Praktikum Programmiermethodik SS 2002
an der Universität Mannheim

Implementierung eines RoboCup-Soccer-Agents

JAIS

Konzeptpapier

Version 0.3
Teil a

1. Mai 2002
2. Mai 2002
5. Mai 2002

Das JAIST - Java Implemented Soccer Team -
ist eine Zusammenarbeit zwischen

Oliver Beck, Stephan Diederich, Claus Kestel, Florian Münz,
Sven Scheffelmeier, Christian Traub und Raul Zanzinger

Ende Definitionsphase (2 Wochen):
Erste Analyse der Problemstellung bis 06.05.2002

Behandlung folgender Punkte

1. Geplante Architektur
2. Strategieüberlegungen
3. Aufteilung in der Gruppe
4. Zeitplanung

RoboCup-Soccer-Agents, Konzeptpapier

Version 0.3, vom 05.05.2002

Basierend auf:

- Vorgaben des LS für PI IV, Universität Mannheim, während des Programmiermethodik Praktikums im SS 2002
- JAIS Version 0.2 (Konzeptpapier), vom 2. Mai 2002

Inhaltsverzeichnis

0. Modifikationen

Die Modifikationen werden in diesem Abschnitt zusammengefaßt.

Version 0.1

Initialversion nach LS Vorgabe

Version 0.2

Layoutänderung

Version 0.3

Inhaltsverzeichnis wurde hinzugefügt und der Programmentwicklungszeitablauf ergänzt

1. Geplante Architektur

Die vorgesehene Architektur umfaßt eine Modularisierung des soccer clients, welche eine Kapselung folgender client-Elemente vorsieht:

1.1 Kommunikations-Modul

Das Kommunikations-Modul wird für die Kommunikation zur Außenwelt - mit dem RoboCup Server zuständig sein.
Es wird für den Verbindungsaufbau, den Empfang, wie auch für das Versenden von Daten zuständig sein.

Anmerkung

Weiterhin wird an das Kommunikations-Modul ein Parser angegliedert, welcher die vom Servers stammenden Daten entsprechend den Anforderungen des zu erstellenden Clients aufbereitet. Die Aufbereitung wird natürlich auch für Daten, die vom Client an den Server gehen, vorgenommen.
Der Parser wird noch eine Klassifizierung vornehmen, die bereits im vornherein klärt, welche Server-Daten, sich an welches Untermodul des Client richtet.

1.2 Weltmodell

Die Aufgabe dieses Moduls wird darin liegen, die vom Kommunikations-Modul stammende Daten für spätere Entscheidungen zu interpretieren. Das bedeutet, daß z.B. aus gelieferten Positionsdaten Abstände, Richtungen, etc. bestimmt werden.

1.3 Visualisierungs-Modul

Dieses Modul dient der Visualisierung und der Beobachtung
eines jedes aufgerufenen Spielers.
Mögliche Fehler lassen sich durch ihn in der Darstellung erkennen und beheben.

1.4 K.I.-Modul

Das Kernmodul des clients stellt dieses Modul.
Die ‘Künstliche Intelligenz’ wird der Spieler werden. Durch ihn und in Ihm werden Entscheidungen, durch gegebene Bedingungen, gefällt und dadurch Handlungen hervorgerufen, welche entsprechend weitergegeben werden.

2. Strategieüberlegungen

Vorläufiges Ziel:
Das Schlagen eines Prototypen des simple-Clients der durch den Lehrstuhl gestellt

wird.

Mittel:

Die Berücksichtigung/Implementierung folgender Punkte in der K.I.:
um es 'kurz' zu machen, ganz im Sinne der portugisischen Gewinner des RoboCup World 2000:

```
1) Stay always in a useful position for the team!
2) Keep track of the ball!

This is enough to develop a good team. However many more
guidelines are needed to develop a really very good team:

3) When the ball is not important scan the field (keeping
    track of teammates and opponents)
4) Don't speak unless you have anything important to say!
5) Communicate important events and information if you believe
    you are the one that know them better!
6) Prepare your actions by predicting it's outcome! (look out
    there and imagine possible scenarios!)
7) At each moment, try to reduce opponent's possible options
     and increase teammate's possible options!
8) To do different things in the field (attack, defend, etc.),
    different team behavior is needed!
9) Don't waste energy unless it is essential for the team!
10) Try to put yourself inside others' minds (teammates and
    opponents) and try to imagine what they are thinking!
11) If you can score, shoot! If not, if you may pass to someone
who can score, pass! If not,  if you can pass to someone
that may pass to someone that may pass to someone that can
    score, pass! If not, ...
12) Don't hold the ball or dribble unless you really do not
    have anything else to do!
13) Don't pass backwards!
14) Forward the ball to the back of the defense whenever
    possible!
15) Cross the ball early from the flanks (wings) to the back of
    the defense!
16) Allow the ball to work for you! Don't you work for the
   ball!
17) Maintain a compact team shape!
18) Use the open space to attack and limit it to defend!
19) Forwards (and other players) should change positions to
    confuse the opponent's defense!
20) Wing forwards should stay wide to bring defender out
   of the middle of the field!
21) When the ball is lost, all players must think defensively!
22) Challenge opponents (in control of the ball) very fast so
    that they may not advance in  the field!
23) Limit opponent's choise by reducing his space and covering
    his pass lines!
24) As a defender don't do risky passes or risky interceptions.
     Be cautious! Let the opponent commit himself first!
25) The closer the play is near your own goal, the tighter the defense must be!
26) What you do without the ball is as important as what you do with the ball!
27) Use players with different capabilities for different positions!
28) Change your playing style according to the game situation and opponent!
29) Be flexible and adaptative. Don't use rigid rules for anything!
30) Train against different opponents and analyze deeply the games!
```

3. Aufteilung in der Gruppe

Definitionen der Anforderungen an Arbeitsbereiche und deren Entwickler:

3.1 System-Architekt (1/8 Kraft), Stephan Diederich

- gemeinsame Interface Deklarationen implementieren
- Interfaces definieren
- CVS Verwaltung, Tasks etc.
- Springer (bzw. springen lassen)

3.2 Dokumentarik/Qualitätssicherung (1/4 Kraft), Raul Zanzinger

- Pflichten-/Lastenheft
- Einzelne Problemdokus liefern
- Programmierrichtlinien
    - Kommentare -> JAVADoc
    - classes, etc.
- Codeprüfung (Q.S.)
- Bekanntgabe von Infos vor Treffen
- Homepage verwalten (mit System-Architekt.), entsprechend Protokolle erstellen/veröffentlichen

3.3 Programmierer (6 5/8 Kräfte), für

- Kommunikation, Oliver Beck, Raul Zanzinger
    - Parser
    - Modulschnittstellen
- K.I., Oliver Beck, Claus Kestel, Christian Traub
    - Entscheidung
    - Handlung
- Visualisierung, Sven Scheffelmeier
- Weltmodell, Stephan Diederich, Florian Münz
- Main Modul, Stephan Diederich

4. Zeitplanung

Die Zeitplanung wird sich an den gegebenen vorläufigen Terminen des Praktikums orientieren.
Das heißt, daß die Phasen folgenden Verlauf nehmen:

4.1 Entwicklungsphasen

- bis 06.05.2002
  Ende der Definition

- bis 20.05.2002
  Ende von Analyse und Design

- bis 03.06.2002
  Implementierung

- bis 24.06.2002
  Ende Implementierung

- bis 08.07.2002
  Schlußphase

4.2 Programmentwicklungszeitablauf

1. MainModul
2. Kommunikationsmodul
3. Modulschnittstellen
4. Weltmodell
5. Visualisierung
6. K.I. Modul