

## **Praktika über Internet - Neue Wege in Lehre und Forschung**

Karsten Henke, Heinz-Dietrich Wuttke, Sven Hellbach

Technische Universität Ilmenau, Institut für Theoretische und Technische Informatik

### **Zusammenfassung**

Der Beitrag stellt mit dem Projekt „FIPS“ das Konzept eines *Web-basierten Praktikums* mit verschiedenen Experimenten vor, die über das Internet gesteuert, beobachtet und ausgewertet werden können. Dabei sollen neben den verschiedenen Realisierungsmöglichkeiten auch die einzelnen Schritte der Versuchsdurchführung - angefangen vom Entwurf des Steuerungsalgorithmus bis hin zur web-gesteuerten Erprobung am realen Modell - anhand des Praktikumsversuches „2D-Laufkatzensteuerung“ vorgestellt werden. Darauf aufbauend werden die im Rahmen des Projektes „FIPS“ entwickelte Systemarchitektur vorgestellt sowie verschiedene Realisierungsmöglichkeiten bezüglich der Abarbeitung des entworfenen Steuerungsalgorithmus aufgezeigt und miteinander verglichen.

### **1 Einleitung**

Mit dem Projekt „FIPS“ (**F**erngesteuertes **I**nternet **P**raktikum **S**chaltsysteme) sollen neue Wege und Möglichkeiten der Fernsteuerung und Fernbeobachtung realer Prozesse (z.B. aus den Gebieten der Steuerungstechnik, Robotik, Fernwirktechnik) aufgezeigt werden, die sich mit der integrierten und interaktiven Nutzung moderner Inter- und Intranet-Technologien ergeben. Bedingt durch die gute Infrastruktur des Internets stehen gut erschlossene und somit einfach zugängige Kommunikationswege und -dienste zur Verfügung.

Das Projekt soll die Studierenden gleichzeitig zum Erwerb von Kenntnissen und Erfahrungen auf den Gebieten der Fernsteuerung, -wartung und -beobachtung motivieren und sie anregen, diese Techniken perspektivisch auch für andere Formen des Teleworking anzuwenden (z.B. Steuerung und Kontrolle sensibler Produktionsprozesse).

Als Beispielanwendung für diese modernen Technologien wird im Institut TTI der TU Ilmenau ein ferngesteuertes web-basiertes Praktikum für das Fach „Schaltsysteme“ realisiert. Es dient den Studenten u.a. zur Vorbereitung auf komplexere Laborversuche mit tutorieller Betreuung an realen Versuchen. Die Studenten können eine Reihe von Grundlagen-Versuchen über das Internet vorbereiten, durchführen und am realen Praktikumsmodell über das Internet analysieren (unterstützendes Lernen). Darüber hinaus können Experimente auch spontan aus einer web-basierten Vorlesung heraus genutzt werden.

Beim Projekt „FIPS“ werden die angebotenen Experimente nicht in virtuellen Welten (abstrahierte Schemata) durchgeführt. Das web-basierte Praktikum beinhaltet reale Laborversuche in realer Laborumgebung, unterstützt durch multimediale Lehr- und Lernmaterialien.

Im folgenden soll mit dem Projekt „FIPS“ das Konzept eines ferngesteuerten web-basierten Praktikums mit verschiedenen Experimenten vorgestellt werden, die über das Internet beobachtet, gesteuert und ausgewertet werden können. Dabei werden neben dem Entwurf der Versuchsanordnung verschiedene Möglichkeiten

- der client- oder serverseitigen Abarbeitung der entworfenen Steuerungsalgorithmen,
- der Zugriffssteuerung bei mehreren Nutzern (Authentifizierung),
- der Datenvernetzung/Kommunikation auf globaler und lokaler Ebene,
- der interaktiven Prozessbeeinflussung (Start/Stop des Systems, Steuerung von Modellparametern) sowie
- der Überwachung der Versuchsanordnung mittels Videokamera sowie zusätzlicher Unterstützung durch chat, mail usw.

gegenübergestellt.

Bei der Umsetzung des Projektes wird besonders Wert gelegt auf

- die Wiederverwendbarkeit der eingesetzten Methoden und Werkzeuge für ähnliche Problemstellungen,
- die Wiederverwendbarkeit der entwickelten Softwaremodule,

was letztlich zu weiteren Synergieeffekten führt.

## 2 Einleitendes Beispiel

Mit dem folgenden Beispiel soll gezeigt werden, wie das *Web-basierte Praktikum* zur Vorbereitung auf ein reales Laborpraktikum genutzt werden kann.

### 2.1 Simulation oder praktischer Versuch

Bei der Vorbereitung auf Laborpraktika werden häufig zur Überprüfung der erarbeiteten Lösungen Simulationen oder Animationen in „virtuellen Welten“ eingesetzt. Dabei wird das physikalische Verhalten sowohl des eigentlichen Modells als auch der Umgebung rechenstechnisch (z.B. als Simulationsmodelle) nachgebildet. Der Benutzer kann seine „virtuelle Welt“ beeinflussen und die entsprechende Reaktion seines Steueralgorithmus analysieren. Dynamische Vorgänge werden dabei animiert und teilweise sogar durch Maschinengeräusche akustisch unterlegt (DynaMit 2000).

Wesentlicher Nachteil dieser Herangehensweise ist allerdings, dass in dem hinterlegten virtuellen Umgebungsmodell *reale Störeinflüsse* (z.B. Ausfall einzelner Komponenten, mechanische Probleme) in der Regel nicht erfasst werden können. Wenn überhaupt, ist meist nur eine Simulation vorher festgelegter Störfälle möglich. Nicht im voraus bedachte Fehlerquellen können dazu führen, dass Fehler der Steuerung nicht bemerkt werden, da die entsprechende Umweltsituation nicht simuliert wurde. Im realen Laborpraktikum (ebenso wie beim *Web-basierten Praktikum*) können jedoch reale Störungen auftreten und begründen gerade dadurch die Besonderheit solcher Praktika. Die Berücksichtigung und Einbeziehung solcher realer Fehlerfälle stellt gegenüber Simulationen einen deutlich engeren Praxisbezug her und soll dem Studenten gleichzeitig auch Anregungen hinsichtlich des Entwurfes sicherheitskritischer Steuerungen geben.

us diesem Grund wurden bei der Konzeption des „FIPS“-Projektes die oben erwähnten virtuellen Welten durch die reale Praktikums Umgebung ersetzt. Schwerpunktmäßig wird deshalb besonders Wert darauf gelegt, dass der Student seine entworfenen Steuerungs Algorithmen

- virtuell im Praktikumsraum, d.h. schon vor dem eigentlichen Laborpraktikum
- unter realen Umweltbedingungen (die er darüber hinaus noch interaktiv beeinflussen kann)

überprüfen und gegebenenfalls korrigieren oder modifizieren kann (siehe Abbildung 1).

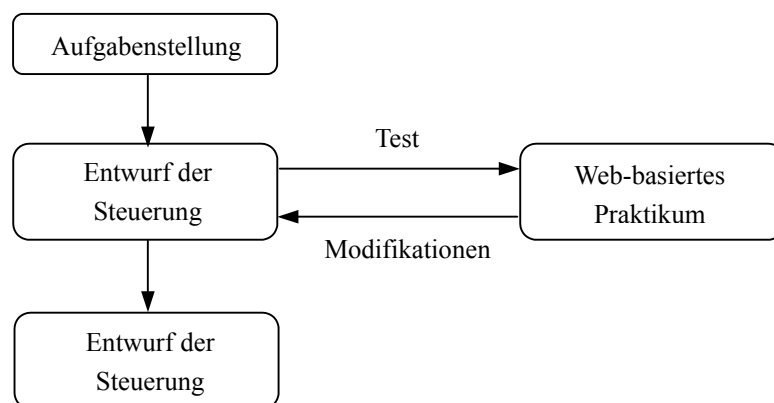


Abbildung 1: Vorbereitung auf ein Laborpraktikum

Anhand des Versuches „2D-Laufkatzen-Steuerung“ aus der Lehrveranstaltung Schaltsysteme soll einerseits die im „FIPS“-Projekt entwickelte Systemarchitektur vorgestellt werden, die zum Betrieb ein *Web-basiertes Praktikum* notwendig ist - andererseits sollen verschiedene Abarbeitungsmechanismen des durch den Studenten erarbeiteten Steuerungsalgorithmus gegenübergestellt werden.

## 2.2 Aufgabenstellung

Als anzusteuern des Labormodell dient eine Laufkatze, die sich in einer Ebene in zwei Freiheitsgraden (links/rechts sowie oben/unten) bewegen kann (siehe Abbildung 2). Eine aktuelle Positionsbestimmung ist nur möglich, wenn sich die Laufkatze an einem Rand oder einer Ecke befindet. Dafür stehen vier Endlagenschalter zur Verfügung, die das Eintreffen der Laufkatze am jeweiligen Rand (links/rechts/oben/unten) signalisieren. Im Praktikum ist durch den Studenten ein Steuerungsalgorithmus zu erarbeiten, der die Laufkatze entlang eines vorgegebenen Weges (siehe Aufgabenstellung) bewegt.

**Der Student hat folgende Aufgabenstellung zu lösen:**

„Gesucht ist eine Steuerung für eine Laufkatze mit zwei Freiheitsgraden, welche durch Auswertung der vier Endlagen-Schalter links/rechts/oben/unten ( $x_l$ ,  $x_r$ ,  $x_u$ ,  $x_o$ ) die vier Motorsteuersignale für die vier Richtungen links/rechts/oben/unten ( $y_l$ ,  $y_r$ ,  $y_u$ ,  $y_o$ ) gemäß dem folgenden Ablauf erzeugt:

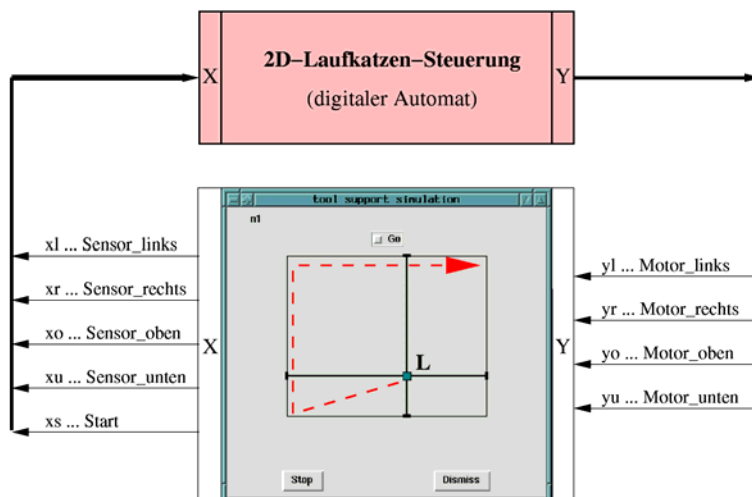


Abbildung 2: Modell „2D-Laufkatze“

- Die Laufkatze *L* soll unabhängig von ihrer Anfangsstellung nach der I/O-Flanke des Startsignals (*xs*) möglichst schnell nach links/unten bewegt werden.
- Danach soll sie am linken Rand nach oben
- und am oberen Rand nach rechts gefahren werden, worauf die Bewegung gestoppt werden soll.

Ein Neustart ist nur mit einer erneuten I/O-Flanke von *xs* möglich.

Zur Realisierung der Aufgabe ist ein Automatengraph zu entwerfen, der den geforderten Ablauf gewährleistet.

Die Aufgabenstellung ist soweit vorzubereiten und mittels des Web-basierten Praktikums auszu-  
testen, dass die Steuerung später im Laborversuch mittels D-Flip-Flops praktisch realisiert werden kann."

## 2.3 Entwurf der Steuerung

In Vorbereitung auf den späteren Laborversuch kann der Student den Entwurf der Steuerung bereits zu Hause von Hand oder unter Einbeziehung verschiedener Hilfsmittel (z.B. GIFT (GIFT 2002), SANE (SANE 2002), Living Pictures (Cate 1999)) durchführen. Es soll an dieser Stelle angenommen werden, dass er als Ergebnis beispielsweise den in Abbildung 3 dargestellten Automatengraph (der die Eingangssignale *xl*, *xr*, *xu*, *xo*, *xs* auswertet und die Ausgangssignale *yl*, *yr*, *yo*, *yu* beeinflusst) entwickelt hat.

Aus diesem Automatengraph wird er einerseits die zur Abarbeitung der Zustandsüberföhrungsfunktion des Automaten notwendigen Gleichungen ermitteln. In diesem konkreten Fall sind zur Umsetzung des Automatengraphen mit 7 Zuständen (bei binärer Zustandskodierung) mit D-Flip-Flops drei *z*-Gleichungen notwendig. Außerdem hat er in jedem Zustand den Wert der entsprechenden Motor-Ansteuersignale festzulegen und danach die vier *y*-Gleichungen zu ermitteln.

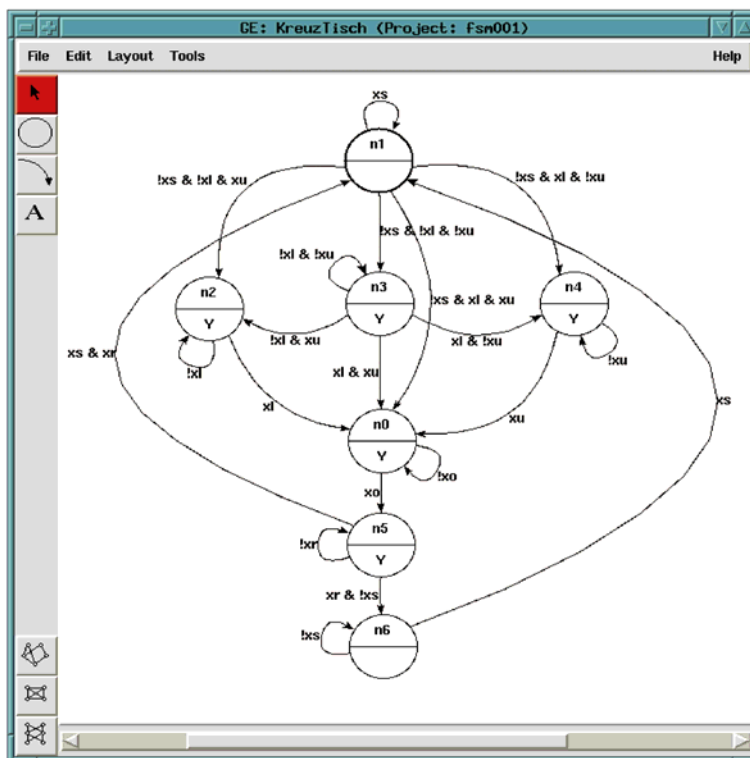


Abbildung 3: Automatengraph der Steuerung

## 2.4 Test der Steuerung

Über das durch den Web-Browsers zur Verfügung gestellte Interface des *Web-basierten Praktikums* (siehe Abbildung 4) kann der Student

- seinen erarbeiteten Steuerungsalgorithmus (in Form der ermittelten  $z$ - und  $y$ -Gleichungen) eingeben,
- den eigentlichen Versuchsablauf steuern (Start, Stop, Reset),
- gegebenenfalls Umgebungsvariablen verändern (so kann er beispielsweise durch Einzelansteuerung der vier Motoren die Laufkatze vor dem Versuchsstart beliebig positionieren, um dadurch verschiedene Spezialfälle zu analysieren) und
- die Abarbeitung durch Überwachung der Modell-/Umgebungsvariablen in einem I/O-Monitor oder aber durch Beobachtung per WebCam direkt am Hardware-Modell verfolgen.

Er hat jederzeit die Möglichkeit, im Fehlerfall seine Algorithmen zu korrigieren und somit schrittweise eine fehlerfreie Lösung zu erarbeiten, die schließlich die Grundlage für sein reales Laborpraktikum darstellt.

### 3 Konzept des *Web-basierten Praktikums*

Anhand von Abbildung 4 soll zunächst die im Rahmen des Projektes „FIPS“ entwickelte Systemarchitektur vorgestellt werden, die zum Betrieb eines *Web-basierten Praktikums* notwendig ist. Anschließend werden verschiedene Realisierungsmöglichkeiten bezüglich der Abarbeitung des entworfenen Steuerungsalgorithmus aufgezeigt und miteinander verglichen werden.

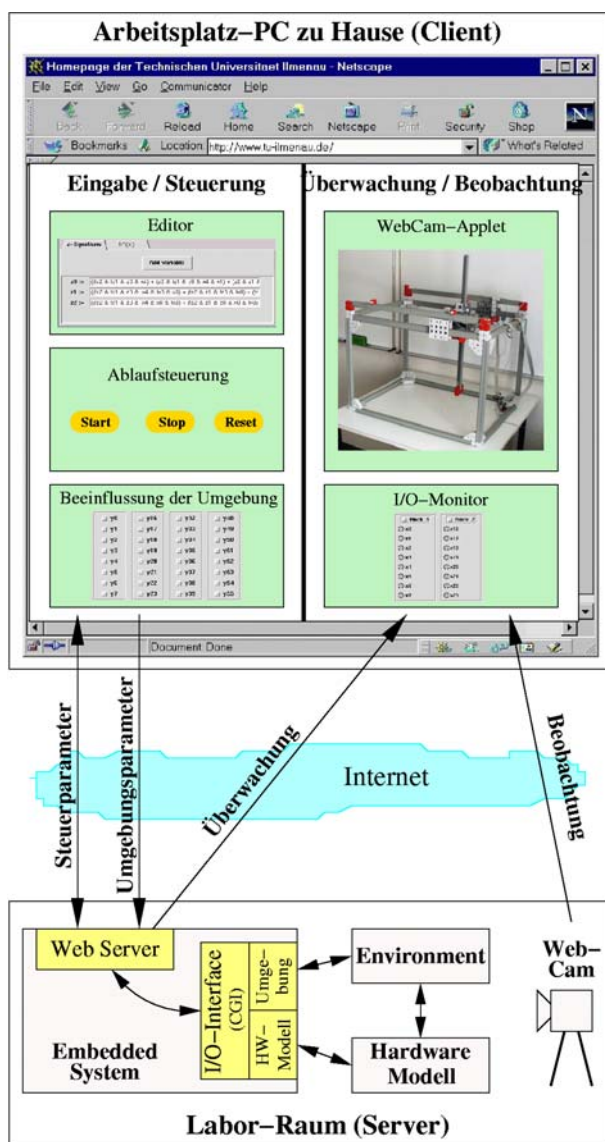


Abbildung 4: Systemarchitektur von „FIPS“

## 3.1 Systemarchitektur

Das Gesamtkonzept beruht aufgrund der Nutzung der im Internet verwendeten TCP/IP-Mechanismen auf einer Client-Server-Architektur. Deshalb muss untersucht werden, welche Systemkomponenten auf der Client-Seite (z.B. am PC des Studenten zu Hause) und welche auf der Server-Seite (im Praktikums-Raum) zur Realisierung eines *Web-basierten Praktikums* notwendig sind.

### 3.1.1 Clientseitige Komponenten

Der Browser muss eine Nutzeroberfläche bereitstellen, die folgende Funktionalität gewährleistet:

- Möglichkeit zur Eingabe des entworfenen Algorithmus und der Ablaufsteuerung
  - **Eingabemöglichkeit** des durch den Studenten entworfenen Steuerungsalgorithmus.  
In dem in Abschnitt 2 vorgestellten Beispiel können die durch den Studenten im Vorfeld ausgearbeiteten und zur Modellansteuerung notwendigen  $z$ - und  $y$ -Gleichungen z.B. über vorgegebene Masken eingegeben werden.
  - **Steuerung** des Versuchsablaufes (z.B. Start, Stop der Abarbeitung des Steuerungsalgorithmus)
  - Möglichkeit der interaktiven **Beeinflussung** von Umgebungsparametern.
- Möglichkeit zur Überwachung und Beobachtung des laufenden Versuches
  - **Beobachtung** des Versuchsablaufes über eine WebCam<sup>1</sup>
  - **Überwachung** der aktuellen Modell- und Umgebungsvariablen über einen I/O-Monitor.

### 3.1.2 Serverseitige Komponenten

Serverseitig sind im Praktikumsraum folgende Komponenten zur Gewährleistung der Funktionalität notwendig:

- ein **Eingebettetes System**<sup>2</sup>, das
  - über den integrierten Web-Server die Kommunikation mit dem Client realisiert,
  - über die I/O-Schnittstelle (in Form von CGI-Scripten) alle zugänglichen Modell- und Umgebungsvariablen austauschen kann.
- das anzusteuernde **Praktikumsmodell** sowie
- eine **WebCam**, die die Beobachtung der gesamten Versuchsanordnung (Modell und Umgebung) ermöglicht.

Die Verwendung des web-basierten eingebetteten Systems demonstriert gleichzeitig die Vorteile des Einsatzes von Single-Chip-PC's für solche Einsatzgebiete und stellt eine kostengünstige und vor allem platzsparende Alternative zu einem vollständigen PC dar.

---

<sup>1</sup> WebCam = Videokamera mit Internetanschluss

<sup>2</sup> Im „FIPS“-Projekt wird ein Single-Chip-PC der Firma Beck (Beck-IPC, Beck 2001) eingesetzt, der über einen eigenen „Mini-Web-Server“ verfügt.

Für die Überwachung der Praktikumsdurchführung (Versuchsbearbeitung) stehen dem Studenten zwei Informationswege zur Verfügung:

- **Überwachung**  
Mit Hilfe der Überwachung können die Signale der entworfenen Steuerung (Ein-/Ausgangsvariablen) und der Umgebung zum Client übertragen werden. Diese sind als Parameter interaktiv veränderbar.
- **Beobachtung**  
Zusätzliche, nicht an den Ein-/Ausgabe-Schnittstellen verfügbare Informationen, die insbesondere im Fehlverhalten relevante Aussagen liefern, werden mit Hilfe einer WebCam erfasst und übertragen.

Mit diesem Konzept sind am studentischen Arbeitsplatz zu Hause nahezu reale Versuchsbedingungen realisierbar.

Die Web-Überwachung ist bei einem fehlerfrei entworfenem Steuerungsalgorithmus und idealem Umgebungsverhalten nicht zwingend notwendig. Da aber im „FIPS“-Projekt von vorn herein eine gezielte oder auch zufällige Beeinflussung von Umgebungsvariablen - sei es z.B. zur Schaffung unterschiedlicher Initial-Bedingungen oder aber auch zur Vorgabe eines realen Fehlverhaltens der Umgebung - vorgesehen ist, ist eine Beobachtung des gesamten Versuchsplatzes zwingend notwendig. Da aber durch die WebCam nicht alle Informationen in ausreichendem Maße dargestellt werden können (z.B. Werte der Endlagenschalter oder interne Signalpegel), ist es notwendig, diese Informationen zusätzlich über einen I/O-Monitor zu überwachen. Auf diese Art und Weise werden redundante Informationen als Zusatzinformationen zur Verbesserung des Verständnisses beitragen.

### 3.2 Abarbeitung des Steueralgorithmus

Im folgenden soll untersucht werden, wie sich die zwei Möglichkeiten des Steuerungsalgorithmus,

- die clientseitige „Fern“<sup>3</sup>-Steuerung sowie
- die serverseitige „vor Ort“-Steuerung

im „FIPS“-Projekt umsetzen lassen.

In beiden Fällen (siehe Abbildung 5) werden die durch den Studenten erarbeiteten und in einem Browserfenster eingegebenen Gleichungen zur Modellansteuerung über ein *Interpreter*-Programm abgearbeitet. Über Client/Server-Verbindungen können zusätzlich Umgebungsvariablen beeinflusst und überwacht (über das Umgebungs-I/O-Interface des Servers) sowie der Versuchsplatz über eine WebCam beobachtet werden. Darauf soll an dieser Stelle nicht im Detail eingegangen werden. Vielmehr stehen die Varianten der „Fern“- und „vor Ort“-Steuerung im Vordergrund.

---

<sup>3</sup> Der Begriff „Fern“ bezeichnet den über das Internet angeschlossenen Arbeitsplatz, „vor Ort“ den Praktikumsraum.



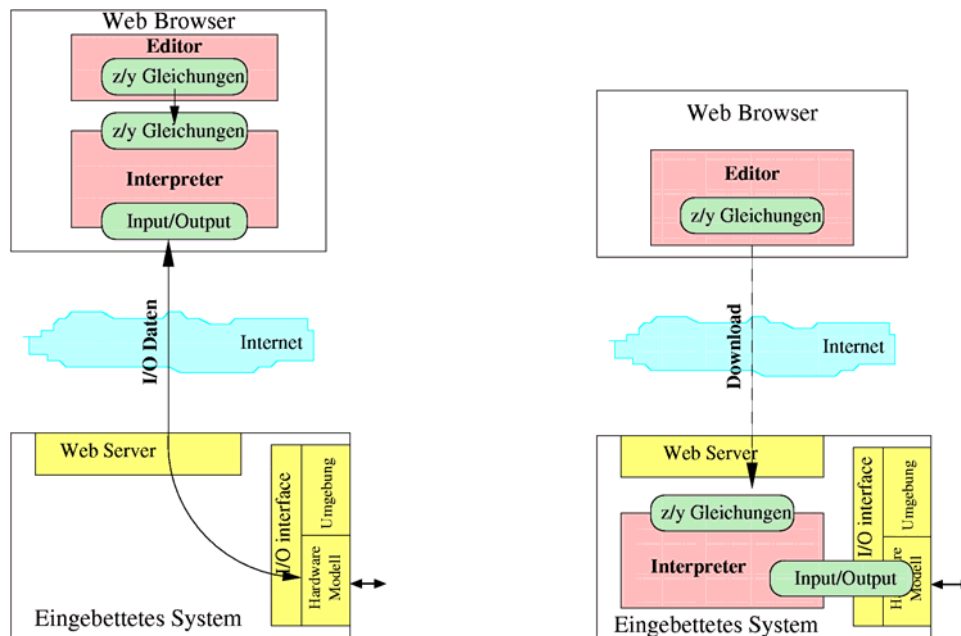


Abbildung 5: Clientseitige „Fern“-Steuerung und serverseitige „vor Ort“-Steuerung

### 3.2.1 Clientseitige „Fern“-Steuerung

Im Fall der clientseitigen „Fern“-Steuerung wird das Praktikumsmodell „aus der Ferne“ durch das im Client ablaufende Interpreterprogramm (welches z.B. als Java-Applet realisiert ist) gesteuert. Über das Internet werden dabei die Ein-/Ausgangssignale des zu steuernden Hardware-Modells übertragen.

Zum Austausch der notwendigen Modellvariablen erfolgt eine ständige Kommunikation zwischen dem Interpreterprogramm und dem Modell-I/O-Interface des Servers. Es ist einerseits notwendig, über ein CGI-Script die Modellvariablen durch das I/O-Interface abzufragen und die erhaltenen Informationen mittels einer Textdatei oder schon als HTML-Code über den Webserver an das Interpreterprogramm weiterzureichen. Um Daten vom Hardware-Modell an den Client zu übertragen, werden CGI-Scripte dazu eingesetzt, um die Ergebnisse des Interpreterprogramms an das angeschlossene Praktikumsmodell weiterzuleiten. Dies kann wahlweise über (die durch HTTP zur Verfügung gestellten) „POST“- oder „GET“-Methoden erfolgen.

Dazu ist es notwendig, die seriell oder parallel anliegenden Informationen in Pakete nach TCP/IP-Standard umzuwandeln. Da TCP/IP keine bidirektionale Verbindung ist, sondern auf Anfragen (requests) beruht, muss der Client in geeigneten Zeitintervallen diesbezügliche Anfragen ständig neu initiieren, um eventuelle Änderungen der Modell- und Umgebungsvariablen zu erkennen. Dabei ist darauf zu achten, dass die gewählte Periode nicht zu kurz ist, um das Netzwerk nicht zu sehr zu belasten. Sie sollte aber auch nicht zu hoch sein, um auszuschließen, dass kritische Meldungen den Interpreter (d.h. den Steuerungsalgorithmus) zu spät erreichen. Detaillierte Untersuchungsergebnisse sollen im Vortrag vorgestellt werden.

### 3.2.2 Serverseitige „vor Ort“-Steuerung

Wird eine Abarbeitung des Steuerungsalgorithmus serverseitig durchgeführt (d.h. das Interpreterprogramm läuft „vor Ort“ innerhalb des Eingebetteten Systems), müssen lediglich einmal die notwendigen  $z$ - und  $y$ -Gleichungen vom Browser über den Web-Server an das Interpreterprogramm weitergereicht werden, welches seinerseits direkt mit dem Modell-I/O-Interface Daten austauschen kann. Da aber eine der wesentlichen Aufgaben des Web-Servers die Verwaltung von Anfragen ist, muss sorgsam abgeschätzt werden, welche und wie viel rechenintensive Prozesse hinzugenommen werden können, ohne dass es zu Ausfällen in den bereits vorgestellten Kommunikationskanälen zum Client kommen kann. Auch dazu werden gegenwärtig Untersuchungen angestellt.

## 4 Zusammenfassung

Erste Erfahrungen aus der Nutzung des *Web-basierten Praktikums* zeigen, dass:

- die Studenten am praktischen Beispiel die Möglichkeiten, aber auch die Grenzen der Fernsteuerung und Fernbeobachtung via Internet kennen und einzuschätzen lernen.
- mit dem *Web-basierten Praktikum* ein Instrumentarium zur Praktikumsvorbereitung für den Studenten in einer neuen Form zur Verfügung steht, was letztlich zu einer Effektivierung des Praktikums führen wird.
- der Student seine web-basierten Experimente in der gleichen, ihm schon aus dem Praktikumsraum bekannten Umgebung durchführt, was auch zu einer Verkürzung der Praktikumszeit (ca. 30%) für komplexere Versuche mit tutorieller Betreuung von realen Laborversuchen und somit zu einer Potenzierung der Laborressourcen insgesamt führt.
- der Student neben der Vermittlung von experimentellem Laborwissen durch den Einsatz der neuen Technologien gezwungen wird, diese auch kritisch zu beurteilen.

### Literatur

- Beck (2002): *Beck-IPC Nutzerdokumentation*. Beck-IPC GmbH: <http://www.beck-ipc.com/chip>
- De Souza, C.P.; Filho, J.T.C (2001): Labnet – Laboratory of remote Access. In: *IASTED International Conference Computers and Advanced Technology in Education CATE2001*. Banff, Canada, 27-29. Juni 2001: Tagungsband, S. 38-40.
- DynaMit (2001): *Erlernen von Eigenschaften dynamischer Systeme mit Multimedia Mitteln*. Ruhr-Universität Bochum: <http://astwww.chemietechnik.uni-dortmund.de/~mume>.
- FIPS (2002): *Ferngesteuertes Internetpraktikum Schaltsysteme*. Technische Universität Ilmenau: <http://www-ihs.theoinf.tu-ilmenau.de/projekte/fips>.
- GIFT (2001): *Graphical Interactive FSM-Tools*. Technische Universität Ilmenau: <http://www-ihs.theoinf.tu-ilmenau.de/projekte/gift>.
- Goshi, K.; Matsunaga, K.; Kuroki, D. (2001): Educational Intelligent Transport System Assist. In: *IASTED International Conference Computers and Advanced Technology in Education CATE2001*: Banff, Canada, 27-29. Juni 2001: Tagungsband, S. 150-154.
- Henke, K.; Peukert, R.; Wuttke, H.-D. (1999): Internet based education – an experimental environment for various educational purposes. In: *IASTED International Conference Computers and Advanced Technology in Education CATE1999*. Philadelphia, PA USA, 6.-8. Mai 1999: Tagungsband, S. 128-132.
- SANE (2001): *Schaltssysteme-Arbeitsblätter im Netz*. Technische Universität Ilmenau: <http://www-ihs.theoinf.tu-ilmenau.de/projekte/sane>.