



Institut für Informatik
Automatisierung und Robotik



University of Applied Sciences

Fachhochschule
EMDEN • LEER

Leicht konfigurierbare Hardware-Abstraktionsschicht für flexible Automatisierung

Kristian Virkus, Matthias Barkhoff

**Uwe Schmidtman, Gerhard Kreuz,
Tobias Stöckmann, Marcel Jovic**

präsentiert auf der Echtzeit 2009
19.-20. November 2009 @ Boppard am Rhein, Deutschland

Leicht konfigurierbare Hardware-Abstraktionsschicht für flexible Automatisierung

Einleitung

Motivation

Lösungsansatz

Architektur

Abstraktion

Beispiel

Echtzeit

Fazit & Ausblick



The background of the slide features a close-up, slightly blurred view of white fabric, possibly a sheet or a piece of clothing, with soft, blue-tinted shadows and highlights that create a sense of depth and texture. The overall color palette is light and airy, with the white fabric contrasting against the pale blue shadows.

Einleitung

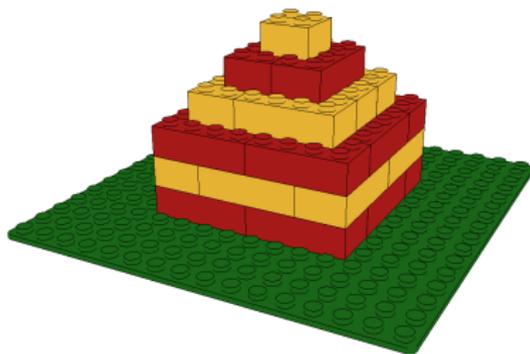
Ziele der “Industrial Automation 2020”

- Innovative Produkte
- Flexibel
- Rekonfigurierbar
- Intelligent
- Umweltfreundlich

Zieldefinition

LEGO-Häuser bauen

- vom Kunden anpassbar
- voll-automatisierte Produktion
- kosten- und energieeffizient



Lösungsansatz

Einsatz eines Agentensystems. Agenten...

- sind Software-Komponenten
- erfüllen eine spezielle Aufgabe
- fällen selbstständig Entscheidungen, basierend auf ihren Erfahrungen
- sind on-the-fly austauschbar
- kommunizieren miteinander
- verbergen ihre Komplexität hinter definierten Schnittstellen

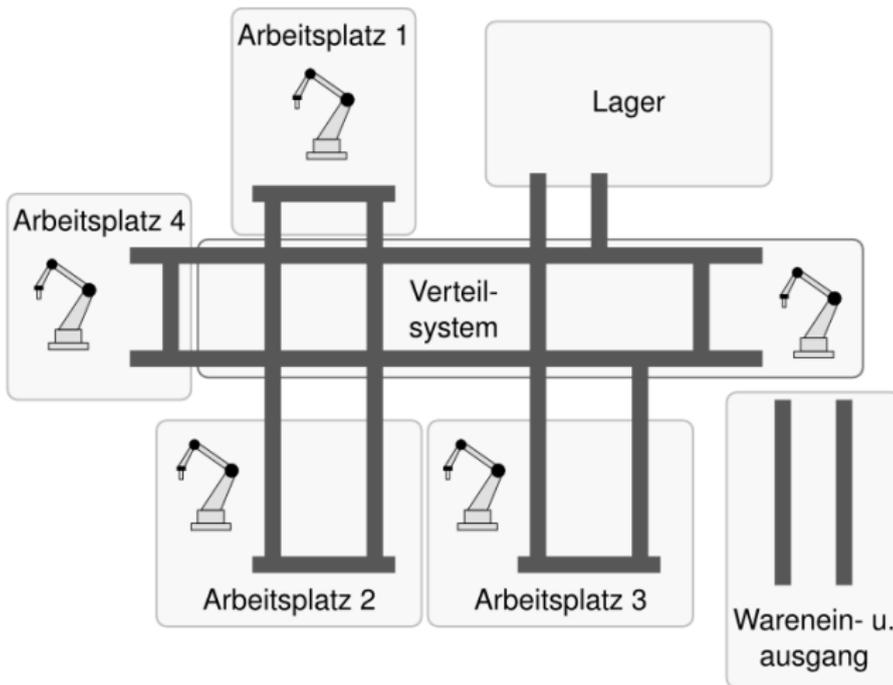
Wie entwickeln und testen wir Agenten?

In einer „echten“ Fabrik!





Layout der Digitalen Fabrik

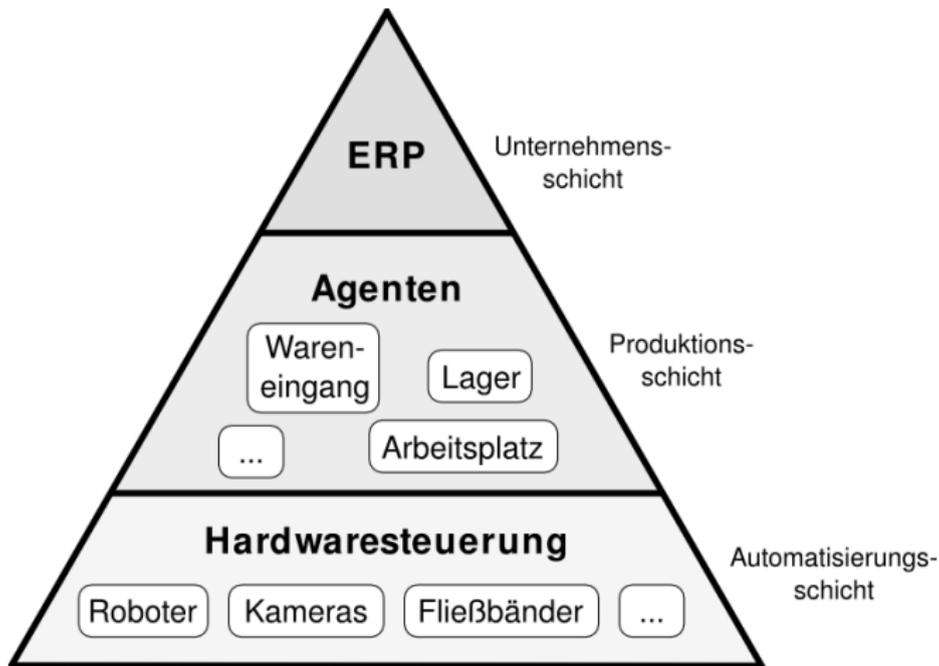


Komponenten der Digitalen Fabrik

- Fließbänder
- Roboter
- Hochregallager
- Kameras
- RFID Sensoren
- Pneumatische Aktoren
- ...

⇒ **viele unterschiedliche I/Os über verschiedene Feldbusse**

Einordnung in die Automatisierungspyramide





Architektur

Wie macht man die Hardware zugänglich?

Einführung einer **Hardware-Abstraktionsschicht**

- physikalische Adressen und Feldbusse in Symbolen abstrahieren (Konfiguration)
- ermöglicht feingranularen Zugriff über Microcodes
- Definition von Tasks, bestehend aus Microcodes
- Taskausführung in Echtzeit
- parallele Ausführung mehrerer Tasks
- zugänglich über textbasiertes TCP/IP-Protokoll

Der Stub als Teil der Hardware-Abstraktionsschicht

- bildet Symbole in echte Hardware-Adressen ab
- als einzige Software hardwarespezifisch
- wird in der Feldebene auf IPCs oder Embedded Systemen ausgeführt
- Hardware-Kommunikation über RTAI (Linux)
- speichert Statistiken in einer Datenbank

Wie wird der Stub benutzt?

1. Telnet-Sitzung zu Stub aufbauen

Wie wird der Stub benutzt?

1. Telnet-Sitzung zu Stub aufbauen
2. zu verwendende Symbole definieren

Wie wird der Stub benutzt?

1. Telnet-Sitzung zu Stub aufbauen
2. zu verwendende Symbole definieren
3. Tasks definieren

Wie wird der Stub benutzt?

1. Telnet-Sitzung zu Stub aufbauen
2. zu verwendende Symbole definieren
3. Tasks definieren
4. Tasks starten

Beispiel: Transportation

Bewege einen Bauteilträger von A nach B

Beispiel: Transportation

Bewege einen Bauteilträger von A nach B

exclusive Fließbandmotor1

exclusive HaltSensor1

Beispiel: Transportation

Bewege einen Bauteilträger von A nach B

```
exclusive Fließbandmotor1
```

```
exclusive HaltSensor1
```

```
begin TransportAB
```

```
code 0 set Fließbandmotor1 100 0
```

```
code 0 wait HaltSensor1 1 10000000000
```

```
code 0 set Fließbandmotor1 0 0
```

```
end
```

Beispiel: Transportation

Bewege einen Bauteilträger von A nach B

```
exclusive Fließbandmotor1
```

```
exclusive HaltSensor1
```

```
begin TransportAB
```

```
code 0 set Fließbandmotor1 100 0
```

```
code 0 wait HaltSensor1 1 10000000000
```

```
code 0 set Fließbandmotor1 0 0
```

```
end
```

```
run test TransportAB
```

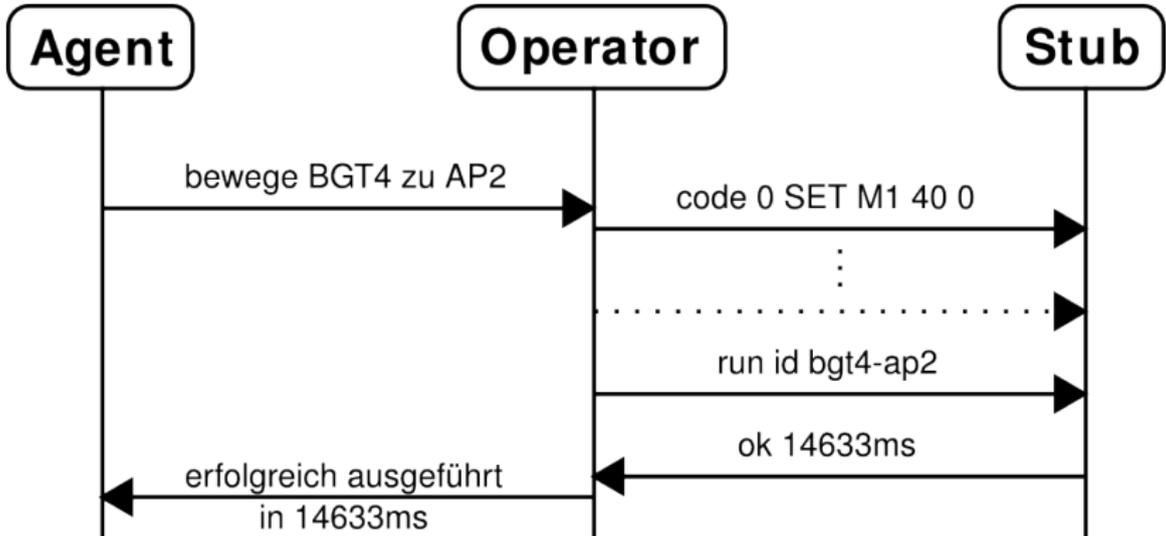
Weitere Abstrahierung

- Microcode immer noch zu komplex für Agenten
- Einführung der Operatoren:
 - Aufgaben zusammenfassen → Dienste anbieten
 - Transport,
 - Montage, ...
 - arbeiten nur auf explizite Anweisung
 - Kommunikation über XML-basiertes Protokoll

Der Pfad-Operator

- findet kürzeste Wege
- verwaltet gleichzeitig mehrere Bauteilträger
- überwacht die Positionen der Bauteilträger
- Wege sind einfach über Konfiguration zu ändern

Kommunikation



Schichten



Echtzeit

Der Stub gliedert sich in zwei Teile:

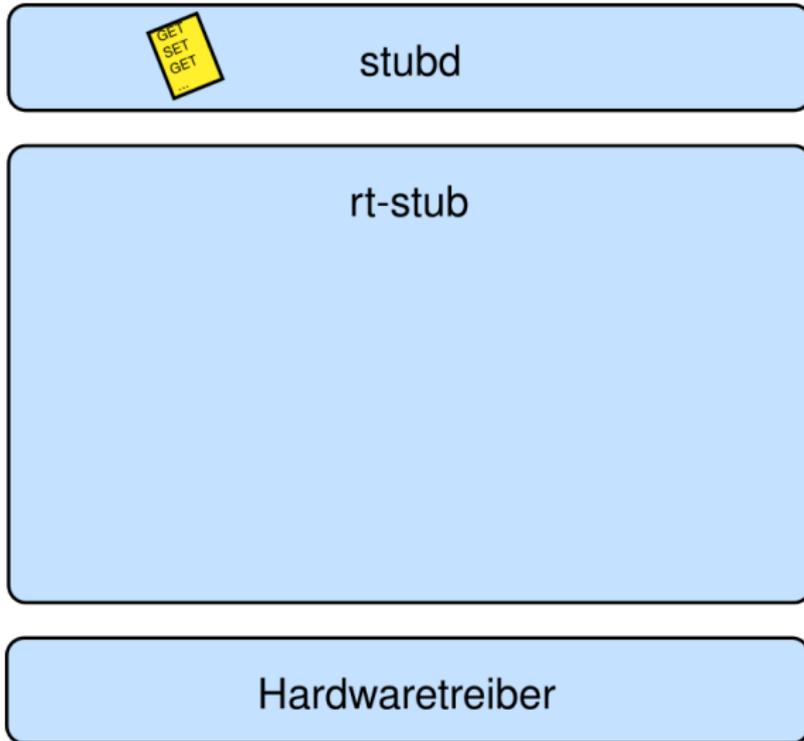
1. rt-stub – RTAI-Linux-Kernelmodul zur Hardwaresteuerung in Echtzeit

Echtzeit

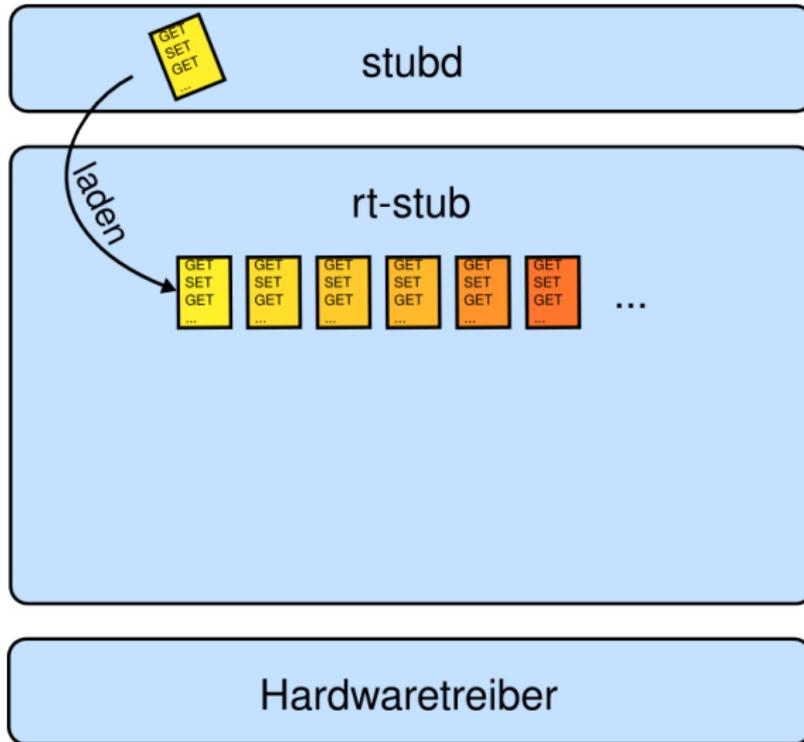
Der Stub gliedert sich in zwei Teile:

1. rt-stub – RTAI-Linux-Kernelmodul zur Hardwaresteuerung in Echtzeit
2. stubd – Userland-Anwendung
 - Anwendungsschnittstelle zur Hardware
 - Symbolverwaltung und Zugriffsrechte
 - Behandlung von Verbindungsabbrüchen
 - fertigt Statistiken an
 - Taskvorverarbeitung
 - Download von Tasks in den rt-stub

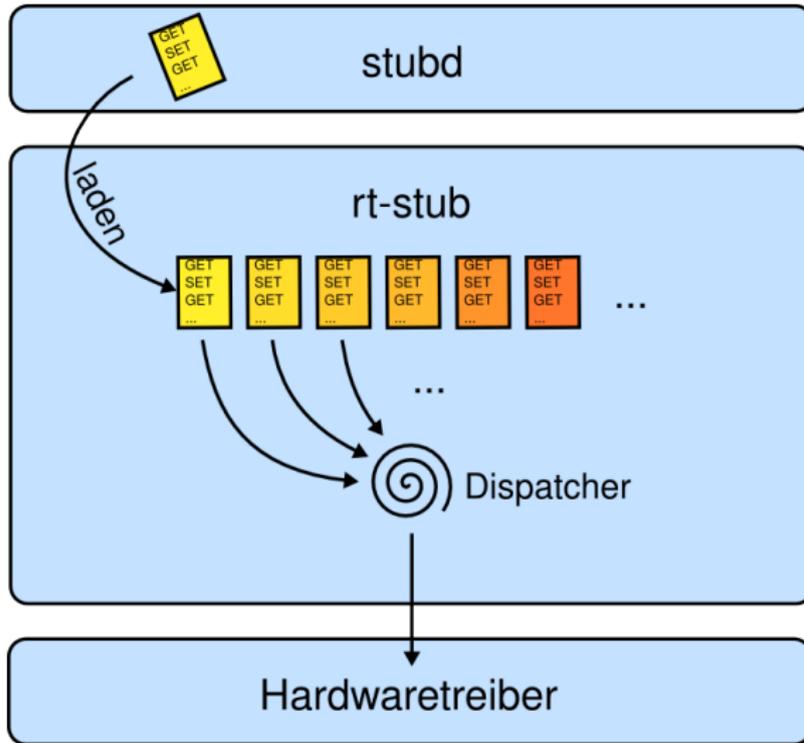
Stub Interna – Schritt 1



Stub Interna – Schritt 2



Stub Interna – Schritt 3



Fazit

Die vorgestellte Hardware-Abstraktionsschicht

- ermöglicht Agenten einfachen Zugriff auf die Hardware
- eignet sich für echtzeitkritische Befehlssequenzen
- ist wiederverwendbar und verteilbar
- liefert Ausführungsstatistiken (Dauer und Energieverbrauch)

Ausblick

- Statistiken zur Optimierung nutzen
- Unterstützung weiterer Feldbusse
- Flexibilität der Tasks erhöhen
- Konfigurationserstellung vereinfachen
- zahlenmäßiger Vergleich mit klassischen Ansätzen
- Agentensystem verfeinern
- ...



Institut für Informatik
Automatisierung und Robotik



University of Applied Sciences

Fachhochschule
EMDEN • LEER

Vielen Dank für Ihre Aufmerksamkeit!

Besuchen Sie uns auf der SPS/IPC/Drives: Halle 7A, Stand 361

In Kooperation mit



Bundesministerium
für Bildung
und Forschung



University of Applied Sciences

Fachhochschule
EMDEN • LEER



Fachhochschule Osnabrück
University of Applied Sciences

www.i2ar.de

Quellen:

- Titel und kapiteltrennende Grafiken basieren auf einer Illustration von Don_Gato (LoFi Photography).