A Software Architecture for Four-Legged Robots

CHRISTIAN GUTTMANN

Email: guttmann@gmx.de

Abstract

A prototype of a software architecture for four-legged robots is described. An integration of higher layer design, basically a model of decision making and global world modelling for robots in a four-legged football domain, was developed. This domain is dynamic and nondeterministic and thus requires dealing with uncertainty. Because the robots play together in one team, aspects of multi-agent development are applied. The suggested SchemeAnalyzer combines reactive and deliberative planning and provides this platform in realtime. This hybrid system, together with Team Sweden’s very first version of a robot soccer architecture, came into use during the RoboCup’99 competition in Stockholm. The higher level decision-maker is inspired by cognitive models of human decision making. Strategic plans are generated and with reactive aspects taken into account, behaviours are chosen based on motivation. Additionally, an introduction of the RoboCup league of Sony four-legged robots, which includes an overview about software architectures from the other eight participating teams in the Sony league, is described.

This thesis corresponds to the effort of 20 full-time working weeks.

Keywords: realtime decision making, multi-agent system, cognitive model, legged robot, RoboCup
Royal Institute of Technology, Stockholm, Sweden
School of Electrical Engineering and Information Technology
Department of Computer and Systems Sciences (DSV)

Stockholm University, Stockholm, Sweden
Faculty of Social Sciences
Department of Computer and Systems Sciences (DSV)

University of Paderborn, Paderborn, Germany
Faculty 17
Department of Computer Science
To my family, who supported me during my whole period of study.
I wrote some parts of my thesis on the remarkable islands of the Stockholm archipelago, which gave me a lot of inspiration and creativity.
Contents

List of Figures

List of Tables

1 Introduction
1.1 Background .......................................................... 1
1.1.1 Team Sweden ....................................................... 1
1.1.2 Robot World Cup (RoboCup) ................................. 3
Pushing the State-Of-The-Art ......................................... 3
The Dream ................................................................. 4
A Landmark Project ....................................................... 4
A Standard Problem ...................................................... 5
RoboCup Rescue .......................................................... 5
Leagues and Rules ......................................................... 5
Basic Rules for the Sony Legged League ............................ 6
Sony Legged Rules ......................................................... 6
1.1.3 Goal for Team Sweden ............................................. 7
1.1.4 Commercial Benefit ............................................... 7
1.2 Objective ............................................................... 7
1.3 Methodology .......................................................... 8
1.4 Structure of the Thesis ............................................... 9

2 The Four-Legged Robot Domain ....................................... 11
2.1 Domain Description .................................................. 11
2.1.1 Agents’ Resources ................................................ 11
2.1.1.1 Description of the Sensors ................................. 12
Colour CCD Camera ...................................................... 12
Infra Red (IR) ........................................................... 14
Feet and Head Switches ............................................... 14
Microphone and Speakers ............................................. 14
Motion Detecting Sensors ............................................. 15
CONTENTS

2.1.1.2 Description of Degree of Freedom (DOF) ....... 15
2.1.2 Characteristics of the Environment .................. 15
    Game ............................................. 15
    Size of Field ................................... 15
    Significant Colours .............................. 16
    Borders and Protecting Walls .................... 16
    Landmarks—Description and Meaning ............... 16
2.2 Software ........................................... 17
    2.2.1 Open-R ....................................... 17
        2.2.1.1 Open-R Architecture ................... 17
            Main Features of Open-R Architecture ...... 17
    2.2.2 Aperice ...................................... 18
    2.2.3 Programming Language ......................... 18
    2.2.4 Examples ..................................... 18
    2.2.5 Robot Simulator .............................. 19

3 Robot Soccer Agent .................................... 21
    3.1 Strategy ........................................ 21
    3.2 Architecture ................................... 22
        3.2.1 Lower Layer ................................ 23
        3.2.2 Middle Layer ............................... 23
            3.2.2.1 Local Perception Space (LPS) ....... 23
            3.2.2.2 Basic Behaviour Module (BBM) ...... 24
        3.2.3 Higher Layer ................................ 24
            3.2.3.1 Local Perception Space To Global Map (LpsToGm) 24
            3.2.3.2 Global Map (GM) ....................... 25
            3.2.3.3 Reactive Planner (RP) ................. 25
        3.2.4 Communication between Modules ............... 26
        3.2.5 Motivation .................................. 26
        3.2.6 Use of Communication ...................... 28
        3.2.7 Multi-Agent Co-operation ................... 28
        3.2.8 Reinforcement Learning .................... 29
        3.2.9 Pronouncers ............................... 29
        3.2.10 Analogy with Intelligent Forms of Life .... 30

4 Algorithms and Implementations ..................... 31
    4.1 Higher Layer—Overview .......................... 31
    4.2 LpsToGm Module .................................. 32
    4.3 GM Module ...................................... 34
    4.4 RP—SchemeAnalyser .............................. 36
CONTENTS

4.4.1 The Command ........................................ 39
4.4.2 Mathematical Formalisation ........................ 39
4.4.3 Time Out Variables ................................. 41

5 Empirical Studies ................................. 43
  5.1 Observations during the Games ................... 43
  5.2 Other Leagues and their Pertinence for the Four-Legged League . 48
  5.3 Experiments with the RoboCup Challenge .......... 50

6 Empirical Results ................................ 53
  6.1 Overview ........................................ 53
  6.2 Complexity, Completeness and Optimality .......... 56

7 Previous and Related Works .................. 59
  7.1 Objectives of RoboCup, AI and Robotics ............. 59
  7.2 Nine Teams ⇒ Nine Approaches ..................... 60
  7.3 ARAIBO .......................................... 61
    7.3.1 Overview ................................... 61
    7.3.2 Specialities ................................ 62
  7.4 UNSW United ..................................... 63
    7.4.1 Overview ................................... 63
    7.4.2 Specialities ................................ 65
  7.5 Upennalizers ..................................... 65
    7.5.1 Overview ................................... 65
    7.5.2 Specialities ................................ 65
  7.6 The CMU Trio .................................... 66
    7.6.1 Overview ................................... 66
    7.6.2 Specialities ................................ 66
  7.7 Les 3 Mousquetaires ................................ 67
    7.7.1 Overview ................................... 67
      7.7.1.1 Locomotion ................................ 67
      7.7.1.2 Vision .................................. 67
      7.7.1.3 Strategy ................................ 68
    7.7.2 Specialities ................................ 68
  7.8 Humboldt Heroes .................................. 68
    7.8.1 Overview ................................... 69
    7.8.2 Specialities ................................ 70
  7.9 McGill Reddogs .................................. 70
    7.9.1 Overview ................................... 70
    7.9.2 Specialities ................................ 71
  7.10 Baby/Tigers-99 .................................. 71
CONTENTS

7.10.1 Overview ............................... 71
7.10.2 Specialities ......................... 72
7.11 Comparison of Software Architectures ............ 72

8 Conclusions and Reflections ..................... 73
  8.1 Conclusions ............................. 73
  8.2 Future Contributions ........................ 74
    8.2.1 Development of the Robot Soccer Agent ... 74
    8.2.2 Usefulness for the Simulator League ......... 74
    8.2.3 Usefulness for other Complex Domains .......... 74
    8.2.4 Autonomous Design Add-on ................. 74
    8.2.5 Deliberative Planning .................. 75
    8.2.6 The Scheme Analyser and Quantum Computers .... 75
    8.2.7 Emotional and Intentional Agents in the RoboCup domain ... 75
    8.2.8 Less Opportunity—More Risk ................ 76
    8.2.9 Sociotronic—Society and Agents ............... 76
    8.2.10 Motivation—equivalent or not? ............ 76
    8.2.11 Just Entertainment? .................... 77
  8.3 Future Work ............................... 77
  8.4 Reflections ................................ 77
    8.4.1 Additional Challenges .................. 78

Acknowledgements ............................... 81

Bibliography ................................ 84

A Press References ............................. 95

B Internet Resources ........................... 97

C Glossary ...................................... 101
List of Figures

1.1 Logo of Team Sweden. .................................................. 2
2.1 Some of the sensors and the use of the tail. ...................... 13
2.2 The visual sensors and the audio output resources. ............ 13
2.3 Description of some additional technical features at a glance. .. 14
2.4 The official RoboCup field for the Sony four-legged robot league. 15
2.5 The differently coloured markers (blue for one team and red for the other) on the robots’ cover distinguish between team mates and opponents. .................................................. 16
2.6 The 50-cm high white borders were built to protect against visual disturbance from the outside of the field. ..................... 16
2.7 A landmark. ................................................................. 17
3.1 Overview of the architecture. .......................................... 22
3.2 System overview and the communication channels between the modules that is implemented in each robot. ....................... 27
4.1 The structure of a Locally perceived object (Lpo). ............... 32
4.2 The structure of a Global map object (Gmo). ....................... 33
4.3 The static objects and their global locations. ..................... 33
4.4 Some typical GMI. ......................................................... 34
4.5 The discretised field. ...................................................... 35
4.6 A description used in the SchemeAnalyser for the “KickBall” situation. ................................................................. 36
4.7 A “goal” script (default). ................................................ 37
4.8 This part of the SchemeAnalyser describes the choice of the behaviour. ................................................................. 38
4.9 The structure command that is sent to the BBM. ................... 39
5.1 The four-legged robot teams. .......................................... 44
5.2 Initial game situation. ................................................. 44
LIST OF FIGURES

5.3 Typical game situation. Team Sweden vs. Osaka University—Sweden lost 2:1. ........................................ 45
5.4 Another game situation. .................................................. 46
5.5 A real game—a robot lost his balance and fell down. .......... 46
5.6 Preparation for the games. .............................................. 47
5.7 A shootout was made, if a game ended in a tie [Sony Rules, 1999]. 47
5.8 Robots are changing the ball, and unfortunately also each other. 48
5.9 Simulator league. ....................................................... 49
5.10 Middle size league. .................................................... 49
5.11 Small size league. ...................................................... 50
5.12 The experimental situation 1. ....................................... 51
5.13 The experimental situation 2. ....................................... 51

6.1 A typical motivation profile. .......................................... 54
6.2 A whole typical motivation flow. .................................... 55
6.3 The executed behaviours and their corresponding motivations. 56

7.1 Logo from the team ARAIBO. ....................................... 61
7.2 System architecture of ARAIBO robot. .............................. 62
7.3 Logo from the Australian team, UNSW United. ................. 63
7.4 System architecture of UNSW United. .............................. 64
7.5 Logo from the Upennalizers. ........................................... 65
7.6 The system architecture from the Upennalizers. .................. 66
7.7 Logo from the CMU trio. ............................................... 66
7.8 Logo from the French team. .......................................... 67
7.9 Logo from the German team. ......................................... 68
7.10 Overview of the architecture of the German team. ............ 69
7.11 Logo from the Canadian team. ..................................... 70
7.12 Logo from team of the University of Osaka. ..................... 72
List of Tables

1.1 Main participants of Team Sweden.......................... 2
1.2 Comparison of chess and the RoboCup domain............... 5

2.2 The technical description of the robot at a glance.......... 12

4.1 The different motivation levels that were used in this version of
    the SchemeAnalyser........................................ 37

6.1 The data table for the chosen example....................... 53
Chapter 1

Introduction

This chapter will provide a comprehensive overview, since this work was done in a cutting edge technology environment. Few four-legged Sony league related papers have been published so far. Hence, this environment needs to be detailed with background information. The interest in using artificial intelligence (AI) systems is increasing in many areas. This is one reason that many standard problems are needed. One current event is RoboCup—an initiative that promotes AI research. We developed a robot agent that played soccer on the RoboCup'99 in Stockholm, Sweden. For this purpose the SchemeAnalyser was applied. The SchemeAnalyser is a concept for decision making on a higher level, which means it decides about complex and strategical actions that will be executed. It keeps track of the environment and it contains a reactive part as well as a deliberative part. The hybrid system of both of these parts provides the robot with a decision module and allows it to make rational decisions. The concept of motivation is used to provide the agent with a base for rational decision making. Integration by implementing a higher level conception into Team Sweden’s software architecture for four-legged football playing robots was done. Additionally, the reader will find a description about the project background.

1.1 Background

1.1.1 Team Sweden

Team Sweden is the Swedish national effort to develop a four-legged robot football team and consists of academic researchers. It combines the scientific interests of professors, undergraduate and graduate students in the field of AI. As we started our project, a team description was made [Boman et al., 1999] that described our background, goals and efforts.
Figure 1.1: Logo of Team Sweden.

Within six weeks we developed a football playing robot team. Table 1.1 shows the structure and members of Team Sweden.

<table>
<thead>
<tr>
<th>Team Captain and Co-ordinator of Team Sweden</th>
<th>MAGNUS BOMAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stockholm University and the Royal Institute of Technology</td>
<td>CHRISTIAN GUTTMANN</td>
</tr>
<tr>
<td></td>
<td>MAGNUS ERICMATS</td>
</tr>
<tr>
<td></td>
<td>ALEXANDER TOLLET</td>
</tr>
<tr>
<td></td>
<td>HÅKAN YOUNES</td>
</tr>
<tr>
<td></td>
<td>JOHAN KUMMENEJE</td>
</tr>
<tr>
<td></td>
<td>DAVID LYBÄCK</td>
</tr>
<tr>
<td>University of Örebro</td>
<td>ALESSANDRO SAFFIOTTI</td>
</tr>
<tr>
<td></td>
<td>KEVIN LÉBLANC</td>
</tr>
<tr>
<td></td>
<td>DANIEL PETERSSON</td>
</tr>
<tr>
<td></td>
<td>MIKAEL KARLSTRÖM</td>
</tr>
<tr>
<td></td>
<td>ZBIGNIEW WASH</td>
</tr>
<tr>
<td>University of Karlskrona/Ronneby</td>
<td>PAUL DAVIDSSON</td>
</tr>
<tr>
<td></td>
<td>RICHARD KRONÉLT</td>
</tr>
<tr>
<td></td>
<td>RICHARD KREJSTRUP</td>
</tr>
<tr>
<td></td>
<td>STEFAN JOHANSSON</td>
</tr>
<tr>
<td></td>
<td>MAGNUS BROBERG</td>
</tr>
<tr>
<td></td>
<td>MARTIN FREDRIKSSON</td>
</tr>
<tr>
<td></td>
<td>PER LIDÉN</td>
</tr>
<tr>
<td></td>
<td>PEDER LÖNN</td>
</tr>
<tr>
<td></td>
<td>ERIK MATTSSON</td>
</tr>
<tr>
<td></td>
<td>MARTIN RUDEFELDT</td>
</tr>
<tr>
<td></td>
<td>GUSTAV SOCHACKI</td>
</tr>
<tr>
<td></td>
<td>MAGNUS WITTSTRÖM</td>
</tr>
<tr>
<td></td>
<td>ANDREAS FOLKLER</td>
</tr>
</tbody>
</table>

Table 1.1: Main participants of Team Sweden.

Christian Balkenius from the University of Lund and Martin Nilsson from the Swedish Institute of Computer Science (SICS) assisted us technically in the beginning of the project. The participants of Team Sweden came from
1.1. BACKGROUND

Italy, Canada, Australia, Sweden, Poland, and Germany.

Because of the distributed nature of this project it was necessary to organise and unify the implementation produced by each of the universities. Where possible, the unifications were made immediately within these six weeks or at the latest in the last preparation phase in Stockholm. In the last ten days before and during the competition, most of the team members worked together in Stockholm, where the RoboCup competition took place. During that time the team members in Stockholm worked without sleep for several days. In spite of some misunderstandings that were caused by the distributed nature of the team, the participation on the RoboCup'99 was a successful attempt of doing team research.

1.1.2 Robot World Cup (RoboCup)

The driving force behind the RoboCup games is the RoboCup Federation, an international organisation that is headquartered in Tokyo. The RoboCup Federation is not only a research organisation; they are also the administrative organ of the annual competition and conference.

The first RoboCup'97 championship games took place in Nagoya, Japan and 40 teams from 25 countries participated. The second international RoboCup'98 games were held in Paris, France and attracted 60 teams from 30 countries. In Stockholm, Sweden the latest RoboCup'99 took place and again the number of participants increased to 90 teams from 35 countries. Thousands of researchers were involved. The composition and structure of the leagues also changed. In Nagoya, the RoboCup included the simulation league and the middle size league. The four-legged robots, sponsored by Sony, played football for the first time, at the RoboCup'99 championships in Stockholm.

The next RoboCup2000 games will be in Melbourne, Australia, hosted by the RMIT University. The next IJCAI will be held in Seattle, United States— together with the RoboCup2001.

Pushing the State-Of-The-Art The RoboCup competition promotes AI research and robotics, by offering a publicly appealing and conspicuous challenge. A significant future application of AI and robotics research is the development of robots that are able to interact and co-operate.

There are different ways to promote engineering research. One way is the development of specific applications. Another way is to set a significant long-term goal. If such a goal is accomplished, it has deep social impact and hence is called a grand challenge project. A soccer-playing robot itself does not generate

\footnote{The distances of the universities comprise up to 600 kilometers.}
significant social and an economic impact. However, the accomplishment is the major achievement of this field. RoboCup is a landmark project as well as a standard problem.

**The Dream** The RoboCup initiative proposed the ultimate goal of the RoboCup competition to be stated as follows:

> By mid 21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, which complies with the official rule of the FIFA [FIFA, 1999], against the winner of the most recent World Cup [RoboCup, 1999].

They suggest this goal to be one of the grand challenges shared by robotics and the AI community for the next 50 years. This goal may sound overly ambitious, by the given state-of-the-art technology today. Nevertheless, it is important that such a long-range goal be claimed and pursued. It took only 50 years from the two Wright brothers’ first aircraft to the Apollo mission to send a man to the moon and safely return him to earth. Also, it took only 50 years, from the invention of the digital computer to the Deep Blue, which beat the human world champion in chess. A speech recognition system that is better than human ears and brain [Berger, 1999] was recently developed. It is recognised, however that building a humanoid soccer player requires a lengthy period and the extensive efforts of a broad range of researchers. The goal might not be accomplished in the near future.

**A Landmark Project** A successful landmark project claims to accomplish very attractive and broadly appealing goals. The most successful example is the Apollo space program [Kennedy, 1961].

The important issue for the landmark project is to set the goal high enough so that a series of technical breakthroughs are necessary to accomplish the task. In addition, the set of technologies that are necessary to accomplish the goal have to form the foundation of the next generation of industries.

The RoboCup committee proposed a more modest goal, “to develop a robot soccer team, which play like human players” [Kitano and Asada, 1998]. This goal can easily create a series of well-directed subgoals. The first subgoal to be accomplished in the RoboCup is “to build a real and software robot soccer team, which plays reasonably well with modified rules”. Even to accomplish this goal will undoubtedly generate technologies, which impact a broad range of industries.
1.1. BACKGROUND

A Standard Problem  Another aspect of the RoboCup is that it is a standard problem so that various theories, algorithms, and architectures can be evaluated. One example for a standard problem is computer chess. Several search algorithms were evaluated and developed using this domain. With the recent accomplishment by the Deep Blue team, whose development beat the world champion Garry Kasparov, the computer chess challenge is almost accomplished. The evaluation of a progress was clearly defined. The rating of the system is a progress measure of the research. However, as computer chess is about to complete its original goal, a new challenge is needed. The challenge needs to promote a set of technologies for the next generation of industries. The RoboCup committee believes that the RoboCup fulfills such a demand.

<table>
<thead>
<tr>
<th>Properties of Environment</th>
<th>Chess</th>
<th>RoboCup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Deterministic</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Episodic</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Static</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Discrete</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

Table 1.2: Comparison of chess and the RoboCup domain.

Table 1.2 describes the difference of domain characteristics between computer chess and the RoboCup based on [Russell and Norvig, 1995]. The RoboCup is designed to meet the need of handling real world complexities, though in a limited world, while maintaining an affordable problem size and research cost. The RoboCup offers an integrated research task, covering broad areas of AI and robotics. Such areas include: realtime sensor fusion, reactive behavior, strategy acquisition, learning, realtime planning, multi-agent systems, context recognition, vision, strategic decision-making, motor control, intelligent robot control, communication and many more.

RoboCup Rescue  RoboCup Rescue is a secondary domain for RoboCup [Kitano et al., 1999]. It is a research and development initiative, promoting innovation. It complements with features missing from RoboCup soccer. RoboCup Rescue has both simulation and real robot aspects.

Leagues and Rules  Robot World Cup competition is the central pillar of the World Cup Initiative’s activity, where researchers can come together and evaluate research progress. Currently, the RoboCup consists of the:

- Simulator League
- Small Robot League
• Full Set Small Robot League with eleven robots per team

• Middle Size Robot League

• Sony Legged Robot League (Sponsored by Sony)

• Humanoid League (From 2002, demonstration may take place before 2002)

• TeleOperation Track (To be announced)

• RoboCup Commentator Exhibition

• RoboCup Junior

• Lego Competition

Basic Rules for the Sony Legged League  The rules of the competition [Sony Rules, 99] are similar to soccer rules [FIFA, 99]:

• Each team is comprised of three robots (including a goalkeeper).

• The playing time is ten minutes per half, with a ten-minute break at halftime.

• The winner team is the team that scores the most goals. In the event of a tie, a sudden death penalty kick competition will determine the winner team.

• If the opposing teams’ robots are damaged or the game is excessively rough (whether intentional or not), penalties may be imposed to the offending robot.

Sony Legged Rules  The following rules are applied at RoboCup competitions:

• The programmer teams may not modify the robots’ hardware.

• The robots may not be manipulated by remote control.

• Additional sensors may not be installed on the robots.

• If the game comes to a halt for ten seconds or more, the referee may pause the game and re-introduce the ball at the nearest feasible point of the football field. If necessary, the referee may change the position of robots during the game.
1.2. OBJECTIVE

1.1.3 Goal for Team Sweden

The main goal for Team Sweden is to develop robots that play intelligent soccer. The first subgoal to be accomplished in the RoboCup is to build real robot soccer teams, which play reasonably well with modified rules. Team Sweden is particularly interested in the problems of

- Dealing with the inherent uncertainty in the four-legged domain.
- Integrating cognitive processes with sensory-motor processes.
- Integrating action and perception in a complex and dynamic environment.

1.1.4 Commercial Benefit

Further development of the Sony entertainment products could be made with our solutions and ideas. Due to the contract with Sony, any idea or program may be published and it can be used for further development. An analysis of the implementations that were made to the robot might be relevant for further development of Sony’s products.

The basis for a useful commercial purpose is that the robot could improve its behaviour. An open architecture is supplied and thus space to add advanced features to the robot, e.g. a communication module could be used to make two robots recognise and play with each other. Maybe with some extra features they would be able to learn from each other. Another example is that competence modules can easily be added to the reactive planner and hence reflect a broader and more complex spectrum of behaviours.

1.2 Objective

The main goal of this thesis is to investigate if a designed higher-layer model—more particularly a concept of decision making (SchemeAnalyser) and global world mapping, can be applied efficiently in the four-legged robot domain.

Several subgoals had to be considered:

1. To partly implement the theoretical concept of the higher layer, mainly the SchemeAnalyser.
2. To integrate it with the system architecture of Team Sweden’s robot system.
3. To develop a module that extracts global information.
The SchemeAnalyser uses the complex global and the local information in order to make decisions under uncertain conditions and in realtime in the four-legged robot domain. To fulfill this objective, an organisational effort about student resources in Stockholm had been pursued.

1.3 Methodology

Another challenge during my thesis work was to establish and improve the communication between the universities that took part in the project. Team Sweden is an international group of researchers who are represented by Stockholm University, the Royal Institute of Technology, the University of Örebro and the University of Ronneby/Karlskrona.

Immediately after the very first general meeting in the middle of June 1999 in Stockholm, we built a virtual discussion group to overcome the distances between the universities. It was necessary to build the appropriate communication base between the Swedish universities. The first six weeks before the competition started, we put a lot of effort into the establishment of communication and into the communication itself. We utilised the Internet for our purposes and used different kinds of communication tools, such as Internet, e-mail, FTP, fax, SMS and sometimes ordinary post. The communication was maintained by using this virtual group extensively. I received 10-15 messages every day and wrote about the same number of messages concerning our work progress in Stockholm to Team Sweden. The team members used these resources from the beginning of the project until the start of the competition.

We used the FTP server that was located in Stockholm, to exchange the latest findings and updates from each University. Further on, I proposed to use a calendar during the whole project to track the workflow of each University group and the members of each group. The calendar was also stored on the FTP server, so that every member could easily access and update it. All of us were able to use this resource efficiently. The calendar had two very useful applications,

1. During the project every member could take a detailed look at what was being achieved and

2. After the RoboCup competition the calendar was very useful to review our own work, as well as to understand the whole project flow.

During this project I was the contact person for technical and organisational issues in Stockholm. In addition, together with my supervisor Magnus Boman,
1.4. **STRUCTURE OF THE THESIS**

I arranged meetings and discussions in Stockholm and Örebro. These arrangements were made together with Alessandro Saffiotti from the University of Örebro and Paul Davidsson from the University of Ronneby/Karlskrona. In the beginning of the project, I was positioned as coach in our team. Later on, Alessandro Saffiotti joined as parallel coach at the University in Örebro. Based on our positive attitudes towards the project progress, we worked together in a close and co-operative manner and were the main contact persons for all Team Sweden members.

In these six weeks, before the competition started, I spent about half of my time on organisational issues. Together with Alessandro Saffiotti and Paul Davidsson we succeeded in building an information network that fit our requirements. I spent another large part of this time reading literature and exploring Internet resources about robotics and AI. During that time, we implemented some rough ideas. We also invested plenty of time getting familiar with the Operative System (OS) Aperios and the Application Programmer’s Interface (API) Open-R. A brief introduction to Aperios and Open-R can be found in section 2.2.1 and section 2.2.2.

Based on this, we programmed robots that play soccer. At the top of our hierarchical architecture, a model of decision making that fits the other approaches of the Swedish team system, was evaluated. Eventually, we were prosperous in showing stable and working robots that played football at the RoboCup'99 in Stockholm. We were on place seven out of nine in respect to the RoboCup Challenge. The team members in Stockholm designed the reactive planner and the higher layer modules, whereas the lower layers where basically produced at the University of Örebro. The University of Karlskrona/Ronneby did most of their research with a robot league simulator, based on a 3D Quake engine. In the final analysis, every team member took active part in any change or optimisation that was done during our project development.

The field and the four robots stayed in Stockholm, so that we could do additional experiments and complete our work. During this time, we improved and remodelled parts of my implementations and compared results.

I was participant on the International Joint Conference on Artificial Intelligence 1999 (IJCAI'99) in Stockholm and could use experiences from this event as an additional inspiration for my work.

1.4 **Structure of the Thesis**

My thesis is structured in three large parts: theoretical background, empirical experiments and results, and conclusions. Chapter two gives a more detailed description about the resources and the domain of the four-legged robot, where
our programs came into use. A reader who is familiar with the RoboCup domain and the robots can skip this chapter. Chapter three is meant to give an overview of the general architecture of the “Team Sweden” robot soccer agent. It examines all important modules that were made during our project and it describes the foundation of the whole robot agent. Algorithms and implementations are explained in chapter four. Motivation-based approaches and their implementations are explored in detail. In chapter five the reader can find details on the experiments made. The results can be found in chapter six. Nine teams participated in the very first official RoboCup legged robots league. All of them used a different approach. These approaches are described in chapter seven. Final discussions and future contributions can be found in the last chapter. Reflections are added as feedback about the project work and context information.
Chapter 2

The Four-Legged Robot Domain

This chapter describes the technical and environmental characteristics of the four-legged robot domain. While approaching an appropriate way of developing a robot agent, we came up on several trains of thoughts and ideas. We started to specify the resources of the robot and characteristics of the robot’s world. Maybe there is no agent or robot architecture that could be called the perfect solution, better than all others are and fitting to every domain. Instead, architectures should be built individually for a class of agent problems [Maes, 1997]. The agents’ resources and the particular characteristics of the task and environment can define a problem class. The next two subsections specify the resources and the characteristics of the four-legged robot domain.

2.1 Domain Description

Most of the description of this domain was defined on the first demonstration and test competition at the RoboCup’98. Some of the parts in these sections are based on the publication from [Veloso et al., 1998] about the experiments and first settings with the first competition in this league.

2.1.1 Agents’ Resources

Table 2.2 introduces the technical resources of the robot.
Table 2.2: The technical description of the robot at a glance.

2.1.1.1 Description of the Sensors

The use of sensors, microphones and speakers that roughly correspond to the sensory organs of a living creature makes it possible for the robots to communicate and to react to external stimuli. The advanced sensor system increases the possibilities for an autonomous behaviour of the robot.

**Colour CCD Camera** The camera is used as the main visual input resource to get information about the environment. It detects the shapes and colours of objects when approaching or avoiding them.

The maximum resolution is 180,000 pixels within a predefined frame, but to make reasonably fast calculations, a frame of 88x60 pixels is evaluated. The main characteristic of the vision system is the design of an image processing software package that allows object retrievals, processed at about 14 or 15 frames per second. The camera generates NTSC video signals and a YUV space is an
2.1. **DOMAIN DESCRIPTION**

![Figure 2.1: Some of the sensors and the use of the tail.](image1)

![Figure 2.2: The visual sensors and the audio output resources.](image2)
Figure 2.3: Description of some additional technical features at a glance.

appropriate format to deal with extracted frames.\footnote{NTSC stands for National Television Standards Committee. The NTSC standard for television defines a composite video signal with a refresh rate of 60 half-frames (interlaced) per second. Each frame contains 525 lines and can contain 16 million different colours.} The YUV space enables to send video signals that can be processed by colour TV and by black and white TV. The gray scale is encoded in the Y plane, while the combination of U and V defines a colour point in the U-V plane.

**Infra Red (IR)** The IR is together with two lights placed behind the black-coloured cover on the head. It is a range finder for detecting the distance to objects.

**Feet and Head Switches** The touch sensors recognise the tactile stimulation, like being patted or hit.

**Microphone and Speakers** The mini stereo microphones detect the direction of a sound source and perceive sound from the environment. The miniature speaker allows the robot to generate a broad spectrum of sound. Sound has to be in the PCM format.\footnote{PCM stands for Pulse Code Modulation, which is the process of digitising analogue sound.}
2.1. DOMAIN DESCRIPTION

**Motion Detecting Sensors**  The acceleration sensor maintains the balance and regains balance when the robot recovers from a fall. The angular velocity sensor detects rotary motion.

2.1.1.2 Description of Degree of Freedom (DOF)

A total of 18 DOFs (one for the mouth, two for the tail, three for the head, and three for each leg) allows the robot to walk smoothly on four legs as well as to perform other complex motions. The tail was basically used for debugging, that means to visualise the activated state by making the robot wag or rotate its tail.

2.1.2 Characteristics of the Environment

![Diagram of the RoboCup field](image)

Figure 2.4: The official RoboCup field for the Sony four-legged robot league.

**Game**  Three robots play against three. This is the minimum number for each team to enable some advanced team play or collaboration tasks.

**Size of Field**  The playing field, shown in figure 2.4, is 2m x 3m and provides with enough space for six robots to dribble, pass and to navigate.
**Significant Colours**  The important objects in this league are coloured so that a robot can autonomously figure out its environment by visual processes in realtime. The ball is orange and the goals are dark blue and yellow. The floor is dark green (see figure 2.4). All robots from one team have either blue or red markers (see figure 2.5) on their bodies.

![Robot Images](image1.jpg)

Figure 2.5: The differently coloured markers (blue for one team and red for the other) on the robots' cover distinguish between team mates and opponents.

**Borders and Protecting Walls**  The field is surrounded by white walls (see figure 2.6) that protects against significant colours from outside the field.

![Diagram](image2.jpg)

Figure 2.6: The 50-cm high white borders were built to protect against visual disturbance from the outside of the field.

**Landmarks—Description and Meaning**  The six poles in each corner of the playing field conduct to self-localisation. Each pole or landmark has two different colours (see figure 2.7). Three different colours are combined to enable the robots to distinguish between the six poles. The robots need information about at least two different landmarks to calculate a global self-location.
2.2 Software

The Sony four-legged robots are equal in the sense of the hardware-design, the operation system and the kind of programming language. From the technical point of view, all teams have the same starting point to build a software architecture. However, the team members, who are engaged to implement the code, have to learn the new software platform as basic requirement. The OS Aperios and the API Open-R are briefly described in the next two sections.

2.2.1 Open-R


2.2.1.1 Open-R Architecture

Sony is advocating Open-R as the standard interface for entertainment robot systems. Sony’s purpose is to create a flexible hardware architecture. Open-R specifies standards for hardware, electronics, and software architectures. By having an Open-R standard, software and mechanical parts of a software architecture can be used to operate any robot no matter which form it has.

Main Features of Open-R Architecture

Interchangeable Hardware Modules for Various Physical Configurations:

The construction of robots that can be reconfigured to suit multiple uses. For example, a four-legged walking robot could be changed into a two-wheeled rolling robot by simply disconnecting the legs and replacing them with a wheeled module.
Hardware-Related Data Transmission for Plug-and-Play Connectivity:
Internal data allow the robot to recognise the configuration of its entire body, its movement capabilities, and its functions. The robot then uses the data to select the most appropriate control signals for co-ordinating its movement add-ons and disseminates them to the individual add-ons.

Interchangeable Software Modules for New Applications: The software programs that control the robot’s movements and responses have also been made modular and thus it is possible to change a robot’s behaviour by simply replacing one memory card with another card suited for a purpose.

Expandability for Accommodation of New Functions: It is possible to insert functional expansion cards. Through the use of wireless control or network expansion cards, for instance, it would be possible to remotely control a robot or even have a number of robots co-operate to perform tasks.

2.2.2 Aperios
The Aperios system is the operative system and has as main feature object communication. This realtime OS developed by Sony is used as the robot’s operating system. Aperios is well suited to applications that need to operate in real-time, for instance tracking a ball with the camera while advancing toward it, all while sending appropriate feedback. Aperios is a completely object-oriented operating system.

2.2.3 Programming Language
The implementations were programmed in a C++ style language. Because of the communication properties between program modules in Aperios, additional parts had to be implemented in our code. Basically, these sections established and organised the message and data communication between the modules in our program architecture.\(^3\)

2.2.4 Examples
The Sony support crew provided all Sony league participants with examples. These examples introduced the robots’ capabilities and they showed the programming style. After reading the introduction and manuals of Aperios and Open-R, we started to understand the robot by executing and modifying the examples.

\(^3\)See figure 3.2 for our communication architecture.
2.2.5 **Robot Simulator**

Sony developed a simulator to arrange the possibility to work with a hardware independent platform. It simulates the technical specifications and provides with a realtime visualisation of the dog and the playing field. The Team Sweden members at the University of Örebro used this simulator before we received the physical robots. Unfortunately, the simulator proved to be too slow to test our ideas.\(^4\) We could not use it for advanced simulations. One reason was that only one robot can be simulated at a time, but six are in the real game. Even simple motions and the corresponding animations were far too slow to produce satisfying results. Hence, the simulator was almost useless for developing our ideas and we did not continue to use it for our test phases.

\(^4\)We used a 500 MHz Pentium II PC, far above standard at that time.
Chapter 3

Robot Soccer Agent

A robust and effective robot was developed. Robust, because it almost never breaks down completely and effective, because the robot eventually scores goals. Developing adaptivity is another very important aspect of designing an autonomous agent. Unfortunately, due to the lack of time, adaptive behaviour did not come into use during the event itself. Research about autonomous agents is carried out in order to understand principles and organisations that underlie robust and effective behaviour [Maes, 1997]. Parts of the lower layers were developed based on the Open-R interface examples and on research that was done at the University of Örebro [Coradeschi and Saffiotti, 1999], [Konolige et al., 1997], [Saffiotti et al., 1999]. I shall try to explain them as best as I can so that my contributions and the higher layer design for this architecture are presented in an understandable manner. Because of close co-operation between the Swedish University teams, many parts were not produced at one physical place or by one person only. For the reader it is necessary to convey a comprehensive introduction of the whole architecture in order to understand specific parts of it.

3.1 Strategy

During RoboCup’99 we were forced to reduce our tactical and strategical ideas. Mainly, due to some unreliable behaviours that were not examined at the time of the competition.

We had the idea of giving roles to each robot. Thus we had two forwards that could move over the whole field and had no constraints while doing that. They should try to charge the ball and put it into the opponent’s goal. Special goalkeeper behaviour was programmed. The goalkeeper was supposed to defend our goal. The idea was that it should use the two landmarks that were placed
on the middle line of the field for recognising their global position. We chose these landmarks instead of the two landmarks that were placed on the own goal line, because they were easier to detect with the camera. A 180-degree camera view was then not necessary. Unfortunately, in most of the games the active goalkeeper behaviour was replaced by a passive one because some modules worked not reliable. A passive goalkeeper in this case means that it defends the goal by just standing in front of it without any motion at all. Sometimes it turned out that this defending strategy is better, because a part of the goal was statically covered and there was less risk to allow a goal. In some other games we let three forwards play and risked leaving an empty goal without any defence.

3.2 Architecture

A graphical abstract overview of the system architecture is shown in figure 3.1.
3.2. **ARCHITECTURE**

Each robot uses a layered architecture for realtime situation assessment and decision making, inspired by the Thinking Cap, the Saphira architectures [Saffioti et al., 1999], and cognitive models [Anderson, 1996].

Basic behaviours are implemented using fuzzy logic techniques for coping with uncertainty. We used a fuzzy logic behaviour-based hierarchical controller for lower layer controls, which earlier was used in wheeled robots [Konolige et al., 1997]. A reactive planner sent directions to the lower level about the activation of behaviours and thus to put them into a rational order.

### 3.2.1 Lower Layer

The lower layer consists of the “commander” module (CMD). The CMD is the interface between the upper layers’ abstract commands and the physical motion functionalities. In our current version, the CMD accepts commands for head motion, commands for physical displacement, and commands for kicking routines. Head motions can be either “scan” commands for scanning in a predefined motion sequence or a “look-to” command that points the head to a particular direction. For physical displacements, triples of the form $<v_x, v_y, v_	heta>$ are used as commands. The components of the triple represent the linear velocities ($v_x$ and $v_y$) along the principal and lateral axes of the robot and the angular velocity ($v_\theta$). The CMD translates these commands into appropriate walking styles. The CMD is also responsible for specialised kicking routines.

### 3.2.2 Middle Layer

The middle layer has the responsibility of building a set of robust behaviours and is the first visual instance in our architecture. This layer is based on a local Cartesian representation of the space around the robot, namely the Local Perception Space (LPS). The Basic Behaviour Module (BBM) uses the information from the LPS to build a set of behaviours.

#### 3.2.2.1 Local Perception Space (LPS)

The LPS stores the recognised objects in a type of short-term memory. Robot-centred co-ordinates represent these objects. Three mechanisms are applied to update the position of each object:

- Perceptual anchoring, which means, whenever the CCD camera and IR detect and scan the distance and co-ordinates of an object,
- Global information, which means, whenever the robot knows where it’s position is and thus extracts more information about the objects, and

---

1In other publications the term Perceptual Anchoring Module (PAM) may also be applied.
• Odometric clamping, whenever the robot moves.

Each object has a fuzzy predicate anchor, which represents the visual perception support and the clamping update. Odometric clamping update means that an object's positions change appropriately while the robot moves. The predicate anchor has values from $[0, 1]$. If an object is not perceived for some time, the elapsed time of moving objects and the moving of the robot constantly decreases its degree of anchoring.

3.2.2.2 Basic Behaviour Module (BBM)$^2$

Basic behaviours are implemented with fuzzy logic techniques. A simple rule-based program style was used to design control strategies. The input from the LPS is used to calculate fuzzy sets of linear, lateral, and rotational velocities. Context Dependent Blending (CDB) is combined with the output fuzzy sets.$^3$

As a result, the fuzzy sets are converted to a triple $<v_x, v_y, v_t>$ of control values that can be sent to the CMD module.

More complex behaviours can be built based on simple behaviours. Behaviours could contain other sub-behaviours and together they are blended by the CDB mechanism. Some complex behaviours were designed in this way, like “align with ball to the goal” and “walk with ball”.

The BBM gives indications about and emphasise the need to recognise a specific object. The current executed behaviour needs knowledge about certain objects. A truth-value of the predicate needed is used to send an variable using the needed channel in figure 3.2 to the perception module (LPS). A truth-value consists of fuzzy proposition “needed ∧ anchored” and is calculated by the LPS. Then the of LPS is focussing the object with the highest truth value.

The controlling is divided, so that the LPS is responsible for all head motion and the BBM is responsible for all leg motion.

3.2.3 Higher Layer

At the top of this hierarchy, the higher layer was established to calculate global information, based on the Global Map (GM) and to make strategical realtime decisions, provided by a Reactive Planner (RP).

3.2.3.1 Local Perception Space To Global Map (LpsToGm)

The higher layer has the perception module (LPS) as the only resource from which it can retrieve information from the real environment. The LpsToGm

---

$^2$In other publications the term Hierarchical Behaviour Module (HBM) may also be applied.

$^3$See Glossary for explanation.
converts these locally perceived objects from the Cartesian representation to the global representation that is ideally globally mapped.

If the LPS is able to anchor at least two landmarks with enough reliability, a global location can be extracted by calculating the relative distances to the landmarks and by fitting the moving objects into the grid. We applied a generalised concept of evidence grids, that is a probabilistic, finite-element representation of a robot's spatial knowledge [Moravec and Blackwell, 1993] [Martin and Moravec, 1996] [Moravec, 1996b] [Elfes, 1989].

3.2.3.2 Global Map (GM)

The global map is a model from the global view of the robot. A global grid matrix was applied to localise the robot itself and the other moving objects. The GM module is able to extract global information. The module uses the fixed objects, landmarks and goals, in the four-legged robot domain to calculate a set of objects on the GM. Thus it extracts a self-localisation and a localisation of all moving objects on a field. Based on this, global information could be gathered.

A unique characteristic of the GM module is that it can gain global knowledge of the robots’ environment. Typical information is e.g. knowledge about the location of the goals, without really seeing them. A determined location of moving objects that are near the robot’s own goal, is also necessary to provide the decision making module with important data. If the LpsToGm can calculate a global position, the GM extracts the relevant information.

3.2.3.3 Reactive Planner (RP)

The RP uses local information from the LPS as well as global information from the GM to make decisions. A so-called SchemeAnalyser is used. The SchemeAnalyser is a motivation-based decision-maker and it was inspired by cognitive models [Anderson, 1996]. It is a hybrid system that combines reactive and deliberative planning. Information that is extracted from the GM and from the LPS is used to describe a current situation and later on—but in the same time cycle—it is used to describe a whole scheme or script. A time cycle corresponds to one second and defines the time that the higher layer has for disposal to generate an appropriate decision. Some parts of the SchemeAnalyser, in our current implementation, had to be written from scratch and they require a good description of a situation from the view of the robot. All comparisons of the descriptions of a real world situation are made in one cycle and thus they are made quasi-parallel. The SchemeAnalyser extracts the description, which fits the current situation best. This extraction fits to the behaviour sets of the BBM
and will be sent to it. The SchemeAnalyser has an attractive time complexity concerning the input vector for this class of domains—it is linear.  

3.2.4 Communication between Modules

The Aperios and the Open-R, compared with usual C++ programming have the additional property to have object communication. All program modules that were described above communicate with each other. Finally, they are all implemented in the same system as one program package. Figure 3.2 depicts the underlying communication architecture of our Robot Soccer Agent. This figure shows many communication channels which have not been described more precisely. It gives an overview about the additional effort of building the object communication of our implemented architecture.

3.2.5 Motivation

Motivation descends from theories in psychology [Bourne and Eckstrand, 1992] and is used as main variable to make decisions in the SchemeAnalyser. Even though, motivation is an oversimplified term used in our architecture.

Motivation from the psychological point of view is the wish to act, or to satisfy a need (like hunger or thirst). A creature striving for a goal does it because of motivation. The motivation determines the human behaviour together with external stimuli, perception and learning processes as well as with certain skills or premises. Because of that, motivation is an action between personal specific and situational specific properties.

Explanations and theories to systematise motivation processes can be categorised in the following four areas of psychology:

1. *Instinct and urge-based models*: As stated by some significant scientists ([Lorenz, 1963], [Lorenz, 1965], [Lorenz, 1966], [Tinbergen, 1951]), behaviour is not changeable by learning processes, it is innate. Freud's arguments about the activation of behaviours are that they are dependent on specific urges ([1940] [1953]).

2. *Incentive and stimulus-based models*: Learning theories, in particular works from C.L. Hull ([1940] [1943]) explain the activation of behaviours by a general urge that is a combination of specific impulses.

3. *Psychophysiological activation models*: These models are based on neurophysiological science. They claim that an organism is striving for an average

---

\(^4\)See chapter 6 for further analysis.
Figure 3.2: System overview and the communication channels between the modules that is implemented in each robot.
activation level and could also be influenced by learning processes. This opinion is shared by Hebb ([1949] [1958]).

4. Cognitive models: The current theories take the view that cognitive processes (e.g. anticipation of results of events, interpretation of a situation, comparison of reached goals and future goals) play a significant role in motivation events. Representatives of cognitive models of motivations are Lewin ([1947a] [1947b] [1951]), Atkinson and Shiffrin [1968], Atkinson and Feather [1966], and Heckhausen [1980].

As a summary, it is stated that early motivation theories are mechanic-orientated (urge and stimulus), while the current theories are based on individual and more active roles. However, from the technical point of view this term is an oversimplified definition. I chose the term “motivation”, but it should be seen as the term “artificial motivation”. Motivation as well as many other terms (emotion, behaviour, ...) have a long history in human and animal psychology.

3.2.6 Use of Communication

Team Sweden intended to use the speakers and the stereo microphones of the robots to enable the robots’ communication and thus make them co-operate in a more advanced way. The idea was to generate simple sound sequences that could safely transmit some significant information, like “I have the ball” or “our goal has to be defended”.

The speakers can generate sounds up to 30 kHz. We estimated a maximum transmission rate of about eight bands. This modest estimation takes into account the noise from the audience on the RoboCup itself and our lack of experience with our robots. We did not use communication at all, because of the lack of time.

3.2.7 Multi-Agent Co-operation

To define a multi-agent system in this domain, co-operation is already shown if the robots recognise each other via visual recognition. That means that they co-operate if they do not tumble into each other and if they realise that the obstacle in front of them is either an opponent, a teammate or the ball. Real multi-agent actions were rare in the legged league of RoboCup'99 in Stockholm. Almost every team had the idea but none of them were really able to execute their multi-agent ideas this year. RoboCup is one approach to make robots interact. Before RoboCup, most robots and agents were developed just as a single system.

\footnote{5In this case the baudrate is equivalent to bits per second.}
3.2. ARCHITECTURE

[Weiss, 1999]. Next year’s RoboCup 2000 in Melbourne will definitely produce robots with more advanced multi-agent and communication features.

3.2.8 Reinforcement Learning

A learning module was discussed in our architecture, but did not come into use. We tried to implement reinforcement learning in our robots. Our ideas were based on the thesis by Åhlander [1999].

The idea was that the robot would explore an unknown area with reinforcement learning for mobile robots. The area was divided into discrete grid fields. Unfortunately, we were not able to implement any algorithm that was described in Åhlander’s thesis for the competition. Advanced neural network approaches seem to require a high effort of implementation work and we did not have enough time to evaluate this idea.

In spite of this, we concluded that the only way to study reinforcement learning in our robots effectively is to use a simulator, because learning algorithms need to repeat situations very often [Haykin, 1994]. To obtain results and to do experiments with reinforcement with the robots in the real world would take months.

3.2.9 Pronouncers

Pronouncers [Boman et al., 1998] could be used to make decisions at the higher layer, because they are fast enough to be applied in real-time domains [Younes, 1998]. A pronouncer is defined to be an entity giving advice to intelligent agents. It is distinguished from a decision module in that it suggests an extrinsic entity, while a decision module suggests an entity intrinsic to an intelligent agent.

The Swedish research group DECIDE at the DSV is developing pronouncers. This concept could be used in our approach. Our first idea was to design a kind of basic pronouncer. That means it has basic advice-giving functionality—a minimum requirement for a pronouncer. The purpose of that pronouncer is to give advice to intelligent agents in decision situations.

To design a decision problem from scratch is extremely difficult for an agent [Boman, 1998a]. Instead they should use template models in advance for decision situations that can be presumed to occur.

The basic functionality is then restricted to comprise the following:

- load template models
- set/modify values
- evaluate the model, and answer with a single recommended alternative.
We did not use them in this year's approach, because the members of the DE-CIDE group were occupied with programming another team in the simulator league. Even though, the use of a decision support entity, like the pronouncer, would help our Robot Soccer Agent to do better decisions.

3.2.10 Analogy with Intelligent Forms of Life

The whole architecture that is explained above is based on some biological and psychological structures of existing forms of life [Beer et al., 1994]. The nervous structure of invertebrates [Reichert, 1990] may remind of some components of our artificial system. During the preparation phase, we often discussed the need of facing the comparison with intelligent forms of life that already exist on this planet.

We decided that the lower layer would roughly represent the low-level design in a neural system. It co-ordinates the communication and the movement of the legs and the head as well as the visual input. The middle layer could be seen as more advanced and autonomous area. Finally, the higher layer is the complex nervous system and it represents more complex psychological issues, such as decision making and world modelling.
Chapter 4

Algorithms and Implementations

We developed the GM and the SchemeAnalyser based on the defined interface between the lower and the upper layer. Before the competition we were forced to carefully design simulated examples and execute them to calibrate the higher layer for the real application. The members of Team Sweden together wrote 14,000 lines of code in about six weeks. Some parts of the code were still incomplete by time of competition. In spite of this, the implementations were developed effectively. I have been involved in many parts of the final implementation. It is not possible to estimate my individual contributions concerning the implementations made by Team Sweden, because we worked in a close and co-operative manner. The implementations of the lower layers are not explained in detail, because this thesis describes and contains basically information about the higher layer architecture. The LpsToGm, the GM and the SchemeAnalyser were parts of the higher layer and they were produced at the DSV in Stockholm.

4.1 Higher Layer—Overview

The higher layer contains three modules,

- \textit{LpsToGm}, to convert Locally perceived objects (Lpo's) to Global map objects (Gmo's).
- \textit{GM}, to extract global information and build a global map.
- \textit{RP}, to make decisions in realtime.

Basically, all modules that were implemented at the higher layer contain the algorithms in the .cc program files and the definitions and declarations in the .hh
library files. However, exceptions exist, but did not impact our final structure significantly.

The LpsToGm was implemented first, because we needed the Gmo’s in our global representation. The GM used these Gmo’s and it was implemented after the LpsToGm. The RP or SchemeAnalyser was implemented step by step in about four weeks, but got most of its code in the last two weeks before the competition.

In general we implemented all modules in parallel because of the lack of time. Nevertheless, the SchemeAnalyser was established as one of the latest modules for the competition, because it needed to be adapted to the lower layer modules. It is one of the essential modules in our system, because it makes the decisions.

4.2 LpsToGm Module

The LpsToGm module is converting the local received objects from the LPS module. Lpo’s have a structure for each object that is shown in figure 4.1. These are converted into the Gmo’s, that is defined in figure 4.2.

```c
typedef struct
{
    short int id; // who I am (index in oru_lpo array)
    short int type; // can be ball, dog, ...
    float rho; // distance of object, mm
    float theta; // azimuth of object, rads
    float anchored; // degree by which the object is anchored
    float needed; // degree by which the object is needed
    int last_seen; // last time we have seen the object
    int last_lost; // last time we have performed
        // an unsuccessful search
} OruLpo;
```

Figure 4.1: The structure of a Locally perceived object (Lpo).

Basically, the LpsToGm takes the local Cartesian values from the robot’s point of view, rho and theta, and converts them to global values based on the global view.

For this the LpsToGm uses the static objects, the two goals and the six landmarks (see figure 4.3). We know the global locations of them, if we receive at least two static symbols with a sufficient anchor level. The two best perceived landmarks are used to calculate the global position.

A few days before the competition, we tried a variation of that algorithm.
4.2. LPSTOGM MODULE

typedef struct
{
  float x; // x-component of position
  float y; // y-component of position
  float anchored; // degree by which the object is anchored
  float needed; // degree by which the object is needed
  int last_seen; // last time we have seen it
  int last_lost; // last time we have performed
                  // an unsuccessful search
  float heading; // heading of object
}
} StockGmo;

/*
   // L3 oppsite L6
   // o--------o
   // |       |  y
   // |       |  ^
   // L2  o   o L5  |
   // |       | +x
   // |       |
   // o--------o
   // L1 outside L4
   // (0,0)
*

// coordinates for the landmarks
#define L1_X 0.0
#define L1_Y 0.0
#define L2_X 0.0
#define L2_Y 1400.0
#define L3_X 0.0
#define L3_Y 2800.0
#define L4_X 1800.0
#define L4_Y 0.0
#define L5_X 1800.0
#define L5_Y 1400.0
#define L6_X 1800.0
#define L6_Y 2800.0

Figure 4.3: The static objects and their global locations.

We modified the algorithm so that all static objects influence the global location weighted by their anchor values. By that, all received static objects influence the global location. After the global position of the robot is calculated, all
other objects are converted to the global format. In our global representation, the static objects then has no meaning anymore. They do not move from the global point of view.

As a matter of fact, the LpsToGm was used in a reduced form in the goal-keeper during the competition, because the LpsToGm was insufficiently tested at that time.

4.3 GM Module

The GM module was devised because global information (GMI) was necessary to make complex decisions. The GMI includes knowledge about the goal positions, without having to see them.

We tried to describe and retrieve significant GMI that could not be extracted by any other module in our architecture. We implemented procedures for each global feature. Some of them are shown in structure 4.4.

```c
//Flag for the objects in the line-up class
#define RIGHT_LINEUP   0  //the object is between me and the goal,  
                        //slightly to my right
#define LEFT_LINEUP    1  //the object is between me and the goal, 
                        //slightly to my left
#define STRAIGHT_LINEUP 2 //the object is between me and the goal, 
                        //fairly in front of me
#define NO_LINEUP     3  //nothing is between me and the object

// penaltyzone definitions
#define NOT_IN_ZONE   0
#define IN_OWN_ZONE   1
#define IN_OPP_ZONE   2

// last line of defence definitions
#define NOT_IN_LLD    0
#define ME_IN_LLD     1
#define MATE_IN_LLD   2
#define LLD_Y        3

// near edge definitions
#define NEAR_EDGE     1
#define NOT_NEAR_EDGE 0

// definition of fieldsquares
#define SQUARES_X     5
#define SQUARES_Y     6
```

Figure 4.4: Some typical GMI.

It also provides the possibility for advanced planning structures, because opponent robots can be placed in a global map. Thus it knows how many of
them there are and at what place. For example, if all opponents would be near our own goal, it might be smart for our robots to defend it and run back as fast as possible.

We could define very detailed information about a line-up, but because of the differing information that we received, we decided to use STRAIGHT_LINEUP and NO_LINEUP to differentiate between the line-up of the players. Additionally, we chose one structure to describe the different features. The reason for this is that if we used a Boolean feature map with indistinct Boolean variables, the state space would be exponential.

**Example:** Any 15 indistinct binary features described by Boolean expressions would give $2^{15}$ states.

In a structure with different features, it would be much less, because of the linear independence that is included in integer expressions and Boolean features. They would be distributed over the whole state space.

Additionally, we tried to determine a discrete global map. We divided the playing field into different zones. Figure 4.5 describes this in detail. During the competition we used the GM in a reduced form in the goalkeeper. The landmark recognition and the implementations for the LpsToGm were incomplete and thus the global location finding was almost insufficient.
4.4 RP—SchemeAnalyser

Basically, the motivation-based SchemeAnalyser makes all higher level decisions about the execution of behaviours on lower layers. This idea came into use because other approaches, e.g. finite state machines do not seem to be practical for complex decision making. Higher level decisions take into account all available information, both global and local information. Interpretations of the environment are made, based on this information. The SchemeAnalyser uses this information to build environmental properties, which are used in further calculation as binary predicates or threshold functions. The environmental properties describe an interpretation and are ideally treated like estimations of real world situations. The point in designing a description in the SchemeAnalyser is that the property-descriptions are subjective. Subjectiveness in this case is determined by the design of different descriptions of situations. A typical description, e.g. of an estimation of a situation, like a "KickBall" description that is shown in figure 4.6.

```c
int Situation_KickBall()
{
    int motivation=0;

    //If Ball near
    if (BbBall.rho < 266.0)
        motivation += MOT_4;

    //If Ball in front of me
    if (ABS(BbBall.theta) < RAD(35))
        motivation += MOT_2;

    //If opponent goal in right position
    if (ABS(BbNet1.theta) > RAD(80))
        motivation += MOT_2;

    return motivation;
}
```

Figure 4.6: A description used in the SchemeAnalyser for the “KickBall” situation.

The result of this interpretation is a summation of part-motivations that increase or decrease depending on the description and the situation. Strength-levels, here MOT_4 and MOT_2 are clustered part-motivations. Table 4.1 describes the used motivation levels.

The descriptions are static, but the interpretations are dynamic, because the
real world situations change over time.\textsuperscript{1}

In this case, if the “KickBall” interpretation has more motivation than the other interpretations, the appropriate behaviour command would be sent to the BBM and executed by the robot agent.

The use of a scheme or a script, as a deliberative part of the SchemeAnalyser did not come into use at the RoboCup as we wished it would. Nevertheless, the idea of a scheme is descended from research in cognitive psychology [Anderson, 1995].

Essentially, it is based on the identification of a scheme or a script and it is meant to be a stereotypical sequence of events. Entries in this script can be changed if a situation is not appropriate. That means if a script is chosen that it is not assured that it will be executed from the beginning to the end. A typical script could be a “goal” script (see figure 4.7). A bold circle indicates a high probability of a behaviour that could be executed.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{goal_script.png}
\caption{A “goal” script (default).}
\end{figure}

The whole idea of the SchemeAnalyser is that it acts as a hybrid system in the final analysis. It combines reactive (situation-part) and deliberative (scheme-part) calculations.

This output of the SchemeAnalyser is motivation-based. In detail, the direct interpretation of the situation is purely reactive. This means, that depending on the situation, a motivation will be made up immediately. These calculations interact with the deliberative part. If a script is chosen, it influences the parallel-calculated interpretations. For example, if a “goal” script is chosen, all motivations used in that script receive additional motivation. The addition of

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Motivation levels & The respective value \\
\hline
MOT\_0 & 0 \\
MOT\_1 & 5 \\
MOT\_2 & 10 \\
MOT\_3 & 15 \\
... & ... \\
MOT\_10 & 50 \\
\hline
\end{tabular}
\caption{The different motivation levels that were used in this version of the SchemeAnalyser.}
\end{table}

\textsuperscript{1}Descriptions can be seen as a generalised form of the operators in the STRIPS language.
// these are the results of interpretations

motivation_interpretation[0] = Situation_SearchBall();
motivation_interpretation[1] = Situation_ReacquireBall();
motivation_interpretation[2] = Situation_SearchNets();
motivation_interpretation[3] = Situation_GoAround();
motivation_interpretation[4] = Situation_GoToBall();
motivation_interpretation[6] = Situation_KickBall();
motivation_interpretation[7] = Situation_AlignWithBall();
motivation_interpretation[8] = Situation_StealBall();
motivation_interpretation[9] = Situation_Defense();

// Search for the best fitting description
// in this situation out of all

max_motivation=0;
for (int i=0;i<NUMBER_OF_interpretations;i++)
{
    if (max_motivation<motivation_interpretation[i])
    {
        best_interpretation=i;
        max_motivation=motivation_interpretation[i];
    }
}

Figure 4.8: This part of the SchemeAnalyser describes the choice of the behaviour.

these particular motivations are:

1. Dependent of the pure existence of the description in this script as well as
2. The order within the script.

For example, if the script “goal” is chosen, the descriptions SearchBall, GoToBall, AlignWithBall and KickBall plus the follow in the correct order. GoToBall follows after SearchBall and increases the motivation of a potential behaviour that can be executed.

The script on the other hand, is chosen based on previous events of executed behaviours as well as special situational demands, like e.g. in the beginning of the game several behaviours are supposed to receive more (or less) motivation. For example, if SearchBall and GoToBall were already executed in a row, the probability to choose the “goal” script compared to “stop and do nothing” script would increase. The same idea applies for interaction schemes with other robots, like passing the ball. However these advanced schemes did not come into use, but could be implemented in a future version.
Because the calculations are basically comparisons, the SchemeAnalyser has a linear time complexity concerning the input space. A further estimation and analysis of the algorithm is made in section 6.2.

One important fact that should be mentioned here is that unlike utility functions, motivations are not just goal orientated. They are not designed to reach a predefined goal, but instead indicate the opportunistic freedom of choosing the behaviour with the highest motivation, best fitting the description of the situation.

### 4.4.1 The Command

Finally, the SchemeAnalyser generates a single command every second. The command structure, shown in figure 4.9 is sent to the BBM that does additional lower layer calculations with it.

```c
typedef struct
{
    int behaviour;
    int object;
    int target;
} OruCommand;
```

Figure 4.9: The structure command that is sent to the BBM.

### 4.4.2 Mathematical Formalisation

The mathematical formalisation of the SchemeAnalyser (SA) should be seen as an attempt to formalise the implementations. By doing this, the SchemeAnalyser can be explained in a more thorough manner and a better understanding is possible.

A description \( d \) is an element of a family of sets of descriptions \( D, d \in D \), while \( \{d_1, d_2, \ldots, d_m\} \) is equivalent to \( D \). A description \( d_1 \), for example, consists of description entries \( \{e_{d1,1}, e_{d1,2}, \ldots, e_{d1,n}\} \). \( E \) is the set of all description entries present in the current SchemeAnalyser, so \( E \) consists of \( \{e_1, e_2, \ldots, e_{n+m}\} \).

An motivational entry can be formalised as a function \( m(e) \) (function 4.1). It is an injective function from a description entry \( e \) to \( \{\text{MOT}_0, \text{MOT}_1, \ldots, \text{MOT}_{10}\} \), the motivational set \( MT \).

All interpretations are made in one time cycle. For \( n \) designed descriptions, \( n \) interpretations are made in quasi-parallel time. A description corresponds to one estimated situation. Then, an interpretation is possible. Mathematically, equation 4.2 is a very basic formula describing an interpretation that is the sum of part motivations in a description.
\[ E \rightarrow MT \]

\[ e \rightarrow m(e) \]

\[ m(e) = \begin{cases} 
MOT - 0 & e = false \\
MT/MOT - 0 & e = true 
\end{cases} \tag{4.1} \]

\[ I_a = \sum_{i=0}^{[D]} (m(e))_a^i \tag{4.2} \]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>Description number (a, a \in {1, 2, 3, \ldots, [D]})</td>
</tr>
<tr>
<td>(I_a)</td>
<td>Interpretation of the situation (a)</td>
</tr>
<tr>
<td>((M(e))_a)</td>
<td>Part motivation within the description (a)</td>
</tr>
<tr>
<td>(d)</td>
<td>Set of description entries</td>
</tr>
</tbody>
</table>

The results (motivations) of these interpretations are compared and thus the situation with the highest motivation out of \(n\) is chosen to be the one that will be executed. The best interpretation is the interpretation with the maximum motivation.

\[ I_{best} = \max(I_i) \tag{4.3} \]

\(d_f\) is a description in which all entries are true. This fact can be formalised in equation 4.4 and 4.5, that is the maximum motivation that can be produced by one description.

\[ d_f = \max(\sum_{i=0}^{[D]} (m(e))_a^i) \tag{4.4} \]

\[ \equiv \]

\[ d_f = \max(I_i) \tag{4.5} \]

\(d_f\) A fulfilled description

For further contribution, it is necessary to know the maximum motivation (equation 4.6) that the SchemeAnalyser can produce. Maximum motivation that is producable, all entries point to \(MT/Mot = 0\). Usually, less motivation is produced so that equation 4.7 is less than or equal to the maximum motivation described in the equation.

This section formulates some basic rules for the SchemeAnalyser. They can
4.4. RP—SCHEMEANALYSER

\[ M_{\{SA\}} = \sum_{i=0}^{|E|} \max(d_f^i) \]  

(46)

\[ M_x = \sum I_x \]  

(47)

be applied for some basic complexity analysis in section 6.2.

4.4.3 Time Out Variables

We also used Time Out variables in the SchemeAnalyser. These variables were devised because sometimes the robots executed the same behaviours even though the situation changed. At the stage it seemed reasonable to force the robot to execute another behaviour after executing one behaviour too long, e.g. if “KickBall” is executed longer than 60 seconds, then the SchemeAnalyser decides to execute “SearchBall” for a while.

We used this idea also to give the robot more motivation for some tactical and strategical moves in the beginning of the game. We used Time Out variables to increase the motivation for some special behaviours, like “KickBall” and “GoToBall”. We assumed that at least one robot is placed a few centimetres in front of the ball in the beginning of a match. This strategy seems logical, but because of some unreliability, the robots had to be rebooted in the middle of a game. After the reboot they were, according to the rules, set to the middle line and scored a very nice goal over half of the field—unfortunately in our own goal.2

The use of a Time Out variable was not planned, but some untested implementation forced us to apply such ad hoc methods. Newer versions of the architecture are supposed not to include this kind of variables.

---

2Maybe one of the nicest goals in the whole competition.
Chapter 5

Empirical Studies

In this chapter the experiments and games are described and the efficiency of the implementations are measured. During RoboCup'99 in Stockholm the real games were observed. Further on, the experiments after the official RoboCup event are described. In chapter six the results are described. We used the idea of the official RoboCup Challenge to study the behaviour of the robot. For both the official games and for the practice games we used the SchemeAnalyser. The most interesting fact and observation that we made during the RoboCup'99 about the Sony four-legged robots was that during the round robin, the semi-finals and the finals some interesting goals were made. Interesting, because if one saw the Challenge on the day before the finals, no robot of any team was able to score a goal within a time of together twelve minutes on a completely empty field. The RoboCup made up a Challenge, to determine one wildcard semi-final opponent.

5.1 Observations during the Games

Team Sweden participated in three official RoboCup games and one practise game. We played against Osaka University and Pennsylvania University. Sometimes our robots tended to play against their own goal. Unfortunately, in most of our games the ball was kicked towards to our own goal without any intentional influences. Depending on the team strategy and programming style, the robots were lined up in different settings. Three blue robots played against three red robots (figure 5.1).

We decided to set the robots as near as allowed to the ball. We wanted to decrease the time that was needed by the robot to reach the ball (figure 5.2).

We played two times against Osaka University and lost two times (figure 5.3).
Figure 5.1: The four-legged robot teams.

Figure 5.2: Initial game situation.
5.1. Observations during the games

The games were often harsh. It happened that goals were scored and not
only the robots did not know why (figure 5.4).

If the robots fell down, the acceleration sensor was activated. They got up
immediately and continued playing. Most of the teams used a reset routine after
the robot fell down because it was unlikely that they would stand up in the same
position as before. Thus they had an incorrect orientation of the environment
(figure 5.5).

The robots were calibrated and prepared for the next game. The colour and
recognition settings in the robot were sensitive and we had to improve them
during every break we had (figure 5.6).

It happened that a shootout was made if a game ended in a tie. The teams
had to score as fast as they can. The team who scored first, won. Figure 5.7
shows the start of a shootout kick.

Games were harsh sometimes. That depended on the unreliable visual con-
ditions. It happened that robots charged the ball and each other. Figure 5.8
shows a game situation that reminds more a fight than a soccer game.

Figure 5.3: Typical game situation. Team Sweden vs. Osaka University—
Sweden lost 2:1.
Figure 5.4: Another game situation.

Figure 5.5: A real game—a robot lost his balance and fell down.
5.1. OBSERVATIONS DURING THE GAMES

Figure 5.6: Preparation for the games.

Figure 5.7: A shootout was made, if a game ended in a tie [Sony Rules, 1999].
5.2 Other Leagues and their Pertinence for the Four-Legged League

In the simulator league (figure 5.9) software plays the sole role. Eleven software agents play against each other but still have to deal with simulated realtime uncertainty. There are thoughts of combining pure software approaches with the real world agents, which have to deal with hardware as well. At the DSV a lot of research about synthetic soccer and social reasoning is currently underway, see [Alm, 1998], [Aberg, 1998], [Danielson, 1999], [Kunneneje and Younes, 1999], [Andreasen et al., 1998], [Boman et al., 1998a], [Lybäck, 1999] for a few simulator league related publications. Research in the simulator league can partly be used in the physical leagues and vice versa, since they have to handle similar problems, e.g. calculation of proximity location. The four-legged robots on the other hand can constrain the highly ambitious approaches in the simulator league. The team “CMUnited-99” from the Carnegie Mellon University, USA received the World Champion title, as in the last year’s competition in the simulator league.

As mentioned in chapter one, there are two other physical leagues. One of them is the middle size league (figure 5.10). The robot’s hardware and soft-
5.2. OTHER LEAGUES AND THEIR PERTINENCE FOR THE FOUR-LEGGED LEAGUE

Figure 5.9: Simulator league.

Figure 5.10: Middle size league.
ware are self-constructed. The most difficult part is to find the right hardware components. In spite of the possibility to design robots with all possible extras, almost all teams design robots with wheels. Legged-robots were not constructed in this league. In the middle size class on the RoboCup'99, the team “CS Sharif” from the Sharif University of Technology, Iran won the games.

The small size league (figure 5.11) has access to a global view. The robots can be manipulated by an external computer resource. “The Big Red” from Cornell University, USA was awarded the RoboCup World Champion title in this year.

5.3 Experiments with the RoboCup Challenge

After the games of the RoboCup ‘99, I made additional studies and experiments with our robots. As mentioned earlier, the RoboCup Challenge decides a wildcard opponent for the semi-finals. All teams (even the teams who are in the semi-finals anyway) usually participate. The ball and the robot have to be on a predefined position on the field. After the line-up is made, the robot should try to make as many goals as possible within a certain time.

This is a complex task. On the Challenge 1999, no team was prosperous
5.3. EXPERIMENTS WITH THE ROBOCUP CHALLENGE

to score a goal. Nevertheless, the team “UNSW United” from the University of Southern Wales, Australia won the Challenge by pushing the ball as near as possible to the goal.

I varied the official RoboCup Challenge, redefining it and made it simpler, so that it fitted my purposes. The robot and the ball were placed quite close to the goal (figures 5.12 and 5.13). The robot should find the ball and kick it in

![Figure 5.12: The experimental situation 1.](image)

![Figure 5.13: The experimental situation 2.](image)

the goal, which it finally did. I did that, because of time constraints. Similar situation was made up, but the robots were positioned differently in the middle
of the field. Additionally, the awareness of the difficulty of the accomplishment of the original RoboCup Challenge contributed to my decision to choose a simpler method.

Generally, these two situations are simpler, but sufficient to do experiments with the SchemeAnalyser. I sometimes modified these situations slightly. That was necessary, because it took an enormous time to find some best alignments and thus would require an experiment with more specified examination rules. Unfortunately, this would take months and additional human resources. Nevertheless, the results that were received can be found in the following chapter “Empirical Results”.
Chapter 6

Empirical Results

This chapter gives an overview of the results that I received during the RoboCup competition and through an additional experimental phase after the RoboCup competition. I stored logfiles that contained all valuable information, like decisions, motivations and environmental properties, on the memory stick in the robot. After the time was up or the robot scored a goal I transmitted the data from the memory stick to the computer and analysed them. The following chapter describes one example analysis about a logfile where the robot scored a goal in about 25 seconds.

6.1 Overview

I chose one example to explain the output and the parallel-calculated motivations from the Scheme Analyser. Table 6.1 depicts the motivations that were calculated by the interpretations.

<table>
<thead>
<tr>
<th>SearchBall</th>
<th>40</th>
<th>40</th>
<th>40</th>
<th>50</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RetrieveBot</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>SearchNets</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GoAround</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GoToBall</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>KickBall</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>AlignWithBall</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SlideBall</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Defense</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.1: The data table for the chosen example.
A motivation profile is the output of all motivations produced by the Scheme-Analyser. The best of them will be sent to the BBM. In all tests, in the beginning of the experiment (immediately after the robot booted) the robot decides to search for the ball. In figure 6.1 shows a typical motivation profile with behaviours and the corresponding motivations visualised in the last cycle.

![Motivation profile](image)

**Figure 6.1:** A typical motivation profile.

After about 25 seconds the robot was able to kick the ball into the goal. All competing motivations are shown in a flow motivation profile. In figure 6.2, the quasi-parallel estimations of nine behaviours in 25 seconds have been plotted. Compared to a human motivation profile it is an oversimplified sketch. Humans might have some thousands of motivations competing with each other.

Your motivation to read this thesis should be higher than going to a party right now. The reasons for that might be formalised in a description and your personal interpretation of a situation. You might find it useful and interesting
Figure 6.2: A whole typical motivation flow.
to read through this thesis while hoping that you read something that you could use for your research. Some entries in your description “read Christian’s thesis” are fulfilled and win, compared to others, like “go to a party”.

Finally, figure 6.3 shows the winning and thus executed behaviours. These are the winning behaviours and their corresponding motivation values.

Figure 6.3: The executed behaviours and their corresponding motivations.

Finally, I did experiments with the experimental situation explained in section 5.3. In some of the experiments the robot was able to kick the ball into the goal within five minutes.

6.2 Complexity, Completeness and Optimality

A small analysis of the algorithm is attempted here. Mathematical formulation of the SchemeAnalyser was made in section 4.4.2 and should be applied as a foundation for an asymptotic analysis.

Algorithms in AI, like e.g. search algorithms, are always exposed to a complexity analysis. Two additional properties also have to be included, namely completeness and optimality.

Completeness means: does the strategy guarantee to find a solution when there is one? It is difficult to answer this question, because the existence of a solution in this domain is unpredictable. Further experiments and theoretical foundations have to be made in order to answer this question sufficiently.
6.2. COMPLEXITY, COMPLETENESS AND OPTIMALITY

Optimality means: does the strategy find the highest-quality solution when there are several different solutions? It is always quasi-optimal with respect to the sensor inputs. Both terms describe a purely goal-orientated system and might be applied to the SchemeAnalyser.

Another idea during the development of the robot soccer agent was to use a finite-state machine for higher-layer decision-making. It came into use, but only as an alternative. It was used as a kind of simple “default plan” and was meant to serve as an ad hoc method. Why did we do this?

Finite-state machines are practical for the core of decision making systems, but not for complex decision making. The finite-state machine has in the worst case, exponential complexity, related to the amount of connections between states. This can be avoided by decreasing the relations between states, however this runs the risk of not covering every case or situation. Additionally, to design most of the states by hand with the ambition to cover and to react to realistic situations, requires an enormous effort.

First thoughts of an asymptotic analysis would be to show that the SchemeAnalyser consists of two parts—the reactive part and the deliberative part. These estimations are dependent on the input/output relationship.

A rough intuitional estimation defines the complexity of the SchemeAnalyser as $O(m + n)$. The reactive-part has the time complexity of $O(m)$, because $m$ comparisons are made in quasi-parallel. The scheme-part has the time complexity of $O(n)$, because the one scheme out of $n$ schemes have to be chosen in one cycle. Thus an estimation can be done.

**Reactive Complexity:** $T = m + k + m$. $k$ is a constant and thus disappears. Thus $T = 2m \Rightarrow O(m)$.

The deliberative part is a choice of the script.

**Deliberative Complexity:** $T = n + k$. $k$ is a constant and thus disappears. Thus $T = n \Rightarrow O(n)$.

Together they prove to be linear in time.

**Action Choice** = Reactive + Deliberative = $O(m) + O(n) = O(m + n)$. 

Chapter 7

Previous and Related Works

This chapter gives a small survey of the state-of-the-art practices in AI and robotics. Robotics goes back to the 70s and since Shakey, the robot built by the Stanford Research Institute (SRI), this area has grown rapidly. I am not able to describe all findings that were made before this work. Instead, the historical AI background and existing theories that lead to some goals of the RoboCup will be reflected. I will shortly introduce some architectures and approaches that were made within the four-legged robot league.

7.1 Objectives of RoboCup, AI and Robotics

Criticism could be made and one might wonder if the objectives of most of AI and robotics projects are overly ambitious and maybe too optimistic. Until now, scientific research [Maes, 1994] [Kaelbling and Rosenschein, 1994] [Brooks, 1994] [Anderson and Donath, 1994] [Rich and Knight, 1991] in world-renowned AI research labs or Robotics Institutes of Technology have never proved that a complex organism with intellectual and omnipotent features, like, e.g., humans, could be realised today or could be realised at any time in the future. As a matter of fact most of the scientific works even show that certain problems are not able to be solved with current technology [Minsky, 1954]. Certain landmark projects and standard problems came up, like the Turing test or the RoboCup, but have not been solved until now. A more realistic estimation seems to be that AI is applicable and useable in very specific domains, but does not succeed very well in broader domains. Most of the scientists approach their ideas and these ideas are merely interesting for limited domains.

But in spite of this, discussions about AI foundation are much more advanced than some scientists might think. The transhumanists [see Internet resources] and extropians [see Internet resources] have opinions about the re-
relationship between human beings, society and new technologies, like robotics, nanotechnology, and many more.

The objective of the RoboCup follows as a series of predictions from world-
known AI researchers and philosophers. Some issues concerning the ultimate relationship between robots and humans are discussed in many books and articles nowadays. Questions like, “Will robots inherit the earth?” or “Will humans reside in robotic form?” appear [Arkin, 1998].

Marvin Minsky claimed, based on his long experience within AI research [1965] that any kind of advanced agents or robots will have a deep impact on our life and worldview. He responds to the question, “Will robots inherit the earth?”—“Yes, as we engineer replacement of bodies and brains using nanotechnology. We will then live longer, possess greater wisdom and enjoy capabilities as yet unimagined.” [1994].

After his fundamental and revolutionary research work in robotics [1900], Hans Moravec foresees an evolutionary process for robot applications in the future ([1996] [1998]) and as he stated once, these robots are our mind children [1988].

The philosopher Stanislaw Lem proposed in his works ([1961] [1964]) the technical evolution that Hans Moravec described more concretely. Robots are going to have a technical evolution that is going to be much more advanced than the human evolution, according to Stanislaw Lem.

Rights for robots are discussed and proposed for the first time in the book “Foundation” by one of the most well known Russian science fiction writers, Isaac Asimov [Asimov, 1971].

7.2 Nine Teams ⇒ Nine Approaches

Several approaches have been made since the first RoboCup competition in the year 1997 in Japan. I want to describe the most significant works in the four-
legged league. All of the teams that participated had their own very individual approach. Most of them applied research that has proved to be effective in similar environments.

Some general comments about the approaches and background events could be made from personal discussion with the participants on the RoboCup'99 events. First of all it should be mentioned that three teams had already experienced the Sony robot league in a preliminary four-legged competition. On the RoboCup'98 one team from the Carnegie Mellon University, the Laboratoire de Robotique de Paris (LRP) and the Osaka University participated. They had better premises and were better prepared concerning the robots’ design, Sony’s technical system and the playing field in general.
7.3 ARAIBO

7.3.1 Overview

![ARAIBO Logo](image)

Figure 7.1: Logo from the team ARAIBO.

The team “ARAIBO” explained in their Team Description [Maeda, 1999] and in their final report for Sony [Kobayashi et al., 1999] that they planned a main strategy, basically that the robots should “find the ball, and get close to it”, and to “manipulate the ball according to the direction of the goal”. To establish this strategy, the modes ball-searching and ball-tracking were developed. These modes correspond to the modes that were served by one of the sample programs (SoccerDog).

If the mode ball-searching is active, the robot turns around and wags its head. If the robot executes an action for too long, the consequence of a threshold value forces the robot to change the walking state to an unconditional random state.

In the ball-tracking mode, the robot approaches the ball and than decides the way to align itself and how to kick. The robot changes the walking modes so that the pan angle of the head becomes zero. If the pixel number of the orange ball exceeds a threshold value it decides what to do next.

The robots from the team ARAIBO make use of a head kicking routine. Together with Team Sweden, they were the other team who kicked with the head during the competition.\(^1\) A leg kicking routine was not used, because the posture of the robot seemed to be too unstable while lifting a leg. The head kick is established through a combination of the swinging motion with the head and a translation of the body with the legs.

\(^{1}\)The difference is that Team Sweden used the head kicking as an additional possibility to kick the ball and not as the only one.
For visual recognition and colour calibration the “Gimp” software is applied in the playing domain. Basically one landmark for a global localisation process during the competition is used. If this landmark is recognised a very rough estimation of the direction is possible. The action to recognise the ball in front of the body and to kick the ball were realised by two kinds of artificial neural networks (ANN). The ANN for the measuring of the distance and position of the ball is done by a Back Propagation algorithm. For kicking the ball, Kohonen’s Self-Organising Maps (SOM) were used as associative memory. It is a learning architecture for the kick.

![System architecture of ARAIBO robot.](image)

Four objects were programmed in their agent architecture (figure 7.2). The object “find ball” calculates information about the position of the ball. The object “LearnToKick” does an estimation about the future position of the ball after kicking. The object “commander” manages the kicking and heading operations. Finally, the object “soccer dog” translates the summarised inputs from the other objects and executes an appropriate action.

### 7.3.2 Specialities

The ARAIBO Team uses some extra strategies and approaches that can be summarised.

- Used two neural networks systems, Kohonen’s Self-Organising Maps (SOM) for learning to kick and back propagation algorithms for recognition support.

- Used a head kick instead of a leg kick. They could kick the ball more effective in the Challenge, because the robots do not have to go around the ball. It kicks while doing this was positioned about 90 degrees to the goal, so that it saved time for another movement.
7.4 UNSW United

![UNSW United Logo]

Figure 7.3: Logo from the Australian team, UNSW United.

7.4.1 Overview

The academic representatives from Sydney in Australia use a number of techniques for learning, vision and planning [Dalglish et al., 1999] [Westendorp et al., 1999].

The aim is to integrate the physical feature of the Sony Legged robots in an effort to allow flexible implementation of control algorithms based on learning and planning.

First of all it should be mentioned that UNSW was the team who won the “RoboCup Challenge’99” and successfully played in the semi-final against the CMU Trio. After this game, they lost against the French team.

In their architecture, three concepts for different areas, namely Vision, Planning, and Acting are described. The UNSW has implemented a strong visual module in their robot. The graphical tasks were newly programmed. They designed a scheme, the Generalised Colour Detection (GCD) to overcome the constraints by Sony’s colour table system. The GCD is able to detect polygons, unlike the table system. This seems to approximate the position of visual objects more realistically. Sony’s colour detection hardware generates a C plane, which contains all information about the colours.

A program, called imageClassifier (ic) is applied to classify images, e.g. the orange ball. After this classification another program, CaliCalc came into use trying to fit a polygon in UV space around the pixels that have been classified as orange, for example. After a sufficient detection of all the colours, an image processor came into use - ProcImage. ProcImage separates the C plane into seven layers, one for each of the detected and classified colours.

A standard blob is a group of four connected pixels. A standard blob algorithm is used on each layer to detect any blob that may have been present in this frame. A blob corresponds to an object, like the orange ball or the goals. If the object recognition is done, they calculated the heading, distance and elevation of
each object. Heading and distance calculations were done by taking the centre of each object, by pre-calculating the values for the size of objects. They have done this by measuring the objects in the real world and thus they were able to calculate the distance of the objects on the playing field. The memory system of the robot was called Remembre. The ProcImage sends the current frames to the Remembre module, where it integrates the past data and attributes a confidence factor to each objects’ position, measured from the point at the base of the neck in the centre of the head of the robot. A confidence factor is a term in an equation indicating the probability that the blob in question is at a certain position around the point indicated the data. They used a trilateration method to localise the robot on the field. Trilateration requires two instead of three landmarks. A planner chooses appropriate actions.
7.4.2 Specialities

- Strong visual system by using several tools. On the RoboCup competition they have to be proven exceptional in recognising the objects on the Challenge.

- Use of a simple goalkeeper architecture that was very effective.

7.5 Upennalizers

7.5.1 Overview

![Upennalizers logo](image)

Figure 7.5: Logo from the Upennalizers.

The team from the University of Pennsylvania focused on two strategic areas, attacking the ball in order to move it toward the goal and to defend the goal [Ostrowski et al., 1999]. Their defending strategy was that the robot should stay between the own goal and the ball while an opponent is approaching the goal. The strategy of the attackers was simple: find the ball, approach it, walk around it until the ball was between the robot and the goal and then charge to the goal. While the robot walked around the ball it looked continuously for the goal that it attacked.

For visual processes they used MATLAB and processed the YUV space. Then they executed a blob and marker detection with a blob detection routine. For determining the distance to the ball and the markers, they used a calibration look-up table.

Their agent architecture (see figure 7.6) consists of five modules: the PredictorWriter, AI, Output, Input, and Locator. The AI object performs higher level decisions to the robot. The Output object communicates with the legs.

7.5.2 Specialities

- Used a defender strategy where the robots should stay between the ball and their own goal while an opponent is attacking.
7.6 The CMU Trio

7.6.1 Overview

![CM-Trio'99 Logo](image.png)

Figure 7.7: Logo from the CMU trio.

Personal discussions with the team members, the team captain of the CMU Trio, Manuela Veloso, and her given seminar on the IJCAI [Stone and Veloso, 1999] implied that essential changes to their architecture were not made. Basically, they used the previous year’s approach [Veloso and Uther, 1998].

7.6.2 Specialities

- CMU has teams in three leagues (Simulator, Small Size and Four-Legged league).
7.7 Les 3 Mousquetaires

Figure 7.8: Logo from the French team.

7.7.1 Overview

The LRP, France used software components, which are intended to make the Sony pet robots behave as powerful, organised soccer players [Boucheif et al., 1999]. These components comprise a locomotion module, a vision module and a strategy module.

7.7.1.1 Locomotion

The robot has been provided with an enhanced walking control system that allows it to move in any direction while keeping balance using quasi-dynamic gaits. A self-recovering behaviour has been implemented in case of a fall during the competition, using onboard accelerometers.

7.7.1.2 Vision

The vision system is the main sensor input. Its aim is to detect, to identify and spot the different objects of the observed scene during the game. Detection corresponds to the extraction of all connected components, the set of pixels in the colour images that represent the objects observed with the camera. Identification is processed in two steps:

1. Consists of finding the set of connected components representing an object within the scene.

2. Consists of associating each connected component with a symbolic label.

The set of defined labels is the following: ball; beacon; own or opponent goal; partner or opponent player; edge of the soccer field. Spotting beacons or goals means to localise them in terms of a horizontal angle (with respect to the head’s direction), and in terms of a rough measurement of the distance between the head and the targeted object.
7.7.1.3 Strategy

A model of a multi agent system is the basis for their design of the software level of the robot. Each robot is considered as an agent. The design of the agents’ model leads to the consideration of both aspects of algorithm design (each algorithm corresponds to a particular process), and process control. The proposed process control is event-driven and is based on considering the processes’ level of priority. The organisation of soccer teams leads us to the elaboration of different agent-to-agent interaction schemes. These interaction schemes are based on an agent’s objective identification. The problem in designing these interaction schemes, concerns identification of conflicts between different agents’ objectives. This difficulty is due to uncertainty and the inaccurate nature of data issued by the environment, making behaviour interpretation difficult. The design of the agents’ processes includes data uncertainty and inaccuracy management, which is based on probability theory. In the proposed schemes, agents that belong to the same team, co-operate if their objectives differ, they are concurrent when their objectives are the same. The agents are antagonists if they belong to different teams and if they share a common objective.

7.7.2 Specialities

The winning team of this year’s four-legged league was the French Team.\(^2\) They showed profound skills in building a soccer playing robot.

- They designed an excellent goalkeeper. It saved the ball from coming into the goal area in several situations.

7.8 Humboldt Heroes

![Humboldt Heroes Logo](image)

Figure 7.9: Logo from the German team.

---
\(^2\) They also won the human world soccer championships 2009.
7.8. HUMBOLDT HEROES

7.8.1 Overview

The Humboldt University from Berlin represents Germany. They have experience in the design of agent architectures for the simulation league, where they have used belief-desire-intention (BDI) approaches [Burkhard et al., 1999a] [Burkhard et al., 1999b]. They developed a couple of basic skills (e.g. for kicking, dribbling, and ball interception). The choice of desires and intentions is based on utility calculations, which lead to individual plans on the base of the available skills. Co-operation is performed using the known behaviours of teammates.

They have distinguished four main parts, which we call Cortex, Brain, Body, and Communication (see figure 7.10). Messages are passed between these modules according to the underlying control structure.

The Cortex uses the Colour Detection Engine to identify the objects in the image by common procedures of image processing and to find the object parameters, e.g. position, width, centre-point. The control of head motion is subject to the Cortex.

![Diagram](image-url)
CHAPTER 7. PREVIOUS AND RELATED WORKS

The software architecture of the Brain is built on mental modelling of agents (BDI), which is used by AT Humboldt in the virtual RoboCup. It transforms the received data into an internal world representation (“belief”). It identifies possible options (“desire”) and makes useful plans (“intention”). The basic structure permits various refinements in the future. The Brain can ask the Cortex for certain information.

Plans computed by the Brain are transmitted to the Body and performed by the available skills. The Body controls the movement of the legs in order to turn, move, kick etc. There exists a direct information flow between Cortex and Body e.g. for keeping track of the ball. This implements some rudimentary layered architecture.

7.8.2 Specialities

- Used their experience in the Simulator league for this league—a BDI approach.

7.9 McGill Reddogs

![McGill Logo](image)

Figure 7.11: Logo from the Canadian team.

7.9.1 Overview

McGill use customised ambulation and vision to achieve dynamic localisation and global map maintenance [Unger et al., 1999] [Unger, 1999]. The McGill’s approach to RoboCup legged robot systems design is based on several different software subsystems:

- A responsive custom-designed motion control and walking engine,
- A custom-designed colour vision system based on Bayesian-analysis of colour-space training data,
- A localisation system based on a hybrid of partial scene reconstruction and appearance-based methods,
7.10. BABYTIGERS-99

- A medium-term strategy planner based on a family of finite-state machines,
- A long-term planner based on switching between different behavioural modes,
- A role switching behaviour,
- A function abstraction layer and pseudo-realtime task scheduling architecture for inter-process communication and control,
- A pose and posture estimation system that also produces odometry estimates, Sensor fusion between multiple measurement sources including odometry and vision.

Localisation is done in a global map according to corrected odometry and vision based methods. This global map will provide information to modify a behaviour state-transition engine as the match evolves. Behaviour modes are switched on the fly to best suit the conditions that are detected, and to modulate design.

A custom ambulation provides a responsive and tuneable mechanism for locomotion. The odometry estimate system receives better feedback while providing flexibility with respect to robot speed and capabilities.

Higher layer decisions are made by the state of the global map and a finite state machine. Some behaviours are, goal attempt, attack/ball handling, ball interception and defence.

7.9.2 Specialities

- Applied long term behaviour by asymmetric mode switching.
- Use of a parallel process control kernel which allows optimal execution of computational processes.

7.10 BabyTigers-99

7.10.1 Overview

The team from the Japanese Osaka University were interested in implementing teaching, self-localisation without 3D-reconstruction and embodied trotwalking [Mitsunaga and Asada, 1999].

Their team strategy is, that one robot chases the ball without stopping to walk. The two other robots search, chase and go around the ball depending on its location. They are taught by a human instructor. They developed a special
embodied trotwalking, which can be controlled easy. The visual recognition is established with an extended CDT program and increases the suggested detection of real objects. The robots estimated their localisation by a calibration free method, which assigns a quantified direction to a landmark. Some behaviours were taught, like

- Look around to see landmarks and the ball.
- Check the data and make an action.
- Estimate the landmarks.
- If estimations are reliable, execute the action.

The architecture of the BabyTigers-99 consists of four modules, the sensor object, the camera object, the actuator object and the main object.

The sensor object receives sensory information, while the camera object takes care of all visual perception and calculation processes. The actuator object sends an action to an internal Open-R object, which executes this action. The main object receives all sensory information and makes a decision. They used three buffers to avoid race conditions without looking.

7.10.2 Specialities

- Applied localisation without 3D-reconstruction.
- Developed an embodied trotwalking.

7.11 Comparison of Software Architectures

After the RoboCup’99, which was the first competition of four-legged robots, it is too early to state any positive or negative aspects concerning the comparison of software architectures. Because of the lack of time and other limitations it might be a vague claim that one architecture is better than another is. In spite of that, the next competitions are supposed to finally reveal features, which prove to be significantly better than others are.
Chapter 8

Conclusions and Reflections

8.1 Conclusions

In general, the complex operative system for the Sony legged robots forced us to do a reduced implementation of our architecture. Because of the lack of time, we were not able to apply the SchemeAnalyser and the other higher modules extensively during the RoboCup competition. Especially, with these robots, it is necessary to be aware of the complex platform that is provided by Sony. Open-R and the Aperios are very well organised systems, but because of their complexity, they are difficult to use. We were forced to fix primitive and trivial problems, such as errors that were due to our lack of experience with the system.

Further conclusions could be drawn. In spite of the reduced application of the higher layer and the SchemeAnalyser, it has been shown that motivation-based decision making is possible in this domain. Additionally, the effective approach and the realtime decision process fits well in our architecture. In the beginning, we emphasised the open architecture and it turned out to be very flexible concerning the expansion of the system. Our architecture is easy to understand and to expand because of its significant modular structure.

The SchemeAnalyser works reasonably well, because it works within our theoretically expected situations, but because of unexpected situations occurring during the games it turned out to be unreliable sometimes. For example, an enemy robot and a teammate robot that are struggling against each other for too long. That is an example where the SchemeAnalyser put a lot of motivation into the GoToBall behaviour, since the robot is stopped by another one. The SchemeAnalyser still decides to go to the ball, but do not take into account the hostile enemy in front of it. We never tested it during a real game, because of the lack of time. We were not able to fix that problem in the SchemeAnalyser during the official games.
We received the seventh place in the RoboCup competition. We evaluated our research ideas and applied our experiences from similar domains. Team Sweden plans to explore these issues in preparing our entry in RoboCup2000.

8.2 Future Contributions

8.2.1 Development of the Robot Soccer Agent

The solution described details a very open architecture, so that in either case the objects as well as the RP can be used for further research. Modules could easily be added. A communication module, e.g. one that uses the sound capabilities of the robot could be used to make the robots speak to each other and thus exchange valuable information.

8.2.2 Usefulness for the Simulator League

It might be an interesting experiment to combine the agents' architecture from a simulator league team with the agents' architecture in the robot soccer leagues. Hence, to actually implement simulator softbots with the real world robots could be an important step to compare "perfectly-designed" software ideas with the "real world". A question, like "Could software developments only be applied on the nice and perfect computer domain or also in reality?" appears.

8.2.3 Usefulness for other Complex Domains

The Internet domain is a very complex, because of its thousands of hosts and millions of users all over the world. It is a demanding task to control the intricate nature of the Internet. The SchemeAnalyser, with some changes in the architecture, could also be used in the Internet domain. Situations could be described depending on what site the agent is at and deliberative motivation influences can be examined. The Internet with its communication protocols to the information is a very attractive domain to promote intelligent systems. A multiagent system can be established and exploring the goal for any kind of resources. They could make business. An example of this idea was served by [Greenwald and Kephart, 1999].

8.2.4 Autonomous Design Add-on

Situation description is made by hand. The next step to develop the SchemeAnalyser could be to add a scheme descriptor. An instance could automatically create the situations and schemes and thus provide us with a robot soccer agent
that is able to adapt to the change from one domain to another. Neural networks could contribute to a scheme descriptor by experience. For the autonomous design, the rule-based descriptions have to be exchangeable and motivation values have to be variable.

8.2.5 Deliberative Planning

For further development, the deliberative part of the SchemeAnalyser could be improved dramatically. Schemes are an important theory about how humans act.

8.2.6 The SchemeAnalyser and Quantum Computers

In the long run, the SchemeAnalyser on quantum computers could be an effective approach, assuming that quantum computers are useable in practice. If the promising prediction and research objectives of quantum computation [PQI, 1990], [Eckert, 1990] come true, the complexity of the SchemeAnalyser would be much less than log (n).¹ Quantum computers represent a completely new way to compute. They might even handle the calculation of NP-problems and thus would offer an revolutionary step for space and time exponential AI problems. I am aware of that this propose is a very debateble one. Maybe quantum computers will never become reality. However, based on recent research results in this area, I could imagine to apply a special designed SchemeAnalyser for quantum computers.

8.2.7 Emotional and Intentional Agents in the RoboCup domain

The RMIT University does research about agents in RoboCup leagues [Ch’ng and Padgham, 1996], [Ch’ng and Padgham, 1997]. They use different approaches based on intentional and emotional agents [Padgham and Taylor, 1997a], [Padgham and Taylor, 1997b], [Padgham and Taylor, 1997c], [Padgham et al., 1996], [Cavedon and Padgham, 1995], partly implements in RoboCup. Since the SchemeAnalyser is an approach using similar ideas, modifications and further aspects could be added. The research that is going on at the RMIT University could further develop the idea of the SchemeAnalyser or a similar approach.

¹No detailed calculation are made.
8.2.8 Less Opportunity—More Risk

Most approaches that I have seen from the other leagues and teams were based on rational decision making. The approaches were often described in logical terms. Modifications could only be made in logical changes. But sometime it might be necessary to "blindly" take risks without having any detail of what will happen. For example, I decided to come to Stockholm as a Master degree student. This decision was not rational at all, because in the final analysis I had to leave friends, lost contacts to my professors, had to end my student job and invested a lot of money to come to Stockholm. A rational decision-maker would definitely decide to stay in my home country, but as human, I found it more exciting to experience Stockholm and to leave Germany. All that makes me believe that additional experiments could be made in this area. To conclude this paragraph, as future consideration I would suggest, to implement a "just do it" approach into the SchemeAnalyser.

8.2.9 Soziotronik—Society and Agents

Research about emotions could serve to translate important terms into Computer Science dictionaries, or versa. A attempt to combine social and technical aspects is implemented in the Soziotronik-project at the "Deutsche Forschungsgemeinschaft" was done.

For example, if a robot is trying to reach the ball and does decisions to its best to achieve this goal, but does not reach it, a frustration level could raise and let the robot make other decisions.²

8.2.10 Motivation—equivalent or not?

The term motivation, like most of the psychological terms that are used in robotics, might be oversimplified. It could be doubtful that psychological used terms are really equivalent to technical terms. For further research I would suggest additional study concerning the equivalence (if there is one) between terms in psychology and terms in Computer Science.

For example, the principle of the motivation-based SchemeAnalyser could easily be compared to a simple decision tree, which combines the evaluation of long-term utility as deliberative part and utility-based reasoning as a reactive part. In this case the differences are that the motivation-based SchemeAnalyser does decisions in one cycle (useful for realtime) without any yes or no decision nodes between. Also, it is not typically goal-orientated, like most decision tree approaches.

²This specific example came up during a discussion with David Lybäck who worked on his thesis about “Transient Diversity in Multi-Agent Systems” [Lybäck, 1999].
8.3. FUTURE WORK

In many AI projects, and also in this case, robotics and AI experts do not attempt to find appropriate analogies in psychological senses. And thus the question “To what degree is the term artificial motivation equivalent with psychological motivation” appears.

8.2.11 Just Entertainment?

To investigate a living together with robots might be another aspect [Asada and Christensen, 1999] [Breazeal and Scassellati, 1999]. Children that I have seen during the RoboCup’99 looked very interested to come in contact with these robots. Interviews by the media reveals that they could imagine to have an artificial pet in their home instead of an organical one. The impact of this part of entertainment industries is from interest, and not only for industry. How much has the AI and robotics industry already influence our social life? How will a scenario look like ten years from now? All these questions are important, useful and necessary when considering AI related research.

8.3 Future Work

I am planning to do further research in the field of psychology and AI. I might take the issue of RoboCup as an example for AI development and the contributions of aspects from psychology.

Research in social cognitive psychology can be extremely challenging from two points of view. The first general question could be, in how far new technologies, such as robotic pets, effect human society and vice versa. The other research formulation could be, how psychological research is applied in the AI and RoboCup in particular.

8.4 Reflections

As a member of Team Sweden, I participated in the first competition of soccer playing four-legged robots. We played against eight academic teams from all over the world.

The RoboCup was the most exciting project I have ever participated in. In such a research competition, scientific potential could be experienced live and valuable and relevant information was exchanged in realtime. Media from all over the world continuously accompanied this event. TV channels, like CNN and Euronews and newspapers, like the Financial Times, showed their interest by broadcasting the games and interviewing the participants. The local channels
displayed an extraordinary interest in this event and spread the idea of RoboCup to the nation.

In reflection I would like to say that the extraordinary and ardent effort of the RoboCup “Team Sweden” members was the most impressive issue. In addition, I felt that national borders and differences disappeared and working together with six other nationalities beside one was inspiring. An additional interesting fact was also to deal with the international composition of the team. In our discussions we enjoyed using language skills to communicate our ideas and to speak with most of the team members using their native language. I conducted and took active part in discussions and meetings in three languages: Swedish, English, and German.

All nine teams worked together in a 60 m² room in the City Conference Center that is located in the inner city of Stockholm. I was involved in a fantastic international multidisciplinary research environment. In addition to all technical tasks, we were faced with almost all kinds of challenges, like improvisation—technically as well as socially, the use of different languages, cultural interest integration, teamwork, dealing with aspects of the competition, working under pressure and with deadlines, working with serious lack of sleep, etc.

I was participant at the IJCAI as well as in the RoboCup. During the nights at the Norra Latin I implemented software to the robots and the next day participating in workshops and tutorials. I have the belief that both events complete each other and I benefited from both the practice and the theoretical part. I am looking forward to participate on the RoboCup2000 as well, since I am going to move to Melbourne. I am going to study at the Royal Institute of Technology (RMIT University), who will organise this event.

I personally would like to add that this thesis, in particular, reflects and summarises remarkable teamwork. If another similar project environment would be offered to me, I would not hesitate to join it.

8.4.1 Additional Challenges

Hopefully, the effects of my thesis contributed to make the research area Artificial Intelligence, popular and accessible to the broad public. The RoboCup is an exception in the world of science [Kitano, 1999]. Everybody who is interested, familiar with this field or not, has the chance to see the state-of-the-art in this research field. We gave interviews on TV, radio and in several magazines, see [TV3, 1999] [Metro, 1999] [DataTeknik3.0, 1999], [Sveriges Radio, 1999], [Spiegel, 1999] for a few examples. Additionally, I wrote a popular-science article that was published in the most well known German computer magazine CT [Guttmann, 1999]. The RoboCup event makes it possible to get into direct contact with
everybody who is interested in this field. Additionally, researchers related to
RoboCup get the possibility to get the attention of sponsors and thus do research
with the financial help from the industry.

Another interesting challenge to me was the nature of this project work. Dealing with the multi-cultural and international composition of the team was
important and because of the distance between the working places in Sweden,
the danger of misunderstanding eachother was high. An example of that was
the Swedish holiday rush in June and July each year. This knowledge becomes
vitally important, since we suffered from a lack of time. Nevertheless, knowing
the cultural and educational background of the other team members to unify
our scientific ideas during this project really helped.

In the academic world it gets more common every day to deal with multi-
culturism. One has to consider cultural and linguistical gaps or even contradic-
tions, if one wants to build an international team of scientists. This is certainly
a requirement to build an effective team with international team composition
and to finally accomplish research goals.

The RoboCup challenges the skills of the researchers and the performance
can directly be compared with the approaches of the other research groups.
Acknowledgements

First of all I would like to thank my supervisor Magnus Boman for his patience and advice. He gave me the unique opportunity to participate in a team of creative researchers, where I had the extraordinary opportunity to improve and contribute my academic skills. He was in every aspect an exceptional supervisor and he was always willing to give me further intelligent advice so that I could improve my thesis work.

I would like to thank my German professor Hans Kleine Buening from the University of Paderborn for revising my thesis and giving important suggestions for my thesis work.

For reading my thesis and giving valuable comments, I would like to thank Tove Söderlund, Neil Piper and Fredrik Hacklin. Additionally, I would like to thank Björn Lingstenz and Göran Eneroth for officially revising my thesis. Both were my academic thesis opponents.

Magnus Ericmats was my main companion in Stockholm during the preparation phase. In the preparation phase before the competition, he implemented most of our ideas of the LpsToGm and the GM module. I am very grateful to Alexander Tolley, who visualised some of our ideas in a professional manner and provided me and thus Team Sweden with helpful information. He was present all the time and made, in particular, photos of Team Sweden and digitised films from certain media.

Together with David Lybäck I had some inspiring discussions and I would like to thank him for his good advice. David Lybäck, Johan Kummaneje and Håkan Younes were patient and helped me during the project and because of research experience within the DSV, they gave me valuable information about an execution of a Master’s Thesis at this Department.

I also want to thank every team member of Team Sweden, the complete list is in table 1.1. But especially I would like to thank Alessandro Saffiotti from the University of Örebro for his ideas and guidance during the RoboCup’99 in Stockholm. I was happy that we had such an experienced researcher in our team. Since he will be the team captain on the RoboCup'19
for Team Sweden, I wish him and his crew all the best and good luck for all the
games in Melbourne. Another important person in Team Sweden for me was
**Kevin LeBlanc**. He supported me in many important aspects concerning the
organisation. Among other things, I am very grateful for his support with the
sketch of our “communication architecture” (figure 3.2). I am sure that without
his support I could not able to complete my work so quickly.

**Mats Wiklund** from the DSV was the press officer of the RoboCup’99 in
Stockholm and I am grateful that he arranged the possibility of contributing
my opinions about the RoboCup to the media. He also removed the press in
time consuming situations.

I am very grateful to the staff of the DSV and Team Sweden who organised
the RoboCup and the IJCAI. I think that the RoboCup and the IJCAI, after
all was a unique and inspiring event for all of us.

The following researchers contributed to the contents of my thesis in a very
special way; it was very inspiring to discuss issues of AI with them.

I want to thank **John Mallary** from the AI Institute from the Massachu-
setts Institute of Technology (MIT) for a very interesting conversation
during the IJCAI'99 in Stockholm. Further on, I had some interesting discus-
sions with **Peter Stone** from the Carnegie Mellon University (CMU). I
would like to thank **Manuela Velloso** from the CMU, because we had a discussion
about the robot legged league before the event and continued this discussion at
the IJCAI event. I would like to thank **Professor Burkhard** from the Hun-
boldt University in Berlin for discussions about our leagues and philosophical
issues.

**Lin Padgham** from the RMIT University, Australia, gave me ideas about
future contributions to my thesis. **Professor Udo Hahn** from the University of
Freiburg for advice and for future contributions within the area of AI.

**Anders Lansner** from the SANS Research, NADA helped us to take Rein-
forcement Learning for mobile robots into consideration. **Magnus Åhländer**
send me an unpublished version of his work, which I could use for my thesis.

**Michael Ritzschke** from the Humboldt University, Germany, **Mike Lawther**
from the University of Southern Wales, Australia, **Yuich Kobayashi** and
**Yusuke Maeda** from the University of Tokyo, Japan and **Richard Unger**
from the Mc Guill University sent me a preliminary version of their Sony report,
which I could apply before my thesis work was finished. Additionally, **Bern-
hard Froetschel** from the Foo Fighters team allowed me to apply pictures of
the Small Size league that they made. Other participants from the RoboCup
teams might not remember me but because of several technical discussions in the
Norra Latin, City Conference Hall they indirectly participated in establishing
this thesis work.
I would like to thank the Sony support staff who were friendly and understandable of the difficulties we had. Masahiro Fujita from the Sony D21 Laboratory was our main contact person during the games and supported us friendly whenever he could spare time.

And last but not least I am very grateful to my family, who supported me financially and morally during the whole time that I worked on my thesis.
Bibliography


[88] [Veloso and Uther, 1998] MANUELA VELoso and WILLIAM Uther. The CM’Trio-98 Sony legged robot team. In Minoru Asada and Hiroaki Kitano,


Appendix A

Press References

At least 150 Swedish press publications have been printed. Only a few press references concerning the RoboCup’99 are listed here. For further information, please contact the RoboCup’99 press officer Mats Wilkund from the DSV. His email is matsv@dsv.se.

[Guttmann, 1999]
CHRISTIAN GUTTMANN, RoboCup’99: Roboter kicken wieder, CT no.16, Heise Verlag, 1999

[TV3, 1999]
CHRISTIAN GUTTMANN and MATS WIKLUND, Interview, Daily News, TV3-station, June 1999

[Metro, 1999]
MATS WIKLUND, CHRISTIAN GUTTMANN and HÅKAN YOUNES, Interview, Metro, June 1999

[Spiegel, 1999]
CHRISTIAN GUTTMANN, Interview: Bocklosigkeit als Zeichen von Intelligenz, Spiegel-Online, 1999

[Datateknik3.0, 1999]
MATS WIKLUND, Datateknik3.0, 1999
[Sveriges Radio, 1999]

Christian Guttmann and Magnus Ericmats, Swedish Radio, 1999
Appendix B

Internet Resources

HomePages

The Open-R Support Page, Sony Inc. (confidential, for official RoboCup Members only)
because of security reasons this IP address can not be published

The “Robot World Cup Initiative” Official Site
http://www.robocup.org

The RoboCup’99-Stockholm Home Page
http://www.ida.liu.se/ext/RoboCup-99/

Spiegel-Online, Live Games on the Internet and Interviews with RoboCup Participants
http://www.spiegel.de/netzwelt/netzkultur/nf/0,1518,31410,00.html

The Sony RoboCup Service Page
http://www.world.sony.com/dream/

Team Sweden Homepage, DSV in Stockholm
http://www.dsv.su.se/~robocup/teamsweden/

Team Sweden Homepage, AASS in Örebro
http://aass.oru.se/Living/RoboCup/
CMU-Trio
http://www.cs.cmu.edu/~robosoccer/legged/legged-team.html

McGill RedDogs
http://www.cim.mcgill.ca/~mumeteam/

The Humboldt Heroes
http://www.ki.informatik.hu-berlin.de/RoboCup/RoboCup99/

UNSW United
http://www.cse.unsw.edu.au/~robocup/

ARAIBO
http://www.arai.pe.u-tokyo.ac.jp/robocup/

Scandinavian RoboCup Committee
http://www.ida.liu.se/~milco/Scand-committee.html

German National RoboCup Committee
http://www.ki.informatik.hu-berlin.de/ARobocup/index.html

Swedish Artificial Intelligence Society (SAIS)
http://www.ida.liu.se/~jalma/sais/

Danish Artificial Intelligence Society (DAIS)
http://www.daimi.aau.dk/~brian/dais.html

Norwegian Artificial Intelligence Society (NAIS)
http://www.dsv.su.se/ijcai-99/

Finnish Artificial Intelligence Society (FAIS)
http://www.uvasa.fi/stes/eindex.html

IJCAI’99 in Stockholm
http://www.dsv.su.se/ijcai-99/
Extropy Institute HomePage  
http://www.extropy.org/

Transhumanism HomePage  
http://www.transhumanist.com/

FTP Server  
Team Sweden Server  
robi.dsv.nu.se, Code Exchange and Calendar

Mailing Lists  
Team Sweden Mailing List  
team_sweden@mail.hh-r.se, access for Team Sweden Members only

RoboCup Mailing List  
majordomo@cs1.sony.co.jp, with a content of “subscribe RoboCup”.

RoboCup European Mailing List  
majordomo@cs1.sony.co.jp, with a content of “subscribe robocup-euro”.

Extropy European Mailing List  
majordomo@extropy.org, with a content of “subscribe exi-euro”.

Extropy Institute Mailing List  
majordomo@extropy.org, with a content of “subscribe extropy-institute”.

Newsgroups
  
  - sci.psychology.theory
  - sci.psychology.research
  - sci.math
  - sci.math.research
  - sci.cognitive
  - comp.robotics.research
• comp.robotics.misc
• comp.ai.fuzzy
• comp.ai.neural-nets
• comp.ai.alife
• comp.ai.edu
• comp.ai.fuzzy
• comp.ai.games
• comp.ai.genetic
• comp.ai.nat-lang
• comp.ai.philosophy
• comp.ai.shells
• comp.ai.vision
• fr.sci.cogni.publication
• fr.sci.cogni.discussion
• swnet.ai.neural-nets
• bionet.neuroscience
Appendix C

Glossary

Open-R

A standard architecture for Robot Entertainment Systems, see chapter 2.2.1 for a more detailed definition.

Aperios

Application layer for Sony Pets Development, see chapter 2.2.2 for a more detailed definition.

Saphira Architecture

An integrated sensing and control system for robotics applications [Konolige et al., 1997].

Thinking Cap

Offspring of a layered architecture based on fuzzy logic originally developed for the robot flakey.

Degrees of Freedom (DOF)

An actor that determines a single motion.

PCM Format

Pulse Code Modulation, a method to digitise analogue sound. Files end with .pcm.
CDB

CDB simplifies the integration of higher-level, symbolic deliberation processes and lower level, numerical control processes.

Animal Behaviour

Studies about the basis of intelligence, biological systems, through the eyes of psychologists, neuroscientists, and ethnologists and examines several representative robotic systems inspired by animal behaviour.

Robot behaviour

Describes the basis for behaviour-based robotics, including the notation, expression, encoding, assembling, and co-ordination of behaviour.

Behaviour-based architectures

Presents a range of robotic architectures employing the behaviour-based paradigm.

Adaptive Behaviour

Addresses how robots can cope with a changing world through a variety of learning and adaptation mechanisms, including reinforcement learning, neural networks, fuzzy logic, evolutionary methods, and others.

Social behaviours

Opens up behaviour-based robotics to the consideration of how teams and societies of robots can function together effectively—raising new issues such as communication, interference, and multiagent competition, co-operation, and learning.

NTSC

Abbreviation of National Television Standards Committee. The NTSC is responsible for setting television and video standards for North America (in Europe and the rest of the world, the dominant television standards are PAL and SECAM). The NTSC standard for television defines a composite video signal with a refresh rate of 60 half-frames (interlaced) per second. Each frame contains 525 lines and can contain 16 million different colours. In addition to the U.S. and Canada, the NTSC standard is used in Central America, a number of South American countries, and some Asian countries, including Japan.
YUV

YUV is the color space used by the PAL color system (it may also be used in the NTSC color system). Y is the luminance component, while U and V are the color components.

LPS

LPS is the Local Perception Space. It takes track of locally perceived objects relatively from the position of the robot.

BBM

Is the abbreviation and means Basic Behaviour Module. This module contains basic behaviours, such as KickBall and GoToBall.

CMD

CMD means Commander, which is the entity converting abstract information from upper layers directly to the internal Aperios interface.

GM

GM is the abbreviation for Global Map and contains a discretised map of the playing field. Once the robot is globally located on the map further global information (GMI) can be extracted.

GMI

GMI means Global Information (originally Global Map Information). It contains several stereotypical global informations, such as global line-ups.

RP

RP is the abbreviation for Reactive Planner. It makes plans but takes reactive aspects into account. In the first version of our Robot Soccer Agent the SchemeAnalyser is the underlying concept.

PAM

PAM is the abbreviation for Perceptual Anchoring Module and another term for LPS.
HBM

HBM is the abbreviation for Hierarchical Behaviour Module and another term for BBM.