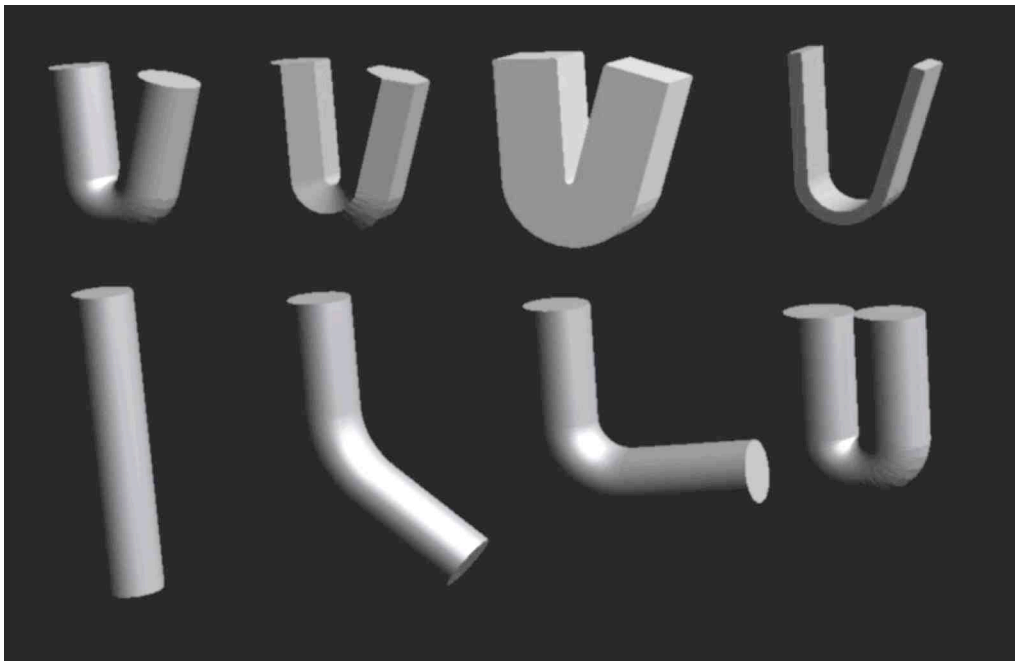


Abbildung 3: Unterschiedliche gekrümmte Formen und Krümmungswinkel.

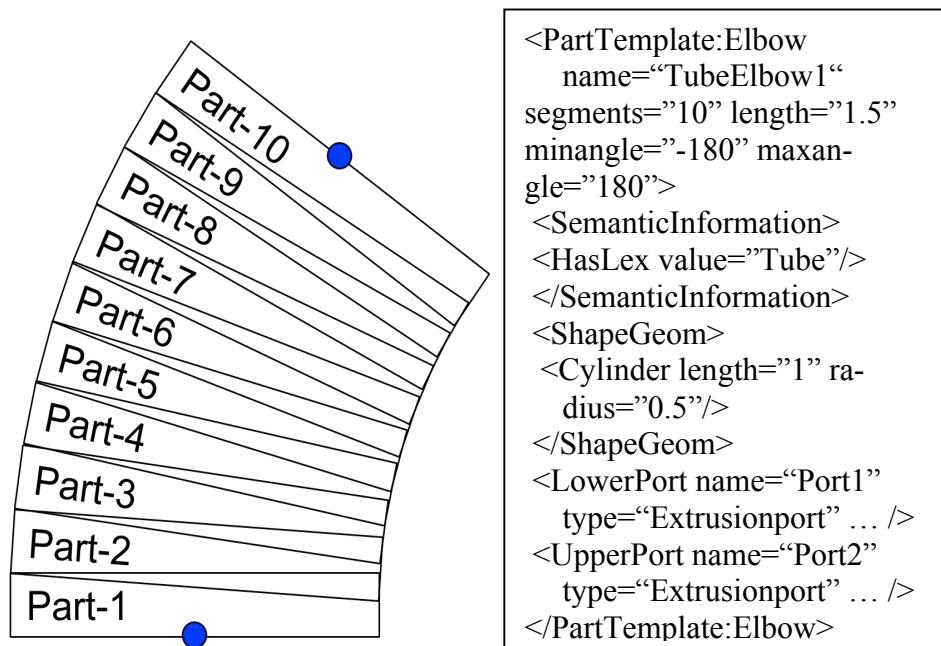


Abbildung 4: Template-Definition eines krümmbaren Rohres mit 10 Segmenten.

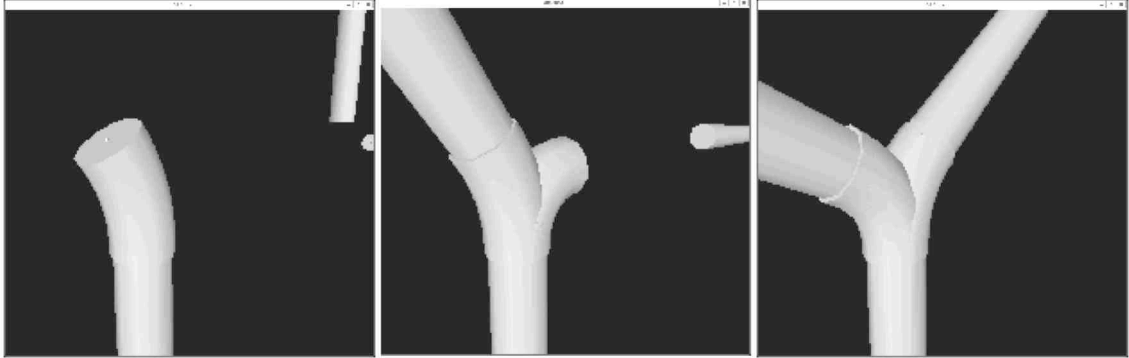
### 2.3 Morphologische Veränderung der Bauteile

Mit Hilfe der im Interaktions-Framework vorhandenen Möglichkeit, Sensoren und Constraints zu schaffen und in der virtuellen Szene zu platzieren, wurde ein intelligenter Annäherungssensor entwickelt. Dieser Sensor ist in dem Bauteil platziert, das eine neue Verbindungsstelle ausprägen soll. Durch den Zugriff des Sensors auf die semantischen Informationen der Szene über *Semantic Entities* kann die Anzahl der eventuell noch benötigten und der vorhandenen, freien Verbindungsstellen (Ports) verglichen werden. Durch dieses Vorgehen kann direkt während der Interaktion in der virtuellen Umgebung auf neue Konfigurationsmöglichkeiten der Ports eingegangen werden. Ist eine kompatible Verbindungsstelle in der Nähe, für die ein weiterer Port geschaffen werden muss und die Verbindungsstelle das erlaubt, kann dieser mittels einer Anpassung der Geometrie und der semantischen Struktur etabliert werden (weitere Informationen hierzu finden sich im nächsten Abschnitt). Durch den flexiblen Aufbau und Erweiterbarkeit des beschreibenden XML-Formates für virtuelle Bauteile (VPML) kann der Annäherungssensor und seine Konfiguration direkt in der Beschreibungssprache definiert werden.

Durch die durchgeführten Arbeiten ist es möglich, krümmbare und skalierbare Bauteile zu erstellen. Für die Ausprägung einer neuen Geometrie werden diese flexiblen Teilstücke eingesetzt, um zum Beispiel eine neue Verbindungsstelle für eine Rohrverzweigung zu generieren (siehe Abbildung 5). Die lokalen Constraints, welche mit Hilfe von Constraint-Mediatoren etabliert werden, sorgen hierbei dafür, dass die vorhandenen und die neu etablierte Verbindungsstelle bei Bedarf weiterhin flexibel gekrümmt



werden können (siehe Abbildung 5 rechts). Zusätzliche Constraints überwachen die Positionen der Verbindungsstellen, um Überlappungen der Ports beim Verbiegen der Bauteile zu vermeiden.



*Abbildung 5: Rohrverzweigung mit optionaler, dritter Verbindungsstelle mit (von links nach rechts) einem belegtem Port, zwei belegten Ports und Ausprägung einer weiteren Verbindungsstelle, drei belegten Ports und veränderten Biegefactoren der flexiblen Teilstücke.*

Die zusätzlichen Geometrien und Eigenschaften der Verbindungsstellen können als optionale Teilstücke in der Beschreibungssprache vordefiniert werden. Dieses erlaubt eine automatische Generierung der flexiblen Geometrien und der überwachenden Constraints direkt aus der VPML-Beschreibung.

### **3 Multimodale Mensch-Maschine-Kommunikation**

Die multimodale Mensch-Maschine-Kommunikation spielte in der Virtuellen Werkstatt von Beginn an eine zentrale Rolle. Es sollte den Benutzern möglich gemacht werden auf natürliche Art und Weise (sowohl mittels Sprache wie auch mit Hilfe von Gesten) mit dem System zu kommunizieren und es instruieren zu können. Hier wurde eine Integration von Sprache und Gestik in dem System realisiert, welche es erlaubt, gleichzeitige sprachliche Äußerungen und verschiedene Arten der Gestik miteinander zu synchronisieren, um diesen eine neue Bedeutung zu geben.

Für diese Integration von Sprache und Gestik wurde eine Erweiterung eines *Augmented Transition Network* implementiert, das sogenannte *tATN* (Latoschik, 2002). Die Erweiterungen ermöglichen über das Hinzufügen von Zeitregistern die parallele Exploration von Alternativpfaden, um so mögliche Integrationshypothesen gleichzeitiger Äußerungen (Sprache und Gestik) zu generieren.

### **3.1 Verarbeitung verschiedener Gestentypen**

In der Virtuellen Werkstatt ist es dem Benutzer möglich durch die Verwendung verschiedener Gestentypen mit dem Benutzer zu kommunizieren. Diese Typen schließen deiktische (zeigende), kinemimische (bewegungsnachahmende) und ikonische (formbeschreibende) Gesten ein. Im Folgenden sollen einige Details und Anwendungsbeispiele für diese verschiedenen Typen gegeben werden, wobei der Fokus hier auf den ikonischen Gesten (Fröhlich, Biermann, Latoschik, & Wachsmuth, 2009) liegen wird.

#### **3.1.1 Deiktische Gesten**

Zeigegesten sind in der virtuellen Konstruktion vor allem für die Auswahl von Teilen von Bedeutung, welche nicht im direkten Greifraum platziert sind. Die Auswahl eines Teiles über deiktische Gesten ist immer mit einer sprachlichen Äußerung koordiniert, um das Teil gegebenenfalls noch weiter spezifizieren zu können.

#### **3.1.2 Kinemimische Gesten**

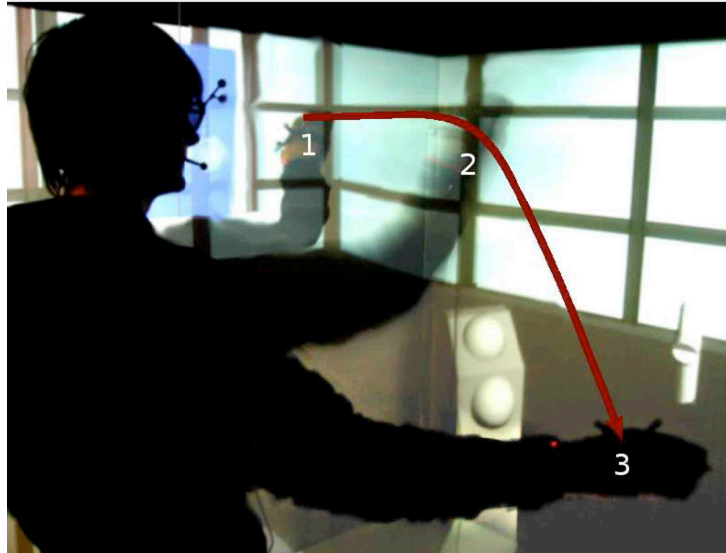
Sogenannte kinemimische Gesten finden in der virtuellen Werkstatt bei der Rotation von Teilen Verwendung. Die Teile werden entsprechend der Handbewegung eines Benutzers rotiert. Hierbei werden sowohl die Rotationsachse der Bewegung, wie auch ihre Geschwindigkeit mit einbezogen. Die von dem Trackingsystem erkannte Trajektorie wird dabei entsprechend auf das virtuelle Teil übertragen, was auch eine Rotation von Teilen erlaubt, welche sich nicht im direkten Greifraum befinden.

#### **3.1.3 Ikonische Gesten**

Durch die Einbindung von *Imagistic Description Trees* (IDTs (Sowa & Wachsmuth, 2005)), welche eine Baumstruktur mit zusätzlichen Formbeschreibungen der Einzelteile enthalten, steht in der virtuellen Werkstatt ein Repräsentationsformat mit hierarchischen Formbeschreibungen zur Verfügung. IDTs werden bei der Generierung neuer Teile, als auch bei der Äußerung einer ikonischen Geste des Benutzers aus den erkannten Formaspekten erstellt. Durch Hinweise auf die semantischen Aspekte einzelner Parameter in der VPML-Beschreibung können für die IDTs relevante Informationen aus den Bauteilbeschreibungen übernommen werden. Die Formaspekte beziehen sich in den Beispielen aus der Virtuellen Werkstatt auf Längenabschnitte und Krümmungswinkel der Bauteile, welche entsprechend der Bauteilhierarchie mehrstufig ausgewertet werden können. Mit Hilfe einer Metrik, welche Ähnlichkeitswerte für Paare von IDTs berechnet, können unterschiedliche IDTs auf Ähnlichkeit, z.B. für die Referenzauflösung untersucht werden.

Die Generierung von neuen Objekten mit Hilfe von sprachbegleiteter ikonischer Gestik wurde in den Demonstrator integriert („Gib mir so [+ ikonische Geste] einen Winkel“). Hierfür wurden entsprechende Erkenner für ikonische Gestik mit Hilfe von Detektor-

netzen implementiert, welche eine Formbeschreibung, die durch einen IDT repräsentiert wird, für z.B. ein gebogenes Rohr aus einer Geste detektieren (siehe Abbildung 6).



*Abbildung 6: Phasen einer ikonischen Geste zur Beschreibung eines gebogenen Rohres.*

Wie in der Abbildung zu sehen ist, werden zwei lineare Bewegungssegmente erkannt und verarbeitet. Zwischen diesen Segmenten wird ein Winkel errechnet, der die Krümmung des Rohres bestimmt. Die Länge der beiden Segmente legt die Länge des Rohres fest. Diese Informationen werden dem virtuellen Bauteil – in Form eines IDTs – als semantische Information zugefügt, damit es später durch eine ikonische Geste referenziert werden kann. Bei der Referenzierung wird ein neuer IDT basierend auf der Geste erzeugt, welches mit dem des Bauteiles verglichen wird. Existieren mehrere solcher Bauteile, wird dasjenige ausgewählt, welches bei dem Vergleich der IDTs den besten Wert erzielt hat.

#### **4 Prozessinteraktionsschemata, Komponentenintegration**

Das entwickelte SCIVE (Simulation Core for Intelligent Virtual Environments) Framework (Latoschik, Fröhlich, & Wendler, 2006) bietet die Möglichkeit verschiedene Simulationsmodule, die unabhängig voneinander – innerhalb ihrer Simulationsschleifen – laufen, miteinander interagieren und ihre Simulationsergebnisse untereinander austauschen zu lassen, sowie eventuell auftretende Konflikte zwischen diesen Modulen aufzulösen. In der Virtuellen Werkstatt kommen eine Grafikkomponente und eine physikalische Simulationsengine zum Einsatz, wodurch es möglich ist, eine physikalisch animierte Szene darzustellen. Die wissensbasierte Simulation in SCIVE wird durch ein semantisches Modul in Form eines funktional erweiterbaren Semantischen Netzes reali-

siert. Auf diesem Netz werden diverse Aspekte einer laufenden Simulation nach dem Prinzip der *semantic reflection* abgebildet. Zu diesen Aspekten gehören beispielsweise die Datenhaltung sowie funktionale Definitionen. Die integrative Repräsentationssprache für das semantische Netz und somit auch für Zusammenführung der einzelnen Komponenten liegt in Form der auf XML basierenden Sprache SNIL (Semantic Net Interchange Language) vor. Ähnliche Ansätze sind unter anderem bei (Lugrin & Cavazza, 2007) und (Kalogerakis, Christodoulakis, & Moumoutzis, 2006) zu finden.

Die Abbildung funktionaler Aspekte auf das vorhandene semantische Netz bezieht auch weitere Methoden der Künstlichen Intelligenz mit ein. Unter anderem wird das *Actor Model* (Hewitt, Bishop, & Steiger, 1973) verwendet, um einzelne Instanzen abzubilden und die Kommunikation unter diesen zu fördern. Das *Actor Model* kann bei dieser Abbildung auf zwei verschiedenen Ebenen betrachtet werden. Zum einen können die einzelnen an der Simulation beteiligten Module, wie etwa die physikalische Simulation oder die Grafik, als einzelne Actors modelliert werden. Diese berechnen ihre Daten in einer eigenen Simulationsschleife und tauschen sich im Anschluss daran über definierte Nachrichten aus. Ein Actor hat prinzipiell die Möglichkeit Nachrichten von anderen zu empfangen, selber Nachrichten zu verschicken und auf empfangene Nachrichten in adäquater Art und Weise zu reagieren. Eine weitere Fähigkeit besteht darin, zur Laufzeit einer Anwendung neue Actors zu erzeugen, was es ermöglicht, auch während der Simulation neue Module, zum Beispiel ein Audiomodul, zu laden und dieses in die Simulation mit einzubeziehen. Zum Anderen können aber auch einzelne Entitäten der Datenhaltung als Actors, mit den oben genannten Fähigkeiten, dargestellt werden, was die Kommunikation zwischen den Daten im semantischen Netz vereinfacht.

Temporale Aspekte, wie zum Beispiel eine bestimmte Reihenfolge von Funktionen, können ebenfalls auf dem Netz abgebildet werden. Hierzu bieten sich die 13 von Allen definierten zeitlichen Intervall-Funktionen (Allen, 1983) an, wobei die Funktionsknoten innerhalb des Netzes durch Kanten verbunden werden, welche die Intervall-Funktionen abbilden. So ist es möglich über ein Traversersystem verschiedene Funktionen in dedizierter Reihenfolge auf dem Netz abzuarbeiten. Erreicht ein Funktionstraverser einen Funktionsknoten, wird ein bestimmter Funktionsaufruf aus dem Funktionsnamen und etwaigen Parametern zusammengesetzt und in der ebenfalls über eine Relation angegebenen Programmbibliothek aufgerufen. Abbildung 7 zeigt ein exemplarisches Netz mit mehreren Funktionsknoten und zeitlichen Relationen zwischen diesen. Die momentan an das SCIVE-Framework angeschlossenen Komponenten umfassen die Grafikengine *OpenSG*, die Physikengine *ODE* (bzw. den *OPAL Wrapper*), eine Anbindung der *FMod* Audio Bibliothek sowie eine Skriptanbindung zur Laufzeit über Python. Die Skriptanbindung wurde über *SWIG* generiert, sodass auch andere Skriptsprachen möglich sind.

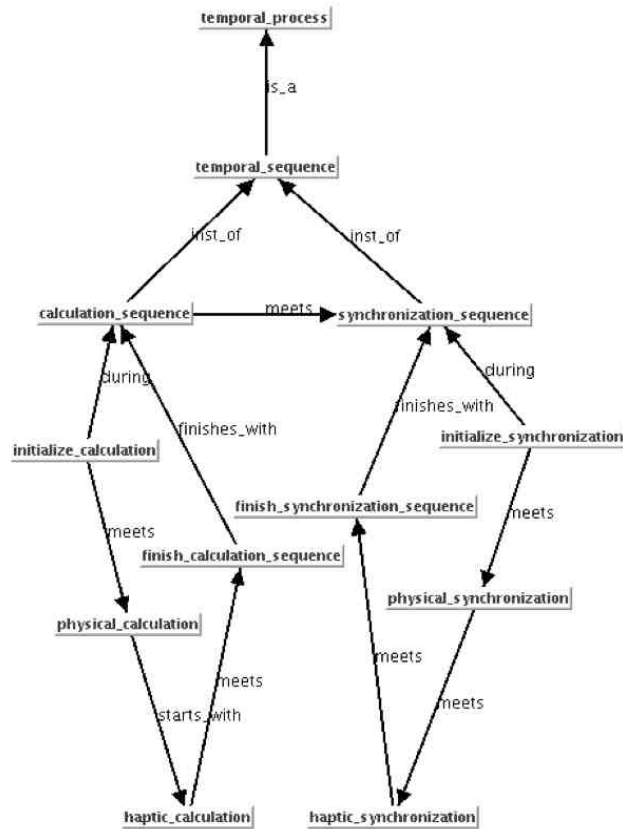


Abbildung 7: Eine exemplarische Funktionsfolge. Allens Primitive dienen als Kanten zwischen den einzelnen Funktionsaufrufen.

## 5 Fazit

Zum Abschluss des Projektes können in der virtuellen Werkstatt komplette Konstruktionen frei gebaut und exploriert werden. Die multimodale Interaktion erlaubt integrierte sprachliche-gestische Eingaben für den Greifraum, Fernraum und insbesondere auch mit ikonischen Gesten. Dabei sind sowohl die Repräsentationen der Bauteile, ihrer Verbindungen und parametrischen Veränderungen, als auch weitestgehend die Anwendungslogiken über deklarative (XML-)Formate beschrieben. Die parametrischen Veränderungen umfassen heterogene Skalierungen und Transformationen der Bestandteile. Falls nötig kann die Morphologie der Bauteile, z.B. für die Generierung neuer Verbindungsgeometrien, während der Konstruktion verändert werden. Die virtuelle Konstruktion erlaubt den vollständigen Aufbau komplexer Zielaggregate, wobei alle Anpassungen über die Parameter im kompletten Aufbau weiterhin möglich sind. Durch die Arbeiten im Bereich der Softwareentwicklung für VR-Systeme wurde die Virtuelle Werkstatt modularisiert und um andere Module erweitert, welche neue Funktionalitäten einbringen, wie zum Beispiel eine physikalische Simulation.

*Hinweis: Das Projekt „Virtuelle Werkstatt“ wurde von der Deutschen Forschungsgemeinschaft gefördert.*

## Literatur

- [All83] Allen, J. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11), 832-843.
- [BJ04] Biermann, P., & Jung, B. (2004). Variant Design in Immersive Virtual Reality: A markup language for scalable CSG parts. *Articulated Motion and Deformable Objects, Proceedings AMDO-2004* (S. 123-133). Berlin: Springer (LNCS 3179).
- [BFLW07] Biermann, P., Fröhlich, C., Latoschik, M. E., & Wachsmuth, I. (2007). Semantic Information and Local Constraints for Parametric Parts in Interactive Virtual Construction. *8th Symposium on Smart Graphics*, (S. 124-134). Kyoto.
- [CP00] Cavazza, M., & Palmer, I. (2000). High-level interpretation in dynamic virtual environments. *Applied Artificial Intelligence* 14(1), 125-144.
- [FBLW09] Fröhlich, C., Biermann, P., Latoschik, M. E., & Wachsmuth, I. (2009). Processing of Iconic Gestures in a Multimodal Virtual Construction Environment. (M. Dias, S. Gibet, M. Wanderley, & R. Bastos, Hrsg.) *Gesture-Based Human Computer Interaction and Simulation*, 187-192.
- [HBS73] Hewitt, C., Bishop, P., & Steiger, R. (1973). A universal modular actor formalism for an artificial intelligence. *IJCAI*, 235-245.
- [KCM06] Kalogerakis, E., Christodoulakis, S., & Moutoutzis, N. (2006). Coupling ontologies with graphics content for knowledge driven visualization. *Proceedings of IEEE VR2006*, (S. 43-50).
- [Lat02] Latoschik, M. E. (2002). Designing transition networks for multimodal VR-Interactions using a markup language. *Proceedings of the fourth IEEE International Conference on Multimodal Interfaces, ICMI 2002*, (S. 411-416). Pittsburgh, USA.
- [LBW05] Latoschik, M. E., Biermann, P., & Wachsmuth, I. (2005). Knowledge in the Loop: Semantics Representations for Multimodal Simulative Environments. *Proceedings of the 5th Symposium on Smart Graphics*, (S. 25-39).
- [LFW06] Latoschik, M. E., Fröhlich, C., & Wendler, A. (2006). Scene Synchronization in close-coupled World Representations using SCIVE. *International Journal of Virtual Reality* 5(3), 47-52.
- [LS01] Looock, A., & Schömer, E. (2001). A virtual environment for interactive assembly simulation. *Proceedings 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI'01), vol. 3 (Virtual Engineering and Emergent Computing)*, (S. 325-332).
- [LA00] Luck, M., & Aylett, R. (2000). Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence* 14(1), 3-32.
- [LC07] Lugin, J.-L., & Cavazza, M. (2007). Making sense of virtual environments: Action representation. *Proceedings of the 2007 International Conference on Intelligent User Interfaces, IUI07*, (S. 225-234). Honolulu, Hawaii, USA.
- [PS03] Peters, S., & Shrobe, H. (2003). Using semantic networks for knowledge representation in an intelligent virtual environment. *PerCom'03: 1st Annual IEEE International Conference on Pervasive Computing and Communications*. Ft. Worth, TX, USA: IEEE.
- [SA02] Soto, M., & Allongue, S. (2002). Modeling methods for reusable and interoperable virtual entities in multimedia virtual worlds. *Multimedia Tools Applications* 16(1-2), 161-177.

- [SW05] Sowa, T., & Wachsmuth, I. (2005). A model for the representation and processing of shape in coverbal iconic gestures. *Proceedings of KogWis05* (S. 183-188). Basel: Schwabe Verlag.
- [ZR01] Zachmann, G., & Rettig, A. (2001). Natural and robust interaction in virtual assembly simulation. *Proceedings Eighth ISPE International Conference on Concurrent Engineering: Research and Application (ISPE/CE2001)*.

## **Autoren**

### **Dipl. Inform. Christian Fröhlich**

absolvierte das Studium der Naturwissenschaftlichen Informatik an der Universität Bielefeld, welches er im April 2006 mit dem Erreichen des Diplomgrades abschloss. Seitdem arbeitete er in verschiedenen wissenschaftlichen Forschungsprojekten an der Universität Bielefeld. Eines davon war die Virtuelle Werkstatt, in der er zwei Jahre als studentische Hilfskraft und zwei Jahre als wissenschaftlicher Angestellter tätig war. Andere Projekte umfassen eine Mitarbeit im Sonderforschungsbereich 673 „Alignment in Communication“ und in dem von der Europäischen Union geförderten Projekt PASION. Im Rahmen seiner laufenden Promotion arbeitet er an der integrierten Modellierung intelligenter virtueller Umgebungen auf der Basis einer verteilten Daten- und Prozessarchitektur.

### **Prof Dr. Ipke Wachsmuth**

studierte Mathematik und Informatik an der Technischen Universität Hannover, wo er 1980 mit einer Dissertation über Zellularautomaten promovierte. Nach Lehr- und Forschungstätigkeiten an der Universität Osnabrück, der Northern Illinois University und bei IBM Deutschland habilitierte er sich 1989 an der Universität Osnabrück über intelligente Wissensorganisation; im gleichen Jahr wurde er auf die Professur für Wissensbasierte Systeme (Künstliche Intelligenz) an die Universität Bielefeld berufen. Er war dort Gründungsmitglied der Technischen Fakultät (1990), Mitinitiator zweier Sonderforschungsbereiche und von Oktober 2002 bis März 2009 geschäftsführender Direktor des Zentrums für interdisziplinäre Forschung (ZiF).

### **Prof. Dr. Marc Erich Latoschik**

studierte Mathematik und Informatik an der Universität Paderborn, dem New York Institute of Technology und der Universität Bielefeld. Nach einigen Jahren in der IT Branche schlug er die wissenschaftliche Karriere ein und übernahm die Leitung des Labors für Künstliche Intelligenz und Virtuelle Realität bei Professor Wachsmuth in Bielefeld. In 2001 promovierte er über "Multimodale Interaktion in der Virtuellen Realität". Nachdem er im Jahr 2008 als Professor für Multimedia an die FHTW in Berlin wechselte, folgte er 2009 einem Ruf auf eine Professur für Multimedia und Visualisierung an der Universität Bayreuth.