



Hasso
Plattner
Institut

IT Systems Engineering | Universität Potsdam



Entfernte und virtuelle Experimente im Distributed Control Lab - Integration in ein transnationales Netzwerk

Dipl.-Inf. Andreas Rasche

Fachgebiet „Betriebssysteme und Middleware“

Prof. Dr. Andreas Polze

- | Fachgebiet "Betriebssysteme und Middleware" am HPI
- | Entfernte Experimente über das Internet: Distributed Control Lab
- | Experimente im Distributed Control Lab
 - ? Lego Mindstorm Roboter
 - ? "Hau-den-Lukas"
 - ? Foucaults Pendel
- | Problematik "Bösartige Programme"
- | Vet-Trend: Ein transnationales Projekte zur Integration virtueller und entfernter Experimente
- | Erfahrungen & Zusammenfassung

Architektur von Betriebssystemen

- | Windows, Mac OS X, BSD Unix, Solaris

Architektur von komponentenbasierten verteilten Systemen

- | Entwicklung verteilter Anwendungen mit Middleware
- | CORBA, J2EE, COM+, .NET, Web-Services

Betriebssysteme für eingebettete und mobile Systeme

- | Windows CE, .NET Compact Framework
- | Echtzeitbetriebssysteme (LynxOS, VxWorks, QNX)

Paradigmen, Entwurfsmuster und Implementierungsstrategien

- | Vorhersagbarkeit in verteilten und eingebetteten Systemen
- | Performance, Fehlertoleranz und Rechtzeitigkeit

Middleware für vorhersagbare Systeme

- | Paradigmen, Entwurfsmuster, Implementierungsstrategien
- | Rechtzeitigkeit, Sicherheit, Fehlertoleranz

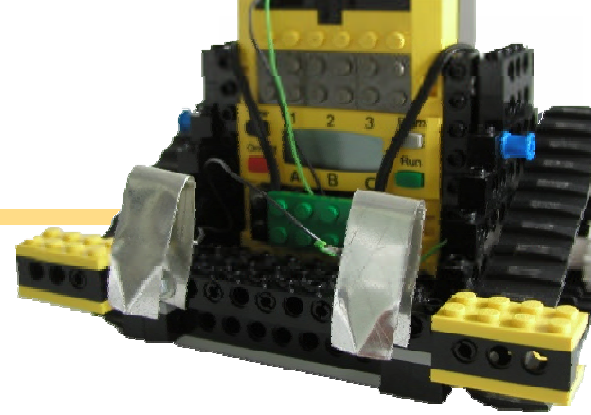
Verteilte Systeme

- | Dynamische Ressourcennutzung in heterogenen dienstorientierten Systemen
- | Standardisierung in Grid-Umgebungen

Eingebettete Systeme

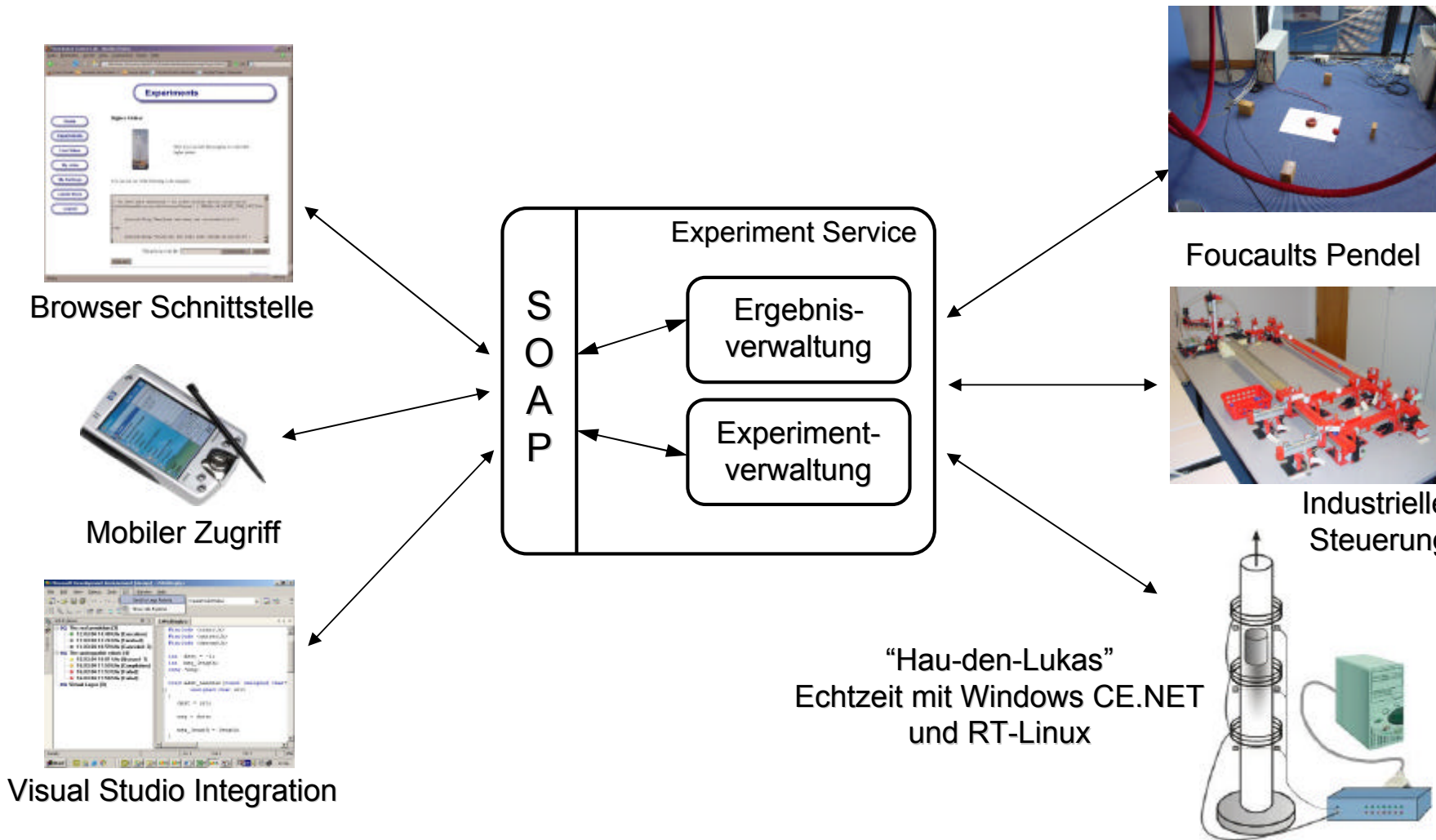
- | Analytische Redundanz und online Ersetzung
- | Experiment-basierte e-Learning Umgebungen
- | Dynamische Rekonfiguration von Komponentensystemen
- | Aspekt-orientierte Programmierung mit C# und .NET

Das Distributed Control Lab



- | Projektstart 2001 am HPI
- | Entfernte Nutzung kostenintensiver Experiment-Hardware über das Internet
- | Praktika zur Programmierung eingebetteter Steuerungssysteme
- | Erweiterbare Architektur zur Verwaltung physikalischer und virtueller Experimente
- | Testumgebung für die Verbindung von Middleware-Komponenten und eingebetteten Systemen
- | Studium von Vorhersagbarkeit, Verfügbarkeit und Sicherheit in Middleware-basierten dynamischen Steuerungssystemen
- | Kontrollierte Ausführung von fremden ("böartigen") Programmen

Das Distributed Control Lab



Experiments

Home

Experiments

Live Video

My Jobs

My Settings

Latest News

Logout

Higher Striker



Here you can enter the program to control the higher striker.

You can use one of the following code examples :

```
// We have hard deadlines - no other thread should interruptus
if(SetThreadPriority(GetCurrentThread( ),THREAD_PRIORITY_TIME_CRITICAL
))
{
    fprintf(flog,"Realtime has been set successfully\n");
}
else
{
    fprintf(flog,"Could not set real time thread priority\n");
}
```

Upload your code file :

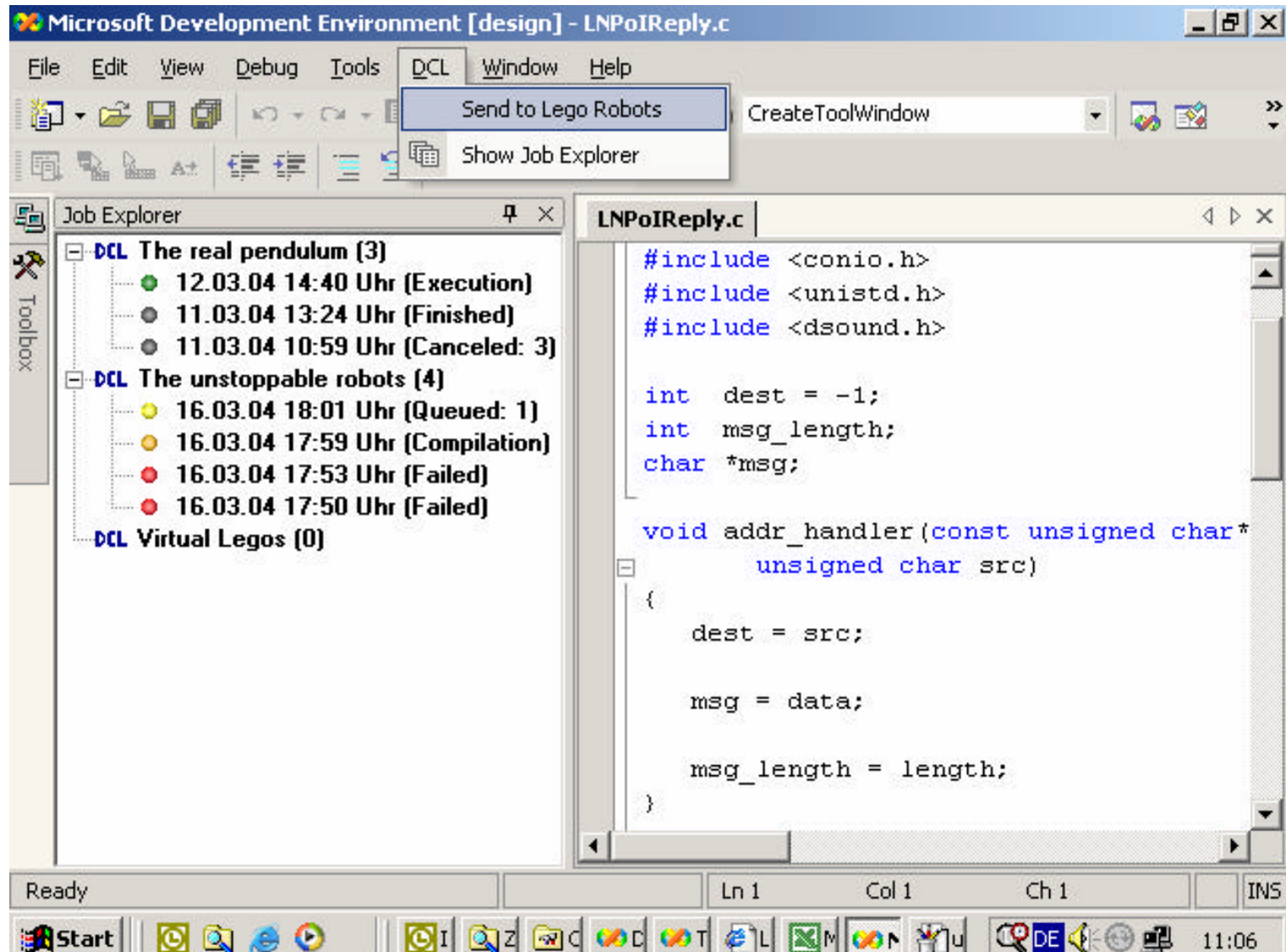
My Jobs

The following jobs are registered for your account in the system :

Name	Location	State	Queued on		
Hau den Lukas - Perform API (Owner : Andreas Rasche)	HPI	Finished	Thu, 09 Nov 2006 15:50:11 GMT	View details	Delete
Hau den Lukas - Perform API (Owner : Andreas Rasche)	HPI	Finished	Thu, 09 Nov 2006 15:49:54 GMT	View details	Delete
Hau den Lukas - Perform API (Owner :)	HPI	Finished	Thu, 09 Nov 2006 15:41:03 GMT	View details	Delete
Hau den Lukas - Perform API (Owner :)	HPI	Finished	Thu, 09 Nov 2006 14:58:25 GMT	View details	Delete
Hau den Lukas - Perform API (Owner : Andreas Rasche)	HPI	Finished	Thu, 09 Nov 2006 14:57:42 GMT	View details	Delete
Hau den Lukas - Perform API (Owner :)	HPI	Failed	Thu, 09 Nov 2006 14:02:47 GMT	View details	Delete
Hau den Lukas - Perform API (Owner :)	HPI	Finished	Thu, 09 Nov 2006 11:52:35 GMT	View details	Delete
Hau den Lukas - Perform API (Owner :)	HPI	Finished	Thu, 09 Nov 2006 11:49:53 GMT	View details	Delete
Hau den Lukas - Perform API (Owner :)	HPI	Finished	Thu, 09 Nov 2006 11:03:51 GMT	View details	Delete
Hau den Lukas - Perform API (Owner : Andreas Rasche)	HPI	Finished	Thu, 09 Nov 2006 09:49:14 GMT	View details	Delete

Delete all entries

Visual Studio Integration



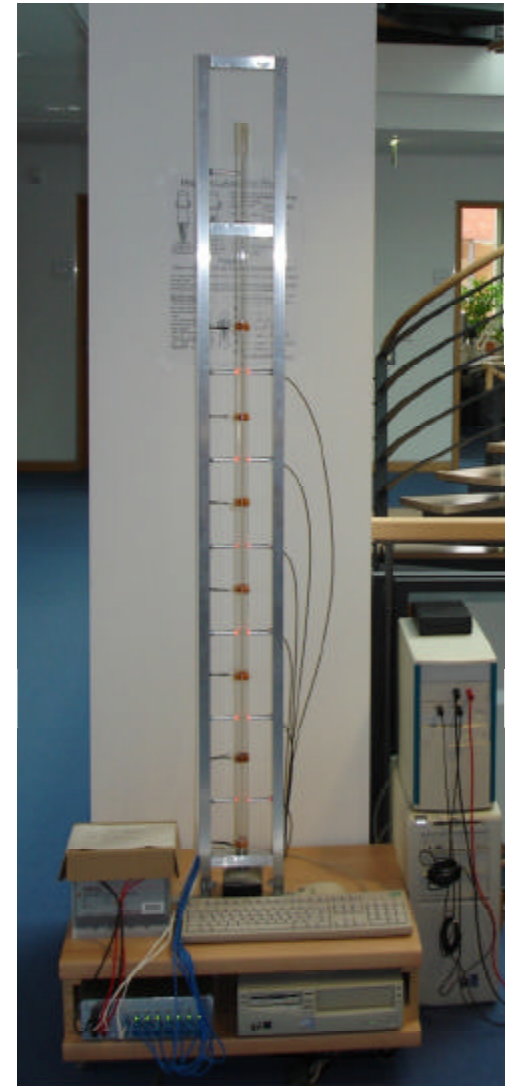
Das „Hau-den-Lukas“ Experiment

Lernziele

- | Benutzung von Echtzeitbetriebssystemen
- | Zeitanalysen von Steuerungsalgorithmen
- | Hardware-nahe Programmierung
- | Entwicklung von Steuerungs- und Regelungsalgorithmen

Daten zum Experiment:

- | E/A: Parallele Schnittstelle
- | 38 kHz Abtastfrequenz
- | 256 Byte dual-ported FIFO
- | 633 MHz Intel CPU
- | Windows Ce.Net & RT-Linux
- | C und C# (Real-Time.Net)



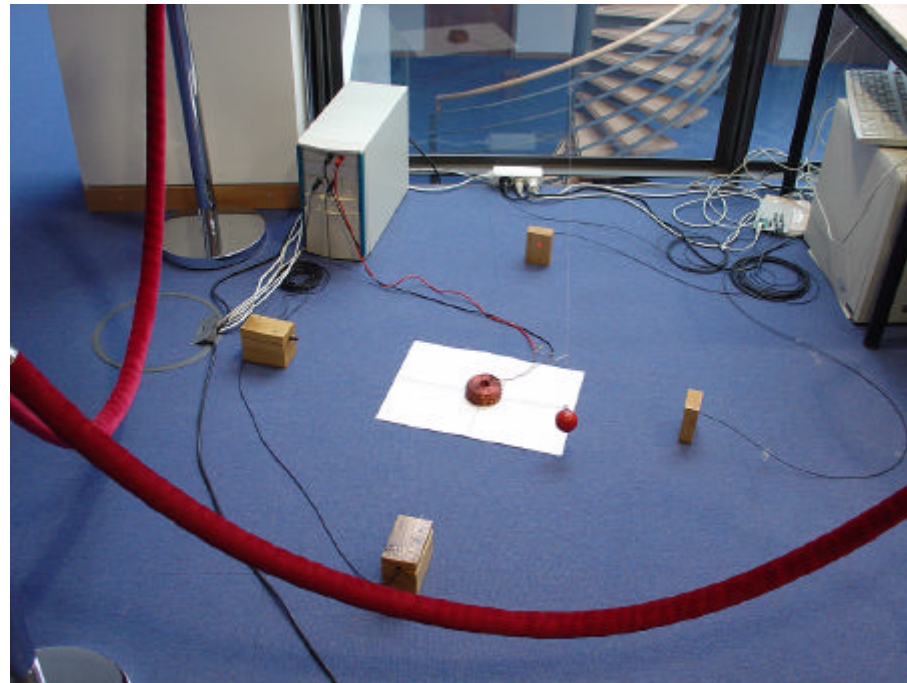
Foucaults Pendel

Lernziele

- ! Entwurf und Implementierung von Steuerungsalgorithmen
- ! Verwendung objektorientierter Hochsprachen (C#)
- ! Genetische Algorithmen

Aufgabendetails:

- ! Aktivierung der Elektromagneten zum Ausgleich von Reibungsverlusten
- ! Auswertung von Lichtschrankensignalen
- ! Programmierung Ereignis-orientierter Software
- ! Filter-basierte Verarbeitung von Datenströmen

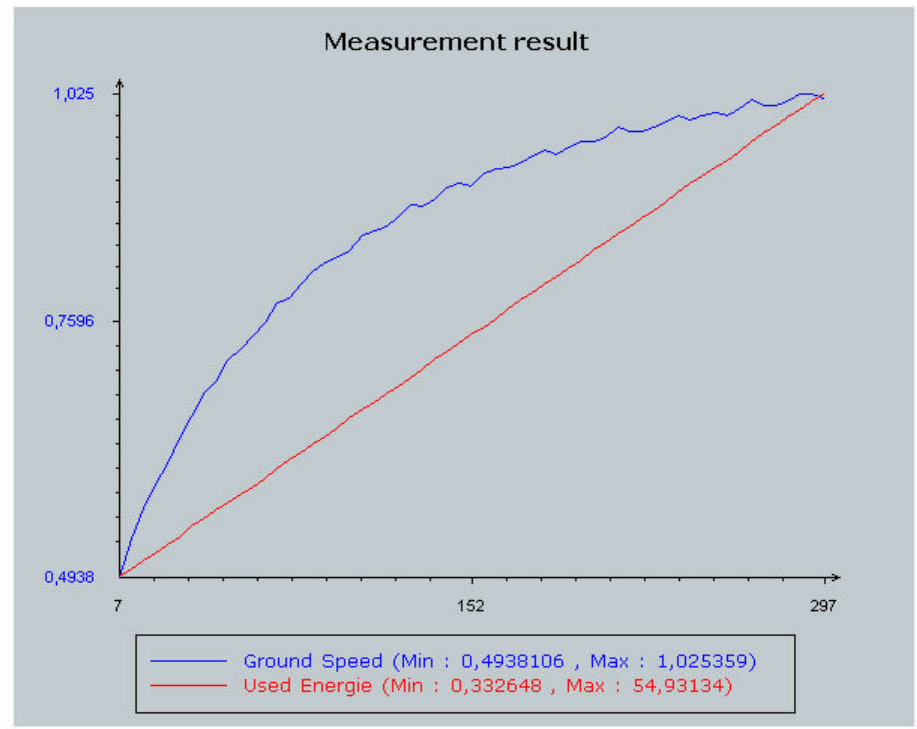


- My Jobs
- My Settings
- Latest News
- Logout

```
long tperiod=0,tau=0;  
SimEvent se=null;  
SimEvent[] last= new SimEvent[2];  
  
bool setevent = false;  
int state = 0;  
  
int mode=0;  
int magnetontime=0;  
  
double s_k  
  
... [incomplete text, use download for full version]
```



State Flow



You can use the download link for viewing or saving the source data of the diagram.



Console Output

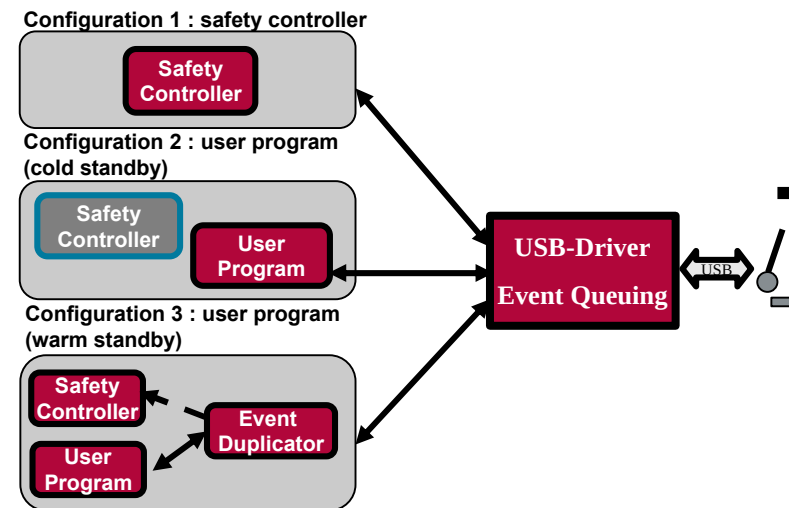
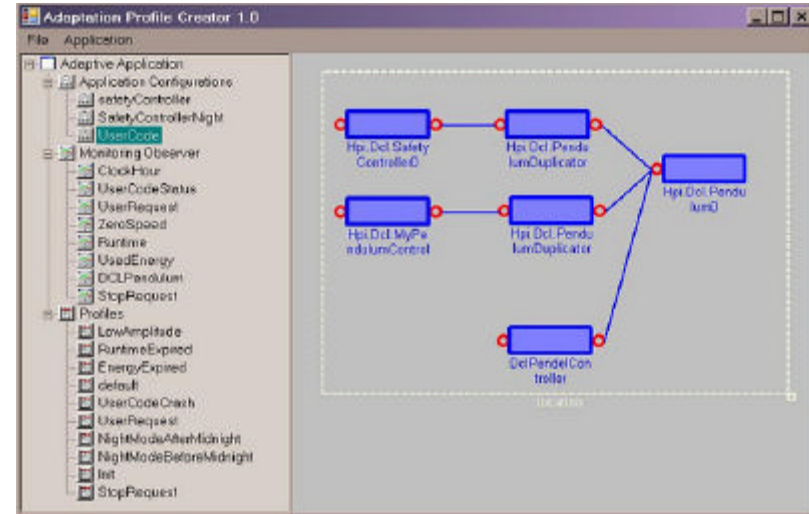
Problematik „Bösartiger Code“

Schutz vor böswilliger Zerstörung von Experimenthardware

- | Quellcodeanalyse
- | Experiment-spezifische Programmiersprachen
 - ? Einschränkung des Sprachumfangs (Rekursion, Zeiger)
- | Sandbox-Ausführung: .NET Code Access Security (CAS)
- | Simulation vor der echten Ausführung
- | **Analytische Redundanz**
 - ? Laufzeitüberwachung der Steuerungsalgorithmen
 - ? Ersetzung bevor ein unkontrollierbarer Zustand erreicht wird
 - ? Dynamische Rekonfiguration von Komponenten-basierten Steuerungen
 - ? Überwachung von dynamischen Umgebungsparametern
- | Einsatz spezieller Hardware (z.B.: Watchdog, Schutzschaltungen)

Dynamische Rekonfiguration als Sicherheitsmechanismus

- ! Konfiguration beschreibt Komponenten und Verbindungen
- ! Dynamische Rekonfiguration: Änderung einer Konfiguration zur Anwendungslaufzeit
- ! Konfigurationsspezifische Schnittstellen werden mit Hilfe von Werkzeugen generiert
- ! Monitoring: Laufzeitanalyse von Amplitude und Temperatur
- ! Ersetzung fehlerhafter Steuerungskomponenten durch verifizierte Sicherheitskomponente (Safety Controller)



Das Distributed Control Lab im Einsatz

Herbst 2006 Gastvorlesung: Blekinge Tekniska Högskola, Schweden

- | Echtzeitprogrammierung mit "Hau-den-Lukas" in Potsdam

Praktika: Vorlesung "Betriebssysteme für Embedded Computing"

- | Programmierung von Echtzeitsteuerungsalgorithmen
- | Evaluierung von Betriebssystemen für Echtzeitsysteme

Schülerinformationstage / Schülerpraktika

- | Programmierung der Lego Mindstorm Roboter / C-Programmierung

DISCOURSE – Testbed für Forschung und Lehre TU, HU, FU, HPI

- | Zugang zu Experimenten im Distributed Control Lab

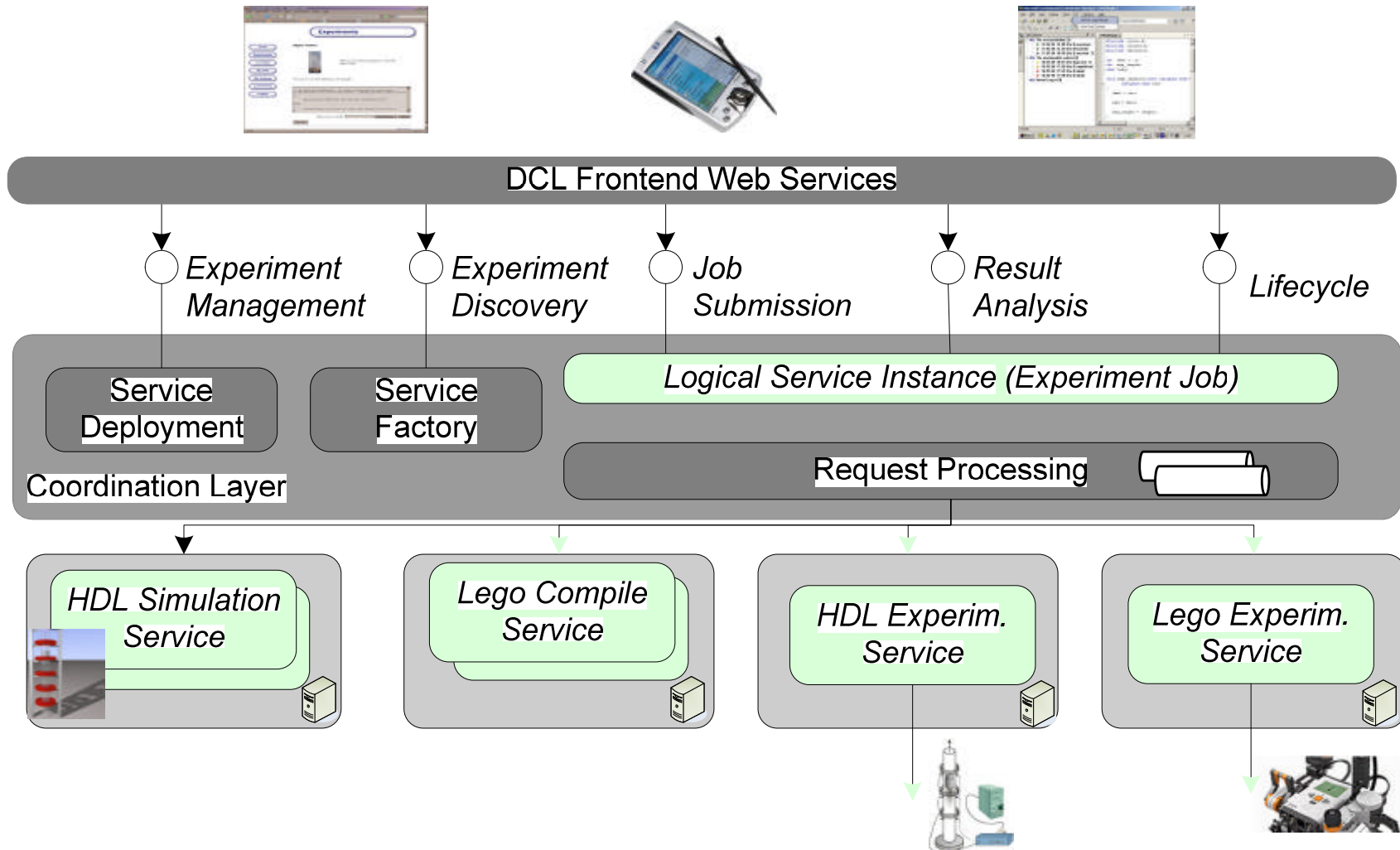
Bachelor Projekt: Integration des Szenarios: "Industrielle Steuerung"

- | Speicherprogrammierbare Steuerungen
- | Fischertechnik Fertigungsstraße
- | Hausautomatisierung



Projekt Vet-Trend

- | „Valorisation of an Experiment-based Training System through a Transnational Educational Network Development“
- | Leonardo Da Vinci II Programm, Transnationale Netzwerke
 - ? Dec. 2006 – Nov. 2008
- | Partner aus Rumänien (Universität Brasov), Italien, Schweden, Portugal, Griechenland, Deutschland
- | Integration von virtuellen und entfernten Experimenten im Bereich der Elektrotechnik und Informatik
- | Schaffung von Standards zur Interaktion verteilter Experimentierumgebungen
- | Pilotinstallationen zwischen den Partnern und Integration in die Lehre (Universität, Berufsausbildung)



Vet-Trend Herausforderungen

- | Sicherheit, Nutzerverwaltung
- | Integrationsebenen (Experiment, Webseite, LMS)
- | Experimentinformationsdienste (Suche, Beschreibung)
- | Integration von Standardsoftware (LabView, Simulink)
- | Feedback (Video, Bilder, Text)
- | Standardisierung beim Experimentzugriff (WebServices)

Zusammenfassung

- | Entfernte Experimente sind eine wichtige Ergänzung für die Lehre
- | Zugriff auf die Experimente über das Internet erlaubt:
 - ? „Rund-um-die-Uhr“ Bearbeitung von Übungsaufgaben
 - ? Effektive Auslastung der Hardware
- | Web-Services sind vielversprechende Technologie für institutsübergreifende Integration von Experimenten
- | Erfahrungen mit dem Distributed Control Lab:
 - ? Experimente erfolgreich im Einsatz seit einigen Jahren
 - ? Positives Feedback von Studenten und Schülern
 - ? Relativ hoher Instandhaltungsaufwand
 - ? Spezielle Vorrichtungen für die Integration von Experimenten nötig (dunkler Keller, Aufladefunktion, Schutzschaltungen)

Weiterführende Informationen:

<http://www.dcl.hpi-uni-potsdam.de>

andreas.rasche@hpi.uni-potsdam.de