

# Pflichtenheft RoboCup-Client

Version 0.2

16. Mai 2002

**Basierend auf:**

RoboCup-Simulationssystem

## Inhaltsverzeichnis

0 Modifikationen.....	3
1 Zielbestimmung .....	3
1.1 Musskriterien.....	3
1.2 Wunschkriterien.....	3
1.3 Abgrenzungskriterien .....	3
2 Produkteinsatz.....	3
2.1 Anwendungsbereiche .....	4
2.2 Zielgruppen.....	4
2.3 Betriebsbedingungen.....	4
3 Produkt-Umgebung .....	4
3.1 Software.....	4
3.2 Hardware .....	4
3.3 Orgware .....	4
3.4 Produkt-Schnittstellen .....	4
4 Produkt-Funktionen.....	4
4.1 Anmeldung bei einem RoboCup-Server .....	4
4.2 Abmeldung bei einem RoboCup-Server .....	5
4.3 Visualisierung des Weltmodells eines Spielers.....	5
4.4 Simulation einer Fußballmannschaft .....	5
5 Produkt-Daten.....	5
6 Produkt-Leistungen.....	6
7 Benutzeroberfläche .....	6
8 Qualitätszielbestimmung .....	6
9 Globale Testszenarien .....	6
10 Entwicklungs-Umgebung .....	6
10.1 Software.....	6
10.2 Hardware .....	6
10.3 Orgware .....	6
10.4 Entwicklungs-Schnittstellen.....	6
11 Ergänzungen/Sonstiges.....	6

## 0 Modifikationen

Im Verlauf des Praktikums werden möglicherweise Veränderungen an diesem Dokument erforderlich werden. Die Modifikationen werden in diesem Abschnitt zusammengefasst.

*Version 0.1* Initialversion

*Version 0.2* Änderungen in Kapitel 1 und 4

## 1 Zielbestimmung

Für das RoboCup-Simulationssystem, das aus einem Server (rcssserver) und einem Monitor (rcssmonitor) besteht, entwerfen wir eine Fußballmannschaft. Diese besteht aus bis zu elf Spielern, wobei jeder als eigenständiger Client implementiert wird. Die Clients kommunizieren ausschließlich mit dem Server über das UDP/IP Protokoll, eine direkte Verständigung der Clients untereinander ist verboten. Der Server übernimmt die Spielkontrolle und übermittelt den Spielern ihre sensorischen Daten. Der Monitor visualisiert das Spielgeschehen auf dem Fußballfeld. Die zu implementierende Mannschaft muss in der Lage sein, eine gestellte Dummy Mannschaft zu schlagen.

### 1.1 Musskriterien

- **Implementierung** eines virtuellen Fußballteams in Java, welches gemäß den RoboCup-Anforderungen spielt. Die Fußballmannschaft besteht aus bis zu elf Spielern, die ausschließlich mit dem Server kommunizieren. Die Spielertypen sind in Bezug auf Positionen (Torwart, Abwehr, Mittelfeld, Sturm) und Fähigkeiten (Sichtfeld, ...) heterogen.
- **Kommunikation:** Es müssen Routinen entwickelt werden, die die Datenpakete empfangen und auswerten, sowie eigene Aktionen in Datenpakete umwandeln und an den Server senden.
- **Kompatibilität:** Der Client ist zur Protokollversion 7.0 des Servers kompatibel.
- **Weltmodell:** Um ein geeignetes Verhalten entwickeln zu können, muss der einzelne Spieler ein Weltmodell konstruieren, d.h. die eingehenden Informationen müssen entsprechend eingeordnet werden.
- **Visualisierung des Weltmodells:** Zur Kontrolle des internen Weltmodells soll dieses mit Hilfe der Swing-GUI-Elemente von Java visualisiert werden.
- **Strategie:** Das "Gehirn" des Spielers. Je nach Spielsituation und Position wählt der Spieler geeignete Aktionen aus.
- **Ziel:** Eine gestellte Dummy Mannschaft zu schlagen.

### 1.2 Wunschkriterien

- Verarbeitung von vielen sensorischen Informationen.
- Erzeugung eines umfassenden Weltmodells, in dem bewegliche Objekte als Vektoren gespeichert werden (d.h. Berücksichtigung von Position, Richtung und Geschwindigkeit).
- Entwicklung einer vorausschauenden Strategie, die insbesondere auch die zukünftigen Positionen der Objekte (Ball, Mitspieler, Gegner) berücksichtigt.

### 1.3 Abgrenzungskriterien

- Keine Implementierung der Coach Funktionen

## 2 Produkteinsatz

Das Produkt dient zur Simulation einer virtuellen Fußballmannschaft. Der Spielablauf läuft ohne Eingreifen von außen ab.

## **2.1 Anwendungsbereiche**

- Simulation einer einfachen künstlichen Intelligenz
- Unterhaltung / Entertainment

## **2.2 Zielgruppen**

- Entwickler von eigenen RoboCup-Clients
- Personen mit Interesse an Fragestellungen der künstlichen Intelligenz
- Technik- und Computerinteressierte Fußballfans

## **2.3 Betriebsbedingungen**

- heimische Umgebung / Büroumgebung

# **3 Produkt-Umgebung**

Das Produkt ist client/server-fähig und kommuniziert mit dem Server mittels UDP/IP.

## **3.1 Software**

- Betriebssystem: Windows 9x/NT/2000/XP, Linux/Unix
- Java VM: Java 2 Standard Edition SDK/RE 1.3.1 oder höher

## **3.2 Hardware**

- PC, Workstation mit Grafikbildschirm
- Prozessorleistung mind. 500 MHz und 128MB RAM

## **3.3 Orgware**

- Internetanbindung

## **3.4 Produkt-Schnittstellen**

- RoboCup-Server (Spielverwaltung)
- RoboCup-Monitor (Visualisierung des Spielgeschehens)

# **4 Produkt-Funktionen**

## **4.1 Anmeldung bei einem RoboCup-Server**

### **Ziel**

Die Mitglieder eines Teams sind bei einem RoboCup-Server als Spieler angemeldet.

### **Vorbedingungen**

Der RoboCup-Server wurde gestartet und der Benutzer kennt dessen Adresse und Port.

Noch kein Spieler ist beim Server angemeldet. Das Team besteht aus elf Spielern inklusive maximal eines Torwarts.

#### **Ablauf**

1. Der Benutzer ruft sein Programm folgendermaßen auf: `java RoboClient teamname servername serverport`
2. Das Programm startet elf unabhängig agierende Spieler und meldet diese einzeln beim Server an.

#### **Alternativen**

- 2a. Der Server hat die Anmeldung aufgrund ungültiger Zugangsdaten abgelehnt. Der Benutzer wird zur erneuten Eingabe aufgefordert.
- 2b. Der Server nimmt keine weiteren Anmeldungen an. Der Benutzer erhält eine entsprechende Mitteilung.

### **4.2 Abmeldung bei einem RoboCup-Server**

#### **Ziel**

Der Spieler ist nicht mehr angemeldet.

#### **Vorbedingungen**

Der Spieler muss beim Server angemeldet sein.

#### **Ablauf**

1. Der Benutzer gibt an, dass er beim RoboCup-Server abgemeldet werden möchte.
2. Der Server meldet den Spieler ab. Der Spieler erhält keine Bestätigung, dass die Abmeldung erfolgt war.

#### **Alternativen**

keine

### **4.3 Visualisierung des Weltmodells eines Spielers**

#### **Ziel**

Darstellung des internen Weltmodells des Spielers mit Hilfe der Swing-GUI-Elemente.

#### **Vorbedingungen**

Der Spieler muss beim Server angemeldet sein.

#### **Ablauf**

1. Auslesen der Daten aus dem Weltmodell
2. Grafische Darstellung mit Swing

#### **Alternativen**

keine

### **4.4 Simulation einer Fußballmannschaft**

#### **Ziel**

Implementierung einer Fußballmannschaft, die selbständig an einem vom Server verwalteten Spiel teilnimmt.

#### **Vorbedingungen**

Korrektes Starten der Clients und des Spiels.

#### **Ablauf**

1. Die Spieler kommunizieren mit dem Server.
2. Basierend auf ihrem Weltmodell und ihrer Strategie führen sie Aktionen aus.

#### **Alternativen**

keine

## **5 Produkt-Daten**

### **5.1 Session-Daten**

Das Spielgeschehen wird auf dem Monitor in Echtzeit visualisiert. Optional kann das Spielgeschehen in einer Logdatei mitgeschrieben werden.

## 6 Produkt-Leistungen

Der Spieler muss in Echtzeit in der Lage sein, mit dem Server zu kommunizieren, sich ein Weltmodell zu konstruieren und aufbauend auf der eigenen Strategie Aktionen auszuführen.

## 7 Benutzeroberfläche

- Die Visualisierung des Spielgeschehens wird vom rcssmonitor übernommen.
- Das Weltmodell des Spielers wird mittels einer Swing-Oberfläche visualisiert.

## 8 Qualitätszielbestimmung

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität		X		
Zuverlässigkeit			X	
Benutzbarkeit	X			
Effizienz			X	
Übertragbarkeit	X			

## 9 Globale Testszenarien

- Allgemeiner Funktionalitätstest (Zurechtfinden des Spielers auf dem Spielfeld und Ausführen sinnvoller Aktionen, Ausschließen von Deadlocks)
- Das Team ist in der Lage, die gegebene Dummy Mannschaft zu schlagen
- Visualisierung des Weltmodells eines Spielers.

## 10 Entwicklungs-Umgebung

### 10.1 Software

analog Produktumgebung

### 10.2 Hardware

analog Produktumgebung

### 10.3 Orgware

- Internetanbindung
- Versionsverwaltungssystem CVS

### 10.4 Entwicklungs-Schnittstellen

keine

## 11 Ergänzungen/Sonstiges

keine