

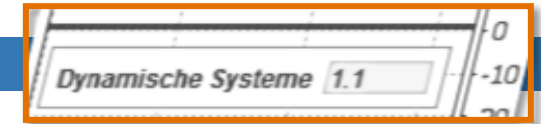
DYNAMISCHE SYSTEME

Softwareprojekt 2

08.06.2013

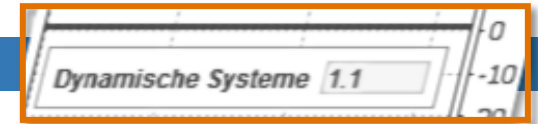
Roger Knecht, David Elsener

Agenda



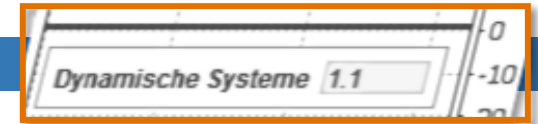
- Füchse essen Hasen essen Karotten
- Hefe-Zucker-System
- Projektidee
- Projektplanung
- Realisierung
- Demonstration
- Reflexion

Füchse essen Hasen essen Karotten



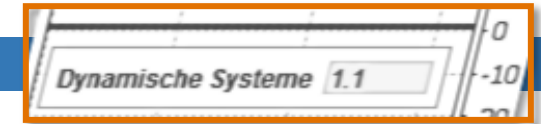
- Hase-Fuchs-System: Simulation eines **Räuber-Beute-Systems**
 - Simulation der Systemveränderungen über die Zeit
 - 3 Populationen: Füchse, Hasen, Karotten
 - 3 Wachstumsfunktionen:
 - Wachstum **Karotten** = $a - b \cdot \text{Hasen}$
 - Wachstum **Hasen** = $c - d \cdot \text{Füchse} + e \cdot \text{Karotten}$
 - Wachstum **Füchse** = $f + g \cdot \text{Hasen}$

Füchse essen Hasen essen Karotten

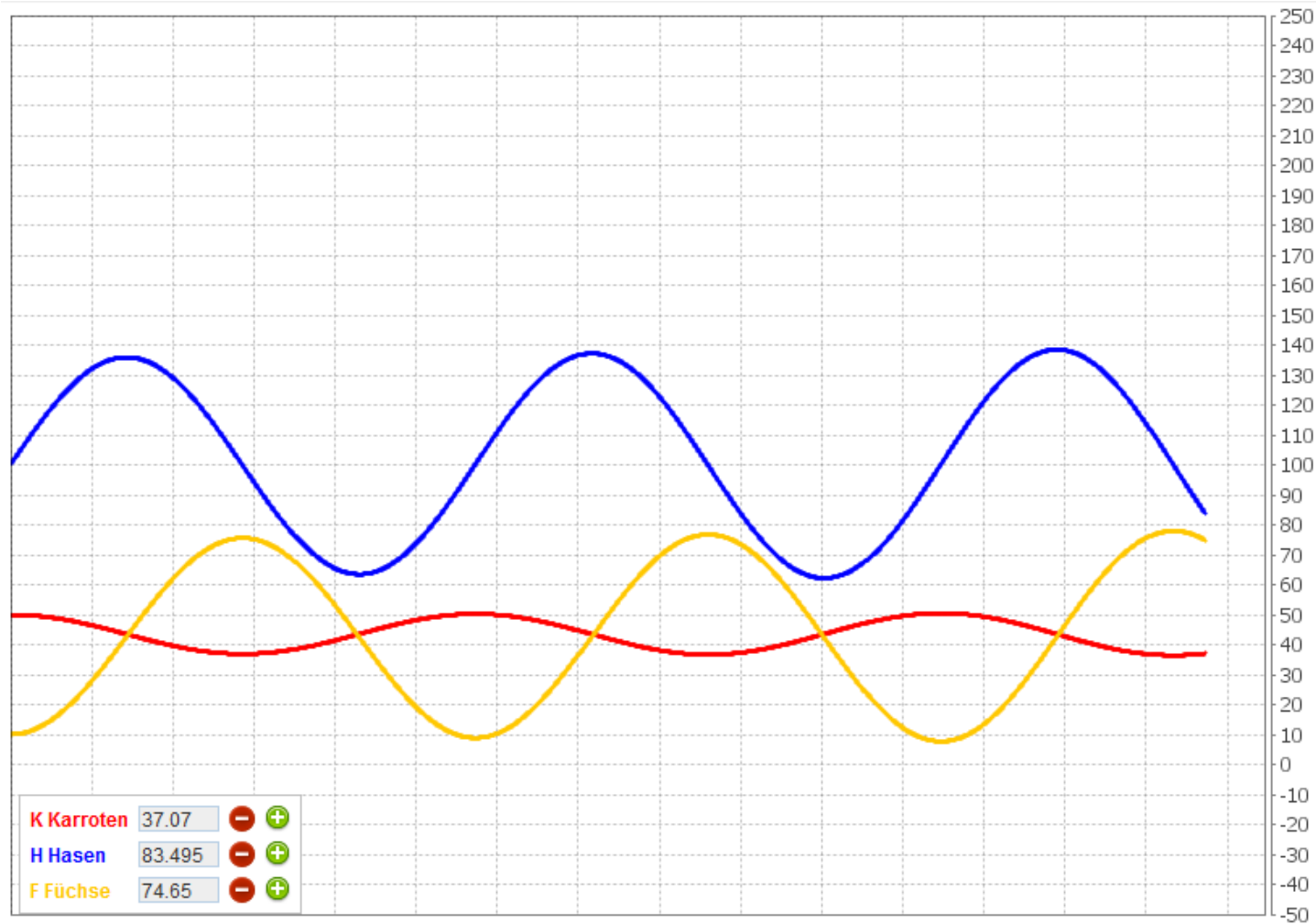


- Hase-Fuchs-System: Simulation eines **Räuber-Beute-Systems**
 - Simulation:
 - Zuweisung von Startwerten der Populationen und Parametrierung von a , b , c , d , e , f und g
 - Berechnungen in möglichst kleinen Schritten:
 - Wachstumsrate berechnen
 - Zur aktuellen Populationszahl Wachstumsrate addieren

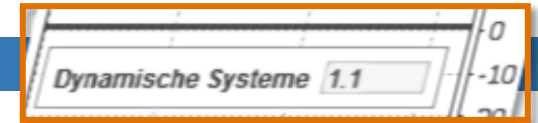
Füchse essen Hasen essen Karotten



- Demonstration



Hefe-Zucker-System

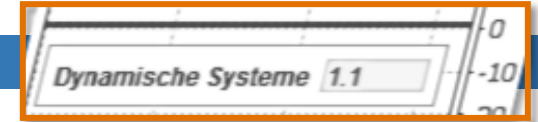


- Hefewachstum: Abhängig von der Zuckermenge und dem Platz, also auch der Hefemenge
- Zuckerwachstum: Abhängig von der Hefemenge und dem Hefewachstum
- Mathematische Definition (Kinetik nach Monod):
 - ▣ Wachstum Hefe (H') = $a \cdot Z + b \cdot H$
 - ▣ Wachstum Zucker (Z') = $c \cdot H + d \cdot H'$



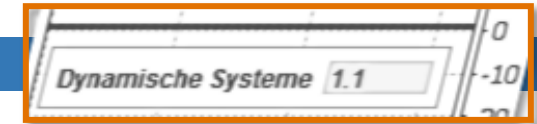
Differentialgleichungen

Projektidee



- Ziel: Programm zur Simulation von beliebigen Dynamischen Systemen
- Haupt-Anforderungen:
 - Konfiguration von Dynamischen Systemen
 - Eingabe von mathematischen Funktionen mit Variablen
 - Visualisierung von beliebigen Simulationen
 - Zusatzfeatures wie Speichern/Laden/Import, Steuermöglichkeiten während einer Simulation

Projektplanung

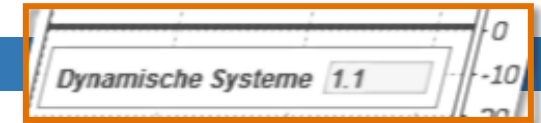


□ Iterationsplanung mit 3 Iterationen

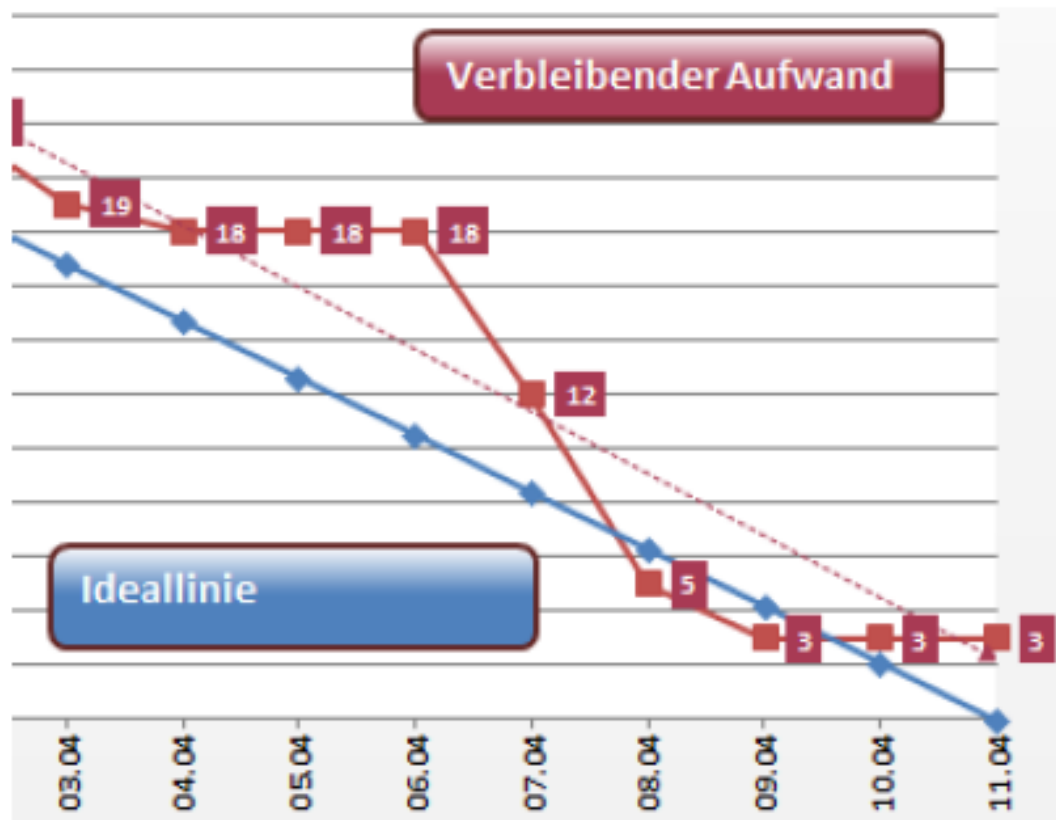
Nr.	Zeitraum	Geplanter Aufwand RK	Geplanter Aufwand DEL	Aufwand total
1	22.03 - 11.04	18.0	23.0	41
2	12.04 - 02.05	5.0	15.0	20
3	03.05 - 24.05	30.0	22.0	52
		53.0	60.0	113

- Taskplanung aufgrund der Anforderungen
- Zuteilung und Priorisierung jeweils vor Iterationsstart
- Burndown-Charts zum Tracken der Fortschritte während einer Iteration

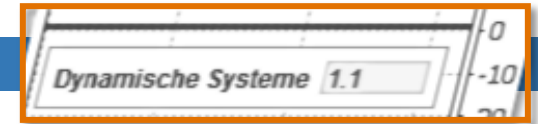
Projektplanung



□ Burndown-Chart: Abschluss Iteration 1

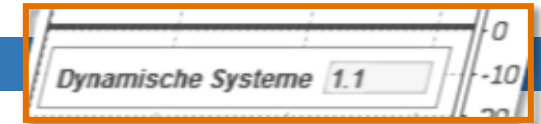


Realisierung



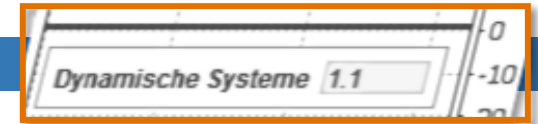
- Tools, Technologien und Frameworks
 - ▣ Java 1.7 (zum Schluss 1.6) und Swing
 - ▣ Eclipse IDE
 - ▣ Git und Github
 - <https://github.com/delsener/ch.zhaw.softwareproj2.git>
 - ▣ Maven 3.0.4
 - ▣ JUnit 4.8.1
 - ▣ JFreeChart 1.0.13
 - ▣ Apache Common JEXL

Realisierung



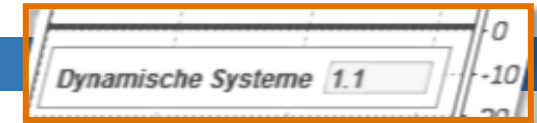
- Projektstruktur / Komponenten
 - ▣ GUI-Komponenten
 - ▷  `ch.zhaw.dynsys.gui`
 - ▣ Logik-/Simulations-Komponenten
 - ▷  `ch.zhaw.dynsys.simulation`
 - ▣ Hilfsklassen für Expression Language (JEXL)
 - ▷  `ch.zhaw.dynsys.el.utils`
 - ▣ Persistenz-Klassen
 - ▷  `ch.zhaw.dynsys.persistence`

Realisierung



- Problemstellung: Eingabe von mathematischen Formeln
 - ▣ **Expression Language:**
 - Unterstützung bei der Interpretation von mathematischen Scripten
 - Scriptsprache
 - Zwei bekannte Java Libraries: Spring Expression Language SpEL, **Apache Common JEXL**

Realisierung

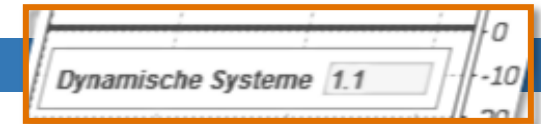


- Problemstellung: Eingabe von mathematischen Formeln

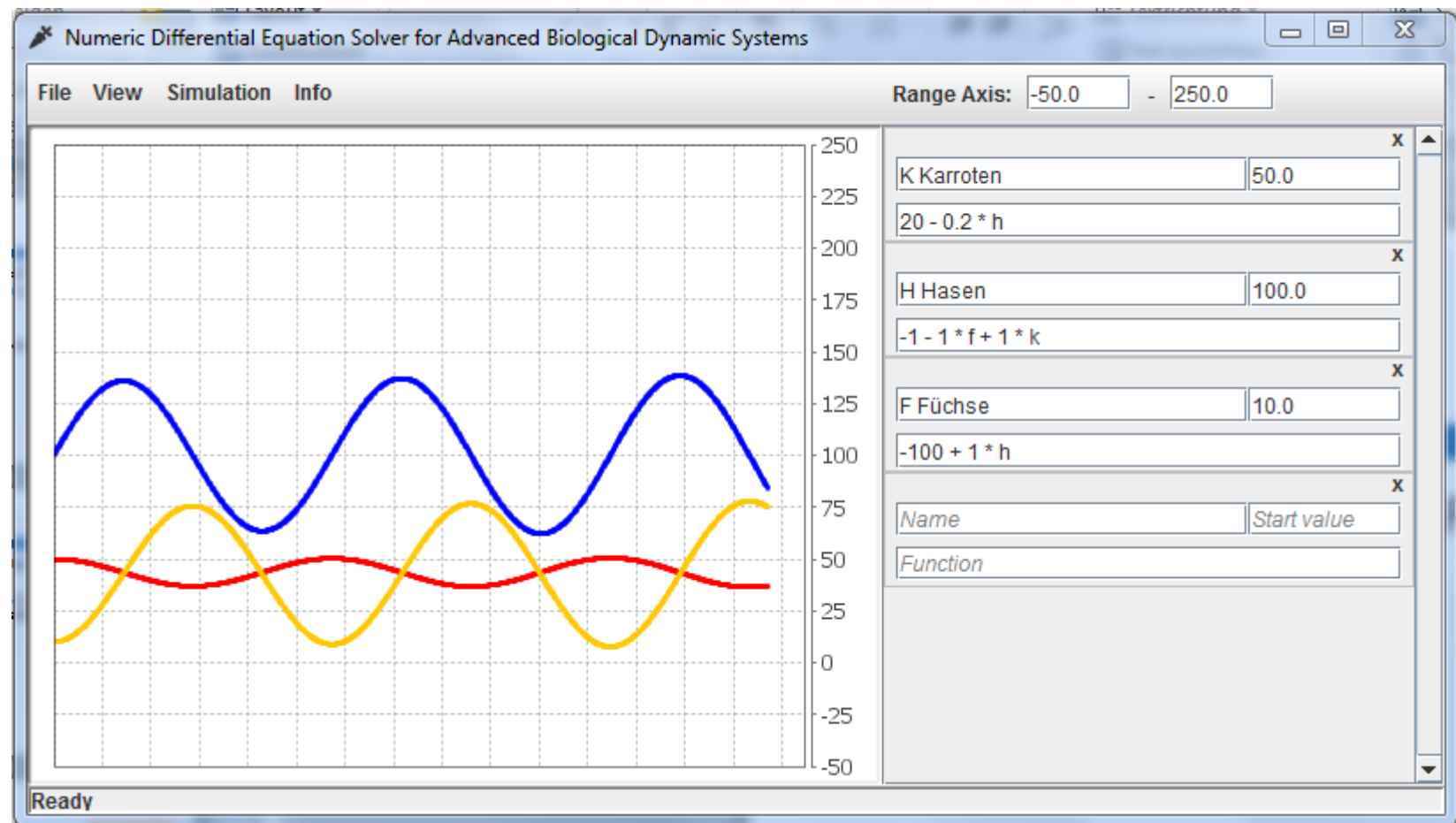
- ▣ Beispiel-Script das von JEXL interpretiert wird:

```
{
var t = -20.0
var t_diff = 0.0
var H = 100.0
var H_diff = 0.0
var Z = 650.0
var Z_diff = 0.0
}
return [
    new("java.lang.Double",5),
    new("java.lang.Double",math:max(-H, 0.0001*H*Z*(20-math:abs(10- t)))),
    new("java.lang.Double",math:min(0, -math:min(Z, H_diff)))
];
```

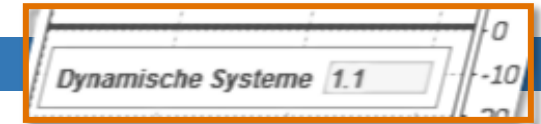
Realisierung



□ Demonstration der GUI-Features



Demonstration



□ Hefe-Zucker-System

▣ Wachstumsfunktionen:

- Wachstum Hefe (H') = $a \cdot Z + b \cdot H$
- Wachstum Zucker (Z') = $c \cdot H + d \cdot H'$

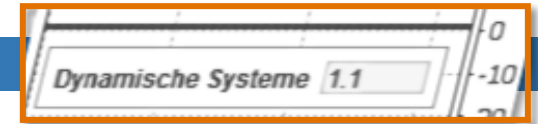
▣ Parametrierung:

- $a = 1, b = 2, c = -1, d = -0.2$

▣ Startwerte:

- Hefe = 1.0
- Zucker = 1.0

Reflexion



- Interessante Projektarbeit
- Applikation entwickeln um ein numerisches Problem zu lösen
- Gute fachliche Unterstützung durch Hr. Heuberger
- Programm funktioniert, ist aber (wie immer) erweiterbar:
 - ▣ z.B. Regelbarkeit der Berechnungsschrittweiten, komplexere grafische Darstellung (weitere Grafen), ...