Batchelorarbeit FS 2012

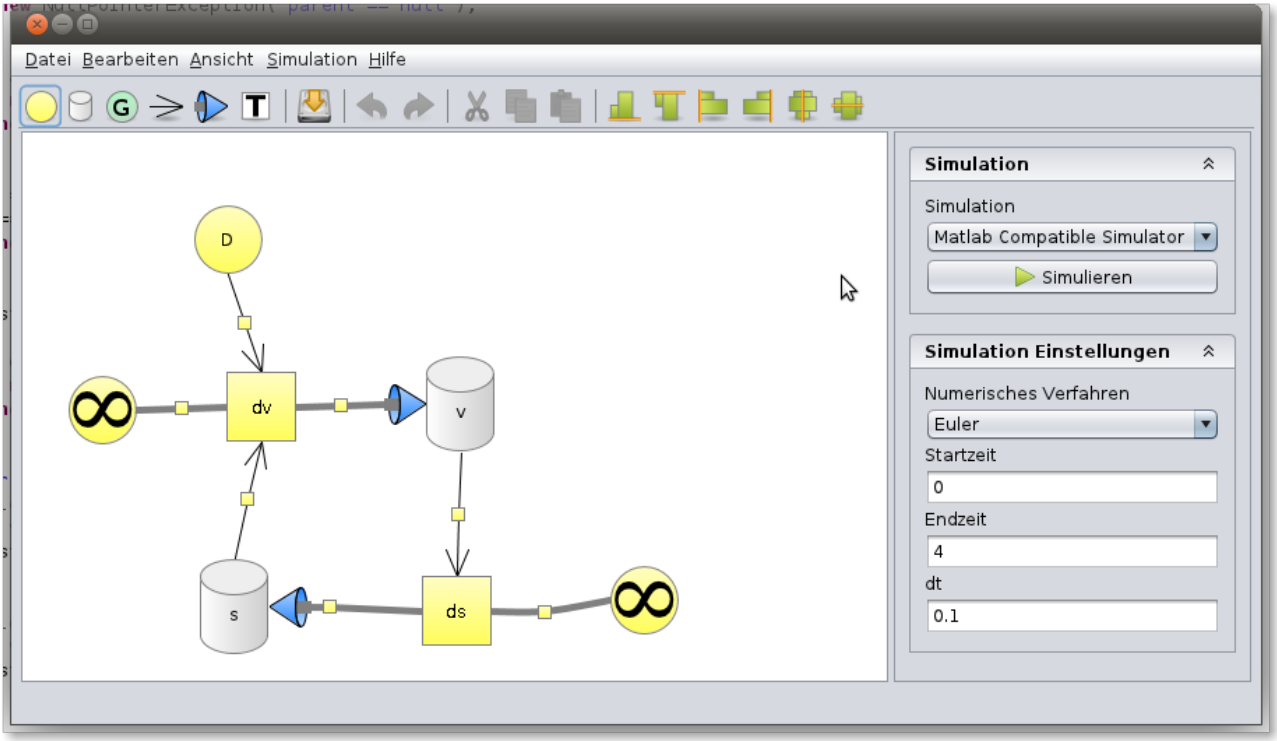
Andreas Bachmann, Andreas Butti

ABGABEDATUM

School of Engineering

Simulation

Betreuer	Stephan Scheidegger, Füchslin Rudolf Marcel
Studiengang	Systeminformatik
Klasse	SI09a



Inhaltsverzeichnis

Andreas Bachmann, Andreas Butti

Abstract

Englische Version von “Zusammenfassung”

Andreas Bachmann, Andreas Butti

Zusammenfassung

Wird erst am Ende der Arbeit geschrieben

Andreas Bachmann, Andreas Butti

Vorwort

In der Forschung aber auch zu Unterrichtszwecken werden viele Theorien vermittelt und Simuliert. Solche Simulationen basieren meistens auf einem Strikt Mathematischen Hintergrund, meinst Integrale erster Ordnung. Forschende Personen im Bereich Physik / Biologie verwenden dabei bevorzugt eine Modellierungssoftware, da es nicht ihr Fachgebiet ist selbst zu programmieren.

Andreas Butti hat sich bei der Verwendung von Simulationstools über deren Plattformabhängigkeit und Benutzerunfreundlichkeit gestört. Als Informatiker kommt man da schnell in Versuchung selbst etwas besserer zu schreiben. Nach einem Gespräch mit dem Physikdozenten, Herr Scheidegger, wurde daraus dann diese BA, die jedoch nicht nur ein Benutzerfreundliches Simulationswertzeug sein soll, sondern auch bisher nicht vorhandene Möglichkeiten für die Simulationen von Biologischen Abläufen, wie das innere einer Zelle, darstellen soll.

TODO: Danksagung

Andreas Bachmann, Andreas Butti

Erklärung betreffend des selbständigen Verfassens einer Projektarbeit an der School of Engineering

Mit der Abgabe dieser Projektarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe).

Der / die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Projektarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum

.....

Unterschriften

.....
Andreas Bachmann

.....
Andreas Butti

Andreas Bachmann, Andreas Butti

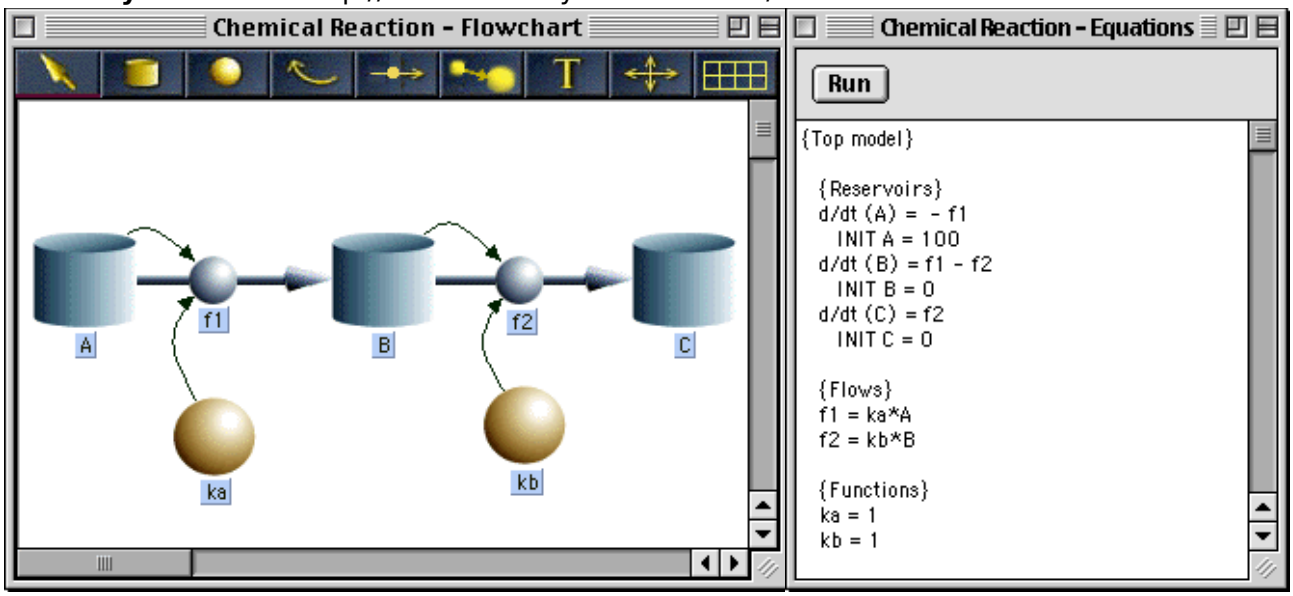
Einleitung

Es gibt bereits viele Simulationstools, wie z.B. Barekely Madonna um nur ein bekanntes Beispiel zu nennen. Diese Tools unterstützten die Modelierung von Differentialgleichungen als Modell, mit Containern und Flüssen. Diese modellart ist schon lange bekannt, und etabliert, daher übernehmen wir dieses Prinzip.

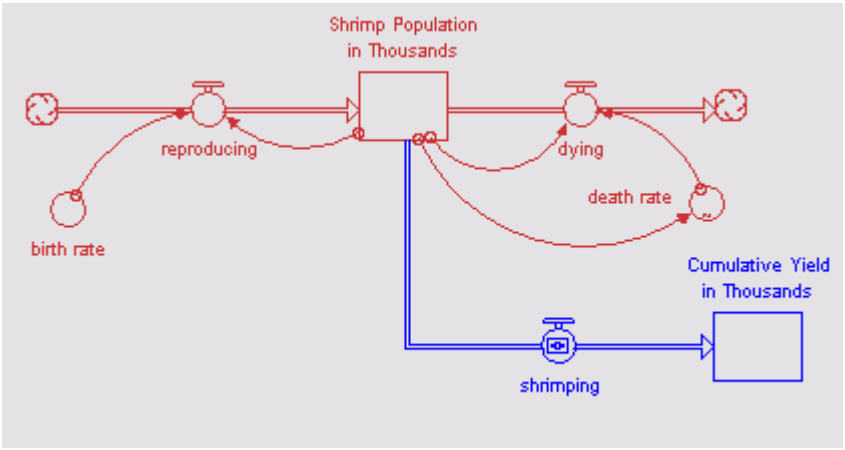
Bereits existierende Tools

Die Screenshots stammen von der Herstellerseite.

Berkeley Madonna: http://www.berkeleymadonna.com/

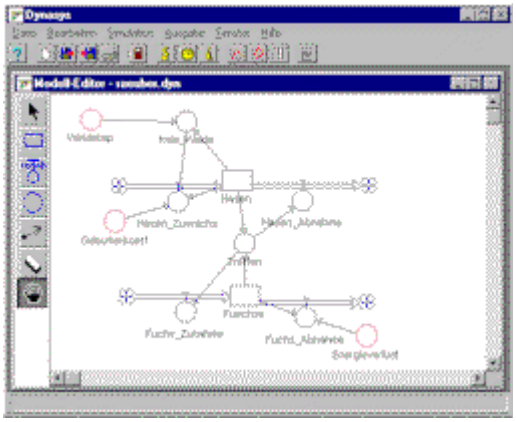


Stella: http://www.iseesystems.com/softwares/Education/StellaSoftware.aspx



Andreas Bachmann, Andreas Butti

Dynasis: http://www.hupfeld-software.de/pmwiki/pmwiki.php,
http://code.google.com/p/dynasys/
Wahrscheinlich das älteste der 3 Tools, unterdessen OpenSource, somit ist der Code einsehbar.



Individualität unserer BA

Mit unserer Simulation ist es zusätzlich möglich in einem XY Modell mehrere Meso Kompartmente abzubilden, ein Meso Kompartment ist das vorhin genannte Flussmodel.

Diese Meso Kompartmente können sich wären der Simulation im XY-Raum bewegen. Es können Dichten angegeben werden, die über den XY Raum verteilt sind, und die Meso Kompartmente können an Ihrer aktuellen Position von der dichte Konsumieren oder dichte Produzieren, somit kann die Umgebung beeinflusst werden.

Mit diesen Fähigkeiten ist es möglich das innenleben einer Zelle oder andere Biologische Prozesse **einfach**, **grafisch** abzubilden.

Die Idee und Vorgabe dieser Simulationsmethode stammt von Herr Scheidegger, und wurde zusammen mit Herr Füchslin und uns ausgearbeitet.

Andreas Bachmann, Andreas Butti

Vorgabe

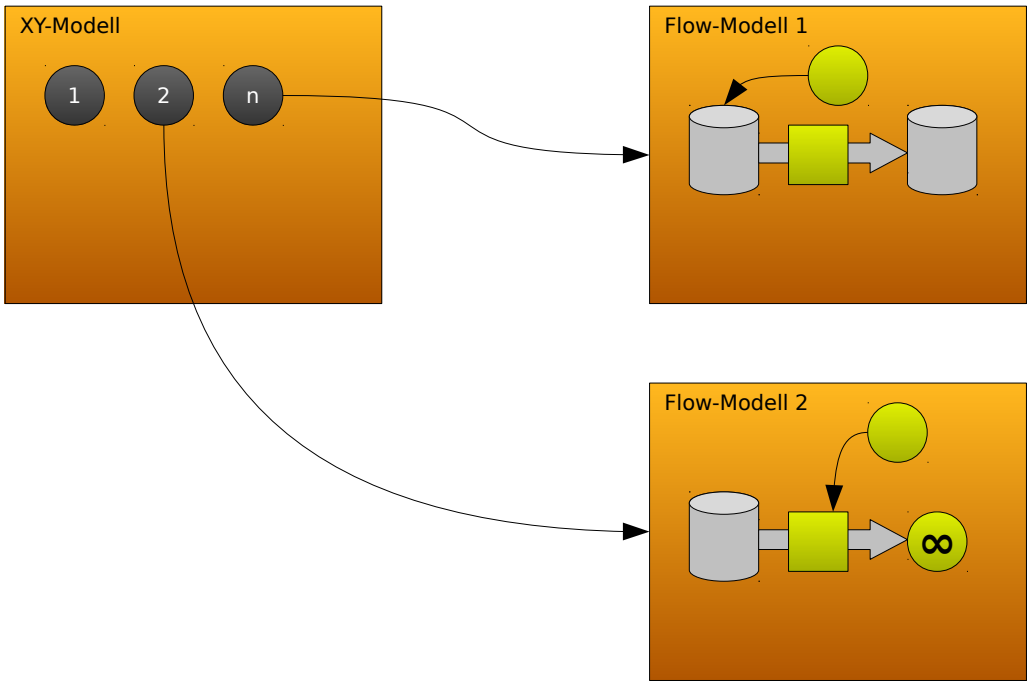
Bestehende graphische Modelleditoren erlauben eine effiziente Modellierung von kompartimentalen Systemen. Dabei unterstützt die graphische Oberfläche die Strukturierung des Modells bzw. des Systems. Dies kann gerade bei der Erfassung von komplexen Systemen den Zugang zu einer adäquaten Systembeschreibung erleichtern. Gerade aber Modelleditoren wie Berkeley-Madonna sind auf eine kompartimentale Struktur des Systems angewiesen. Räumlich strukturierte bzw. verteilte Systeme lassen sich nur schwer und in vereinfachter Form abbilden. Die Verwendung oder Kopplung verschiedener Simulationswerkzeuge kann für gewisse technische Systeme in Betracht gezogen werden (z.B. elektrische Schaltung mit Komponenten, bei denen die Wärmeabstrahlung und oder Wärmeleitung räumlich modelliert werden). Bei vielen Systemen lässt sich aber durch eine solche Kopplung das System nicht abbilden. Bei biologischen Systemen z.B. können sich Kompartimente bewegen (bei Zellen z.B. Chemo- und Haptotaxis). Zudem zeichnen sich biologische Systeme durch hierarchische Kompartimentstrukturen mit Unterkompartimenten aus. Ein weiterer Aspekt betrifft die Möglichkeit, dass Kompartimente in biologischen Systemen fusionieren oder sich teilen können.

Anforderungen Das zu entwickelnde Modellierungswerkzeug soll an die intuitive graphische Oberfläche bestehender Modellierungswerkzeuge für kompartimentale Simulationen anknüpfen. Folgene Aspekte sollen konzeptuell untersucht und wenn möglich implemetiert werden: - Hierarchische Kompartimente

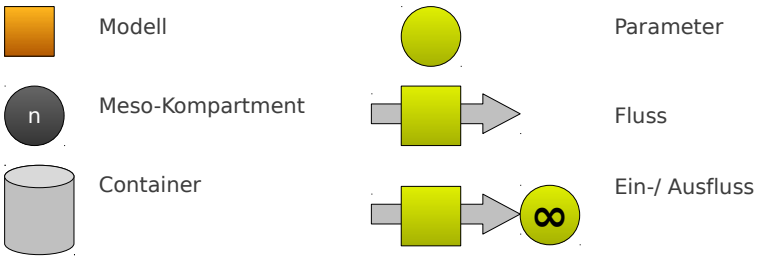
- Hierarchische Kompartimente - Räumliche Positionierung von Kompartimenten, welche die Wechselwirkung der Kompartimente auf gleicher Stufe beeinflussen kann und somit Einführung von Koordinaten (erster Schritt 2-Dim.) bzw. orthogonales Grid und beschreibung von Gradienten (z.B. für Änderung der räumlichen Position von Kompartimenten aufgrund von z.B. Gradienten) - Ausgabe eines Codes in einer Markup language (z.B. SBML), welcher von einem bestehenden Solver ausgeführt werden kann (z.B. Matlab)

Andreas Bachmann, Andreas Butti

Übersicht



Legende



Es kann entweder ein Herkömmliches «Flow-Modell» erstellt werden, das bereits bekannt ist da es vom Konzept her identisch bereits von vielen Simulationstools angeboten wird.

Oder es kann ein «XY-Modell» erstellt werden, das dann Meso Kopartments beinhaltet, diese befinden sich an einer Position (X / Y) und verweisen auf ein Modell («Flow-Model-X»).

Im «XY-Modell» könne sich «Dichten» befinden, das sind Stoffe die eine gewisse Konzentration an einer gewissen Stelle aufweisen.

Ein Meso Kompartment kann sich wären der Simulation im «XY-Modell» bewegen, es kann dichten Konsumieren oder Produzieren.

Andreas Bachmann, Andreas Butti

Theoretische Grundlagen

Simulationsverfahren...

Euler

Runge-Kutta

Werkzeuge und Hilfsmittel

Programmiersprachen

Als Programmiersprache kam Java zum Einsatz. Java ist Plattformunabhängig und sehr angenehm zum Programmieren.

Es existieren gute Bibliotheken für grafische Darstellungen, und Java war bereits beiden Studierenden bekannt.

Markup Language

Als Markup Language für die Simulation kam der Matlab Syntax zum einsatz, der ebenfalls vom Open Source Tool «octave» verarbeitet werden kann.

Entwicklungsumgebung

Bei Java ist man nicht fest an eine Entwicklungsumgebung gebunden. Das Projekt kann mit Hilfe der Ant-Buildfiles automatisch kompiliert werden, somit war es kein Problem das Andreas Bachmann IntelliJ und Andreas Butti Eclipse als Entwicklungsumgebung verwendet hat.

Testing

Die Zeit hat leider nicht gereicht eine komplette Testumgebung einzurichten. Das Automatische Testen einer GUI-Applikation ist auch nicht ganz trivial. Trotzdem wurden einzelne TestCases für JUnit geschrieben, jedoch nicht für die GUI.

Auf nachfrage war auch den Unterrichtenden Java Dozenten an der ZHAW keine Lösung bekannt die für GUI Applikationen allgemein funktionieren würde. Da sowieso zuwenig Zeit vorhanden war wurde das automatische Testing der GUI dann Ersatzlos gestrichen.

Andreas Bachmann, Andreas Butti

Vorgehen

MVC

Tests und Validierung

Test der Matheengine, Vergleich mit Berkeley-Madonna

Andreas Bachmann, Andreas Butti

Resultate

Beschreibung, Screenshots, Beispielsimulationen mit Ergebniss-Diagramm

Andreas Bachmann, Andreas Butti

Diskussion und Ausblick

Was Fehlt noch, was muss noch gemacht werden?

Interpretation und Validierung der Resultate

Rückblick auf Aufgabenstellung, erreicht bzw. nicht erreicht
Legt dar, wie an die Resultate (konkret vom Industriepartner oder weiteren Forschungsarbeiten; allgemein) angeschlossen werden kann; legt dar, welche Chancen die Resultate bieten

Verzeichnisse

Quellverzeichniss

Gloassar

Meso-Kompartment: Ein Teil des XY-Simulationsmodells

School of Engineering

Anhang

Aufgabenstellung

Bachelorarbeit 2012 - FS: BA12_scst_3

Allgemeines:

Titel:	Entwicklung eines graphischen Editors zur Modellierung von Systemen mit dynamischer Modellstruktur
Anzahl Studierende:	1

Betreuer:

HauptbetreuerIn:	Stephan Scheidegger, scst
NebenbetreuerIn:	Rudolf Marcel Füchslin, furu

Zugeteilte Studenten:

Diese Arbeit ist zugeteilt an:

- Andreas Bachmann, bachman0 (SI)
- Andreas Butti, buttiand (SI)

Fachgebiet:

- **CP** Comutational Physics
- **NAT** Naturwissenschaften
- **SOW** Software

Studiengänge:

- **SI** Systeminformatik
- **UI** Unternehmensinformatik

Zuordnung der Arbeit:

- **ZAMP** Zentrum für Angewandte Mathematik und Physik

Infrastruktur:

benötigt keinen zugeteilten Arbeitsplatz in der ZHAW

Interne Partner:

Es wurde kein interner Partner definiert!

Industriepartner:

Es wurden keine Industriepartner definiert!

Beschreibung:

Bestehende graphische Modelleditoren erlauben eine effiziente Modellierung von kompartimentalen Systemen. Dabei unterstützt die graphische Oberfläche die Strukturierung des Modells bzw. des Systems. Dies kann gerade bei der Erfassung von komplexen Systemen den Zugang zu einer adäquaten Systembeschreibung erleichtern. Gerade aber Modelleditoren wie Berkeley-Madonna sind auf eine kompartimentale Struktur des Systems angewiesen. Räumlich strukturierte bzw. verteilte Systeme lassen sich nur schwer und in vereinfachter Form abbilden. Die Verwendung oder Kopplung verschiedener Simulationswerkzeuge kann für gewisse technische Systeme in Betracht gezogen werden (z.B. elektrische Schaltung mit Komponenten, bei denen die Wärmeabstrahlung und oder Wärmeleitung räumlich modelliert werden). Bei vielen Systemen lässt sich aber durch eine solche Kopplung das System nicht abbilden. Bei biologischen Systemen z.B. können sich Kompartimente bewegen (bei Zellen z.B. Chemo- und Haptotaxis). Zudem zeichnen sich biologische Systeme durch hierarchische Kompartimentstrukturen mit Unterkompartimenten aus. Ein weiterer Aspekt betrifft die Möglichkeit, dass Kompartimente in biologischen Systemen fusionieren oder sich teilen können.

Anforderungen Das zu entwickelnde Modellierungswerkzeug soll an die intuitive graphische Oberfläche bestehender Modellierungswerkzeuge für kompartimentale Simulationen anknüpfen. Folgene Aspekte sollen konzeptuell untersucht und wenn möglich implemetiert werden: - Hierarchische Kompartimente

- Hierarchische Kompartimente - Räumliche Positionierung von Kompartimenten, welche die Wechselwirkung der Kompartimente auf gleicher Stufe beeinflussen kann und somit Einführung von Koordinaten (erster Schritt 2-Dim.) bzw. orthogonales Grid und beschreibung von Gradienten (z.B. für Änderung der räumlichen Position von Kompartimenten aufgrund von z.B. Gradienten) - Ausgabe eines Codes in einer Markup language (z.B. SBML), welcher von einem bestehenden Solver ausgeführt werden kann (z.B. Matlab)

Dateiformat .simz
XSD, Beschreibung etc.

Dokumentation

Übersichtsdiagramm, komplettes Klassendiagramm => CD

Inhalt CD

Verzeichnissstruktur

Code kommt NICHT in den Anhang.

count of lines: 66920 (Ohne Leerzeilen)

count of lines: 79380 (Mit Leerzeilen)