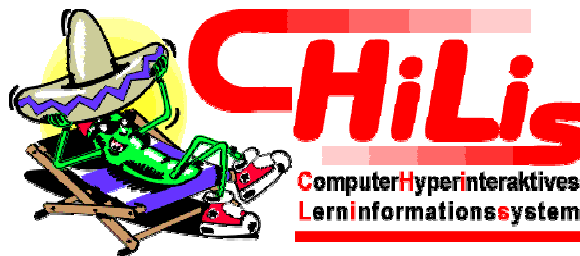


Interaktive Animationen und Visualisierungen, eine neue Qualität und „Spielwiese“ für die Mechanik

Ergebnisse der interdisziplinären Projektgruppe CHiLis



Projektgruppe CHiLis¹
Prof. Dr. Reinhard Keil-Slawik
Thorsten Hampel

Heinz Nixdorf Institut
Universität-GH Paderborn
33102 Paderborn
chilis@uni-paderborn.de

Der Einsatz neuer Medien in der Lehre kann seine volle Qualität erst in der Betonung von Fähigkeiten entfalten, die herkömmliche Medien nur bedingt bieten. Erst eine Kombination unterschiedlichster Medien und neuer Visualisierungstechniken oder die Berücksichtigung eines hohen Grades von Interaktivität kann den Einsatz von Multimedia in der Lehre zu einer breiten Akzeptanz führen. Die interdisziplinäre Projektgruppe CHiLis erforscht an der Universität-GH Paderborn zwei spezifische Charaktereigenschaften der Erstellung von JAVA Animationen und Visualisierungen. Zum einen steht die Untersuchung der Möglichkeiten einer dreidimensionalen Visualisierung komplexer interaktiver Versuchsaufbauten der Mechanik im Vordergrund, zum anderen wird die Leistungsfähigkeit der einfachen und komponentenbasierten Gestaltung von Animationen aus dem Bereich der Grundlagenvorlesung zur Mechanik untersucht. Der folgende Beitrag stellt zwei Zwischenergebnisse der Projektgruppenarbeit vor.

¹ Christian Saalborn, Elke Stanger, Ingo Fuß, Lars Fleigl, Lars Steinfurth, Marc Sinemus, Mathias Preiß, Thorsten Kontorzik, Adelhard Tuerling, Frank Albracht

viLAB – Visualisierung von komplexen Versuchen in Java3D

Einführung

Komplexe Versuchsbauten in Bereichen der Mechanik sind oftmals schwer zu verstehen und stellen somit eine Hürde für Mechanikstudenten im Hauptstudium dar. Die Gruppe viLAB der Projektgruppe ChiLis versucht exemplarisch einen Versuch zu visualisieren. Gewählt wird das Polariskop, ein Versuch aus der Spannungsoptik. Ein Polariskop dient zur Analyse von Materialproben auf mögliche Fehler und besitzt bereits eine breite Akzeptanz in der Industrie. Hierbei wird bestimmt gerichtetes Licht durch die Probe geschickt und anhand der mit einer Kamera aufgenommenen Lichtbilder, den sogenannten Isochromaten, lassen sich Veränderungen der Materialprobe erkennen. Dabei läßt man eine Kraft auf die Probe wirken. Dieser Versuch wird gewählt, da er sich offenbar gut visualisieren läßt und eine interaktive Änderung der Kraft an der Probe mit der Maus möglich ist. Als Implementationssprache soll eine plattformübergreifende Sprache gewählt werden, um eine Verbreitung im Internet und gegebenen Universitätsnetzen zu gewährleisten. Es wird entschieden, 3D als Visualisierungsmittel einzusetzen, um einerseits neue Wege der Visualisierung aufzuzeigen, als auch die Motivation des Lernenden hoch zu halten, der Gelerntes in einer spielerischen Atmosphäre verifizieren soll.

Spezifikation des Prototyps

Zunächst liegt es nahe auf der Basis von VRML zu entwickeln, da dieses bereits als ein Standard in der Internetwelt vorliegt. Die Nachbildung der Versuchsbauten wird damit problemlos möglich. Eine Navigation wird durch die verschiedenen Player redundant. Bald tauchen jedoch Probleme in Bezug auf Animationen, Interaktion und Berechnungen auf. Daher entschließt man sich eine eigene 3D-Engine in Java zu implementieren, mit deren Hilfe man dreidimensionale Szenen darstellen kann. Inmitten der Spezifikationsphase wird die Spezifikation der Java-3D API von Sun Microsystems zur JavaOne in San Francisco vorgestellt. Schnell wird klar, daß man damit alle gesteckten Programmierziele und Anforderungen erfüllen kann. Der Versuchsaufbau läßt sich problemlos dreidimensional nachbilden, so daß Ansichten aus beliebiger Perspektive möglich sind. Auch lassen sich reale Versuchsergebnisse durch Einstellung assoziierter Parameter, was optional durch Dialogelemente oder interaktiv im 3D-Fenster erfolgen kann, simulieren.

Die einfache Graphstruktur der 3D-Szene sowie die Objektorientiertheit von Java schlechthin ließen nun eine parallele Entwicklung eines Prototypen zu. 3D-Szenen werden in Java3D in einer baumartigen Struktur gespeichert. Einzelne Szenen lassen

sich getrennt implementieren und durch verbinden der Teilbäume miteinander verknüpfen.

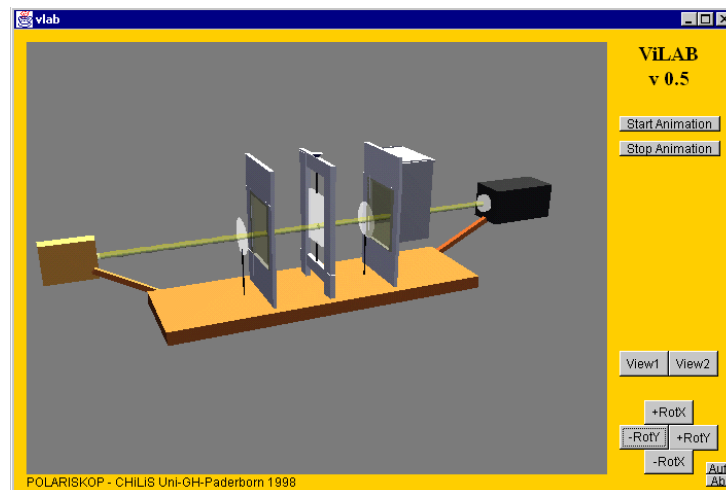


Fig 1 Das Polariskop als 3D Modell im Viewerfenster

Implementation

Mit Hilfe der Buttons im Viewer (Fig.1) kann der Student das Objekt in alle Richtungen drehen, sowie verschiedene Viewpoints anfahren. In dem Viewer ist ein Canvas3D-Fenster, in welchem Java3D die entsprechende Sicht rendert. An der angebrachten Probe in der Mitte des Polariskopes kann man Kräfte durch Ziehen mit der Maus verändern.

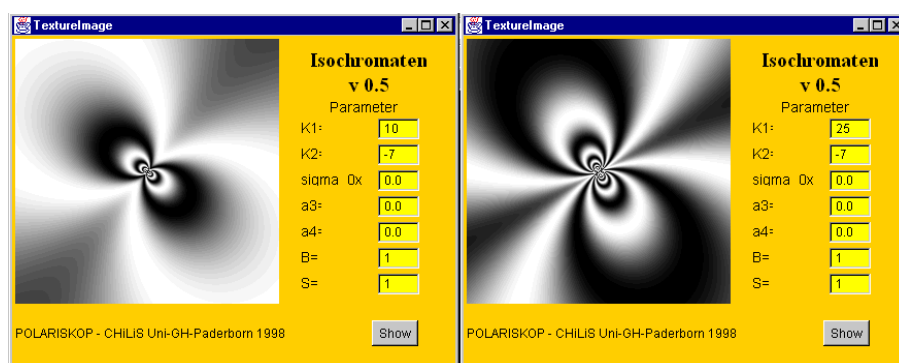


Fig 2 – Das Parameterfenster mit zwei verschiedenen Wertevorgaben

Das zweite Fenster (Fig 2.) enthält die Werte der verschiedenen Parameter. Sie lassen sich in sinnvollen Bereichen ändern. Eine Änderung der Werte führt nach Betätigung des Berechnungsbuttons zur Anzeige des aktualisierten Isochromatenbildes. Exemplarisch wird hier der Kraftparameter K_1 geändert, so daß sich die Ausgabe des Isochromatenbildes ändert.

Das vorgestellte Applet läuft dank des neu verfügbaren JavaPlugins in den gängigen Browsern. Hierzu ist lediglich eine Konvertierung der assoziierten HTML-Seite mittels des mitgelieferten HTML-Konverters notwendig.

Schnittstelle zu VRML

Zur Verbindung von weiteren Applets in dieser Thematik zu einer Oberfläche dient ein Labor, welches in VRML implementiert wurde. Dort finden sich einzelne Versuche wieder, die von dort aus selektiert werden können. Desweiteren finden sich dort auch Hilfen in Form von Auszügen aus Lehrbüchern und Links zu themenverwandten Websites in aller Welt.

mechANik-Beans – eine Spielwiese für die Visualisierung und Animation von Problemstellungen der Mechanik

Im Rahmen der interdisziplinären Projektgruppe „CHiLis“ wird neben der Entwicklung interaktiver 3D Visualisierungen für das Hauptstudium des Maschinenbaus ein zweiter Ansatz zur Erstellung von Animationen für den Bereich des Grundstudiums erforscht. Geplant ist nicht die Realisierung einer Anzahl von vollendeter, in sich geschlossener Javavisualisierungen, vielmehr wird angestrebt, eine Komponentenbibliothek für die Erstellung interaktiver Animationen zu entwerfen. Die Idee, einen „Mechanikbaukasten“ für Studenten und Tutoren des Fachbereiches Maschinenbau für den Einsatz in Übungen der Mechanikvorlesung im Grundstudium zu entwickeln, wird maßgeblich durch den Gedanken des „Steinbruch des Lernens“ geprägt. Die Metapher, Multimedia als eine Art „Steinbruch“ bestehend aus unterschiedlichsten Medien betrachten zu müssen, beschreibt treffend unseren Ansatz weniger die Entwicklung von aufwendigen und teuren „Hochglanz-Multimedia“ zu forcieren, als vielmehr Dozenten zur selbständigen Entwicklung und Konzeption von Multimedia anzuleiten.

Die JavaBean Technologie wird zur Umsetzung der Idee ausgewählt, um von Plattformen und von Entwicklungsumgebungen unabhängig zu sein. Außerdem wird durch diese Technik der Einsatz der Animationen in Web-Browsern möglich und somit eine Lehre über das Internet realisierbar.

Jedes mechanische Element, wie z.B. Balken, Lager, Seil, Gewicht u.s.w., wird jeweils als eine JavaBean implementiert. Somit lassen sich diese Elemente in einer

visuellen Entwicklungsumgebung (z.B. VisualAge von IBM) zu komplexen Versuchsaufbauten zusammenstellen, wobei die mechanischen Funktionen der einzelnen Beans genutzt werden können. Die Visualisierung von traditionellen Übungsaufgaben gewinnt auf diese Weise eine neue Qualität. Die Illustrationen von mechanischen Sachverhalten beschränkt sich in der klassischen Mechaniklehre lediglich auf die abstrahierte Abbildung eines gegebenen Problems, ein direkter Bezug zur real existierenden Problemstellung wird meist nur mangelhaft deutlich. Die von der Projektgruppe entwickelte objektorientierte Java Komponentenbibliothek erlaubt nun auf einfache und interaktive Weise mechanische Sachverhalte durch Kombination von einzelnen Beans zu visualisieren. Ziel ist es hierbei, den zusätzlich erforderlichen Java-Quellcode auf ein Minimum zu reduzieren. Ergebnis sind universell verwendbaren Komponenten (JavaBeans), die als Bauteile kombiniert ein kompaktes System zur Darstellung von mechanischen Problemen schaffen. Neben reinen Visualisierungsaufgaben werden durch definierte Schnittstellen zur JAVA Sprache auch mathematische Lösungen produziert. Für die Anwender der fertigen Applikation bedeutet der Einsatz von Beans eine homogene Oberfläche und immer wiederkehrende Objekte, die in ihrem Verhalten konform sind. Somit ist die Einarbeitungszeit für den Lernenden minimal. Der Tutor hingegen, der mechanischen Sachverhalt visualisiert, kann sich vollkommen auf die Funktionalität des Aufbaus konzentrieren und schnell Alternativen entwerfen, um verschiedene Anordnungen zu testen.

Im folgenden Beitrag wird die Visualisierung eines Flaschenzuges aus Seilen und Rollen durch die entwickelten Basiskomponenten beschrieben.

Ein Flaschenzug aus mechANik-Beans

Ein Vorteil von JavaBeans, ist der universelle Einsatz in visuellen Programmierungsumgebungen. Durch den funktional gebundenen Einsatz der Beans, ist es auch Entwicklern (Tutoren) mit wenig Java Erfahrung möglich, Applikationen zu erstellen. Für den Tutor einer Mechanikvorlesung ist es wichtig, sich auf die Konstruktion der Lösung einer mechanischen Aufgabe zu konzentrieren. Sie/Er sollte sich nicht primär mit Java Grafikbibliotheken beschäftigen müssen, um zu einer optisch ansprechenden und mechanisch korrekten Animation zu gelangen.

Durch die entwickelten JavaBeans, bestehend aus den Grundelementen der Mechanik, kann eine normierte Optik über einfaches "Zusammenklicken" in einer visuellen Programmierungsumgebung erstellt werden. Das Erscheinungsbild einer auf diese Weise konstruierten Visualisierung entspricht den in der Mechanik Theorie üblichen Vorgaben.

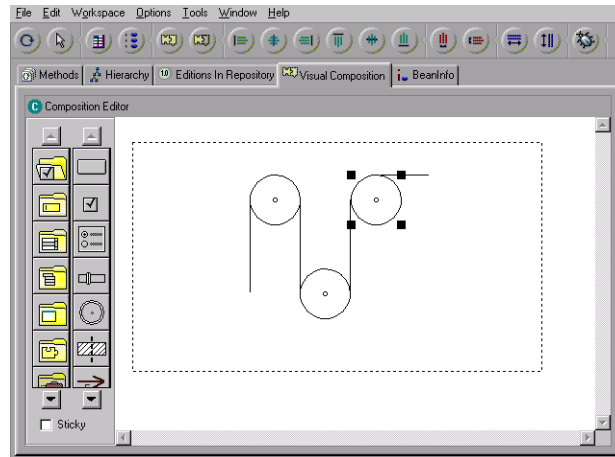


Fig. 3. Einsatz der mechANik-Beans am Beispiel eines Flaschenzugs

Dem Entwickler einer Animation oder Simulation wird hierdurch die Möglichkeit eröffnet, sich von rein technischen Realisierungsproblemen zu lösen und mehr didaktische Aspekte in das Blickfeld der Betrachtung zu stellen. Die Komponentenbibliothek bietet dem Lernenden nicht nur ein starres System, sondern eine „Mechanikspielwiese“ mit aktiven und agierenden Elementen. So werden funktionelle Aspekte des Systems in die Relevanz der Betrachtung mit aufgenommen.

Die Implementierung einer mechanisch, mathematischen Schnittstelle, hat bei der Lösung von gestellten Problemen den Vorteil, eben dieses Schrittweise in formaler Form durch visuelle Unterstützung nachzuvollziehen.

Die entwickelten mechANik-Beans sind durch diese Schnittstelle interaktiv in der Lage, ein mechanisches System mathematisch zu repräsentieren, soweit eine Reduzierung auf bestimmte Parameter möglich ist. Die mathematische Berechenbarkeit komplexer mechanischer Aufgaben ist für die entwickelten Beans aus diesem Grunde nur begrenzt realisierbar. Wir haben die visuelle Programmiertechnik gewählt, da hier die Funktionalität der zu verwendenden Tools relativ schnell ersichtlich ist und diese verständlich eingesetzt werden können. Der Einsatz von JavaBeans ist auch in der Hinsicht vorteilhaft, da sie universell einsetzbar sind. Der Programmierer kann sich direkt von der Funktionalität des Systems überzeugen, da er es nicht nur in Augenschein nehmen, sondern auch die gegebenen Funktionen schon zur Entwicklungszeit überprüfen kann. Die Funktionalität kann begrenzt oder ganz entzogen werden, was dann einer reinen Illustration entspricht.

Die Fig. 3 zeigt das Auswählen der Komponenten (3 Rollen und ein Seil) aus der Bean-Liste in der Entwicklungsumgebung VisualAge von IBM. Die mechANik-Beans werden per „Drag and Drop“ auf die visuelle Entwicklungsumgebung gezogen und angeordnet. Die einzelnen Bauteile werden mit der Maus vergrößert oder verkleinert um sie an den Versuchsaufbau anzupassen.

Fig. 4 illustriert das Verbinden der einzelner Komponenten, d.h. die eigentliche Funktionalität der Animation wird auf diese interaktive Weise hergestellt. Die notwendigen Verknüpfungen sind auch für in der visuellen Programmierung wenig vertraute Anwender sehr schnell zu erlernen, da Funktionen und Eigenschaften der Elemente, einfach aus kontextsensitiven Menüs selektiert werden können.

Konkret wird das obere rechte Seilende mit einem Balken verbunden. Damit existieren zwei feste und eine lose Rolle. Der Schieberegler wird mit dem losen Seilende verknüpft. Schließlich wird der Kraftpfeil an der unteren Rolle befestigt um die resultierende Kraft anzuzeigen.

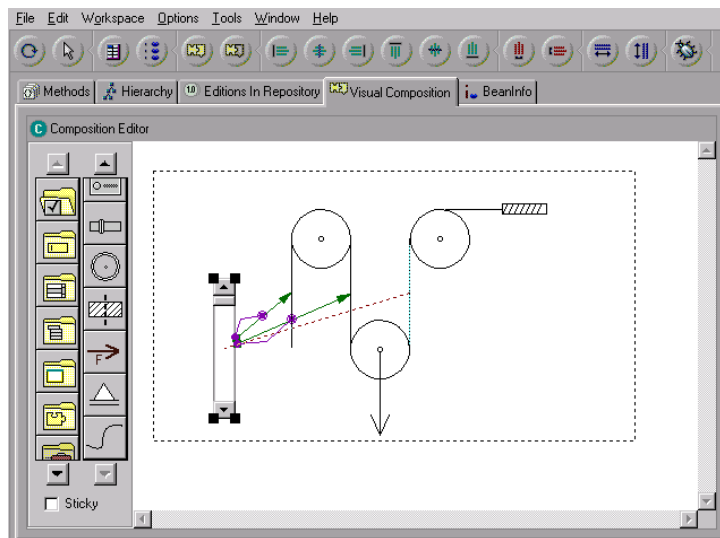


Fig. 4. Verbinden der mechANik-Beans mit Standard Oberflächenelementen

Die Fig. 5 zeigt den abschließenden Schritt der Entwicklung einer Animation, die Verknüpfung der mechanischen Programmelemente mit Standardbeans (Scrollbars, Eingabefeldern u.s.w.). Durch einen universellen Ansatz der Konzeption der mechANik-Beans lassen sich sämtliche Standard-Grafikelemente der Java Bibliotheken nutzen - ein wichtiger Aspekt für den praxisnahen Einsatz der entwickelten Grundelemente. Eigene Formeln, die für einen mechanisch korrekten Versuch zwingend notwendig sind, können ohne Probleme in Form individueller Klassen und Methoden eingebunden werden. In dem Beispiel des Flaschenzuges besteht eine Verknüpfung zwischen der Kraftformel, dem Schieberegler und dem Ergebnisfeld.

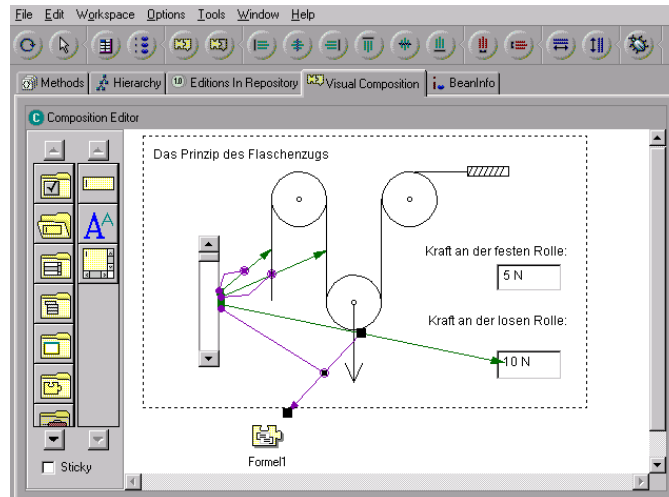


Fig. 5 Visuelle Programmiertechnik hilft maßgeblich eine Semantik der Applikation zu realisieren

Die Fig. 6 stellt das endgültige lauffähige Applet mit allen benutzten Beans dar. An dem Schieberegler wird die Kraft verändert. Dadurch ist es möglich die auftretenden Kräfte zu berechnen und im Ergebnisfeld darzustellen. Durch Hinzufügen einer kurzen Java-Methode bewegt sich die lose Rolle des Flaschenzugs durch Ziehen mit der Maus am Seilende bzw. Vergrößern der Kraft am Schieberegler auf und ab. Es wird mit minimalem Aufwand eine für den täglichen Lehrgebrauch ausreichend komplexe Visualisierung geschaffen.

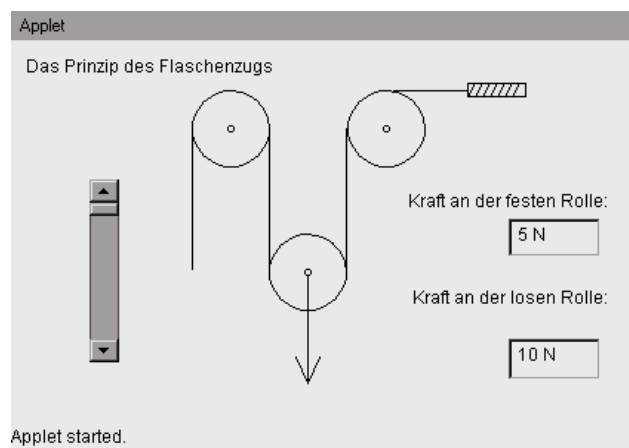


Fig. 6 Das fertige aus mechANik-Beans konstruierte Applet

Literatur

- Brenneke, A., Engbring, D., Keil-Slawik, R., Selke, H.:* Das Lehren mit elektronischen Medien lernen - Erfahrungen, Probleme und Perspektiven bei multimedialgestütztem Lehren und Lernen. In: *Wirtschaftsinformatik* 39 (6), 563-568 (1997).
- Brenneke, A., Keil-Slawik, R.:* Notes on the Alltagspraxis of Hypermedia Design. In: Maurer, H. (Ed). *Educational Multimedia and Hypermedia. Proceedings of ED-MEDIA 95*, Graz, Austria, June 17-21 (1995).
- Hampel, T.:* Analyse des Einsatzes und der Alltagspraxis von Multimedia in Forschung und Lehre der Technischen Mechanik - Entwicklung eines hypermedialen Animations-, Simulations- und Informationssystems -Diplomarbeit (HSII) für den Studiengang Informatik, presented for review, Prof. Dr. Reinhard Keil-Slawik, Universität- GH Paderborn, (1996).
- Ferber, F., Herrmann, K.P.:* Einsatz von Multimedia in der experimentellen Spannungsanalyse am Beispiel bruchmechanischer Untersuchungen mittels der Schatten- und Spannungsoptik, GESA-Symposium, 10.-11. Oktober 1996, Schliersee
- Brenneke, A., Keil-Slawik, R.:* *Einsatz elektronischer Lehr- und Lernumgebungen in der Software-Ergonomie Ausbildung. In: Liskowsky, R., Velichkovsky, B. M. und Wüschmann, W. (eds.): Software-Ergonomie '97 Usability Engineering: Integration von Mensch-Computer-Interaktion und Software-Entwicklung.* pp.83-92, Stuttgart: Teubner (1997).
- Manogg, P.:* Anwendungen der Schattenoptik zur Untersuchung des Zerreißvorganges von Platten. Dissertation, Ernst-Mach-Institut, Freiburg, (1964)
- Radaj, D.:* Zur Didaktik und Geschichte der Bruchmechanik. *Materialprüfung* 12, Nr. 7, 236-237, (1970)