

# Integration von Physik und Informatik in *Cyber-Physical Systems*

Peter Marwedel

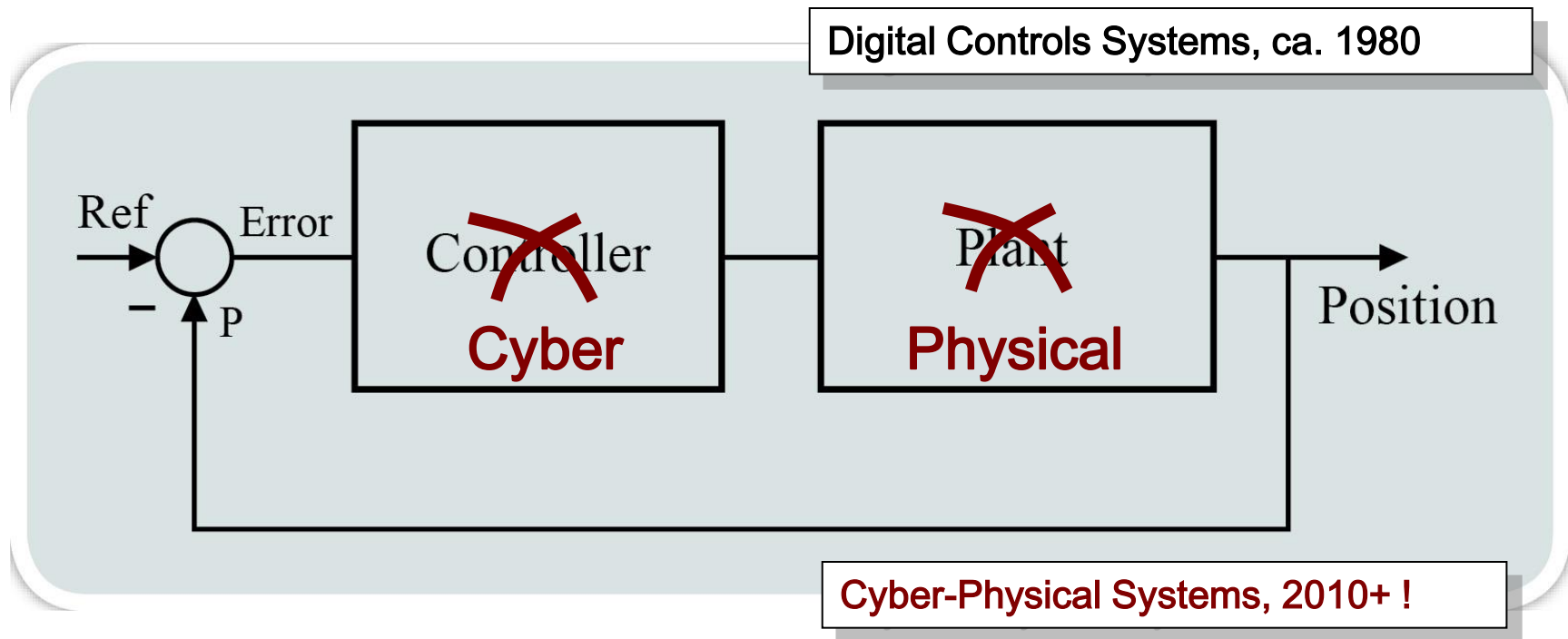
TU Dortmund, Informatik 12



2016年03月14日

Photos/Graphics: P. Marwedel + Microsoft

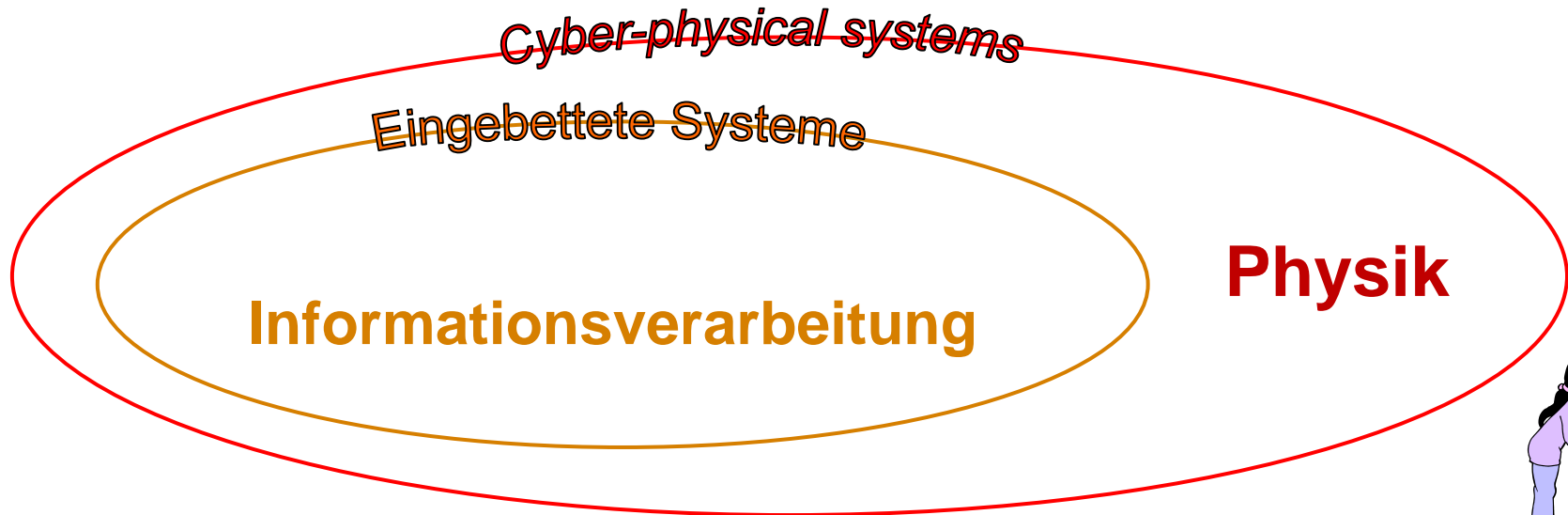
# Was ist ein *Cyber-Physical System*?



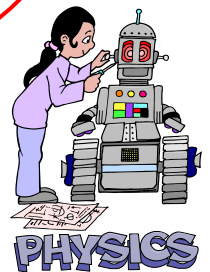
Hier: positivere Betrachtung

# Cyber-physical systems und Eingebettete Systeme

- **Eingebettete Systeme (ES)**: informationsverarbeitende Systeme, die in ein umgebendes Produkt eingebettet sind [Peter Marwedel, 2003]
- **Cyber-physical systems (CPS)**: integrations of computation with physical processes [Edward Lee, 2006].



$CPS = ES + \text{physikalische Umgebung}$



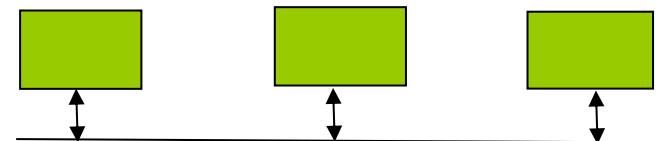
# Definitionen

- **National Science Foundation (NSF):**  
*... (CPS) ... depend upon the **synergy** of **computational** and **physical components**. ..will be **coordinated, distributed, and connected**, ...*
- **Akatech:**  
*... represent **networked, software-intensive embedded systems in a control loop**, ...*
- **Europäische Kommission**  
*... (CPS) refer to **next generation embedded ICT systems** that are **interconnected and collaborating***
- **CPS Hub NRW ..**

[cps-vo.org](http://cps-vo.org)

Akatech: Cyber-Physical Systems.  
<http://www.acatech.de/>

<http://bit.ly/13I7985>

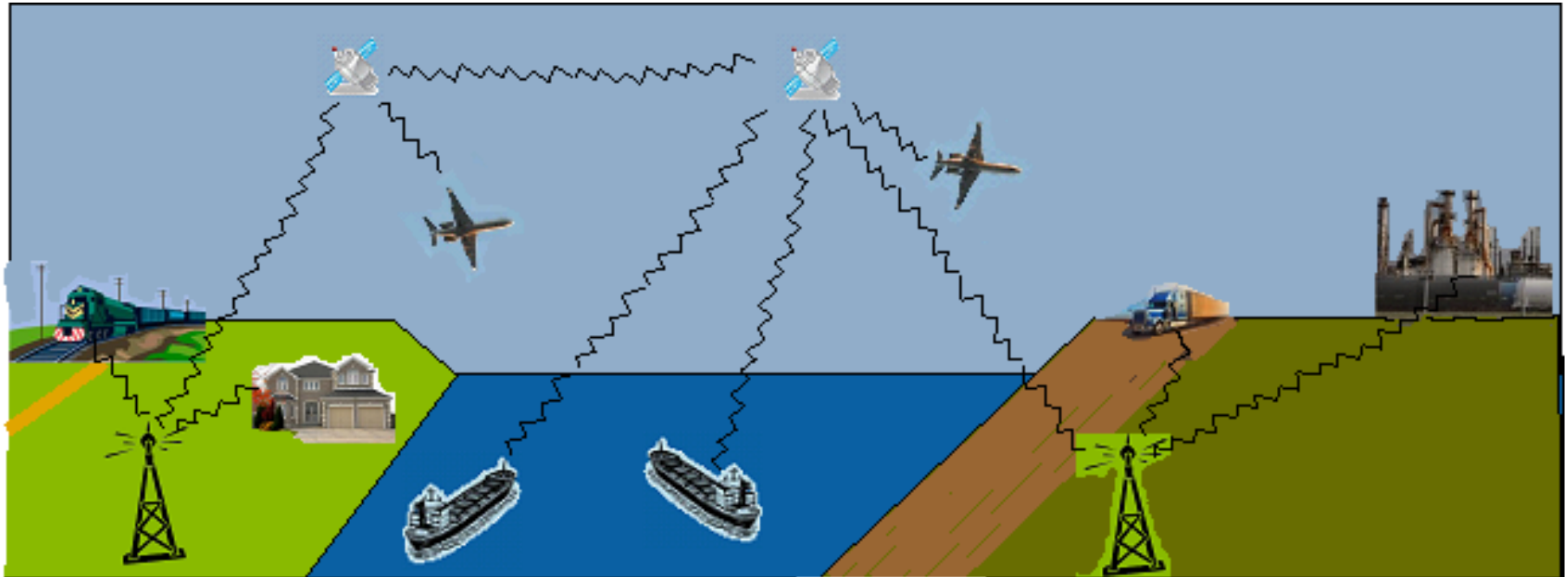


# Gliederung

---

- Definitionen
- ➔ ■ Chancen und Anwendungsbereiche
- Herausforderungen
- Modellierung
  - Behandlung in der Ausbildung in Dortmund
  - *Timed Automata*
  - Modelica
- Zusammenfassung

# Anwendungsgebiete (1): Vernetzte Systeme



# Anwendungsgebiete (2): Verkehrswesen

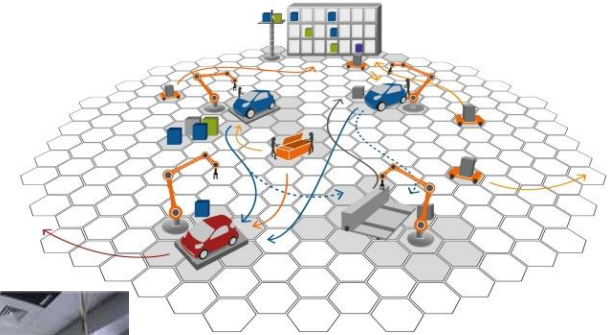
- Automobile
- Luftfahrt
- Eisenbahnen
- Wasserstraßen



Viele eingebettete Systeme, zunehmend vernetzt

# Weitere Anwendungsgebiete (1)

- Fabrikautomatisierung (Industrie 4.0)
- Gesundheitswesen
- Intelligentes Haus
- Robotik





# Weitere Anwendungsgebiete (2)

- Energieversorgung
- Katastrophenschutz
- Öffentliche Sicherheit
- Stabilitätsüberwachung
- Physikalische Experimente



© umweltbundesamt.de



© www.spiegel.de



© P. Marwedel



© CERN

# Globale Sicht auf CPS

---

- Attraktiver Begriff, der fast alle künftigen IT-Anwendungen jenseits der Büroautomatisierung umfasst
- In den USA stärkere Betonung der Integration der Physik
- In Europa etwas stärkere Betonung der Verteiltheit
- Innerhalb Europa's starkes Interesse in Deutschland
- Abgrenzung zu *Internet of Things (IoT)* [Edward Lee]:
  - CPS sind ein Entwurfsproblem,
  - IoT ist eine konkrete Realisierungstechnik

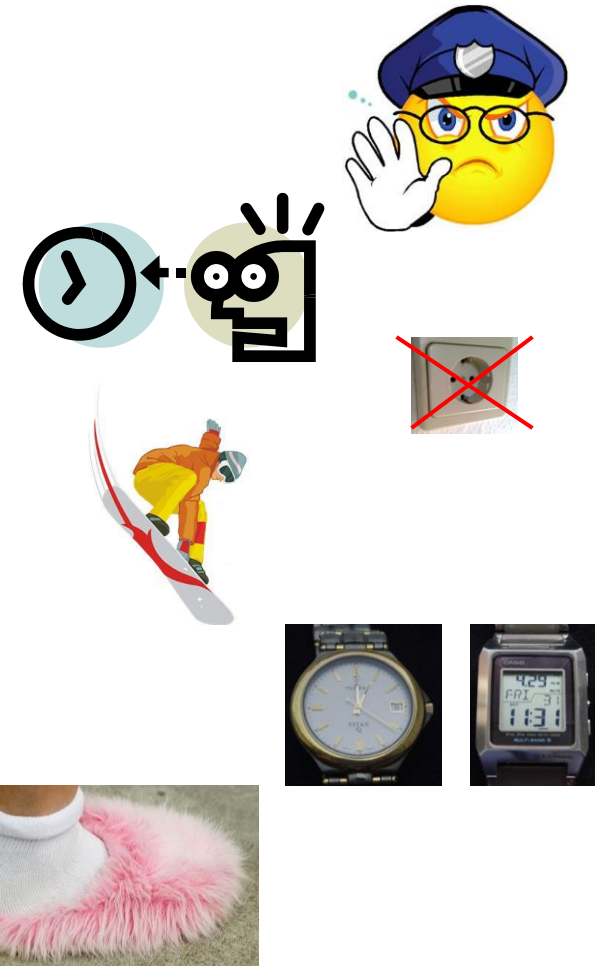
# Gliederung

---

- Definitionen
- Chancen und Anwendungsbereiche
- ➔ ■ Herausforderungen
- Modellierung
  - Behandlung in der Ausbildung in Dortmund
  - *Timed Automata*
  - Modelica
- Zusammenfassung

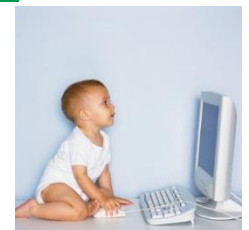
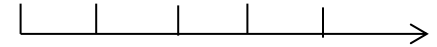
# Herausforderungen (1)

- Sicherheit, Privatheit
- Einhalten von Zeitschranken
- Energieeffizienz
- Dynamik
- Heterogenität
- Unscharfe (*fuzzy*) Verifikation



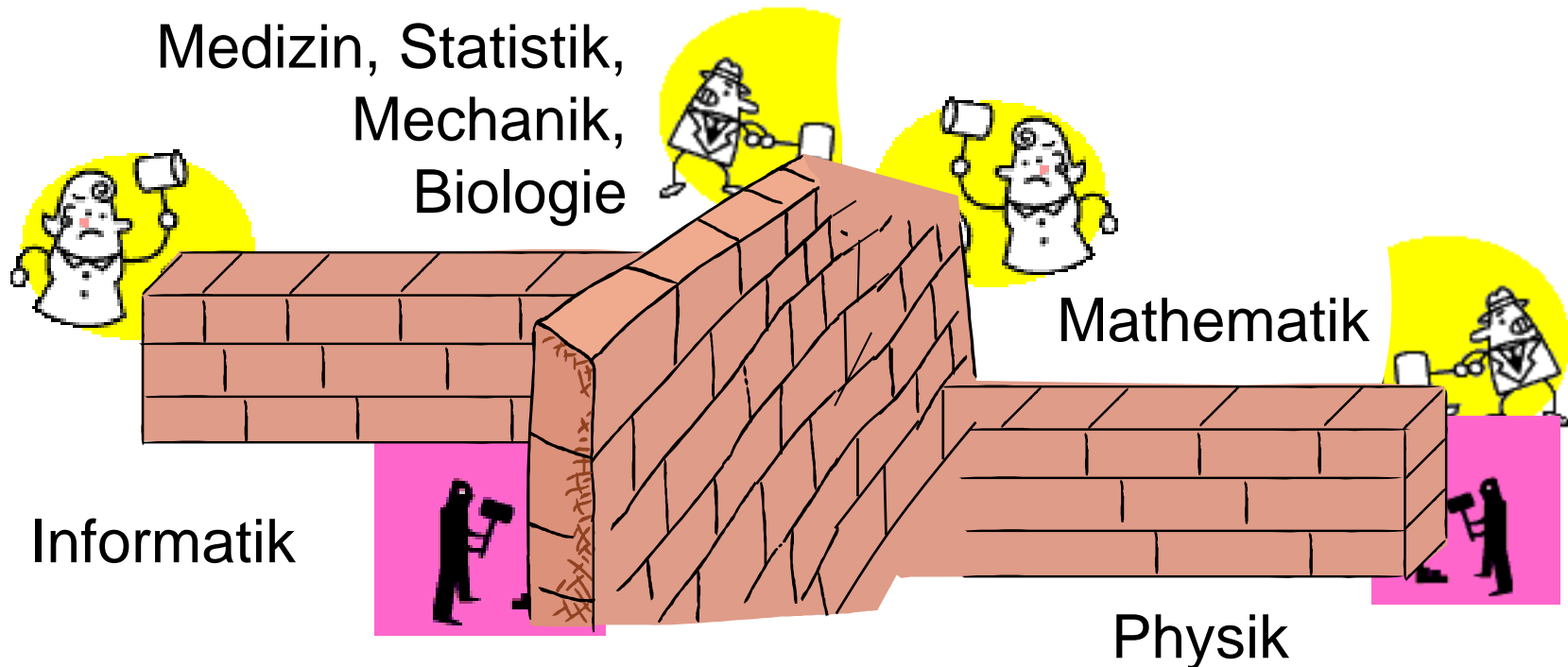
# Herausforderungen (2)

- Periodische Abtastung zu hinterfragen
- Große Datenvolumina
- Rechtliche Fragen  
(Haftbarkeit, Geistiges Eigentum)
- Nutzen für die Gesellschaft
- Akzeptanz



# CPS-Entwurf ist interdisziplinär

Beim Entwurf ist Wissen aus vielen Disziplinen erforderlich  
→ Wände zwischen Disziplinen müssen abgebaut werden.



Im folgenden Fokus auf Integration der Physik

# Reelle vs. Festkommazahlen



## Physik:

- Viele physikalische Größen beschrieben durch ein Paar (reelle Zahl, Einheit)
- Unendlich viele reelle Zahlen

## Cyber:

- Gleitkommazahlen als Approximation
- Endliche Menge an Gleitkommazahlen



## Unpassend!

W. Taha, R. Cartwright: Some Challenges for Model-Based Simulation, The 4th Analytic Virtual Integration of Cyber-Physical Systems Workshop, Dec. 2013, Vancouver

# Herausforderungen: Zeno-Verhalten

## Beispiel:

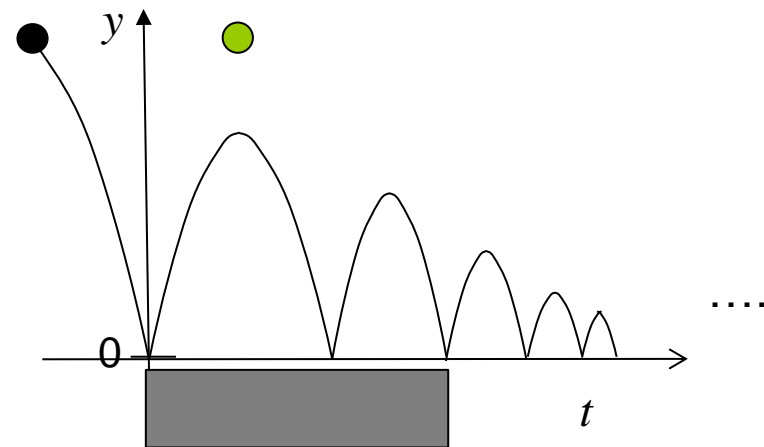
- Teilelastischer Stoß einer Punktmasse

Ideales Model eines Stoßes bei  $t=0$ :

$$y > 0, t < 0: \ddot{y}(t) = -g$$

$$y = 0: \dot{y}(0) = -e \max_{t < 0}(\dot{y}(t))$$

Wiederholte Stöße



- Durch die Dämpfung  $e$  ist jeder Zyklus kürzer als der vorhergehende
- Sei  $\Delta$  ein Zeitintervall
- $\exists$  eine obere Schranke für die Anzahl der Stöße in  $\Delta$ ?

**NEIN**

- Ein System zeigt **Zeno-**Verhalten  $\Leftrightarrow \exists$  unbeschränkte Zahl von Ereignissen in beschränkter Zeit.
- Simulation?



# Gliederung

---

- Definitionen
- Chancen und Anwendungsbereiche
- Herausforderungen
- ➔ ■ Modellierung
  - Behandlung in der Ausbildung in Dortmund
  - *Timed Automata*
  - Modelica
- Zusammenfassung

# Grenzen der üblichen „*Threads*“, Bedeutung von Modellen

---

*Even the core ... notion of “computable” is at odds with the requirements of embedded software.*

*In this notion, useful computation terminates, but termination is undecidable.*

*In embedded software, termination is failure, and yet to get predictable timing, subcomputations must decidably terminate.*

*What is needed is nearly a reinvention of computer science.*

Edward A. Lee: Absolutely Positively on Time, *IEEE Computer*, July, 2005

☞ Suche nach anderen Modellierungsansätzen.

# Einige Berechnungsmodelle

Mehr Details in Büchern und Videos

Kommunikation/ lokale Berechnung.	Gemeinsamer Speicher	Nachrichtenaustausch Synchron   Asynchron	
Undefinierte Komponenten	Einfacher Text, <i>use cases</i>   Sequenzdiagramme		
Differentialgleichung	Modelica, Matlab, ....		
CFSM	StateCharts		SDL
Datenfluss	<i>Scoreboarding + Tomasulo Algor. -&gt; Comp.Architecture</i>		Kahn Netzwerke, SDF
Petri-Netze		C/E nets, P/T nets, ...	
Diskretes Ereignis-(DE) Model	HW-Beschreibungssprache*	i.W. nur experimentelle Systeme	
<i>V. Neumann threads</i>	C, C++, Java	C, C++, Java mit Bibliotheken CSP, ADA	

\* Klassifikation basierend auf Implementierung mit gemeinsamen Datenstrukturen

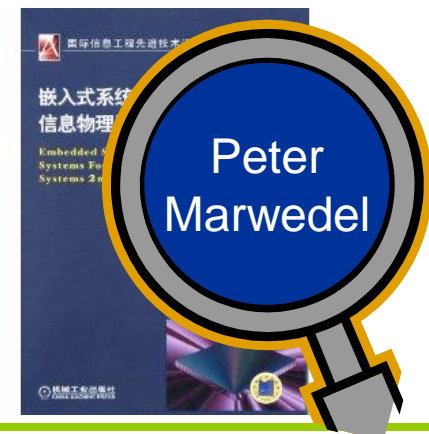
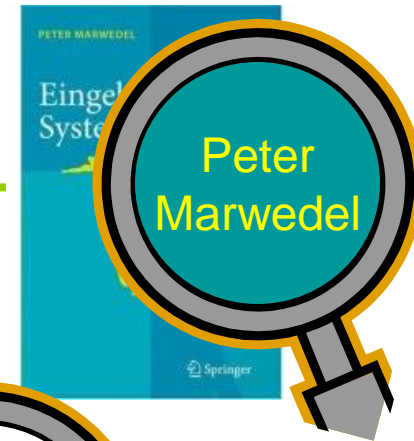
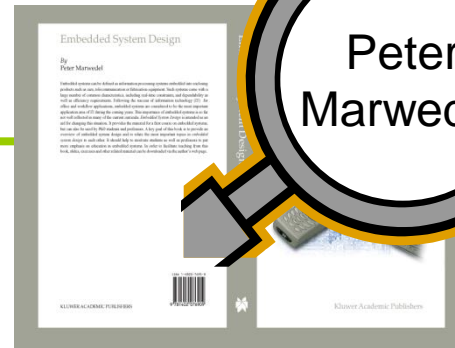
# Lehrbücher

## ■ 1. Ausgabe

- Englisch
- Deutsch, 2003 & 2007
- Chinesisch, 2006
- Mazedonisch, 2010

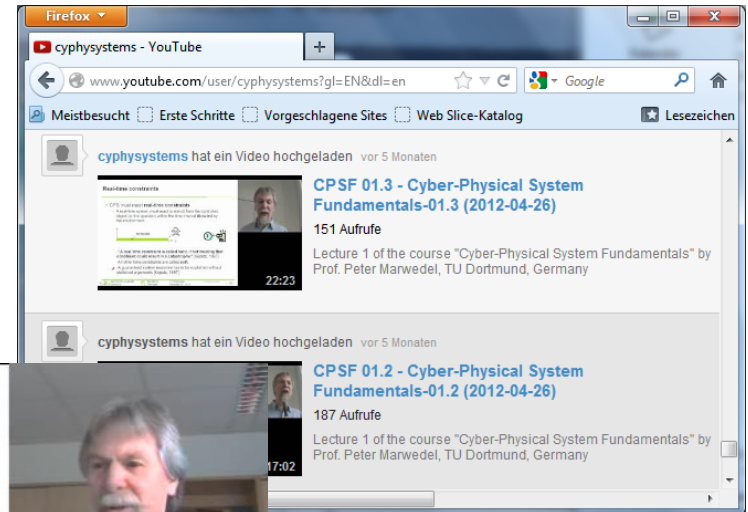
## ■ 2. Ausgabe, mit CPS

- Englisch, Dez. 2010/Jan. 2011
- Übersetzte chinesische Ausgabe, 2013
- Vertrag für deutsche Ausgabe



# Folien und Videos (auf Englisch) verfügbar

Videos verfügbar unter <http://peter.marwedel.eu> oder direkt <http://www.youtube.com/user/cyphysystems>



# Einige Berechnungsmodelle

Kommunikation/ lokale Berechnung.	Gemeinsamer Speicher	Nachrichtenaustausch Synchron   Asynchron	
Undefinierte Komponenten		Einfacher Text, <i>use cases</i>   Sequenzdiagramme	
Differentialgleichung		Modelica, Matlab, ....	
CFSM	StateCharts		SDL
Datenfluss	<i>Scoreboarding</i> + <i>Tomasulo Algor.</i> -> <i>Comp.Architecture</i>		Kahn Netzwerke, SDF
Petri-Netze		C/E nets, P/T nets, ...	
Diskretes Ereignis- (DE) Model	HW-Beschreibungssprache*	i.W. nur experimentelle Systeme	
<i>V. Neumann threads</i>	C, C++, Java	C, C++, Java mit Bibliotheken CSP, ADA	

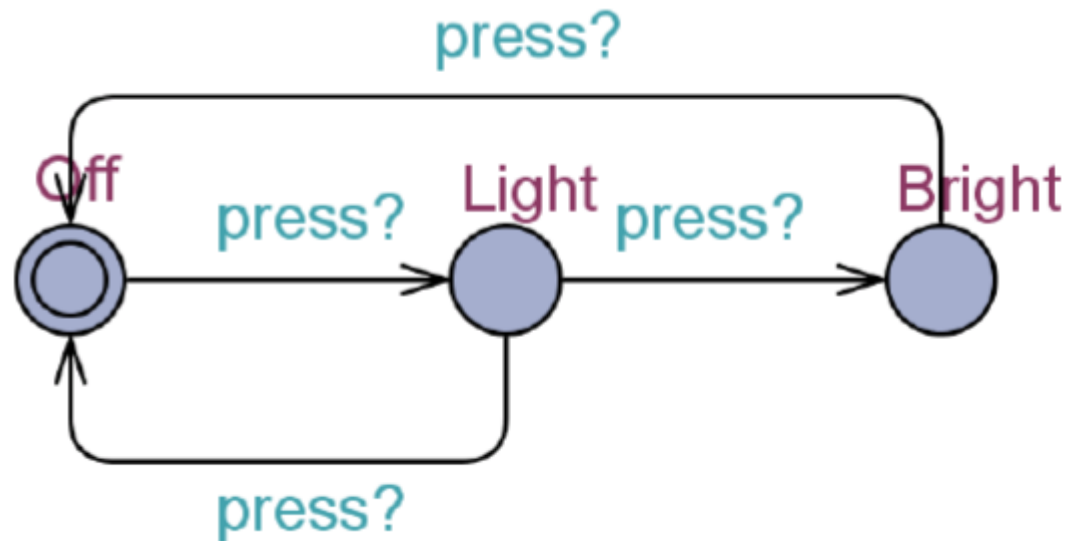
\* Klassifikation basierend auf Implementierung mit gemeinsamen Datenstrukturen

# Gliederung

---

- Definitionen
- Chancen und Anwendungsbereiche
- Herausforderungen
- Modellierung
  - Behandlung in der Ausbildung in Dortmund
- ➔ • *Timed Automata* (Hier: Schwergewicht auf Modellierung)
- Modelica
- Zusammenfassung

# Eine einfache Lichtsteuerung

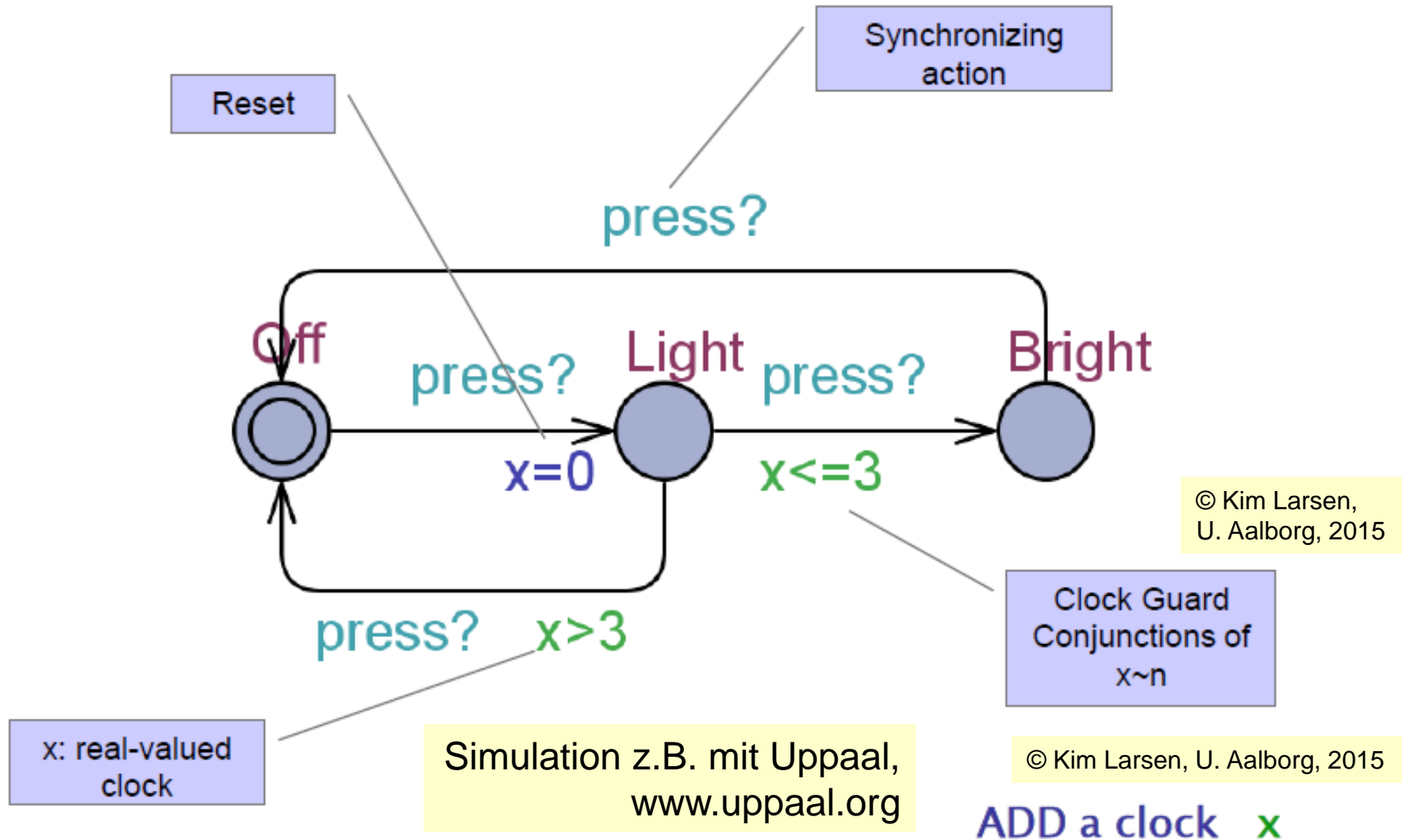


Was heißt „kurz“ nacheinander drücken ergibt helles Licht?

© Kim Larsen, U. Aalborg, 2015



# Timed automata: Schritt in Richtung einer Integration der Zeitmodellierung mit Taktzählern



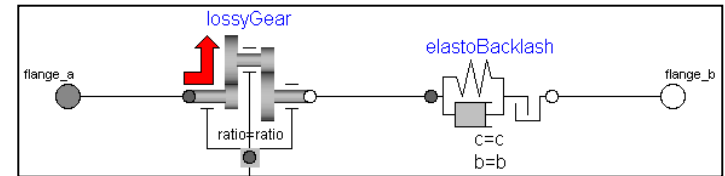
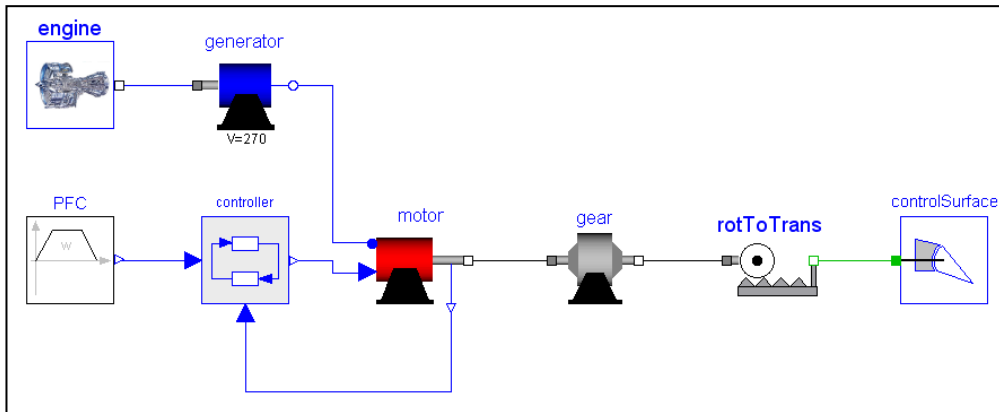
# Gliederung

---

- Definitionen
- Chancen und Anwendungsbereiche
- Herausforderungen
- Modellierung
  - Behandlung in der Ausbildung in Dortmund
  - *Timed Automata*
- ➔ • Modelica
  - Zusammenfassung

# Modelica

Sprache zur Modellierung von gemischten IT- und physikalischen Systemen



$$0 = \mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{y}(t), t)$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$$

$$\mathbf{y}(t) = \mathbf{f}(\mathbf{x}(t), t)$$

- Modellierung über Gleichungssysteme
- Abbildung auf numerische Integration

```
equation
    0 = flange_a.tau + flange_b.tau;
    phi_rel = flange_b.phi - flange_a.phi;
    w_rel = der(phi_rel);
    ...
```

<http://openmodelica.org>

# Modelica-Modell des teilelastischen Stoßes

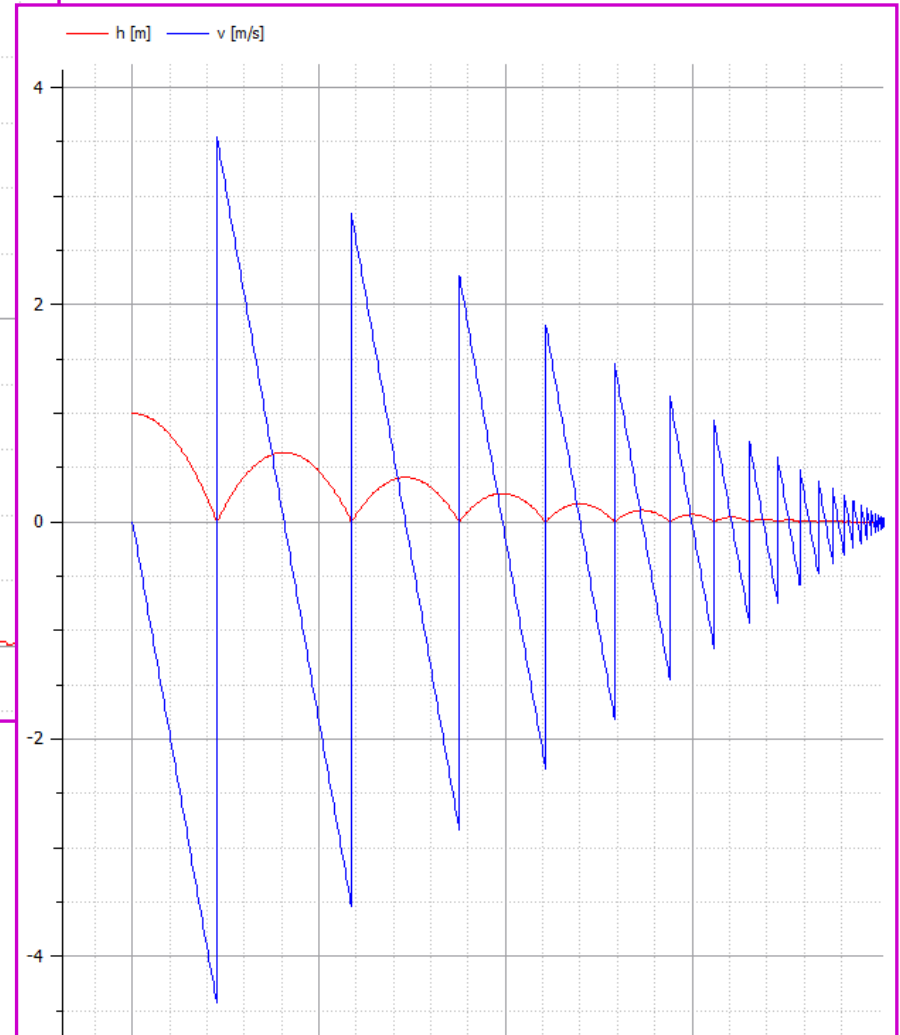
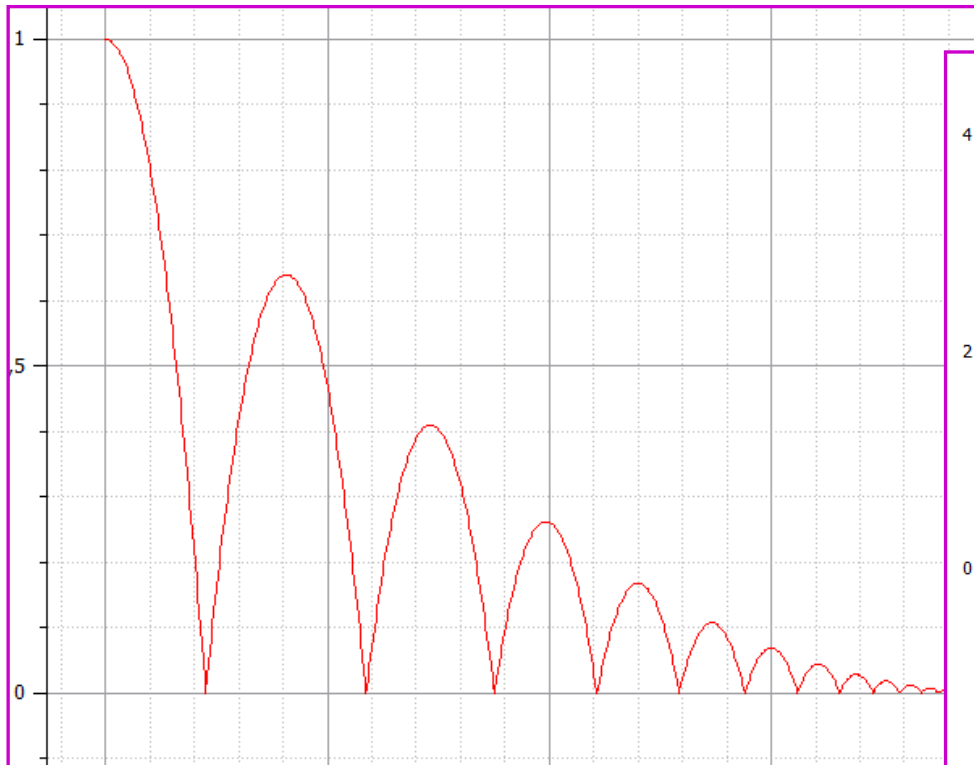
```
model Ball
  type Height = Real(unit = "m");
  type Velocity = Real(unit = "m/s");
  parameter Real e = 0.8 "Restitution";
  parameter Height h0 = 1.0 "Initiale Höhe";
  Height h;
  Velocity v;
initial equation
  v = 0.0;
  h = h0;
equation
  v = der(h);
  der(v) = -9.81;
  when {h <= 0.0} then
    reinit(v, -e * v);
  end when;
end Ball;
```

System von Differentialgleichungen, kann mittels Gleichungslösern numerisch gelöst werden

Wenn die Bedingung erfüllt ist, wird ein Ereignis erzeugt, welches Variable neu initialisiert

In Anlehnung an <http://book.xogeny.com/behavior/discrete/bouncing/>

# Modelica: Ergebnisse

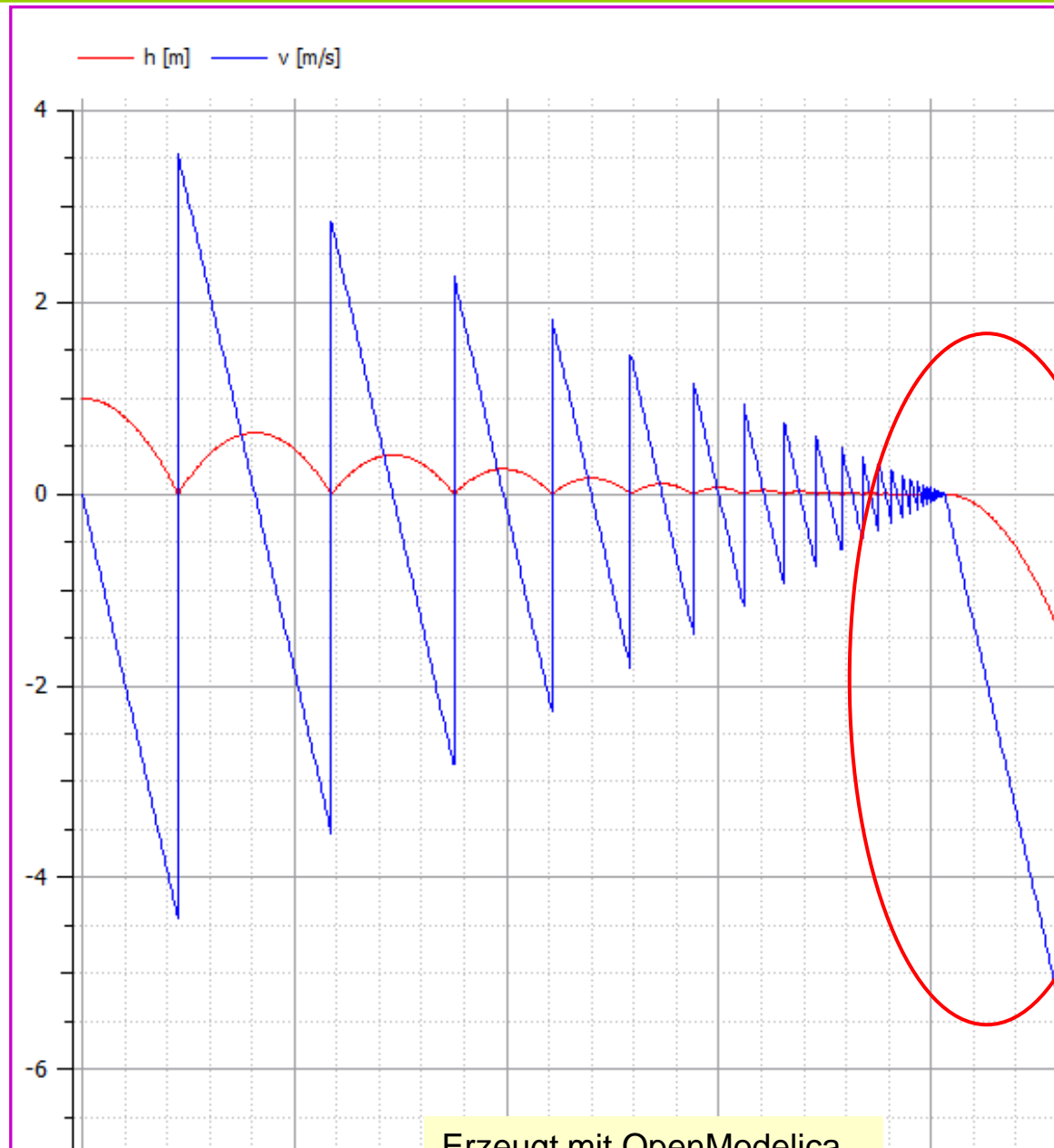


Erzeugt mit OpenModelica  
Connection Editor v 1.9.3

# Modelica: Konsequenzen von Zahlendarstellung, Simulationsmodell und Zeno-Effekt

## Der Ball durchschlägt den Tisch!

Für  $h \approx 0$  wird das Ereignis  
“ $h$  wird  $\leq 0$ ” nicht mehr  
erzeugt, deswegen entfällt  
die Umkehrung von  $v$ .



Erzeugt mit OpenModelica  
Connection Editor v 1.9.3

# Modelica: klebriger Ball

model KlebrigerBall

```
type Height = Real(unit = "m");  
type Velocity = Real(unit = "m/s");  
parameter Real e = 0.8 "Restitution";  
parameter Height h0=1.0 "Initiale Höhe";  
constant Velocity eps=1e-3 "kleine Geschwindigkeit";  
Boolean klebt;
```

Height h;

Velocity v;

initial equation

```
v = 0.0;
```

```
h = h0;
```

```
klebt = false;
```

equation

```
v = der(h);
```

```
der(v) = if klebt then 0 else -9.81;
```

```
when {h <= 0.0} then
```

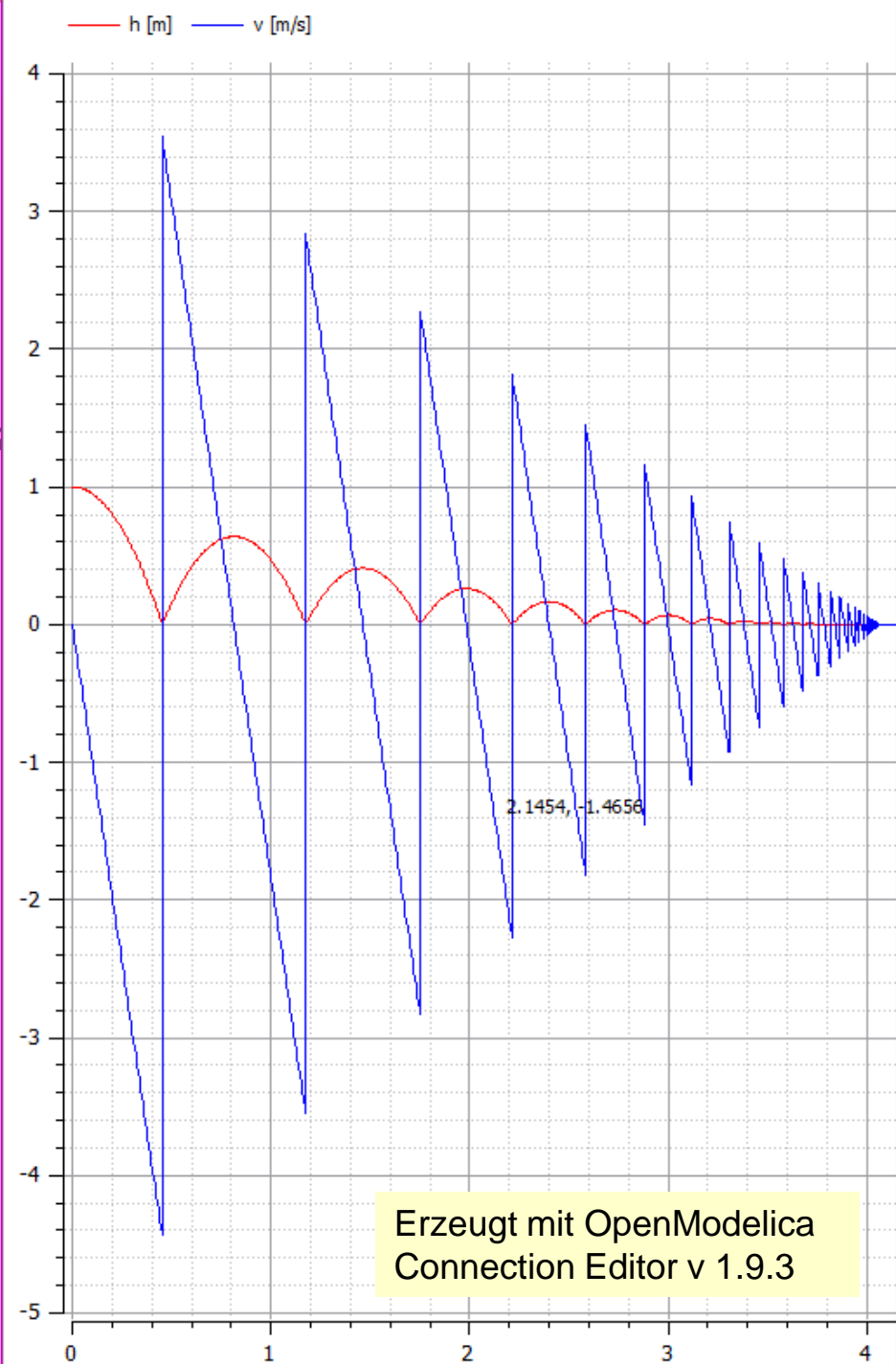
```
  klebt = abs(v) < eps;
```

```
  reinit(v, if klebt then 0 else -e*v);
```

```
end when;
```

```
end KlebrigerBall;
```

Bei  $|v| < \text{eps}$  wird die effektive Beschleunigung auf 0 gesetzt



# Zusammenfassung

---

- Bei künftigen IT-Systemen Trend zur Integration mit Gegenständen des realen Lebens → *CPS*
- *CPS* ist attraktiver Begriff, der eine Vielzahl von IT-Anwendungen abdeckt.
- Viele Chancen, aber auch viele Herausforderungen, hier dargestellt anhand der Integration mit der Physik
- Chancen für eine attraktive Lehre (auch mit HW!)
- Mögliche Berechnungsmodelle beinhalten
  - *timed automata*
  - Differentialgleichungen, z.B. mit Modelica beschrieben