

# Mechanismen-Engine

**Projektleitung**  
Prof. Dr.-Ing.  
Stefan Gössner

**Zeitraum**  
2009 – 2010

**Förderung**  
Fachhochschule  
Dortmund  
Forschungsbudget

**Kontakt**  
Prof. Dr.-Ing.  
Stefan Gössner  
Fachbereich  
Maschinenbau  
Fachhochschule  
Dortmund  
Sonnenstraße 96  
44139 Dortmund  
Tel.: 0231 9112-203  
E-Mail: stefan.goessner  
@fh-dortmund.de

Das Projekt entstand aus der Idee, den Nachweis zu führen, dass es mit vertretbarem Aufwand möglich sei, die Konzepte einer Physik-Engine zur Implementierung einer Simulations- und Analyse-Software für ebene Mechanismen zu übernehmen und jene Mechanismen-Engine als reine Webanwendung – insbesondere ohne Einsatz proprietärer Plugins – zu gestalten.

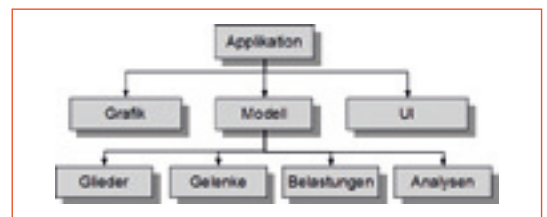
Physik-Engines werden primär von Video- und Computerspielen genutzt. Besonders das natürliche Verhalten der Bewegung starrer Körper wird hierin durch effiziente Echtzeitalgorithmen nachgebildet, wobei die nähere Betrachtung solcher Software eine große Ähnlichkeit zum Aufbau von MKS-Systemen offenbart. In Computerspielen ist eine große Zahl an bewegten Körpern die Regel. Diese sind eher selten durch Gelenke miteinander verbunden, sondern interagieren durch Oberflächenkontakte und Kollisionen. Die Stabilität der Simulation hat einen extrem hohen Stellenwert, damit sich die Objekte aus der Sicht des Spiele-nutzers nicht in unerwarteter Art und Weise bewegen. Die Genauigkeit der Bewegungsgrößen ist von sekundärer Bedeutung. Es genügt, wenn die Bewegung visuell plausibel bleibt. Besonders hier unterscheiden sich die Anforderungen an eine Mechanismensimulation stark von denen an die einer Spielesimulation.

Die Bewegung starrer Körper wird hierbei mittels generalisierter Koordinaten auf Grundlage der Bewegungsgleichungen nach Newton-Euler beschrieben. Es wird ein geschwindigkeitszentrierter Ansatz gewählt, mit dem Vorteil, dass ein lineares Gleichungssystem resultiert. Die Lösung jenes Gleichungssystems erfolgt mittels des iterativen Gauss-Seidel Verfahrens und die Integration auf der Basis der einfachen aber effektiven und recht stabilen semiimpliziten Euler-Integration. Der Notwendigkeit eines Stabilisationsverfahrens wird durch den Einsatz von Pseudoimpulsen Rechnung getragen. Diese allgemein impulsbasierte Vorgehensweise ist effizient und gleichzeitig ressourcengnügsam, so dass eine Webanwendung auf dieser Grundlage auch auf kleinen mobilen Geräten mit akzeptabler Performanz laufen kann.

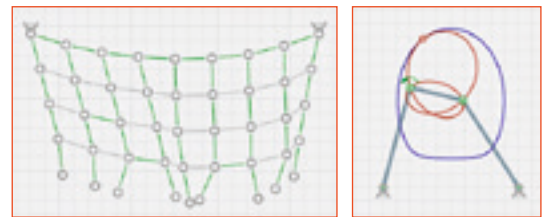
Das Strukturmodell eines Mechanismus wird zunächst mittels eines einfachen textuellen Formats beschrieben. Dieses beinhaltet Informationen zu

- Modell (Gravitation, Darstellungsform)
- Gliedern (Geometrie, Masse)
- Gelenken (beteiligte Glieder, Antriebe)
- Belastungen (beteiligte Glieder, Parameter)
- Analysen

Die Gliedgeometrie wird auf das Notwendige beschränkt. Sie dient unter anderem zur Ermittlung von Schwerpunktlagen und Massegrößen. Ein Geometrieeditor bzw. ein Import aus geometrieverarbeitenden Systemen heraus spielt gegenwärtig keine Rolle. Die vektorgrafische Darstellung des Modells ist zwar für den Betrieb der MecEngine nicht erforderlich, aus Anwendersicht jedoch unerlässlich.



Softwarearchitektur



Simulation von Netz und Polbahnen des Gelenkvierecks

Als Anwendungsumgebung wird eine Webplattform gewählt. Gegenwärtig ist das primär der Browser, jedoch zeigen aktuelle Entwicklungstendenzen eine Ausweitung auf Betriebssystemebene. Die verwendeten Webtechnologien sind HTML5, CSS und Javascript. Die 2D-Vektorgrafik basiert auf der vom Canvas-Element bereitgestellten Programmierschnittstelle.

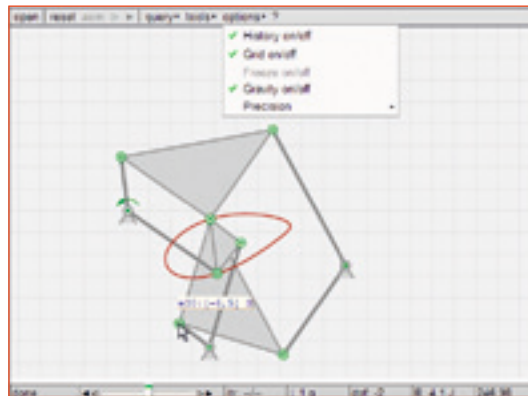
Eine einfache Benutzeroberfläche der Webanwendung wird mittels reinem HTML5 und CSS gestaltet.

Die Implementierungssprache Javascript ist eine dynamische, imperative Sprache mit C-Syntax und funktionalen Eigenschaften. Ihr Objektkonzept basiert nicht auf Klassen sondern auf Prototypen. Die Ausführungsgeschwindigkeit ist in den letzten Jahren erheblich gesteigert worden und mittlerweile auch für wissenschaftlich, technische Anwendungen akzeptabel.

Die Anwendung besitzt einen einfachen Objektaufbau in Anlehnung an das bewährte Model/View/Controller-Prinzip. Das Applikations-Objekt kontrolliert Benutzereingaben und sonstige Ereignisse, stellt Funktionen zum Import und Export von Modelldaten bereit, und orchestriert das

Zusammenspiel von Grafiksystem und Mechanismenmodell speziell im Hinblick auf die zeitgesteuerte Simulation und Animation.

Das Grafik-Objekt bettet das Canvas-Element ein und stellt zusätzlich notwendige Funktionalität sowie Interaktionsmöglichkeiten bereit.



Oberfläche der Webanwendung - Robert'sche Ersatzgetriebe

Das zentrale Modell-Objekt verwaltet neben spezifischen Eigenschaften, wie Gravitation und Timer, die jeweilige Mechanismenstruktur in getrennten Containern für Glieder, Gelenke, Belastungselementen und Analyseobjekten. Die innere Simulationsschleife ist hier implementiert.

Als primäres Beschreibungsformat wurde JSON gewählt. Dieses einfache, populäre, menschen- und maschinenlesbare Webformat ist hier eine bessere Alternative zu XML, harmonisiert vorzüglich mit Javascript und kann mit wenig Aufwand in andere Darstellungen konvertiert werden.

Die Anwendung erlaubt die Modellierung, Simulation und Analyse ebener Mechanismen auf Basis einer Reihe vordefinierter Gelenktypen und Kraftelemente. Der impulsbasierte Ansatz ist robust und liefert neben den Lage- und Bewegungsgrößen generell auch die Gelenkkräfte – selbst in statischen Strukturen. Der Zusammenbau eines Mechanismus ist auch aus völliger Unordnung

heraus möglich, Verzweigungslagen werden konsistent zur Trägheit der Massen weiter behandelt und invalide Getriebebestellungen irgendwie glaubhaft dargestellt.

Es zeigt sich bei den getesteten Beispielmechanismen ein zufriedenstellendes bis sehr gutes Geschwindigkeitsverhalten. Dabei hat es sich bewährt, die Ergebnissenauigkeit konkurrierend zur Geschwindigkeit seitens des Benutzers in drei Stufen gering / mittel / hoch einstellen zu lassen.

Die Anwendung selbst ist letztlich eine leichte Schale um eine wiederverwendbare Javascript-Bibliothek und erzielt dann insgesamt eine recht geringe Größe von etwa 170 KB. Sie qualifiziert sich als reine Webanwendung ohne serverseitigen Anteil für einen Einsatz als Webservice im Sinne dessen, was gegenwärtig als Cloud Computing bezeichnet wird. Genauso gut kann die MecEngine im Intranet, letztlich jedoch auch problemlos als lokale Anwendung direkt auf dem Rechner des Nutzers eingesetzt werden. Als Rechnerhardware sind kleine tragbare Geräte bis hin zu üblichen Tischcomputern adressiert.

Die Webapplikation stellt keinen Ersatz für die mittlerweile ausgereiften, leistungsstarken Hilfsmittel im Arbeitsumfeld des Ingenieurs dar. Vielmehr gilt als sinnvoller praktischer Einsatzbereich die webbasierte Lehre und die technische Dokumentation.

/1/ Gössner, S.: Eine Physik-Engine zur webbasierten Mechanismensimulation – Ergebnisse einer Studie. 9. Kolloquium Getriebetechnik – Chemnitz 2011

/2/ Gössner, S.: Web based Mechanism Simulation and Analysis utilizing a Physics Engine. Front Trends – Webtechnology Conference, Warsaw, 2010.

/3/ Gössner, S.: Ein kinematisches Modell zur Analyse materialflußtechnischer Bewegungsabläufe. Verlag Praxiswissen, 1992.