

# SEMANTISCHE MODELLIERUNG EINES VIRTUAL REALITY SYSTEMS MIT GESTENSTEUERUNG

Stephan Rogge<sup>1</sup>, Ronald Römer<sup>2</sup> und Christian Hentschel<sup>1</sup>

<sup>1</sup> Lehrstuhl  
Medientechnik

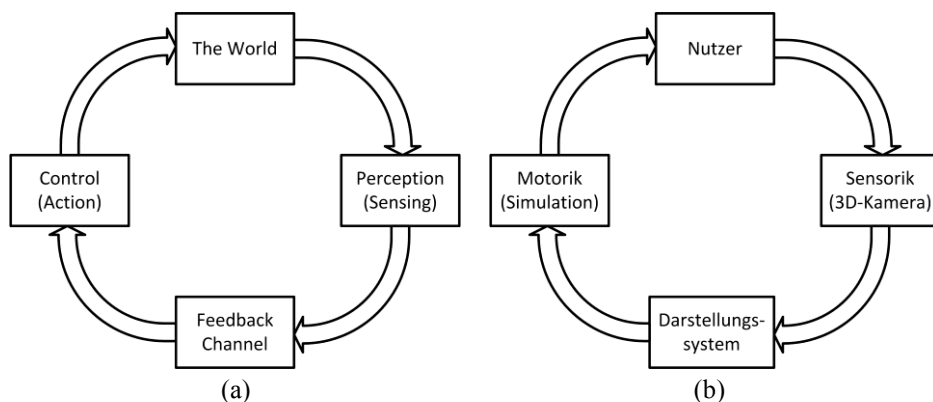
<sup>2</sup> Lehrstuhl  
Kommunikationstechnik

Brandenburgische Technische  
Universität Cottbus  
stephan.rogge@tu-cottbus.de

**Abstract:** Im Zuge eines industriellen Forschungsprojektes wird derzeit an einem interaktiven Prototyp zur aerodynamischen Optimierung von Triebwerksschaufeln für ein Virtual Reality System (VRS) gearbeitet. Hauptaufgabe dieser Anwendung ist die interaktive Modifikation der Schaufelgeometrie und die Bereitstellung einer 3D-Visualisierung der computergestützten Strömungssimulation. Dabei soll der Benutzer Geometriemodifikationen an Turbinenschaufeln mit Hilfe von Gesten direkt im VRS vornehmen können. Nach einer solchen Modifikation erfolgt die Strömungssimulation automatisch und die 3D-Visualisierung wird aktualisiert. Dieser Prototyp dient als Grundlage für die Entwicklung einer semantischen Gestensteuerung. Es wird der Prototyp eines VRS mit seinen wichtigsten Funktionen vorgestellt und auf das Konzept der zugrunde liegenden semantischen Systemkomponente eingegangen.

## 1 Einführung

Die Interaktion in einer virtuellen Umgebung benötigt alternative natürliche Dialogsysteme. Eine mögliche Form der Informationsübertragung zwischen Mensch und Maschine kann visuell erfolgen. Das System überträgt die Informationen zum Menschen über einen Bildschirm und nimmt sie vom Menschen über ein Kamera-System auf. Kommen Mustererkennungs-algorithmen wie z. B. die Frameworks *Kinect SDK* [1] oder *OpenNI* [2] zum Einsatz, können Nutzergesten als Eingabe interpretiert und mit entsprechenden Aktionen versehen werden.



**Abbildung 1** - Schema eines humanen kognitiven Systems (a) nach Haykin [3]. Adaptiert auf eine virtuelle Umgebung mit 3D-Kamera-System zur Gestenerfassung (b).

Dieses Grundprinzip wurde bei der Entwicklung eines industriellen Prototyps zur interaktiven aerodynamischen Turbinenschaufeloptimierung [4] angewendet. Die Anwendung trägt den Namen *IBO – Interactive Blade Optimisation* – und nutzt ein kamerabasiertes

Gestensteuerungssystem, was Schaufelgeometrie-Modifikationen ermöglicht. Die Auswirkungen auf die Simulation der Aerodynamik werden unmittelbar im Darstellungssystem angezeigt, so dass der Nutzer deren Begutachtung direkt im Virtual Reality System (VRS) vornehmen kann, ohne die Interpretation der Simulationsergebnisse der Maschine zu überlassen.

Legt man Haykins' Definition eines humanen kognitiven Systems [5], veranschaulicht in Abbildung 1 (a) dem IBO-System zu Grunde, so lassen sich die einzelnen Komponenten in Abbildung 1 (b) wiederfinden. Im Wesentlichen ist der derzeitige Prototyp mit einer naiven Gestensteuerung realisiert. Da das System davon ausgeht, dass jede Geste zuverlässig und korrekt erkannt wird, können unbeabsichtigte Ausführungen von Aktionen durch falsch erkannte oder mehrdeutige Gesten die Folge sein.

Die Verwendung von Gesten als Eingaben hat allerdings zur Folge, dass das System mit unsicheren Informationen umgehen muss. Dieser Umstand lässt sich berücksichtigen, indem Gesten nicht als Befehle aufgefasst werden, sondern als Beobachtung [6] zu verstehen sind, von denen das System die Intention des Nutzers ableiten kann.

Die Herangehensweise, Gesten nicht direkt als Befehl zu interpretieren, wird auch durch eine semantische Dialogmodellierung [7] aufgegriffen. Zeichen (Gesten) werden nicht nur als Beobachtungen aufgegriffen, sondern einer Bedeutung (Semantik) zugeordnet. Dies setzt eine Beschreibung der Modellwelt voraus, die die Entitäten und Funktionen zunächst identifiziert.

Der vorliegende Beitrag befasst sich mit den ersten Schritten einer semantischen Modellierung der Gestensteuerung. Der Fokus liegt zunächst auf der Erstellung eines statischen Modells. In Abschnitt 2 wird das Gesamtsystem und der bestehende Prototyp IBO vorgestellt. Das beschriebene Szenario wird dann in Abschnitt 3 in ein statisches Modell überführt. Des Weiteren werden die weiteren Schritte für die dynamische Modellierung in Abschnitt 4 angegeben. Abschnitt 5 schließt den Beitrag mit einer Zusammenfassung und einem Ausblick ab.

## 2 Szenario

Im Rahmen eines industriellen Forschungsprojektes mit Rolls-Royce Deutschland wurde eine virtuelle Umgebung konzipiert und errichtet. In ihr kommt ein Prototyp zur interaktiven aerodynamischen Turbinenschaufeloptimierung [4] zum Einsatz, welcher im den folgenden zwei Abschnitten vorgestellt wird.

### 2.1 Systemvorstellung

Die virtuelle Umgebung trägt den Namen „ivi“ und steht für *Intuitive Virtual Interaction*. Sie besteht aus zwei Rückprojektionswänden, die mit einem Innenwinkel von  $112^\circ$  errichtet wurden. Die Anordnung erhöht die Gesichtsfeldabdeckung des Betrachters mit virtuell genierten Bildern, was wiederum das Maß an Immersion erhöht. Außerdem werden durch den größeren Winkel der zwei Projektorenpaare in den Randbereichen visuelle Artefakte verringert.

In Abbildung 2 ist eine schematische Darstellung des Systems dargestellt. Für jede Projektionswand (2,6 m breit und 1,6 m hoch)

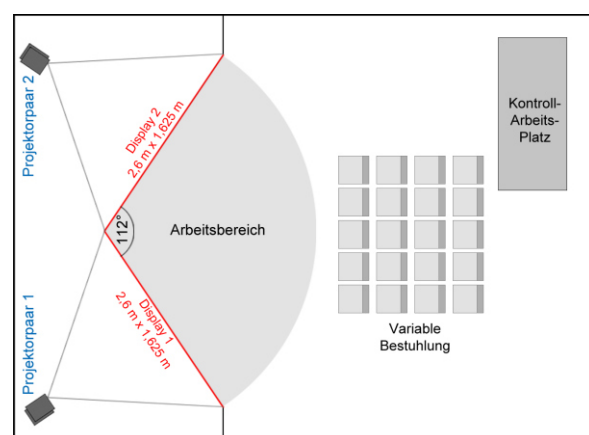


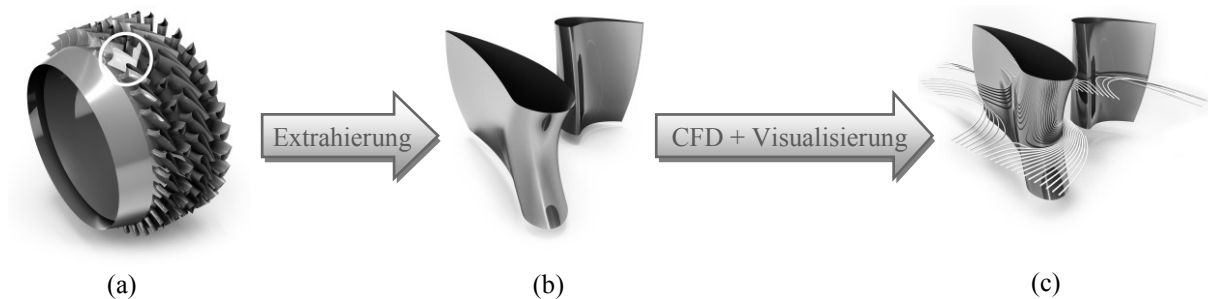
Abbildung 2 Schema des VR-Systems

werden zur passiven stereoskopischen Anzeige zwei Projektoren mit einer Auflösung von 1920 x 1200 Bildpunkten verwendet.

Gemäß des in Abbildung 1 (b) dargestellten Schemas übernimmt das Darstellungssystem die Funktion des Feedback Channel. Die Sensorik besteht aus einer 3D Kamera zur Erfassung von Handgesten sowie einem Infrarot-Tracking-System zur Lokalisierung des Betrachters. Dies ermöglicht die Erzeugung perspektivisch korrekter Darstellungen von dreidimensionalen Bildinhalten. Des Weiteren lassen sich somit virtuelle Objekte „vor“ die Projektionswände platzieren, wodurch diese für den Betrachter virtuell im Raum schweben und „greifbar“ werden.

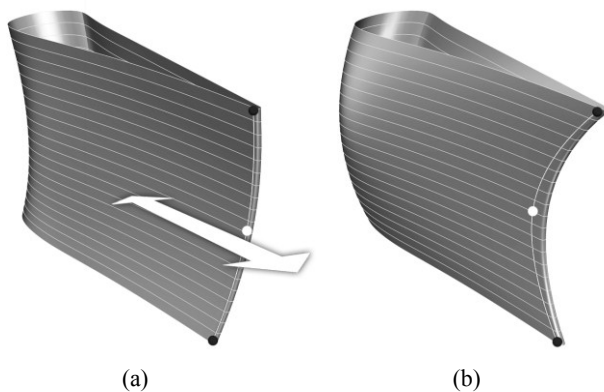
Eine für die Modellierung wichtige Information ist die Transformation vom Koordinatensystem der Handpositionserkennung in das der virtuellen Szene. Somit kann die Handposition des Nutzers in die virtuelle Szene überführt werden.

## 2.2 Interaktive aerodynamische Turbinenschaufel-Optimierung



**Abbildung 3** - Beispiel Turbinenstufe nach einer Extrahierung: Eine zweistufige Turbine (a) wird auf eine einzelne Stufe (b) mit einem Stator (vordere Schaufel) und Rotor (hintere Schaufel) reduziert. Im Anschluss kann mit einem vorgegebenen Parametersatz eine *Computation Fluid Dynamic* (CFD) Berechnung durchgeführt und mit Stromlinien (c) visualisiert werden.

In Abbildung 3 (a) ist beispielhaft eine Hochdruckturbinenstufe dargestellt, wie sie in einem Strahltriebwerk vorzufinden ist. Turbinen können aus mehreren Stufen bestehen. Eine Stufe wiederum setzt sich aus einer Stator- und Rotorschaukel Abbildung 3 (b) zusammen. Da Schaufeln desselben Typs und derselben Stufe identisch sind, wird bei einer aerodynamischen Simulation (*Computation Fluid Dynamic*, CFD) pro Stufe nur ein Stator und Rotor verwendet (*Extrahierung*), was redundante Berechnungen erspart.



**Abbildung 4** - Grafische Darstellung einer Stator-Schaukel vor (a) und nach (b) einer *lean*-Modifikation. Die weiße Kugel repräsentiert den *lean*-Modifikator.

Die Form einer Turbinenschaufel lässt sich durch ein sogenanntes *stacking*-Verfahren definieren. In Abbildung 4 ist eine grafische Repräsentation eines Stators dargestellt. Es werden mehrere Profilsektionen (helle Linien) über verschiedene Radialhöhen verteilt und mit einer Oberfläche verbunden. Üblicherweise werden während des aerodynamischen Schaufelentwurfs einzelne Sektionen um einen Schwerpunkt rotiert (*skew*) oder in Umfangsrichtung verschoben (*lean*), damit gezielte Strömungseffekte erzielt werden.

Die Modifikation der Geometrie lässt sich mit Hilfe von drei Kontrollpunkten, im Folgenden *Geometriemodifikatoren* genannt, durchführen. In Abbildung 4 werden sie als Kugeln dargestellt und verlaufen auf einer *Spline*-Kurve entlang der Schaufel-Hinterkante. Nach jeder Geometriemodifikation wird die CFD neu ausgeführt und das Ergebnis unmittelbar visualisiert. Der Nutzer kann nun auf Grundlage der veränderten Stromlinien seine nächsten Design-Entscheidungen treffen und abermals Geometriemodifikationen vornehmen.

Dieser Prototyp mit dem Namen IBO ist als Erweiterung für die Visualisierungsanwendung *ParaView* [8] entwickelt worden. Grundsätzlich wurde bei der Realisierung darauf geachtet, dass alle Parameter und Funktionen durch aussagekräftige geometrische Objekte repräsentiert werden. So kommen im Fall der lean- und skew-Modifikatoren Kugeln zum Einsatz.

Neben der Möglichkeit die Schaufelgeometrie zu verändern, kann der Nutzer auch die Visualisierung begutachten. Dies umfasst einerseits die Steuerung einer virtuellen Kamera, was im Wesentlichen einer Rotation um einen festen Rotationspunkt entspricht, andererseits kann die Erzeugung der Stromlinien beeinflusst werden.

Die Visualisierung des Strömungsfeldes geschieht mit Hilfe der ParaView-Erweiterung *StreamTracer*. Sie integriert über das gesamte Strömungsfeld und stellt den zurückgelegten Weg grafisch dar. Der Startpunkt der Integration lässt sich mit entsprechenden Modifikatoren kontrollieren.

### **3 Statische Modellierung**

Der in Abschnitt 2 vorgestellte Prototyp wurde in ein statisches Modell überführt. Es dient dazu eine Vereinbarung zwischen Nutzer und System zu treffen, welcher Ausschnitt der Welt mit welchen Begriffen beschrieben wird [7]. Dazu wird zunächst ein *Entity-Relationship* (ER) Modell nach Chen [9] erstellt. Anschließend werden die identifizierten Objekte (Entitäten) mit Aktionen in einem integrierten Struktur- und Funktionsmodell zusammengefasst.

#### **3.1 Entity-Relationship-Modell**

Das in Abbildung 5 gezeigte ER-Modell stellt die Zusammenhänge des Prototyps dar. Eine Turbine besteht aus ein oder mehreren Stufen. Eine Stufe hat genau zwei Turbinenschaufeln, die entweder einem Stator oder einem Rotor entsprechen. Jede Schaufelgeometrie wird durch das Darstellungssystem gezeichnet und kann durch mehrere Geometriemodifikatoren modifiziert werden.

Die Geometriemodifikatoren dienen zur Kontrolle der lean- oder skew-Parameter und werden daher als eigenständige Entitätstypen gehalten. Sobald es zur Veränderung der Schaufelgeometrie kommt, wird die Simulation aktualisiert.

Ein weiterer Bestandteil der Turbine ist das Strömungsfeld. Es umgibt die Schaufelgeometrie und wird durch ein oder mehrere StreamTracer visualisiert. Die Visualisierung wird ebenfalls vom Darstellungssystem gezeichnet und lässt sich über entsprechende Modifikatoren steuern.

Schaufelgeometrie und Strömungsfeldvisualisierung werden in einer virtuellen Szene organisiert und vom Darstellungssystem angezeigt. Teil der virtuellen Szene ist eine virtuelle Kamera, die über einen Modifikator gesteuert werden kann. Somit kann der Nutzer durch die Visualisierungsszene navigieren und die für ihn interessanten Bereiche näher in Augenschein nehmen.

Das Darstellungssystem übernimmt die Aufgabe der Systemausgaben. Bei Nutzerinteraktionen werden die Systemantworten über diesen „Kanal“ gegeben. Die Sensorik übernimmt hier die Aufgabe der Handpositionserfassung.

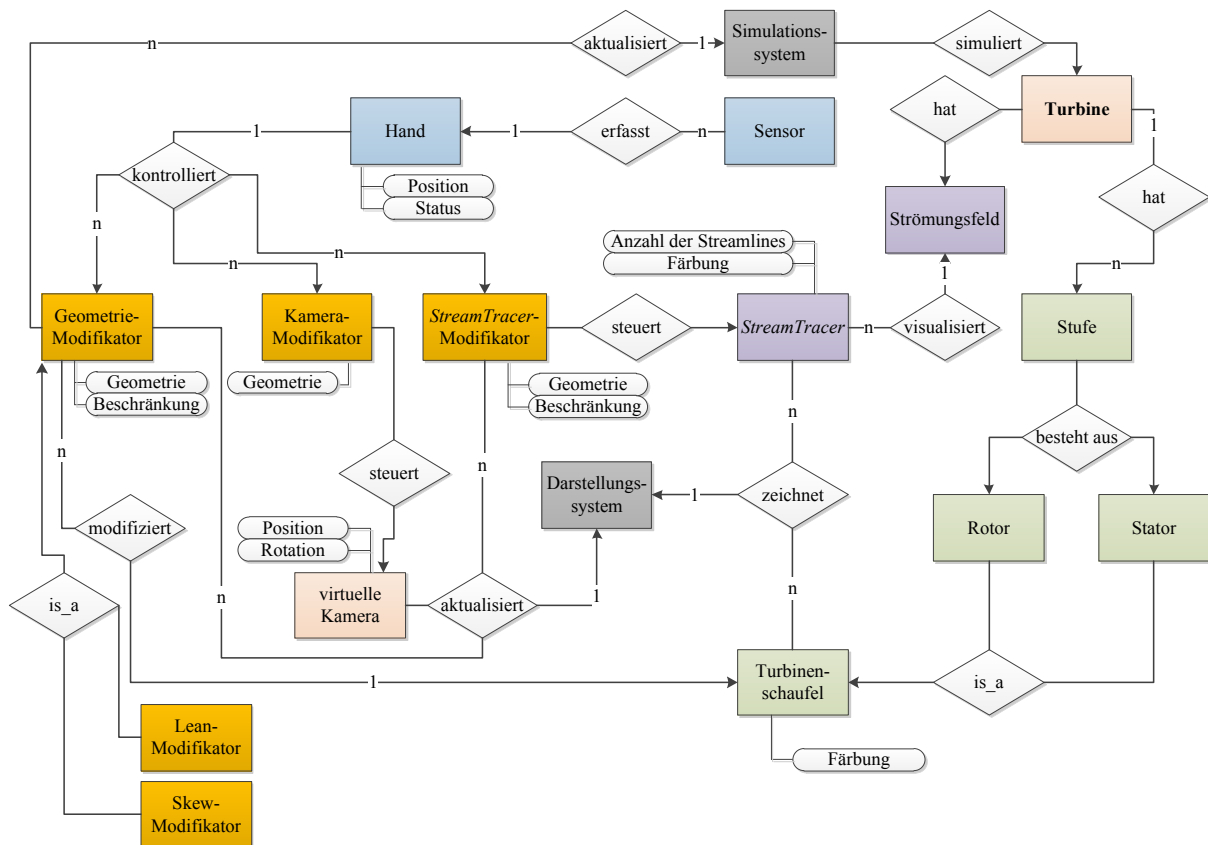


Abbildung 5 – ERM nach Chen [9] zur interaktiven aerodynamischen Turbinenschaufel-Optimierung.

### 3.2 Integriertes Struktur- und Funktionsmodell

Analog zu Karnagel et al. [10] wird das ERM aus Abbildung 5 in ein integriertes Struktur- und Funktionsmodell überführt. Da das ERM nur Entitäten und deren Beziehungen untereinander beschreibt, jedoch die funktionalen Aspekte nicht berücksichtigt, müssen die Verbindungen zwischen den Objekten und den Funktionen hergestellt werden. Als Grundlage der Modellierung dient die Prädikatenlogik: „WAS wird WIE WOMIT gemacht?“.

Funktionen werden durch Aktionen (WAS) und Parametern (WIE) in ein Funktionsmodell gefasst. Sobald Aktionen mit den Parametern verknüpft sind, können Aktions-IDs ( $A_x$ ) gebildet werden. Abbildung 6 stellt das Funktionsmodell dar.

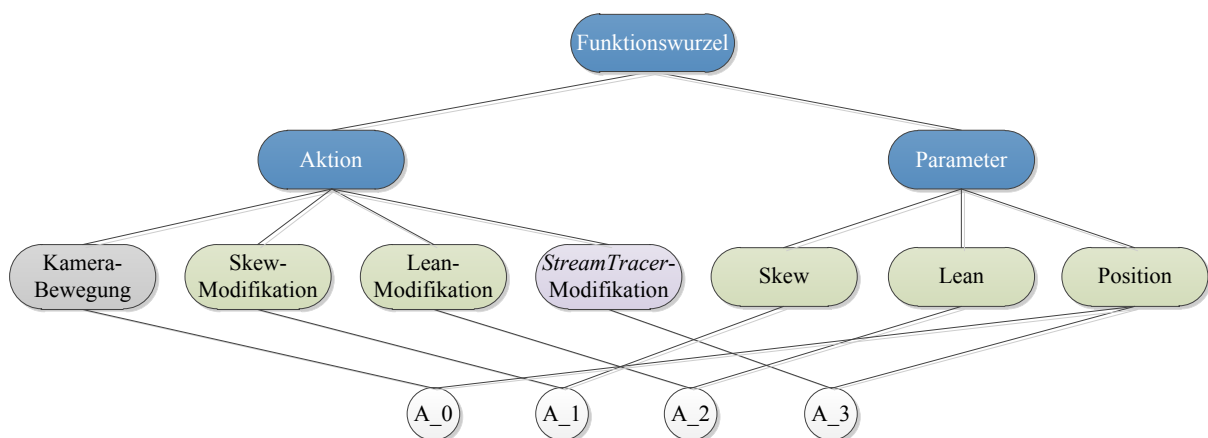
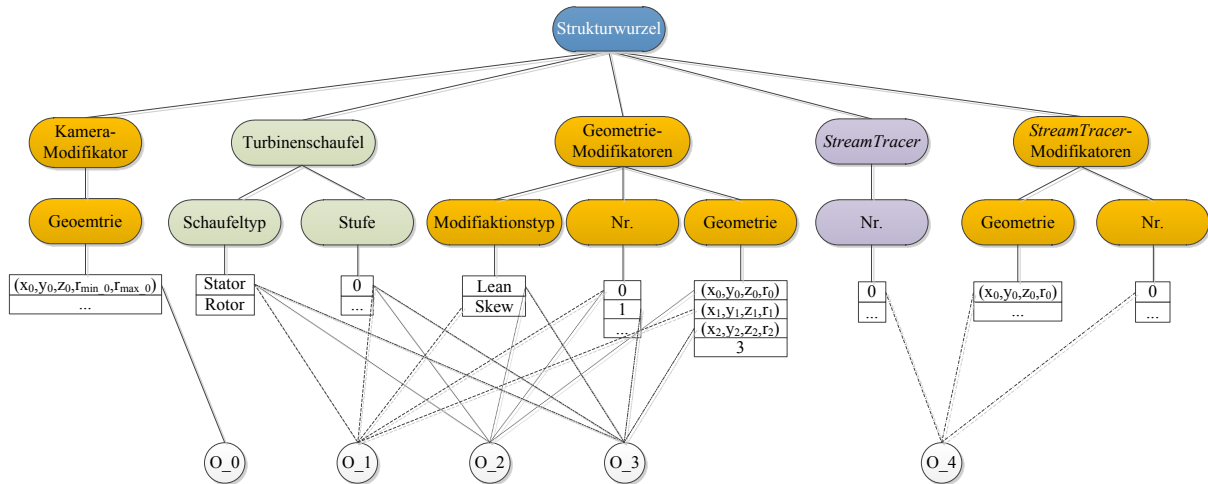


Abbildung 6 - Funktionsmodell der definierten Funktionen mit den erwarteten Parametern.

Die Aktions-ID  $A_0$  steht für die Aktion „Kamera Bewegung“ und benötigt den Parameter „Position“, was im Wesentlichen der detektierten Handposition entspricht. Der gleiche Parameter wird auch von der Aktion „StreamTracer-Modifikation“ ( $A_3$ ) erwartet.

Im Strukturmodell werden nur Objekte spezifiziert, auf denen eine Aktion ausgeführt werden soll. Somit werden Objekt-IDs ( $O_x$ ) nur für die Modifikatoren der Kamera, der Geometrie und des StreamTracers erzeugt. In Abbildung 7 ist das Strukturmodell dargestellt.



**Abbildung 7** - Beispiel eines Strukturmodells: Jeder Attributs-Typ kann eine Menge von Attributwerten (eckige Kästchen) besitzen. Durch einzelne Attributwerte lassen sich Objekte identifizieren.

Wird zum Beispiel ein Geometriemodifikator definiert, müssen die Attribute *Schaufeltyp*, *Stufe*, *Modifikationsstyp*, *Nr.*, und *Geometrie* mit Werten belegt sein. Die ersten beiden Attribute dienen zum Identifizieren der Schaufelgeometrie, auf der die Modifikation angewendet werden soll. Der *Modifikationsstyp* legt die Modifikationsart (lean oder skew) fest. Da mehrere Modifikatoren gleichen Typs eine Schaufel modifizieren können, dient das Attribut *Nr.* zur weiteren Identifizierung. Das Attribut „Geometrie“ ist Bestandteil jedes Modifikators. Es beschreibt die Position und das Volumen der grafischen Repräsentation und wird benötigt um im späteren Verlauf die einzelnen Objekte selektieren zu können.

Objekt-ID  $O_1$  entspricht demnach einem skew-Modifikator für einen Stator der ersten Turbinenstufe (Stufe 0). Da er als Kugel dargestellt wird, kann sein Geometrie mit einer dreidimensionalen Koordinate und einem Radius angegeben werden.

Objekt_ID	Aktions_ID
O_0	A_0
O_1	A_1
O_2	A_2
O_3	A_2
O_4	A_3

**Tabelle 1** - Kombinierte Assoziations- und Verknüpfungstabelle

Nach der Bildung von Aktions- und Objekt-IDs werden diese in einer kombinierten Assoziations- und Verknüpfungstabelle eingetragen (siehe Tabelle 1). Eine Assoziation besteht immer dann, wenn einer Objekt-ID eine Aktions-ID zugeordnet wurde. Grundsätzlich kann jedes Objekt mit jeder Aktion verknüpft werden. Da jedoch auf einem Objekt nicht alle Aktionen ausgeführt werden können, beschreibt die Verknüpfungstabelle, welche

Assoziationen erlaubt sind. Somit fungiert die Verknüpfungstabelle als eine Art Filter, das auf die Assoziationstabelle angewendet wird.

Jeder Objekt\_ID wird in diesem Fall nur eine Aktions\_ID zugeordnet. Somit ist eine Aktion ausgewählt, sobald ein Objekt selektiert werden konnte. Im Gegensatz zu dieser Vorgehensweise werden in Karnagel et al. [10] zunächst die Aktion identifiziert und dann die dazugehörigen Objekte ermittelt.

Die Assoziations- und Verknüpfungstabelle komplettiert die statische Modellierung. In welcher zeitlichen Reihenfolge die Auswahl eines Objektes erfolgt, ist Gegenstand der dynamischen Modellierung.

#### 4 Dynamische Modellierung

Nachdem alle Objekte und Funktionen im statischen Modell identifiziert worden sind, wird im dynamischen Modell festgehalten, wann welche Aktion auszuführen ist. Dies umfasst im Wesentlichen den Prozess der Objekt-Selektion, da somit die Aktion auch identifiziert ist.

Nach Haykin [5] verfügen kognitive Systeme über Modelle ihrer Umwelt. In diesem Fall wird ein Hand-Zustandsautomat definiert, da der Nutzer durch seine Hand im System repräsentiert wird. Wie im ERM aus Abbildung 5 dargestellt, wird die Hand durch zwei Attribute spezifiziert. Neben der *Position* (dreidimensionale Koordinate) gibt der *Status* darüber Auskunft, ob die Hand *geöffnet* oder *geschlossen* ist. Für die Überwachung des Hand-Status ergibt sich der in Abbildung 8 dargestellte endliche Automat.

Da die Handposition in den Koordinatenraum der virtuellen Szene überführt wurde, lässt sich mit dem Geometrieattribut entscheiden, ob die Hand innerhalb eines bestimmten Objekt-Volumens ist. Befindet sich die Hand innerhalb eines virtuellen Objektes und ist geschlossen, gilt dieses Objekt als selektiert und die mit ihm assoziierte Aktion wird ausgeführt, bis der Nutzer seine Hand wieder öffnet. Dies entspricht der Segmentierung einer Geste. Wann ein Objekt selektiert ist und wann eine Aktion auszuführen ist, entscheidet ein weiterer Zustandsautomat, der das Systemverhalten modelliert. Dieser ist jedoch noch nicht im Konzept realisiert.

Derzeit wird basierend auf einem einfachen Algorithmus [4] eine „harte“ Entscheidung zwischen offener und geschlossener Hand getroffen. Dies führt häufig dazu, dass das System fälschlicherweise annimmt, dass die Hand geschlossen sei und ungewollt Aktionen ausführt. Des Weiteren variieren über verschiedene Nutzer hinweg die Eingaben zum Beispiel für das „Schließen“ und das „Öffnen“ der Hand. Zwar meinen alle Nutzer das gleiche, drücken dies aber mit unterschiedlichen Handgesten aus.

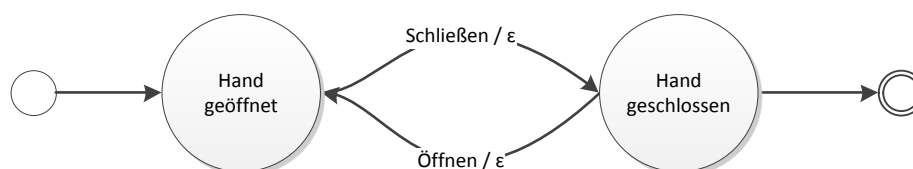


Abbildung 8 - Beispiel eines einfachen endlichen Automaten zur Überwachung des Hand-Status.

Um dem entgegenzuwirken, werden zukünftig für die Zustandsautomaten *Utterance Meaning Pairs* (UMP) [11] als Eingaben verwendet. Sie kommen der Forderung Youngs [6] nach, Gesten als Beobachtungen und nicht als Befehle aufzufassen. Dazu ist die Hand zunächst durch aussagekräftige Merkmalsvektoren zu beschreiben. Mehrere dieser Vektoren werden einer Äquivalenzklasse gleicher Bedeutung zugeordnet. Somit kann z. B. das Schließen der Hand durch unterschiedliche Hand-Ausprägungen semantisch modelliert werden. Welche

Ausprägungen einer Bedeutung zugeordnet werden, muss dem dynamischen Modell antrainiert werden. In Experimenten werden die UMP's bestimmt.

## 5 Zusammenfassung und Ausblick

Es wurden die ersten Schritte zu einer semantischen Dialogmodellierung für eine bestehende gestengesteuerte Virtual-Reality-Anwendung zur interaktiven aerodynamischen Turbinenschaufeloptimierung vorgestellt. Im Wesentlichen lag der Fokus auf der Erstellung des statischen Modells, um einen Ausschnitt der Welt zu modellieren und sicherzustellen, dass System und Nutzer über zueinander kompatible Begriffe und Zielvorstellungen verfügen.

Neben der Überführung des konzipierten ER-Modells in eine Merkmals-Werte-Relation ist ein dynamisches Modell zu erstellen. Die dynamische Modellierung ist konzeptionell noch nicht vollständig ausgearbeitet. Es zeichnet sich jedoch schon ab, dass für Nutzermodell und System zwei getrennte Zustandsautomaten zu verwalten sind. Die Eingaben dieser Zustandsautomaten werden auf UMP's abgebildet.

Ein weiterer wichtiger Aspekt ist die Ausarbeitung von geeigneten Versuchen zum Erstellen der UMP's und zur Modell-Verifikation, so dass die semantische Modellierung einen robusteren Umgang mit unsicheren Informationen gewährleistet.

## Danksagung

Diese Arbeit ist Teil des Projektes „VIT III: Virtuelles Triebwerk“ in Kooperation mit Rolls-Royce Deutschland. Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Ministeriums für Wirtschaft des Landes Brandenburg und der EU gefördert. Die Verantwortung für den Inhalt der Veröffentlichung liegt bei den Autoren.

Die Autoren danken Prof. Dieter Bestle und Philipp Amtsfeld vom Lehrstuhl Technische Mechanik und Fahrzeugdynamik der BTU Cottbus für die enge Kooperation bei der Einbindung der CFD-Analyse in das Gesamtsystem.

Des Weiteren bedanken sich die Autoren bei Prof. Marius Swoboda, Roland Parchem und Marcus Meyer von Rolls-Royce Deutschland für die technische und inhaltliche Unterstützung.

## Literatur

- [1] Kinect for Windows: <http://www.microsoft.com/en-us/kinectforwindows>
- [2] OpenNI: <http://www.openni.org/>
- [3] Haykin, S.: Foundation of cognitive dynamic systems. IEEE Lecture, Queens University, 29 January 2009, [http://soma.mcmaster.ca/papers/Slides\\_Haykin\\_Queens](http://soma.mcmaster.ca/papers/Slides_Haykin_Queens)
- [4] Rogge, S.; Amtsfeld, P.; Hentschel, C.; Bestle, D.; Meyer, M.: Using gestures to interactively modify turbine blades in a Virtual Environment. IEEE International Conference on Emerging Signal Processing Applications (ESPA), 2012, Las Vegas, pp. 155 -158
- [5] Haykin, S.: Cognitive Dynamic Systems: Perception-Action Cycle, Radar and Radio. Cambridge University Press, 2011
- [6] Young, S.: Cognitive User Interfaces. Signal Processing Magazine, IEEE, 2010, vol. 27, pp. 128 -140
- [7] Huber, M.; Kolbl, C.; Lorenz, R.; Römer, R.; Wirsching, G.: Semantische Dialogmodellierung mit gewichteten Merkmal-Werte-Relationen. Elektronische Sprachsignalverarbeitung, 2009, pp. 25-32



- [8] ParaView: <http://www.paraview.org>
- [9] Chen, P. P.-S.: The entity-relationship model - toward a unified view of data. ACM Trans. Database Syst., ACM, 1976, vol. 1, pp. 9-36
- [10] Karnagel, K.; Römer, R.; Rogge, S.; Lindemann, J.: Semantische Modellierung von System-Benutzer-Interaktionen am Beispiel einer Home-Entertainment Steuerung. Elektronische Sprachsignalverarbeitung, 2012, wird noch veröffentlicht
- [11] Wirsching, G.; Kolbl, C.: Language Modeling with Utterance-Meaning Pairs. Technical Report, Angewandte Informatik, Universität Augsburg, 2011