FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Artificial Intelligence Methodologies Applied to the Analysis and Optimization of Soccer Teams' Performance

Pedro Manuel Henriques da Cunha Abreu

Programa de Doutoramento em Engenharia Informática

Supervisor: Luís Paulo Gonçalves dos Reis (Prof. Dr.)

Second Supervisor: Júlio Manuel Garganta Silva (Prof. Dr.)

2011, March

For my Twin Brother and for my Parents

Resumo

Nesta dissertação foi desenvolvido um estudo sobre como melhorar o desempenho de uma equipa de futebol. Os autores tentaram responder à seguinte questão de investigação: Como melhorar o desempenho de uma equipa de futebol através do cálculo de estatísticas finais de jogo. Para isso dividiram a sua contribuição em quatro fases distintas.

Na primeira fase, foi usado um sistema de localização com o objectivo de colectar as posições cartesianas dos jogadores ao longo de um jogo de futebol. Baseada nesse tipo de informação foi ainda desenvolvida uma ferramenta de visualização e extracção de informações de alto nível. Contudo, como se veio a constatar, os sistemas de localização ainda não apresentam maturidade suficiente para serem aplicados neste tipo de cenários. Juntamente com este facto, a escassez de dados de futebol humano levou a que neste estudo fossem utilizados dados de futebol simulado, mais precisamente dados da competição da simulação 2D do RoboCup. No final desta fase e com vista a uma maior percepção dos conceitos inerentes ao jogo de futebol, bem como as relações entre si, foi desenvolvida uma ontologia, posteriormente validada através de um inquérito online, abrangendo investigadores e alunos universitários da área do desporto.

Numa segunda fase foi desenvolvida uma ferramenta capaz de calcular automaticamente aproximadamente 60 estatísticas através dos ficheiros de *log* do RoboCup. Depois deste desenvolvimento, e tendo presente o objectivo do RoboCup, foi realizado um estudo comparativo entre duas realidades (humano e robótico), tendo sido detectadas algumas diferenças significativas.

A terceira fase do projecto (designada por fase *offline*) consistiu em perceber quais as estatísticas (previamente calculadas) que mais influenciavam o resultado final dos jogos. Para isso, foram utilizados dois algoritmos de *feature selection* (MARS e RreliefF) e seleccionado um conjunto de ficheiros de *log* da competição 2D do RoboCup entre 2006 a 2009. No final, o algoritmo MARS apresentou os melhores resultados e por isso foi usado na fase seguinte deste projecto. Esta consistiu em escolher quais as equipas que iriam defrontar a equipa do caso de estudo (FC-Portugal). O critério visou escolher a melhor, a pior e a equipa mediana em termos de resultados finais da competição de simulação de 2009. Após esta escolha, foram simulados vários jogos entre essas equipas e seleccionado o conjunto de estatísticas que mais influenciava o resultado final (utilizando para isso o algoritmo MARS). O passo seguinte consistiu em agrupar esse conjunto de estatísticas em K *clusters*, utilizando um algoritmo de *cluster*. A partir daí, foram treinados classificadores que conseguissem predizer o grupo que melhor caracterizava um conjunto de dados de entrada. Para isso, três classificadores foram utilizados - *Support Vector Machines* (SVM), *Bagging* e *Random Forest* (RF). Finalmente, a melhor estratégia a escolher mediante a similaridade do comportamento da equipa adversária foi escolhida, tendo como filtro de escolha a optimização da diferença de golos (marcados e sofridos). É importante salientar ainda que a estratégia de equipa foi composta por dois factores: formação da equipa e um conjunto de jogadas estudadas.

A quarta fase (designada fase *online*) consistiu em predizer o grupo onde os dados de entrada mais se assemelhavam. O modelo de predição usado foi treinado na fase anterior do projecto. Finalmente, uma dada estratégia foi escolhida para um determinado comportamento do adversário.

Para isso foi usado o grupo e a melhor estratégia por *cluster* (obtidos na fase anterior) com o intuito de escolher a estratégia que melhor optimizava a diferença de golos para um conjunto dados de entrada. Os resultados finais obtidos provaram inequivocamente que o uso do algoritmo SVM com uma periodicidade de análise de 500 e 2000 ciclos melhorou substancialmente uma abordagem mais clássica (que consistiu em não realizar alterações ao longo do jogo).

Abstract

In this dissertation, a study about the improvement of a soccer team was executed. Basically, the author tries to answer the research question "How to improve the performance of a soccer team through the calculation of final game statistics". For that, the author divided his contributions in four parts.

In the first phase, a tracking system was used to collect cartesian coordinates of players throughout a soccer game, and an automated statistical framework based on that information was developed. However, as noted later, there is not a tracking system that is capable of performing such operation. Because of that, and due to the lack of human soccer data availability, in this study the author used only robotic information, more precisely the RoboCup 2D simulation league log files. In order to understand the existing relations between several soccer concepts, an ontology was developed and validated by academics and soccer students through an online survey.

The second phase consisted in the creation of a new statistical framework based on the Robocup log files. Through that tool, the author was able to automatically calculate almost 60 final game statistics. Having the RoboCup main goal in mind, a comparison study between the human and robotic soccer was performed and some statistical differences were analyzed.

The third phase of the project (named offline phase) consisted in understanding which are the final game statistics that most influence the final game results. For this, two feature selection algorithms were used (MARS and RreliefF) and a large set of games of the 2D RoboCup competition between 2006 and 2009 were chosen. At the end, as the MARS algorithm presented better results, it was used in the next phase. That phase consisted in choosing which is the robotic team to improve its performance and what are the three opponents. The FC Portugal was chosen as the case study for this project and the three opponents were chosen according to their final position in the RoboCup 2D 2009 competition (the best, the middle and the worst team was the chosen criteria). After that, we repeated the experiment done previously to calculate the final game statistics (using the game statistics tool) through many simulated games between those teams and selected which are the statistics that most influence the final game results (using only the MARS algorithm). After that, a cluster algorithm was used to group the data calculated by the statistical tool into *k* clusters (using only the statistics selected by the MARS algorithm). The next step consisted in training a classifier that can predict the group that better characterize a given input. For that, three distinct classifiers were used – Support Vector Machines (SVM), Bagging and Random Forest (RF). Finally, the expected best strategy for a group of opponents with similar behaviors according to the maximum of the scored goals was defined. It is important to note that a team strategy is composed by two factors: team formations and set plays.

The fourth phase (named online phase) consisted in predicting the group for which the given data is expected to be more similar with. The model used for prediction was trained in the previous phase. Finally, a certain team strategy is assigned for a particular way of playing of the opponent. For that, we used the predicted group (obtained in the previous step) and the best strategy per cluster (obtained in the previous phase) in order to obtain the strategy that optimizes the difference of goals scored for a given runtime input data. The final results proved that the SVM algorithm with a 500 and 2000 cycles of periodicity increase the FC Portugal performance in comparison with the classical approach (that did not include any changes during the game).

Résumé

En cette dissertation a été développé un étude sur comment améliorer le dégagement d'une équipe de football. Basiquement l´auteur a essayé de répondre au question suivant d´investigation "comment améliorer le dégagement d'une équipe de football en travers du calcul des statistiques finales du jeu". En ce sens, il a divisé sa contribution en quatre parties. Dans une première partie un système de localisation a été utilisé pour collecter les coordonnés cartésiennes des joueurs et un système de calcul des statistiques finales du jeu basé en ce type d´information a été aussi développé. Cependant comment on analyser postérieurement, il n´y a pas encore un système de localisation capable de réaliser ce type d'opération. Ce type de constatation en même temps que la manque disponibilité des dés de football humain a emporté l´auteur à suivre un autre chemin utiliser des dés de foot robotique plus exactement dés provenants de la modalité 2D de la compétition RoboCup. Dans une deuxième phase du projet a été développé un outil de création de statistiques qu´il a utilisé comme base les fichiers de log de la compétition du RoboCup. Afin de comprendre les relations existant entre les concepts de soccer de plusieurs, une ontologie a été élaboré et validé par des universitaires et des étudiants de football grâce à un sondage en ligne. Comme ça a été possible calculer automatiquement plus de 60 statistiques finales du jeu. En outre (d´ autre part) et prennant soin à l´objectif primordial de la compétition RoboCup un étude comparatif entre le football humain et robotique a été réalisé et quelques différences ont été analysées. La troisième phase (désignée phase offline) a constitué en compreendre lesquelles des 60 statistiques calculées préalablement ont plus influé le résultat du jeu; vers cela ont été utilisés deux algorithmes de sélection de variables (MARS et RreliefF) et un ensemble élargi des jeux de la compétition 2D du RoboCup entre 2006 à 2009. Au final de cette étape l´ algorithme MARS a présente meilleurs résultats et pour cela il à été utilisé une autre fois dans une phase postérieure de projet. L'étape suivante a consisté à choisir l' équipe que nous voulions améliorer le dégagement et lesquelles seraient ses apposants. En ce sens l' équipe du F.C.Portugal a été choisir pour être le "case study" de cette thèse et ses opposants ont été choisis selon ses classifications dans la compétition de 2009 (RoboCup) on a choisi la meilleure et la pire équipe selon la classification finale et aussi l' équipe qui est restée au millieu et la table (barême). En suivant, l´algorithme du groupement a été utilisé pour grouper les dés calculés en groupes différents et la démarche suivante s´est composée à entraîner un classificateur capable de prédire le meilleur groupe pour un determiné ensemble de dés. Pour cela ont été utilisés trois algorithmes, Random Forest(RF), Support Vector Machines (SVM) et Baging. Finalement la meilleure stratégie à adopter pour une certaine comportement de l' équipe adversaire a été choisi. C´est important signaler que cette stratégie est définie par l'ensemble des conps étudiés et par la choix de la formation utilisée par l' équipe. La quatrième phase (phase online) consiste à prévoir le groupe qui présente la plus grande similarité avec les dés analysés. Le model de prédiction usé a été entrainé dans la phase antérieure du projet. Finalement une certaine stratégie a été choisi par une déterminée façon de jouer de l´opposant. Pour cela on´a utilisé le groupe de prédiction (obtenu dans la phase antérieure)essayant de cette manière optimiser la différences des coups dans la perspective de l´ équipe F.C.Portugal. Les résultats obtenus

ont confirmé que l´utilisation de l´ algorithme SVM avec le périodicité de 500 à 2000 cycles a été capable d' améliorer le dégagement de l' équipe F.C.Portugal en comparaison avec l' abordage classique (qui n' englobais pas quelque altération se stratégie pendant tout le jeu).

Acknowledgements

During the development of this dissertation, many are the people that directly on indirectly contributed for it.

First, I would like to thanks my two advisors Professor Luís Paulo Reis and Professor Júlio Garganta for their feedback and constant encouragement during this work.

A special thanks to Professor Mendes Moreira for all the transmitted data mining knowledge and for supporting me to achieve the best of my ability.

In LIACC I need to highlight the decisive contribution of my friend Daniel Silva for all the support and confidence in me throughout this thesis. Without Daniel it would have been much more difficult. Also, I like to thanks my master degree students that I supervised – José Mendonça Moura, João Portela and Mário Faria for their enthusiasm during their thesis, which decreased the time spent in their orientation and consequently increased the quality of this dissertation. Finally one last word to my colleagues in I120 room, Rodrigo Braga, Marcelo Petry, Rosaldo Rossetti, Joaquim Silva and Claudia Chituc for the excellent working environment and Professor Eugénio Oliveira for the excellent working conditions offered.

Outside LIACC, I would like to thanks some FADEUP researchers such as Israel Teoldo, Daniel Barreira, João Brito and Daniel Castelão for their contribution and engagement in some collective research works and Pedro Silva, Paulo Pereira and Miguel Azevedo for the long meetings talking about one common passion – soccer – and for the excellent season in Gondomar Football Club in the U10 category. A final thanks to João Canas Ferreira and João Correira Lopes for providing the LaTeX template used in this thesis and Diana Pereira and Maria José Guerra for their contribution in the english and French revision (respectively), which increased the quality of this dissertation.

As this thesis uses the simulator developed for the 2D simulation Robocup competition, a special thanks to Itsuki Noda and Hakyama for their efforts in the development of this platform as well as for the promotion of the RoboCup competition. A last thanks for Alan Mackworth for the initial idea of creation this challenge.

On a more personal note, I would like to thank all my family, especially my grandparents who unfortunately passed away during this process. Also a special thanks to my parents and my twin brother; they were my constant support during this process where the moments of euphoria often mingled with moments of anxiety. Their constant believe in my abilities have led me to develop the best possible work and because of that this work is devoted entirely to them.

Finally this work is supported by the Portuguese Foundation for Science and Technology (FCT), under Doctoral Grant SFRH / BD / 44663 / 2008.

"If you know/study your enemy in a hundred battles,
you will win all of them"

Sun Tzu

Contents

1 Introduction **1**
1.1 Context and Motivation . . . 1
1.2 Objectives and Approach . . . 2
1.2.1 Thesis Question . . . 2
1.2.2 Approach . . . 3
1.3 Contributions . . . 3
1.4 Structure of the Thesis . . . 4

2 Autonomous Agents and Multi Agent Systems **7**
2.1 Agent Definition . . . 8
2.2 Agent Attributes . . . 10
2.3 Environments . . . 12
2.4 Agents classification . . . 13
2.5 Agent Architecture . . . 14
2.5.1 Layers Architecture . . . 16
2.5.2 BDI Architecture . . . 16
2.5.3 Generic Architecture of a Social Agent . . . 17
2.5.4 Autonomous Agents Aplications . . . 17
2.6 Multi Agent Systems Concept . . . 18
2.7 Distributed Artificial Intelligence Versus MAS . . . 18
2.8 Multi Agent Systems Motivation . . . 19
2.9 Communication in MAS . . . 20
2.9.1 Architectures subsystems of communication . . . 20
2.9.2 Communication Characteristics . . . 21
2.9.3 Communication Languages . . . 23
2.10 MAS Learning . . . 28
2.10.1 Different Learning Types in MAS . . . 28
2.10.2 Credit or Blame Determination . . . 28
2.10.3 Important Features in Learning . . . 29
2.10.4 Learning Algorithms . . . 30
2.11 MAS coordination . . . 33
2.11.1 The need to coordinate agents . . . 34
2.11.2 Cooperative versus Competitive MAS . . . 34
2.12 Conclusions . . . 35

3 Ontologies 37
3.1 Fundamental Concepts about Ontologies . . . 38
3.1.1 Ontologies Definition . . . 38
3.1.2 Ontologies Main Objectives . . . 39
3.1.3 Ontologies Componentes . . . 39
3.1.4 Ontologies Types . . . 40
3.1.5 Ontologies versus Other Similar Notions . . . 42
3.1.6 Ontologies and Semantic Web . . . 44
3.2 Ontologies Languages . . . 45
3.2.1 Evolution of the Ontologies Language Representation . . . 46
3.2.2 Classical Languages . . . 46
3.2.3 Markup Languages . . . 47
3.2.4 Languages Comparison . . . 49
3.2.5 OWL Language and its extensions . . . 49
3.3 Ontology Tools . . . 52
3.3.1 Features of Ontologies Tools . . . 52
3.3.2 Development Ontologies Tools . . . 53
3.3.3 Tools Comparison . . . 55
3.3.4 Protege Tool . . . 55
3.4 Methodologies for Ontologies Development . . . 56
3.4.1 Methontology . . . 56
3.4.2 101 Methodology . . . 58
3.4.3 Upon Methodology . . . 58
3.4.4 O4IS Methodology . . . 59
3.4.5 Brief Methodologies Comparison . . . 60
3.5 Conclusions . . . 60

4 Domain Application Analysis: Human and Robotic Soccer 61
4.1 Collective Sports Games . . . 62
4.2 Human Soccer . . . 66
4.2.1 Human Soccer History . . . 66
4.2.2 Soccer Game Evolution Trough the Years . . . 70
4.3 Robotic Soccer . . . 73
4.3.1 RoboCup History . . . 73
4.3.2 RoboCup Federation . . . 75
4.4 RoboCup Leagues . . . 75
4.4.1 Simulation Leagues . . . 77
4.4.2 Small Size Robot League . . . 77
4.4.3 Middle Size Robot League . . . 78
4.4.4 Humanoid League . . . 79
4.4.5 Standard Platform League . . . 79
4.4.6 The RoboCup Rescue . . . 80
4.4.7 The RoboCup@Home . . . 81
4.4.8 The RoboCup Junior . . . 83
4.4.9 The RoboCup Demonstrations . . . 84
4.4.10 Other Associated Challenges . . . 85
4.5 The RoboCup Soccer Simulator . . . 91
4.5.1 Simulation System . . . 92
4.5.2 The Simulated World and the Game Rules . . . 95

4.5.3 Server Communication Protocols 96
4.5.4 Agent Perception 97
4.5.5 Agents Action 100
4.5.6 Heterogeneous Agents 104
4.6 Conclusions 104

5 Tracking Systems **107**
5.1 Literature Review 108
5.1.1 Generic Tracking Systems 109
5.1.2 Sports Video Analysis 110
5.1.3 Object Base Approaches 111
5.1.4 Global Comparison 112
5.1.5 Commercial Available Vision-Based Analysis Systems 113
5.2 Project Description 117
5.2.1 Electrical Infrastructure 117
5.2.2 Global Architecture 118
5.2.3 Database Model 120
5.2.4 Map Editor 121
5.2.5 Real-Time and Historical BI Application 121
5.3 Results 122
5.4 Conclusions 125

6 SocOn - A Soccer Ontology **127**
6.1 Languages used in Team Training 128
6.2 Soccer Concepts Comparison 128
6.3 Structural Ontology Definition 129
6.3.1 Main Class 129
6.3.2 Main Conceptual Relations 130
6.3.3 Data Types 131
6.3.4 Representation of Game Space 132
6.4 Objects Definition 132
6.5 Regions Definition 133
6.5.1 Field Region 133
6.5.2 Variable Regions 136
6.5.3 Team Regions 136
6.5.4 Periods Definition 136
6.5.5 Events Definition 137
6.5.6 Situations Definition 139
6.5.7 Players Definition 140
6.5.8 Players Position 140
6.5.9 Players Attributes 141
6.5.10 Teams Definition 143
6.6 Results 144
6.6.1 Ontology Development Process 145
6.6.2 Concepts Validation 149
6.7 Conclusions 154

7 Statistics Generation 155
7.1 Soccer Analysis Tools . . . 156
7.2 Statistics Definition . . . 157
7.2.1 Soccer Field Regions . . . 159
7.2.2 Soccer Heuristics . . . 160
7.3 Soccer Scientia Tool . . . 167
7.3.1 Soccer Scientia Modules . . . 167
7.3.2 Results . . . 170
7.3.3 Conclusions . . . 173
7.4 Soccer Server Statistical Extracting Tool (SSSET) . . . 174
7.4.1 Results . . . 174
7.4.2 Conclusions . . . 179
7.5 Comparison Study Between Robotic Simulation 2D League versus Human Soccer 179
7.5.1 Method . . . 179
7.5.2 Data Analysis . . . 180
7.5.3 Results . . . 181
7.5.4 Goal Events . . . 181
7.5.5 Extra Time Comparison . . . 184
7.5.6 Final Games View . . . 185
7.5.7 Discussion . . . 186
7.6 Conclusions . . . 189

8 Opponent Modelling 191
8.1 Literature Review . . . 191
8.2 FCPortugal Team Description . . . 194
8.2.1 High-level decision and Cooperation . . . 194
8.3 Methodology . . . 202
8.3.1 Simulation . . . 202
8.3.2 Feature Selection . . . 204
8.3.3 Clustering . . . 205
8.3.4 Training Classifier . . . 205
8.3.5 Selection of the Best Strategy per Cluster . . . 205
8.3.6 Prediction . . . 205
8.3.7 Assignment of the Best Strategy . . . 206
8.4 Experimental Results . . . 206
8.4.1 Feature Selection Algorithm . . . 206
8.4.2 Clustering . . . 211
8.4.3 Training Classifer . . . 212
8.4.4 Assignment of the Best Strategy . . . 213
8.5 Conclusions . . . 214

9 Conclusions and Future Work 215

A Soccer Concepts Survey 219

References 233

List of Figures

2.1 Multi Agent Systems and related fields (adapted from [KV00]) 8
2.2 Taxonomy of autonomous agents (adapted from [KV00]) 14
2.3 Basic Architecture of an Autonomous Agent (adapted from [KV00]) 14
2.4 A BDI Agent Architecture (adapted from [GHN+97]) 16
2.5 (a) Multi Agent System (b) Distributed Problem Solving (adapted from [Bud05]) 19
2.6 Black Board (adapted from [Bud05]) 21
2.7 Message Passing (adapted from [Bud05]) 21
2.8 ACL progression since the agents early days (adapted from [Sin03]) 24
2.9 The design space of agent communication languages (adapted from [Sin03]) . . 25
2.10 Classification (linear separable case) 34

3.1 Relations between the Ontologies Types regarding their generality (adapted from [Gua98]) 41
3.2 Semantic Spectrum (adapted from [CFLGP07]) 43
3.3 Semantic Web Architecture [BL00] 45
3.4 Classical languages for ontologies representation (adapted from [CFLGP07]) . . 47
3.5 Markup languages for ontologies representation (adapted from [Cor05]) 49
3.6 Different Ontologies life cycle phases (adapted from [FGPJ97]) 57
3.7 Different Ontologies phases using UPON Methodology [NMN05] 59

4.1 Read and Edwards Sistematization (adapted from [RE92]) 64
4.2 Sub-Categorization of Net Games and Wall Games with examples (adapted from [HB02]) 65
4.3 Sub-Categorization of Invasion Games with examples (adapted from [HB02]) . . 65
4.4 Sub-Categorization of Base Games with examples (adapted from [HB02]) 65
4.5 Picture carved in stone dating from ancient Greek 67
4.6 Picture representative of the ball fill for the *Giocco del Calcio* 68
4.7 Picture representative of a match in the England Soccer League Competition . . . 69
4.8 Formation Systems Evolution before 1931 70
4.9 Formation Systems Evolution after 1931 72
4.10 General aspect of RoboCup 2009 - Graz 74
4.11 RoboCup Soccer Leagues 75
4.12 Small Size Robot League 78
4.13 Middle Size Robot League 78
4.14 Humanoid League 79
4.15 Sony Four-Legged League 80
4.16 Standard Platform League 81
4.17 Rescue Simulation League (Agent Simulation - (a) and Virtual Robots -(b)) . . . 82

4.18 Rescue Robot League 82
4.19 Robocup@Home 82
4.20 Dance Challenge 83
4.21 Soccer Challenge 83
4.22 Rescue Challenge 84
4.23 CoSpace Demo Challenge 84
4.24 Festo Logistics Competition 85
4.25 Mixed Reality Competition 85
4.26 Virtual RoboCup Viewer of the Bielefed, Germany University 87
4.27 Magic Box Viewer (a) and RoboLog Viewer(b) 87
4.28 Venue Viewer 88
4.29 Caspian Viewer 89
4.30 RA3DM Viewer (a) and Avan Viewer(b) 89
4.31 MAICC Viewer 90
4.32 UML diagram of the objects in the simulation [CFH+03] 92
4.33 Soccer Server Architecture 93
4.34 Soccer Monitors 94
4.35 Frame View Monitor 94
4.36 The flags and lines in the simulation ([CFH+03]) 95
4.37 The visible range of an individual agent in the soccer server ([Sto98]) 98
4.38 Example of the Goalkeeper Catchable Area (adapted from [CFH+03]) 102

5.1 An Individual statistical treatment of a player using ProZone System 114
5.2 Team Statistical Treatment during a soccer game using ProZone System 115
5.3 Events Capture Through Video Cameras in Amisco System 115
5.4 Example of a 2D representation of an image captured during a soccer match (a) and also a representation of players movements during a match using Amisco System (b) 116
5.5 Example of a Event Detection using eSport System 116
5.6 Electrical Infrastructure and WI-FI Network 118
5.7 System's Global Architecture 119
5.8 Database Model 120
5.9 The Apex's engine 122
5.10 Soccer Exercise Conducted 123
5.11 Signal Strength and Noise for WI-FI Network 123
5.12 Box density over an exercise 124
5.13 Player density in game field 124
5.14 Striker Position during a pass in three consecutive instances (T(0), T(1), T(2)) 125

6.1 Class Hierarchy of the Main Developed Concepts. 130
6.2 Main Proprieties-Object presented in the Soccer Ontology. 130
6.3 Main Concepts between the Ontology Conceptual Areas. 131
6.4 Ontology Input Data. 131
6.5 Representation of Game Space. 132
6.6 Hierarchical Representation of Object Types. 133
6.7 Field region definition using 14 distinct areas. 134
6.8 Field region definition using 36 distinct areas. 135
6.9 Soccer field official dimensions. 135
6.10 A Team Region. 136

6.11 Hierarchical Representation of the Actions Classes. 138
6.12 Hierarchical Representation of the Situation Classes. 139
6.13 A Player Classification by Matrix Position. 141
6.14 Different Tactical Parameters for a Soccer Team. 144
6.15 SWRL Question for Identified the Field Regions of the Simulated Objects (a), Field Regions for Simulated Objects (b) 146
6.16 Field Region Detection for Simulated Objects 147
6.17 Representation of Team Proprieites. 147
6.18 Representation of a Static Game Situation. 148
6.19 SWRL Question for Identified Team Tactics. 148
6.20 Team Tactical Results. 149
6.21 Midfield Regions Divisions to Validate the Crossing Concept. 150
6.22 Defensive Actions for a Specific Game Situation. 151
6.23 Game situation using two teams: blue and red team. 153
6.24 Game situation using two team: orange and green team. 153

7.1 SoccerScope2 Interface. 156
7.2 Team Assistant Interface. 157
7.3 Soccer Game Structure. 158
7.4 Soccer Field Regions Divisions. 160
7.5 Example of Distance Algorithm 163
7.6 The Goal Opportunity Triangle Area – the black dot represents the attacking player 165
7.7 The Three Soccer Scientia Main Modules – marked with a red line 167
7.8 An example of a Force Model 168
7.9 An example of a team successful pass representation through the soccer field . . 169
7.10 An example of the frequency of players successful passes 169
7.11 *Kick* Player Detection 172
7.12 Comparison Between the Three Groups of Games Analyzed. 173
7.13 Zone Dominance for the Top Three Tournament Teams–The Direction of Team's attack is marked with white arrow 175
7.14 Successful and Missed Passes for all the Teams in the RoboCup 2D Simulation League 176
7.15 Average of Wing Variation per Game 177
7.16 Average of Wing Variation per Game 177
7.17 Average of Goals Scored and Goal Opportunities per Team 178
7.18 Ratio Between Goals Scored and Goal Opportunities per Team 178
7.19 Frequency of scoring by period of time (first and second half) and by reality . . . 182
7.20 Type of offense style used to score a goal by reality 182
7.21 Action prior to goal by reality 183
7.22 Frequency of Set Pieces types 183
7.23 Area from which the goals were scored 184
7.24 Pass Statistic in Human game finals by country 186
7.25 Different Types of Shots in Human Game Finals 186
7.26 Median of Pass Statistic for RoboCup games between 2006-2009 187
7.27 Pass Statistic for RoboCup Finals by Team 187
7.28 Shot Statistic in Robocup Finals by Type 188

8.1 FC Portugal's Strategical Model (adapted from [RL01]) 195
8.2 Gouraud Shading Interpolation [RLML10] 196

8.3 Example of 1-4-3-3 formation definition 197
8.4 Example of a Kick-off Set Play involving 4 steps 198
8.5 Example of a Kick-off Set Play involving 2 steps 199
8.6 Example of a Free Kick Set Play direct to the goal 199
8.7 Example of a Free Kick Set Play where the ball possession is first priority 200
8.8 Example of a Goal Kick Set Play involving 7 steps 200
8.9 Example of a Goal Kick Set Play involving 3 steps 201
8.10 Project Architecture 202
8.11 Relation between Input Variables and Target Variable divided by groups (Ball Possession, Goals, Passes, Set Pieces, Shots respectively) 207

A.1 A Crossing Game Situation Analysis. 220
A.2 Defensive Game Situation for a Specific Game Period. 222
A.3 Team and Players Position - First Game Situation Analysis. 224
A.4 Team and Players Position - Second Game Situation Analysis. 226
A.5 A Static Game Situation. 229

List of Tables

2.1 Comparison of different Agent's definition . . . 9
2.2 Classification of Agents (adapted from [KV00]) . . . 13

3.1 Comparison between the different types of ontology . . . 43
3.2 Evolution of the World Wide Web (adapted from [CFLGP07]) . . . 45
3.3 Comparison between the three OWL sublanguages . . . 50
3.4 OWL classes descriptions and their constructors . . . 51
3.5 OWL proprieties characteristics and their constructors . . . 52

4.1 Comparison Between Commentator Systems (adapted from [ABTi+00]) . . . 91
4.2 Client Command Protocol ([CFH+03]) . . . 96
4.3 Client Action Protocol ([CFH+03]) . . . 97
4.4 View Widths available in the visual information . . . 98
4.5 Heterogeneous Players Parameters ([CFH+03]) . . . 104

5.1 Technology Comparison . . . 113
5.2 Professional Performance Tools Comparison . . . 117

6.1 Soccer concepts comparison between the analyzed approaches. . . . 129
6.2 Players Position Classification Through the Team Regions. . . . 140
6.3 Cartesian Coordinates Values for the Simulated Objects. . . . 145
6.4 Queries Representation of Teams Proprieties. . . . 146
6.5 Evaluation of Players Characterization. . . . 151
6.6 Assessment of the risk degree of a team. . . . 152
6.7 Defensive Line Style Concept using as reference the distance between the last defender and his goal line. . . . 152
6.8 Team-Width Concept. . . . 152
6.9 Positioning Results of the four teams. . . . 154
6.10 Formation Results of the four teams. . . . 154

7.1 Comparison of the Robotic Soccer Analysis Tools. . . . 157
7.2 An example of a collective team performance . . . 170
7.3 Average Frequency of Events . . . 170
7.4 Accuracy Results for Successful and Missed Passes . . . 171
7.5 Accuracy Results for End Line, Intercepted and Target Shots . . . 171
7.6 Accuracy Results for Goals and Outsides . . . 172
7.7 Accuracy Results for Offsides and Intercepted Offsides . . . 173
7.8 Generic Comparison between Human versus Robotic Soccer . . . 181
7.9 Extra Time Comparison . . . 184

8.1 The Passes Group 203
8.2 The Shots Group 203
8.3 The Set Pieces Group 204
8.4 The Goal Group 204
8.5 The Ball Possession Group 205
8.6 Statistics that Composed the Mars Function 208
8.7 The Highest Correlation Coefficient Table for Soccer Variables 209
8.8 Statistics that Composed the Mars Function without 1-BadDefDefensive 209
8.9 Statistics that Composed the RreliefF Function 210
8.10 Statistics that Composed the RreliefF Function after seven iterations 210
8.11 Performance Comparison Table between the Best and the Worse Teams 211
8.12 Statistics that Composed the Mars Function 212
8.13 Ranks of the Friedman test 213
8.14 Ranks of the Friedman test 214

Abreviations and Symbols

ACL	Agent Communication Language
AI	Artificial Intelligence
API	Application Programming Interface
CSG	Collective Sport Games
Cycl	Cyc Language
DAI	Distributed Artificial Intelligence
DAML	Darpa Agent Markup Language
DARPA	Defense Advanced REsearch Projects Agency
DB	Data Base
DL	Descriptive Logic
DPS	Distributed Problem Solving
DPRE	Dynamic Positioning and Role Exchange
ETL	ElectroTechnical Laboratory
FIPA	Foundation for Intelligent Physical Agents
HTML	HyperText Markup Language
KIF	Knowledge Interchange Format
KQML	Knowledge and Query Manipulation Language
KSE	Knowledge Sharing Effort
MAS	Multi Agent System
OCML	Operational Conceptual Modeling Language
OIL	Ontology Inference Layer
OKBC	Open Knowledge Base Connectivity Protocol for Knowledge Base Interoperation
OpenGL	Open Graphics Library
OWC	Onto Web Consortium
OWL	Web Ontology Language
PAI	Parallel Artificial Intelligence
RDF	Resource Description Framework
RuleML	Rule Markup Language
SBSP	Situation Based Strategic Positioning
SE	Software Engineering
SGI	Silicon Graphics, Inc.
SGML	Standard Generalized Markup Language
SHOE	Simple HTML Ontology Extension
SQL	Structured Query Language
SQWRL	Semantic Query-enhanced Web Rule Language
SWRL	Semantic Web Rule Language
TAC	Trading Agent Competition

UML	Unified Modeling Language
UPON	Unified Process for Ontology building
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	eXtensible Markup Language
XOL	Ontology Exchange Language

Chapter 1

Introduction

The aim of this chapter is to provide an overview about the performed work reported in this thesis. After a brief exposition about the context and motivation of this work, the approach used and the scientific contributions produced are presented. Finally the thesis structure is exposed.

1.1 Context and Motivation

Soccer is one of the Collective Sports Games (CSG) with more participants and supporters all over the world. It is played by over 240 million players in 1.4 million teams and 300 thousand clubs around the world [AYA+08]. In this particular sport, two teams, with eleven players each, try to reach the objective of scoring at least one more goal than the opposing team, thus achieving the victory in the match. In competitive sports, and consequently soccer, the level of performance is determined by a set of complex intercorrelated variables: Technique (coordination abilities, kinetic skillfulness), Tactics (cognitive and planning abilities), Psychological factors (motivation, desires, willingness) and Fitness [Wei97]. The behavior of players and the decision making processes can range from the most simple reactive behaviors, such as running towards the ball, to complex reasoning that take into account the behavior and perceived strategies of both team-mates and opponents [Jon98].

According to Grehaigne et al. [GBD97], the essence of the game can be described as: A team must coordinate its actions to recapture, conserve and move the ball so as to bring it within the scoring zone and to score a goal. Training a soccer team is mainly a task of enhancing team performance by providing feedback about the performance of the athletes and team [HB02]. Human observation and memory are not reliable enough to provide accurate and objective information from athletes in high-performance competitions. In most team sports, an observer is unable to view and assimilate the entire action taking place on the playing area, due to its attention to the game critical areas, which makes most of the peripheral play action to be lost [HF01]. During a soccer match, the coach can become the recipient of a great amount of information; as a result, he might not be able to evaluate and objectively exploit all the technical and tactical elements that may come along [FM91]. Emotional factors, such as stress and anger, or even more subjective aspects such as prejudice, can also lead to a decrease in concentration, and consequential misinterpretation of the game reality [CWR07]. In consequence of that, many are the coaches that tried to collect all this information through automatic performance analysis systems.

In recent years, the growing need and interest in performance analysis have led to new forms of match analysis techniques. Modern-day techniques include video-based statistical analysis systems, video-based tracking and electronic tracking systems [CWR07][AVM+09]. Nowadays, there are many types of analysis software, capable of calculating a large amount of game statistics. However, they still transfer this data to the training session and, not rarely, different software systems, analyzing the same data produce different results, as shown by Randers et al. [RMH+10]. This means that depending on the analysis software being used, the user can obtain contradictory information about his next opponent. Due to the fact that, in many situations, throughout the soccer competitions, the time to prepare the team for the next match (opponent) is too short and, considering that it is impossible for a human to process all the information produced by the analysis software, throughout the years, the variable *scored goals* has, for natural reasons, received considerable attention in notational analysis [JJM04]. However, to increase this number, a combination of a large set of complex variables must appear in the soccer match. In consequence of that, the technical staff of a professional soccer team includes a large number of scouts, who are responsible for elaborating detailed reports, where the manner in which the opponent plays is completely described. Unfortunately, these reports need to be processed by a field coach in order to do the bridge between observation reports and the training session, for instance.

Sun Tzu in his book The Art of War [Gri63] claims that *If you know/study your enemy in a hundred battles, you will win all of them.*

Normally, in the preparation of a match, a soccer coach tries to adapt his strategy in order to increase the possibilities to defeat his opponent. In this project, machine-learning techniques were used to construct the best behavior online model for our team in order to increase its performance according to a specific opponent.

1.2 Objectives and Approach

1.2.1 Thesis Question

The main question addressed in this thesis is

How can we improve our team performance using final game statistics?

More specifically, this thesis contributes to a team model capable of improving the agent´s behavior and consequently team performance, in a specific domain where:

1. There is a need for real-time decision-making;

2. There is noise in sensors and actuators;

3. There is a set of agents (players) with the same high-level goal: *Teammates*;

4. There is a set of agents (players) with a conflicting high-level goal: *Opponents*;

This thesis contributes with a successful method to construct a team model – using machine learning techniques – implementing effective behaviors in agents in a soccer simulated domain.

1.2.2 Approach

The general approach for answering the thesis question is based in the creation of a team model capable of improving our team (FC Portugal) performance. For that, we need to use an environment that was capable of simulating a match between two distinct teams (with different goals and different ways of play), allowing real time changes in what concerns to the players behavior and supporting noise features for both teams.

RoboCup competition and more precisely the official simulator that supports the 2D simulation league totally fit the above characteristics. Regarding the main goal of this thesis, and due to the online availability of the binaries of the teams, as well as the log files of the competition matches, this environment proved to be an excellent testbed for this thesis. Also, it is important to note that all of the thesis contributions were originally developed in simulated robotic soccer. However, the author believes that it is possible to generalize this approach to human soccer environments.

An initial assumption was that the soccer coach had no possibility to improve team performance during the game, without prior knowledge of his opponent. Because of that, our development process consisted in two modules: offline (constituted by 5 steps) and online mode (constituted by 2 steps). The offline mode starts with the calculation of final game statistics capable of representing the opponent´s behavior. Then, feature selection algorithms were used to construct a subset of those statistics that most influence the final match result and the produced data was grouped into k clusters. After that, 3 data mining techniques were trained to predict the group that better characterize a given input data and finally, the expected best strategy for a group of opponents with similar behaviors was defined. On the other hand, using a data mining technique, the online mode starts to predict the group for which the given data is expected to be more similar with (using the training model produced in the offline phase) and finally, the expected best strategy for a particular way of playing of the opponent was implemented in our team.

1.3 Contributions

To increase the understanding of the scientific contributions of this thesis, the contributions were split into two groups:

1. **Initial work** that includes the work developed before the final choice of data used in this thesis.

   (a) **A Wi-Fi based Location System** was used to track a set of players during a soccer match (in an outdoor space). Basically, this experience consists in tracking players in a Penalty Area identifying their preferential movements, and the most populated area, among other features. Despite the achieved results being satisfactory, this experience proved that in this domain, the past developments were not sufficiently good to produce a tracking system capable of automatically following the different soccer agents during a match.

(b) **A Soccer Ontology** was developed allowing the representation of a large set of soccer concepts and their relationships. The validation of the concepts presented in this ontology was executed using inquiry validation technique and the results were very promising.

2. **Improve Our Soccer Team performance** that includes the works that use the RoboCup 2D simulation league log files as their base.

   (a) **A Cartesian Framework** was developed, capable of automatically calculating final game statistics using only the players and the ball cartesian coordinates. In the event detection process, a sequential analysis was used, and it proved to be an excellent approach for this type of problem. To validate this approach, the 2D simulation league log files between 2006 and 2009 were used and, at the end, the results were very satisfactory, attending the given use of such low initial information.

   (b) **A Statistical Generator** was implemented capable of automatically calculating a large set of game statistics defined by a soccer expertise board. To implement those statistics, all the information presented in the log files was used. Having the main goal of the RoboCup competition in mind, this tool was also used to execute a brief comparison study between two realities: human versus robotic 2D simulated soccer. For that, a set of log files of the 2D simulation league competition between 2006 and 2009 were selected as well as some European and World Cup games finals. The results showed that these two realities still present some significant differences in some final game statistics.

   (c) **Construction of a Team Modeling** capable of improving our team performance during a soccer match. This construction includes two different modes and the use of final game statistics (calculated with the statistical generator), the use of two feature selection algorithms as well as 3 data mining techniques and an online coach. At the end it is clearly proven that the approach that uses the team modeling increases our team performance.

1.4 Structure of the Thesis

The remainder of this thesis is organized in six parts, as follows:

1. **Part I: Literature review**

(a) **Chapter 2** provides an extensive survey about two areas that are transversal to the thesis. While other chapters contain their own literature review section describing the most relevant research to their contents, this chapter surveys the field of autonomous agents and multi agent systems;

(b) **Chapter 3** provides an extensive survey about the ontologies development which was extremely important in the initial phase of the project helping to define many soccer concepts and their relationships;

2. **Part II: RoboCup and Human Soccer Overview**

   (a) **Chapter 4** introduces a contextualization of the thesis study environment including an exhaustive RoboCup competition overview as well as the history of human soccer and its evolution throughout the times;

3. **Part III: Initial Work**

   (a) **Chapter 5 and 6** expose the initial development work in this thesis including the use of a tracking system based on Wi-Fi technology in a very reduced soccer game situation and the development of a soccer ontology which constitutes one of the first approaches to understand soccer concepts and their relationships.

4. **Part IV: Development of Statistical Tools**

   (a) **Chapter 7** describes the development steps of two statistical tools: the first one based its knowledge exclusively in players and ball cartesian coordinates and the other in the information gathered by the soccer server. For this work, a soccer expert board was constituted to establish the set of calculated statistics. At the end of this phase, a comparison between a robotic and human reality was performed using the calculated statistics provided by the second statistical tool.

5. **Part V: Team Modeling**

   (a) **Chapter 8** presents the three phases used to create a robotic team model capable to improve its performance against an opponent. In this chapter, two feature selection algorithms were used as well as three data mining techniques. To compare the final results produced by those techniques, the Friedman test was used. Finally, it was proved that through final game statistics, it is possible to improve the performance of a robotic team;

6. **Part VI: Concluding Remarks**

   (a) **Chapter 9** summarizes the contributions of this thesis and presents some future work trends.

Chapter 2

Autonomous Agents and Multi Agent Systems

Troughout the years the term Agent has been used in many ways due to the lack of a scientific standard definition accepted in the research community. A possible agent definition is a computer system, within a given environment, which has the perception of that environment through sensors, has capacity of decision, act autonomously in this environment through actuators, and has capacity for high-level communication with other agents and / or human, to perform a given function for which it was designed [Rei02]. This environment could be physical (university campus, hospitals, soccer fields among others), a simulated environment or even a computer.

Normally, through its sensors and actuators, an agent is capable to interact in an autonomous way with its reality. Regarding physical agents its sensors could be cameras, microphones, proximity sensors (based in technologies like infrareds or ultra sounds) and the actuators could be its arms or legs. For the virtual agents that definition is a bit different and complex presenting a high variation depending on the agent type ant the environment. Nevertheless, an example could be a virtual soccer agent that through its sensors is capable of detecting its teammates and execute a play that could consist in a combination of moves. An example of that domain application is the RoboCup competition explained in the chapter 4 of this thesis.

Multi Agent Systems (MAS) are constituted by multiple agents that presented autonomous behavior and interact with the other system agents. These agents must present two main characteristics: act in an autonomous way to achieve its goals and being capable to interact with other agents including features like coordination, cooperation, competition and negotiation. The concept of MAS arises from the area of distributed AI as illustrated in Figure 2.1.

Since the 90's the MAS area won increasing notoriety [Woo02]. This notoriety is represented by an increased number of books, journals and conferences and also with the creation of an Excellence Network for Computation based on Agents (AgentLink) which the main goal is to promote and coordinate the MAS research in Europe. Another example of the increasing development in

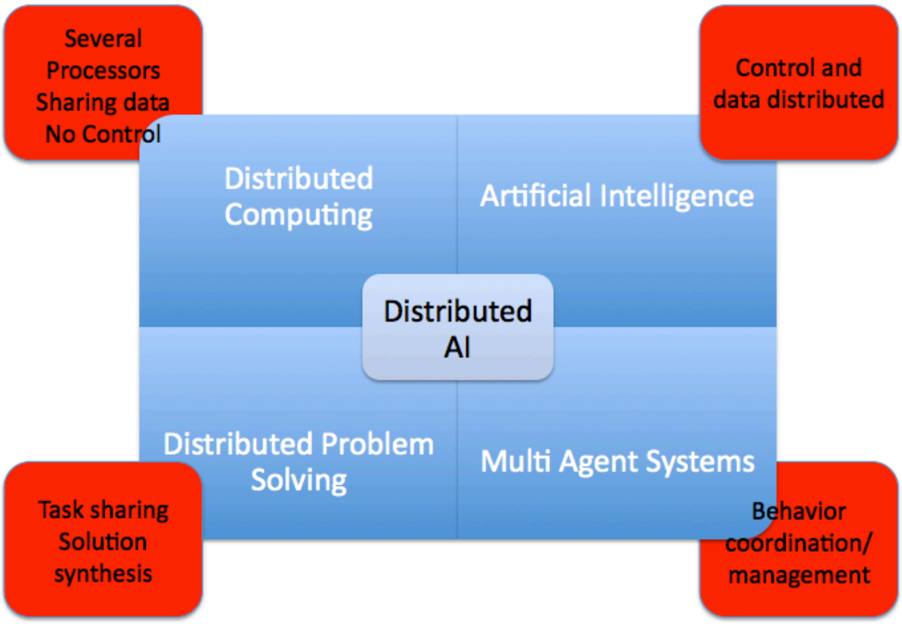


Figure 2.1: Multi Agent Systems and related fields (adapted from [KV00])

this area is the creation of international competitions in different areas like RoboCup [KTS+97] or TAC(Trading Agent Competition) [WGSW02]. In the first competition a set of autonomous agents (physical or virtual) compete in a soccer match following soccer standard rules. In the second competition a set of autonomous agents compete in a simulated travel agent environment supplying travel packages to their costumers.

This chapter presents the autonomous agent and MAS concept. First we analyze and compare some agents definitions and the major attributes of an agent were also presented: autonomy, mobility, reactivity, among others, as well as their major application environments and their principles of research domains. After that, a brief agent classification is presented as well as the principles of agent architectures. In what concerns MAS, an overview concerning to the agents' communication, agents coordination and learning in MAS is also presented.

2.1 Agent Definition

By doing a brief search in the agent definition literature a set of generic properties can be detected Table 2.1.

One of the most well accepted agent definition was proposed by Wooldrige and Jennings [WJ95] which define an agent as a computational system based on software that has the following properties:

1. **Autonomy** — The agents operate without any direct intervention of human beings or any other agent and has some kind of control in its action and its internal state;

Definitions \ Capabilities	Execute a set of predefined tasks	Persistence	Autonomy	Capable of understand and interact with the Environment	Capable to Communicate using a High level language
Russell and Norvig [RN09]				x	
Maes [Mae96]	x		x	x	
Hayes-Roth [HR95]				x	
Franklin and Graesser [FG97]	x	x	x	x	
VanLehn [Van91]				x	
Pickett et al. [Pic00]	x				
Rudowsky [Rud95]	x		x		
Smith et al. [SCS94]		x			
Genesereth and Ketchpel [GS94]	x			x	x
Reis [Reis03]	x		x	x	x

Table 2.1: Comparison of different Agent's definition

2. **Reactivity** — The agents are capable to quickly react to the environment changes;
3. **Pro-Activity** — The agents are capable to not only react to the environment changes but also to demonstrate different behavior according to its goals;
4. **Social Abilities** — The agents are capable to interact with other agents using agents' communication languages. Normally, these languages are designated as *ACL-Agent Communication languages*.

To construct a completely Autonomous agent is an important challenge in the Artificial Intelligence area. However, the total achievement of this goal is not totally desirable (in an extreme situation the agents could dominate the world) and some researchers believe that although the agent autonomy is an important issue it is desirable that an agent can follow a human being's orders [Rud95].

In what concerns to the agents reactivity, it is important that the agents are able to adapt to environmental changes. However, this adaptation must be considered in the terms of the medium and long goals. The pro-activity capability is easy to achieve in a static environment. However for dynamic environments and specially the ones that an agent does not have complete access, to achieve this capability is a complex task.

Finally the social capacity includes an interaction process between agents that not rarely have different goals and different environment perception. This process may involve coordination, cooperation, competition and sometimes negotiation. This capacity combined with the pro-activity and reactivity capacities occupied a major role in collaborative work where a set of agents have the same goal [RL01]. The same authors also present another set of properties that it is desirable for an agent:

1. **Mobility** — An agent must be capable to move from one place to another. This capacity occupies a major role in a environment like a computer network;
2. **Trusted** — An agent must always exchange reliable informations with other agents;

3. **Benevolence** — The agents must collaborate with the other agents to fulfill a common goal or to demonstrate a productive behavior to accomplished its individual goals;

4. **Knowledge and Belief** — The agents must have informations regarding the information and also the capability to process those informations. The belief is related to the agent's notion about a specific fact. This feeling could change over the time;

5. **Intention and Obligation** — Intentions are related to the long time agent goals and obligations are related to the previously commitments made by the agent;

6. **Rationality** — The agent's behavior is based on its knowledge and, through its capabilities the agent will take its options optimizing the number of achieved goals.

Other common properties for agents are:

1. **Intelligence** — The agent's state is formalized by its knowledge (goals, assumptions, plans) and also by its interaction with other agents (through symbolic language). The agent has also capacity to solve new problems and adaptability to new situations;

2. **Character** — An agent has a credible personality and sometimes has also an emotional state;

3. **Learning Capacity** — Based on its previous experiences, this capacity allows the agent to acquire new knowledge;

4. **Temporal Continuity** — Normally an agent is a process that is executed through the time.

2.2 Agent Attributes

In the previous section, we discussed a set of agent characteristics that could be used to classify the agents by classes or topologies. The choice of these attributes varies according to the features that developers want to included in the their agents.

One of the most consensual agent characteristics (in terms of researchers perspective) is the agent autonomy. For Nwana [Nwa96] the autonomy principles are related to the agents decisions (follows their own rules) without any human influence. Other authors claim that the autonomy increases with the pro-activity increase [WJ95].

Huhns and Singh [HS97] divided the agent's autonomy in five levels:

1. **Absolute Autonomy** — The agent is not predictable and has entire control of its perceptions, actions and reasonings;

2. **Social Autonomy** — The agent knows the other system's agents and relates with them (specially in coordination or negotiation situations) without loosing its autonomy;

3. **Interface Autonomy** — Sometimes there is a system where the absolute autonomy is impossible to achieve. In this case the agent will prefer to maintain an autonomy related to the exterior interface;
4. **Execution Autonomy** — It is related to the freedom that the agent has to execute its action in the environment;
5. **Project Autonomy** — Related to autonomy of implementation that a developer puts in his developed agents.

Mobility is an attribute that represents the capacity that an agent has to move within a network. This attribute is specially interesting when the problem consists in searching for a specific information through the web. Normally it brings stimulating challenges: not allowing the network overloading, the agent's security (sometimes the agent could be a virus agent– more related to the mobile agents [KJ97]) and, finally in the physical robots area challengers like: trajectories' definitions, location methodologies, sensorial integration which constitute important issues.

The reactivity is another agent characteristic that could have two different meanings: the first one is related to the adaptation capacity that the agent has resulting from an environment change, the second meaning is when an agent uses a reactive planning and uses condition-action rules to define its behavior. Another characteristic is Pro-Activity that can be classified as an initiative capability representing an independent behavior. Actions are defined according to the environment changes but also the generic agent goals. If in an environment there is more that one agent, it will be necessary to establish a communication process. This process can englobe communication between different agents but also between agent and human or with the environment itself [FG97]. To established this process the agents need actuators (network adapters, sound columns among others) and knowledge that allows the transportation of those messages. Also they need to share/use the same communication protocol.

Social Ability and Cooperation are other two common characteristics. The first one represents the capacity to interact with other agents. The agents must be capable to share the messages semantic and use the same ontology (concept explained in chapter 3) according to the environment. Normally the term cooperation is defined as agent collaboration (teamwork) to achieve a common goal. Some authors also argue that this collaboration is the main reason for the existence of the multi agent systems [Nwa96]. To achieve that common goal the agent must have the capability of communicating with other agents and in some cases with humans through a communication language [WJ95].

Normally a developer programs the agent's features according to the current state of the environment. However, this environment can suffer many transformations and the agent cannot stop executing its tasks. Because of that, learning from previous experiences could be the key factor to help agents follow their own path. In this entire process, of course sometimes the agent will not achieve the success, nevertheless it can learn with its successes and failures to improve its future decisions. The learning process is very complex. Sometimes it is difficult to identify a failure and use that information to improve the future. To facilitate that process a leader/coach agent could be used to improve the knowledge of the other agents providing reliable instructions [LPTW96]. This new agent is also important due to its general vision about the environment allowing the detection of failures.

2.3 Environments

An agent is included in an environment and not rarely, the characteristics of that reality will directly influence the development of the agent. Because of that a careful analysis should be executed in terms of tasks complexity, possible agent actions amount others. In this context a simulated environment could be more complex that a real one.

Russell and Norvig [RN09] divided the environment proprieties into four groups:

1. **Accessible vs Inaccessible** — An accessible environment is the one where an agent obtains through its sensors, complete, precise real time informations. However the majority of the environment is inaccessible like the internet or real physical environments;

2. **Deterministic vs non Deterministic** — In a Deterministic environment there are certainties about the execution of an action;

3. **Static vs Dynamic** — In a Dynamic environment, before an agent executes an action (in his think process) the environment is in constant changing. On the other hand, at the same stage, the Static environment remains without changes;

4. **Discrete vs Continuous** — An environment will be classified as Discrete if there is a finite number of perceptions and possible actions. Otherwise this classifications will changes to continuous. It is important to note that an environment could be classified as continuous (in perception terms) and discrete (in action terms) and vice versa.

The Continuous environment could be verified according to three levels:

1. **World State** — The state of the world can be discrete or continuous. For instance in a board game, like chess, this state is discrete and in some simulated environments, where there are a number of finite states and exists a representation of a set of discrete variables, could also be considered as discrete;

2. **Agent Perception** — In spite of that an environment could be classified as continuous the agent's perception could be discrete. An example of that situation is a robot with boolean proximity sensors in a physical environment;

3. **Agent Action** — Similar to perception the agent's action in a continuous environment could be classified as discrete.

2.4 Agents classification

Over the years many synonyms for the expression *Intelligent Agent* emerged. Some of those are robots, software agents (softbots), knowbots, taskbots or userbots. The agent implementation is a complex task due to the large number of attributes available (previously explained) which turns this task into one of the most complex one in terms of development. Indeed the choice of these attributes is made according to the agent's environment. Some authors have used these attributes to classify the agents in terms of its topologies. A topology is a possible agent classification according to common attributes. Nwana [Nwa96] proposed a topology constituted by seven dimensions:

1. **Mobility** — The agents could be mobile or static. One important note is that a mobile agent could be temporarily located in one machine different from the source machine;
2. **Reasoning Model** — The agents could be a deliberative agent or reactive;
3. **Agent Function** — The main function that an agent performs, could be an information search on the web among others;
4. **Autonomy** — Related to the autonomy degree of the agent;
5. **Cooperation** — Cooperative actions performed (or not) with the other agents;
6. **Learning** — Inclusion (or not) of the learning skills in the agent;
7. **Hybrid Characteristics** — Combination of two or more different philosophies in the same agent;

In 2000 Kim [KV00] presented a new agent classification (Table 2.2). Doing a brief comparison between these two agent classification it is important to note that Kim presents a new agent property (Character) and is related to a new agent research area: emotional agents.

Property	**Meaning**
Reactive	responding to environmental variations in a timely fashion
Autonomous	able to exercise control on its own actions
Goal-oriented	pro-active purposeful, not acting simply in response to the environment
Temporally continuous	continuously running
Communicative	communicates with other agents, perhaps including people
Learning/adaptive	behavior changes as per experience
Mobile	able of transporting itself
Flexible	non-scripted actions
Character	emotional state and personality

Table 2.2: Classification of Agents (adapted from [KV00])

After defining his topology, Nwana [Nwa96] defined seven agents categories according to his architecture and function: collaborative agents, interface agents, mobile agents, information agents, reactive agents, hybrid agents and smart agents. On the other hand Franklin [FG97] and Kim [KV00] presented a taxonomy that divided the autonomous agents into three main groups: biological agents, robotic agents and computational agents (Figure 2.2).

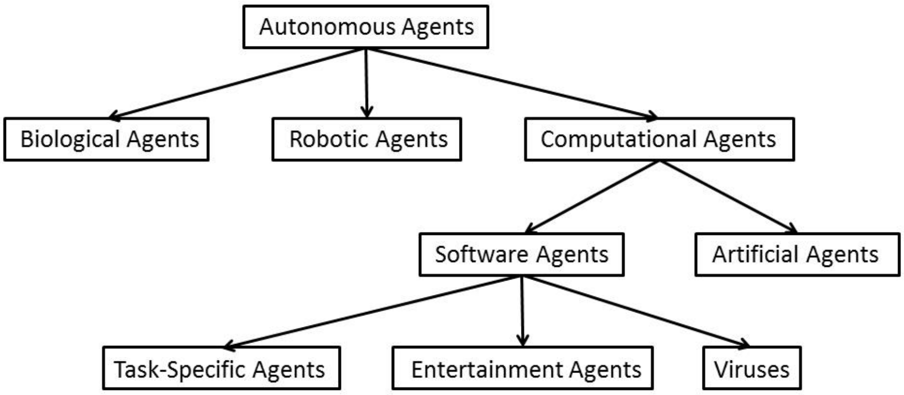


Figure 2.2: Taxonomy of autonomous agents (adapted from [KV00])

2.5 Agent Architecture

A software architecture could be defined as a component configuration that is a part of a system and also the connection that coordinates the activities between the components [AEGO96]. Figure 2.3 illustrates that concept. However, sometimes these architectures did not englobe only the software but also the hardware (an example of that are the robotic agents). Some authors like Knapik and

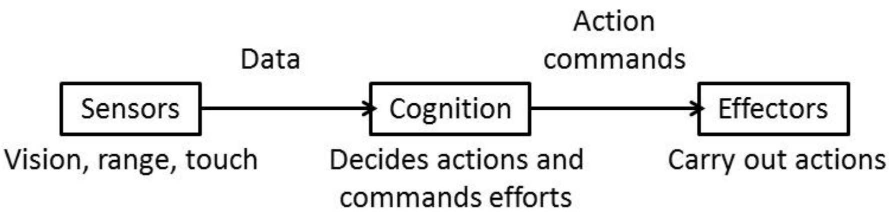


Figure 2.3: Basic Architecture of an Autonomous Agent (adapted from [KV00])

Johnson [KJ97] considered that the discussion involving the quality of the agent architecture is very subjective due to the fact that it is directly involved to specific application aspects (goals). Despite the fact that a a good agent architecture could be considered good for a specific application and bad for another, Mowbray [T.95] defined some generic concepts for a good architecture:

1. **Simplicity** — All architecture concepts and shape should be easy to understand, implement and maintain;

2. **Features** — Select the best architecture and development tools for a given problem;

3. **Expansivity** — The developer should be able to add new features to the architecture;

4. **Portability** — An architecture should have a portable implementation avoiding non-standard solutions.

Over the years many authors presented more simplest agent architecture definitions. For Russell and Norvig [RN09] an architecture agent could be divided in four distinct groups:

1. **Simple Reflex Agents** — The agent used a pre-established set of condition-action rules. In a situation, through a perception, the agent will execute pre-established actions. This kind of agent could be considered the most simple one due to the fact that its behavior is purely reactive;

2. **Model-based Reflex Agents** — Through its perception these agents dynamically update their model of the world. In consequence of that, these agents reactivity will be directly influenced by their past experiences;

3. **Goal-based agents** — Compared to the previous one, this new type of agents keeps the world model updated and also the goals list that it wants to achieve. This type of agents is more flexible because its behavior will change according to its goals;

4. **Utility-based agents** — To measure the level of agent satisfaction according to the achieved goals it is normally used a utility measurer. This measure increases when the agent is closest to reaching its goal and could be extremely useful in situations like goals conflict or even when there is some action uncertainty. Normally these agents are considered more rational because they are capable to evaluate the utility of some actions.

Russell and Norvig [RN09] in their description demonstrate that an agent that based its decisions on facts and learns with its world interactions becomes more effective in pursuing its goals. Also the same authors presented more complex agent architectures that included: learning capacities, decision agents, and agents with planning capacities. Another authors[WJ95] present a more generic division for agent architecture:

1. **Deliberative Architecture** — The agent's decisions are executed following a logical reasoning. The agent possesses an internal world representation and an explicitly mental state that could be changed through symbolic reasoning;

2. **Reactive Architectures** — For this kind of architecture an agent can develop intelligence through environment interactions without any kind of pre-established model [Bro91];

3. **Hybrid Architecture** — The purely reactive agents presents a significant limitation which consists in implementing object-oriented behaviors. On the other hand the purely deliberative agents based on complex symbolic reasoning mechanism become sometimes incapable

of an immediate reaction. A hybrid agent combines these two components in a multi layer architecture. The main idea is to characterize the agents features into hierarchical layers where sometimes the reactive layer has priority over the deliberative one allowing a faster answer to the most important events detected in the environment.

Finally another type of architecture that emerged more recently was the BDI (*Belief-Desire-Intention*) architecture created in 1991 by Rao and Georgeff [RG91]. This architecture is essentially deliberative where the internal agent processing state is defined through a set of mental states: beliefs, desires and intentions.

2.5.1 Layers Architecture

According to the control flow in this type of architectures two layers could be considered:

1. **Horizontal Layers** — Where the software layers are connected to the input sensors and the output actions;

2. **Vertical Layers** — The input sensors and output actions are split by at least one layer.

The main advantage of using a horizontal architecture layer is its conceptual simplicity (because if an agent needs to have different behaviors the developer only needs to implement n different layers). However there is competition between layers and normally the developers use a mediator to solve that problem. Unfortunately that use can strangle the process. In a vertical layer architecture there is no problem because the control flow for each layer is sequential until the last layer generates and executes an action.

2.5.2 BDI Architecture

The agent's beliefs describes the world's state and is related to the agent's beliefs in every moment. The agent's desires describes the agent's desires in each moment which could be conscious or unconscious and reliable resulting an learning agent process. Finally, the agent's intentions are related to the tasks that the agent previously selected to fulfill (Figure (2.4)). Through the relations

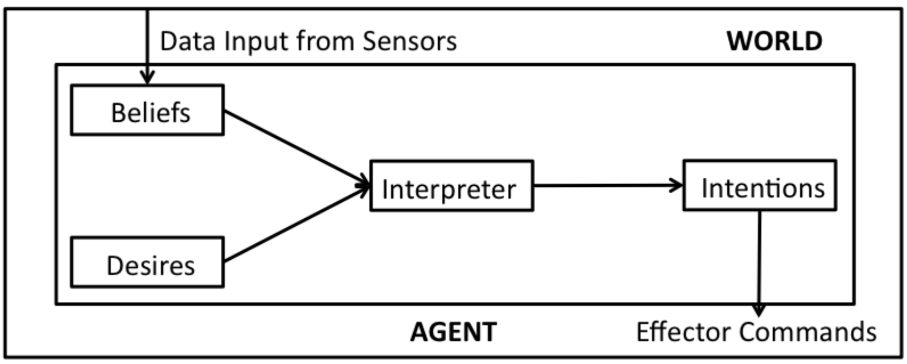


Figure 2.4: A BDI Agent Architecture (adapted from [GHN+97])

of these mental states Rao and Georgeff [RG91] identified three types of agent behavior:

1. **Strong Realism** — The intentions set is a subset of desires set which is a subset of beliefs. If an agent did not believe in nothing it will never have the intention or desire to have it;

2. **Realism** — The beliefs set is a subset of the desires set which is influenced by the intentions set. If a realism agent believes in something, then it will desire and intent to have this thing;

3. **Weak Realism** — The agent does not desire something unless this thing is credible; does not intend nothing unless this thing is desirable; does not intend nothing unless, this thing is credible.

2.5.3 Generic Architecture of a Social Agent

A complete architecture but also a very complex one is the social deliberative agent. This agent possesses explicit models for the other agents and capacity to maintain these models updated through its perceptions and received communications. The models included beliefs, goals and other agents plans.

2.5.4 Autonomous Agents Aplications

Over the years the autonomous agents applications have emerged in many different areas:

1. **Industrial Applications** — Related to telecommunications, control process, electrical energy distribution, air traffic control [Cas09] and transportation system [Ros97];

2. **Search Agent Information** — With the outstanding internet growth in the past years, many thousands of gigabytes information has been uploaded on the web. In consequence of that agent applications that were capable to search a specific information, occupies an important rule in the market [CSSdO09]. Also more personal applications like electronic email manager etc.;

3. **Electronic Business** — Over the years, the electronic business applications through the web increased. Examples of such applications are B2C (Business to Consumer) applications [CO08] and B2B (Business to Business) [RO01];

4. **Medical Applications** —The health area was a standard area for the AI applications. Expert systems like the medical diagnose system MYCIN [BS84] constituted one of the most well succeeded application in AI. Other applications more related to task issues have emerged like an intelligent wheelchair system which main goal was to optimize the different tasks (in terms of time and cost) that a wheelchair has in a hospital facility [BPOR08];

5. **Agent Simulation** —The areas that are included in this task are air flight [SSRO09], soccer [RL01], rescue [CCR$^+$06], traffic control [FRBR09], retail among [VAM09] others;

6. **Robots Control** —The majority of the cases in which the robots control is executed by humans are very difficult and so the autonomous agents were a possible solution for this problem. Agent based architectures for mobile robots control included [NO97] [LLR00];

7. **Cientific Research Competitions** —Human beings need competition to improve their skills. In consequence of that in the past years many were the competition challenges that were emerged with different goals such as Robocup [KTS+97] [KAK+97], TAC-Trading agent Competition [WGSW02] and Portuguese competition Micro-Rato [AFA00].

2.6 Multi Agent Systems Concept

A MAS is a computational system where two or more agents interact or work together to perform specific tasks or achieve common goals. The MAS research is focused on creating small and big semi autonomous agent societies capable to allow agents interaction and fulfill its own goals [Les99]. Normally these systems are distributed ones and in consequence of that coordination tasks occupy an important rule. Over the years many coordination methodologies were proposed and they can be divided in two groups: Their Methodologies for competitive agents domains, where the agents main purpose is to achieve their own self-interests and Methodologies for cooperative agents domains where the agents main goal is related to the interests of all the system's agents (more collective approach).

A MAS is constituted by multiple agents with different perceptions and action capacities in a specific environment. Each agent is capable to influence a distinct part of the world [Jen00]. These influence zones are related to the agent types in the world and its goals.

2.7 Distributed Artificial Intelligence Versus MAS

Normally AI is presented as a part of Computational Science related to the Intelligence systems (systems that include language comprehension, learning, reasoning and problems resolution [BF81]). A more deeply overview in this area could be found at Russell and Norvig [RN09].

The development of powerful processors machines together with the proliferation of computers networks lead to the emergence, in the past years of the DAI area [BG88]. Nwana [Nwa96] also supports this sentence and divides this development into two phases: the first phase (1977 until 1990) that works essentially in deliberative agents (explained before) and the second phase (since the 90's) more related to the development of agent topologies or classes focused on agent intelligence and autonomy.

DAI is essentially focused on problem resolution where different agents work to solve sub tasks using a higher level communication process. The problem resolution could be divided in two groups [BG88] (Figure (2.5)):

1. **DPS (Distributed Problem Solving)** — The problem resolution could be seen as a set of modules. In each module a set of agents were cooperative changing informations and efforts to achieve their goals;

2. **MAS** — In comparison with DPS, the goal in a MAS is to include an intelligent behavior through the coordination of a set of autonomous agents. These agents could work together (with common goals), or alone. In this last situation the agents will have independent goals and their existence is independent of the other agents [DM90]. Also there is another DAI area designated as PAI (Parallel Artificial Intelligence) however, this area is more related to performance issues rather than the agent characterization. So, some authors excluded the PAI from the DAI area.

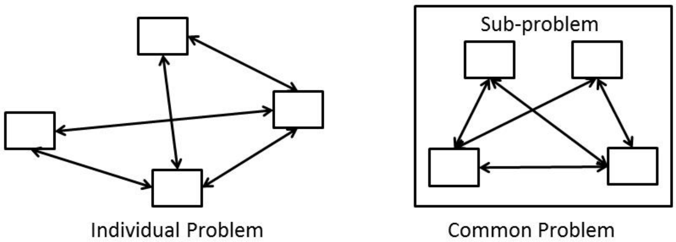


Figure 2.5: (a) Multi Agent System (b) Distributed Problem Solving (adapted from [Bud05])

2.8 Multi Agent Systems Motivation

The MAS motivation is related to the fact that most of the problems are distributed ones. The MAS motivation can be listed as:

1. The problem dimension is to high to be solved by an unique agent;

2. Allow the interconnection and interoperability of multiple legacy systems;

3. Provide a natural solution for geographical and/or distributed problems;

4. Confer simplicity in the conceptual project;

5. Allow cooperative man-machine interface in which both act as agents in the system;

6. Supply problem resolutions in which the experts and the knowledge (to solve the problem) are distributed;

MAS presents higher problem performances where the knowledge is distributed. So there are more reasons to use MAS [SV97]:

1. Scalability, allowing the increase of the number of the intervenient agents in an open system;

2. Robustness, because the use of different agents will decrease in theory with the system lacks;
3. Individual tasks simplicity dividing the global problems in different sub problems;
4. Information Privacy maintenance in individual knowledge of each agent;
5. Individual Intelligence and social behavior study because MAS allows the interoperability between agents;
6. Due to the domain's nature (spatial distribution of the agents);
7. The different tasks parallelism allocates to the agents a more faster execution.

2.9 Communication in MAS

Over the times the communication between entities occupied an important rule in the Computational Science area. Normally in order to provide an agent with communication capabilities it is common to add a module in its architecture that is subdivided in perception components (message reception) and actions (message sending). Also this module is connected to a central agent module (intelligence module) allowing to have access to the received messages and also to define which messages to be sent.

2.9.1 Architectures subsystems of communication

Following Huhns e Stephens ([HS99]) an agent communication subsystem is divided in two groups:

1. **Direct Communication** — In this type of architecture an agent communicates directly with others agents without any intervention. In this process an agent shares its specifications and needs with other agents allowing an agent to take its own communication decisions. The main disadvantage of this architecture is the absence of a coordinator that can avoid the systems blocks (e.g. when the agents decide to send messages at the same time);
2. **Assisted Communication** — With this architecture there is a coordinator agent that is responsible to forward the agent's messages to its receivers. However, this agent does not store any information regarding to other agents and their addresses. Also this agent role (compared to the previous architecture analyzed) will substantially decrease the complexity agent communication process and it will partially solve the communication problem (previously mentioned). The main disadvantages of this approach are the centralization of all the communication process which could lead to a bottleneck situation. Also, if the coordinator agent does not execute its actions (e.g. resulted of a software bug) all the communication system will stop working.

The communication process could be implemented in two ways: Black Board (share information) for al communication agents and Message Passing between agents and its modules.

An agent can communicate its information through a single black board sharing its knowledge with the agent community (Figure 2.6). On the other hand it can adopt a more private and faster

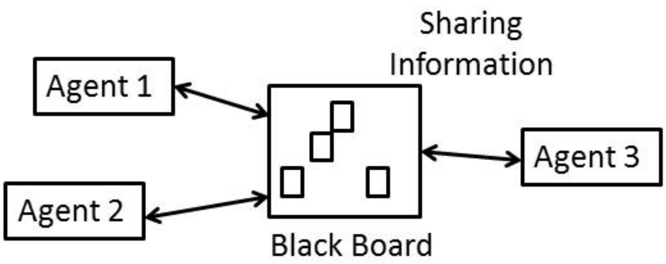


Figure 2.6: Black Board (adapted from [Bud05])

approach (message passing). The communication process is performed directly between agents without the use of a centralized storage information board (Figure 2.7). Normally the first imple-

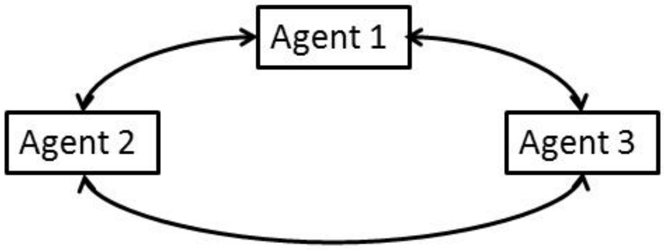


Figure 2.7: Message Passing (adapted from [Bud05])

mentation is more difficult for distributed solutions and needs a complex synchronization system between the agents. In a message passing implementation with an assisted communication or a direct communication architecture it is extremely important to ensure that the messages receiver will always receive its messages (the coordinator agent could occupy an important role in this process). The black board approach does not present this difficulty but presented some privacy and speed issues.

2.9.2 Communication Characteristics

A communication process has two major goals: share knowledge, information, beliefs or plans with other agents and coordinate activities with other agents. However, to execute this process, a communication language must be established according to:

1. **Syntax** — Grammatical structural language that contains rules related to words combination in major units.;
2. **Semantics** — It is the language study in the meaning perspective of words and statements;

3. **Vocabulary** — A set of the symbols used;
4. **Pragmatic Focus** — A set of action and interpretation rules used in the communication;
5. **Speech Domain Model** — Meaning that a symbol set has in a specific context;

An agent communicates with other entities using a specific message, in a certain moment included in a context. So it is extremely important to do the interpretation of the messages according to its context.

2.9.2.1 Characterization of the Messages Meaning

Generically, it is possible to characterize the messages meaning following the characteristics ([Sin97]):

1. **Descriptive versus Prescriptive** — Phenomenon or behavior description. The behavior description is a complex task to understand by the agents. So, the agents communicate changing information about activities and behaviors;
2. **Personal Meaning versus Conventional** — The agent interpretation of a message could be different compared with the other agents (personal meaning). In a MAS (specially in an open environment) it is highly recommended the use of conventional meanings due to the addition of new agents;
3. **Subjective versus Objective** — A meaning is objective when it is interpreted externally (in an agent perspective) and has an explicit effect in the environment;
4. **Sender Perspective versus Society** — Independent of the message's conventional meaning, it can be expressed in the sender or society;
5. **Semantic versus Pragmatic** — The communication pragmatic is focused on How the agents use the communication (how they act according to the communication process). The main facts are: the agent's state, the environment, the syntax and the communication semantic;
6. **Contextuality** — The message interpretation should take into consideration the agent's state, the environment state and also its history. This interpretation should be directly influenced by the communication context;
7. **Identity** — The message meaning depends on the audience and each agent's role in the system;
8. **Coverage** — A communication language can have a small dimension although it should be sufficiently comprehensive in order to facilitate the agents communication process;
9. **Cardinality** — A private agent message should have a different meaning for a public domain.

2.9.2.2 Message Types

Huhns and Stephens [HS99] defend that an agent is capable to send and receive two types of basic messages: assertions and questions. According to the communication capacities it is possible to consider four types of agents:

1. **Basic Agent** — The agent is capable to accept exterior information in an assertion shape but it is unable to execute other communication forms;

2. **Passive Agent** — The agent is capable to accept assertions, exterior questions and answering it in an assertion shape;

3. **Active Agent** — The agent is capable to accept assertions and has the capacity to construct questions and assertions;

4. **Broker Agent** — The agent works as broker between other agents and has the capability to create and accept assertions as well as questions.

2.9.2.3 Protocols and Communication Levels

Usually the protocols' definition involves various levels [HS99]. The inferior level is related to the interconnection agent level while the intermediate level defines the transmitted information format (syntax). Finally, the superior level defines the information specification (semantic).

Generically, a protocol follows a specific data structure constituted by [HS99]:

1. A sender;

2. A receiver;

3. An used language;

4. Coding and decoding functions for the language;

5. Actions that a receiver must execute.

2.9.3 Communication Languages

In the late 80's the Knowledge sharing Effort (KSE) was founded in the USA financed by DARPA (Defense Advanced Research Projects Agency) which had the goal to develop protocols for the exchange and information representation between autonomous information systems. The KSE developed two main products([FWW$^+$94], [LF97]):

1. The **KQML(Knowledge and Query Manipulation language)** — The KQML is an external language for agent communication. The major goal of the KQML is to characterize the needed information for the full context understanding;

2. The **KIF (Knowledge Interchange Format)** — The goal of the KIF language is to represent the knowledge of a certain domain of speech. Primarily KIF was developed to define the messages context expressed in KQML.

Despite several KQML dialects are still being used until today, over the years many are the agent communication languages that have emerged. In the early 90's France Telécon developed Arcol [BS97] (includes a small set of KQML primitives). Similar to KQML, all primitives are assertions or directives but in this particular case they can be composed. After the creation of Arcol, FIPA (Foundation for Intelligent Physical Agents) created its standard language.This language uses the Arcol model and semantics, however it decreases some Arcol performance condition complexity. In 1995 the ACL (Agent Communication language) language has emerged supporting societies constituted by a set of autonomous and heterogeneous agents. Also ACL did not specify the language that an agent must use to specify a content (open dialects). Figure 2.8 illustrates the evolutions of this language through the years.

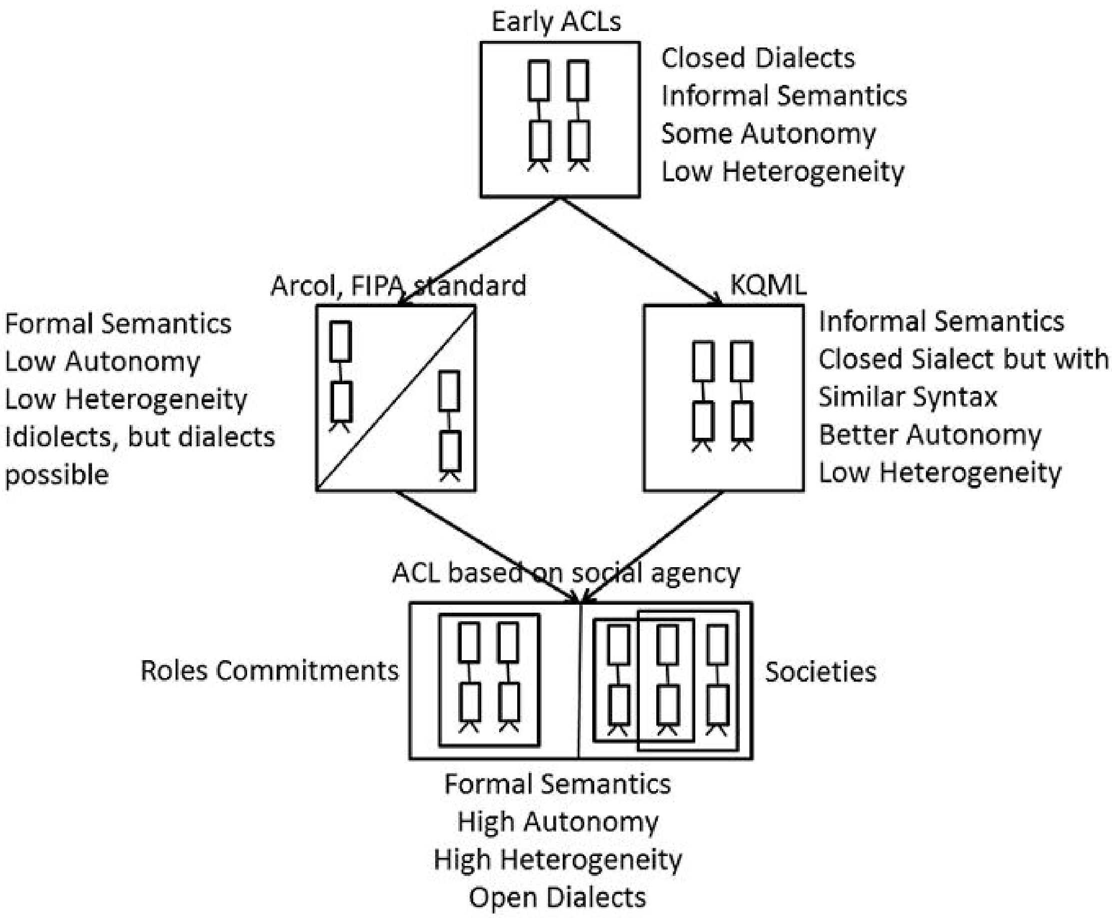


Figure 2.8: ACL progression since the agents early days (adapted from [Sin03])

In a MAS each agent must use the same communication language as well as the discussion concepts meanings used. So it is necessary to established an ontology to specify the meaning of different goals and discussed concepts. In this section elements of meaning perspective related to the previous analyzed languages will be presented. Our analysis will be based on Figure 2.9 which represents the design space of agent communication languages. The region in the lower left represents existing ACLs, which follow a mental agency model and the upper right region represents the desired goals which dictate a social agency model [Sin03].

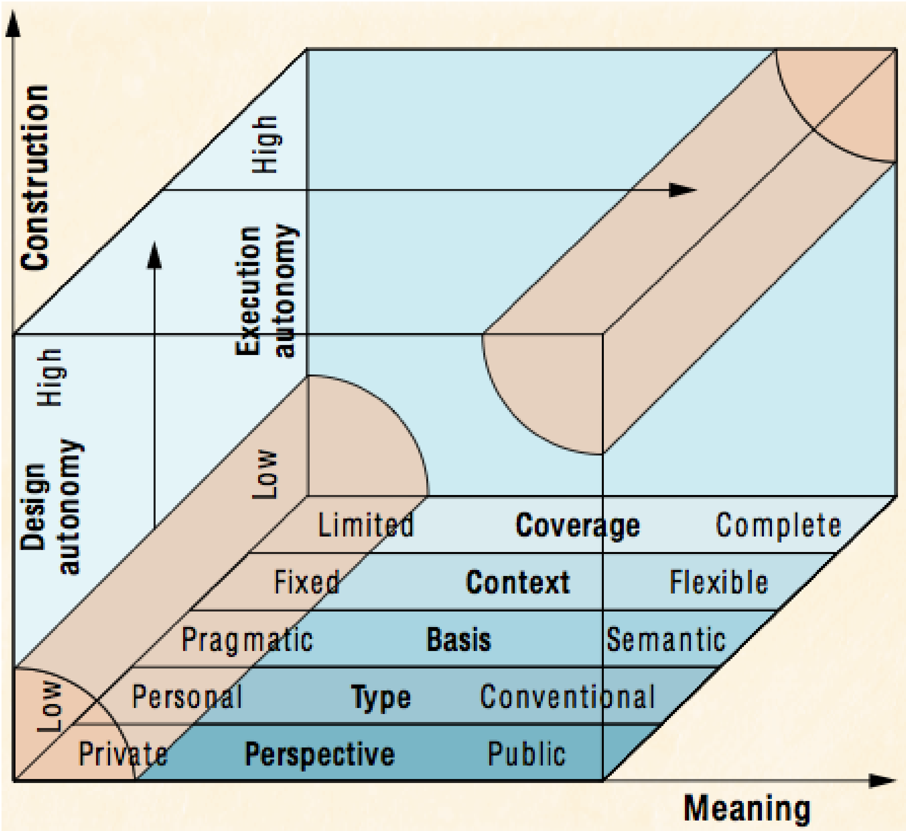


Figure 2.9: The design space of agent communication languages (adapted from [Sin03])

2.9.3.1 Elements of Meaning

In a MAS, agents can cooperate to achieve a common goal or compete to a specific goal. Nevertheless, ACLs must be flexible enough to accommodate abstractions such as negotiation. However in the majority of cases this task is very complex. In consequence of that the meaning of the communication must be analyzed attempting in different parameters such as perspective, type of meaning, basis (semantics or pragmatics), context, and coverage (the number of communicative acts included). Figure 2.9 shows those elements (the region in the lower left characterizes existing ACLs, such as KQML and Arcol).

1. **Perspective** —Normally there are three distinct entities in a communication process (the senders, the receivers, and the societies–the other observers). The first two entities are related to the private perspective (the individual agent perspective) while the last entity is

linked to a public perspective (the MAS perspective). Figure 2.9 shows that Arcol and KQML only concern the private perspective. In fact these languages are only concerned to the sender's perpective which goes in the opposite direction compared to the human discourse perspective (which treats the sender and receiver as equal entities). The ACL language must be normative (following some standards) ensuring that different agents implementation maintain the ACL's meaning. This feature is also important mainly for the understanding of distinct agents' implementation environments. Finally, in order to be testable the ACL's semantics must have a public perspective;

2. **Type of Meaning** — The meaning is the combination between the personal and conventional perspectives. Personal communication is more related to the interpretation of the communication acts of the sender and the receiver. Both Arco and KQML emphasize the personal meaning which can lead to problems. The conventional perspective is focused on usage conventions (a language can be seen as a system of conventions);

3. **Semantic versus Pragmatics** — As previously analyzed, communication can present five different characteristics (syntax, semantic, vocabulary, pragmatic focus, speech domain model). The meaning is the combination of semantic and pragmatic focus. Pragmatics included external considerations to the proper language as well as to the environment. As in showed in Figure 2.9 both Arcol and KQML emphasize pragmatics. In Arco it is assumed that an agent makes only sincere contributions as well as the other system's agents. Consequently it is not possible to use Arco in environments that this feature does not grant such as electronic business;

4. **Context** — Normally to understand a language it is necessary to analyze its context (the agentś physical or simulated environment). As illustrated in Figure 2.9 both Arcol and KQML presented fixed context mainly because both languages have many constraints and they are very inflexible. An example of that is the conclusion of a negotiation process: An agent A sends a message to an agent B with the confirmation of a business like "Ok, so the price will be 10 euros". Despite it beliefs in Agent B agrees. In Arcol the Agent A will be unable to do that communication. Because it would violate a language requirement that agent A believes Agent B does not believe the price is 10 euros;

5. **Coverage of Communication acts** — When a set of autonomous and heterogeneous agent exchange information, the meaning of this exchange is characterized by communicative acts. These acts could be divided in:

   (a) **Assertives** — which inform: *The shop is closed*;

   (b) **Directives** — which request or query: *Shut the door or Could you please help me with this task?*

   (c) **Commissives** — which promise something: *I will do this task*;

   (d) **Permissives** — which give permission for an act: *You may go to the shopping centre*;

(e) **Prohibitives** — which ban some actions: *You may not go to that place*;

(f) **Declaratives** — which cause events in themselves: *I name this box the Golden box*;

(g) **Expressives** — which express emotions and evaluations: *I wish to be the champion of this competition*.

As illustrated in Figure 2.9, all primitives in Arcol and KQML languages are assertives or directives (presenting a limited coverage). These limitations should constitute an important challenge for the agent's developers mainly because in some environments agents need them (all the acts) to enter into and manage more complex social relationships.

2.9.3.2 Agent Construction

Every ACL semantics include an agent model which can emphasize an individual agent mental state or the social communication aspects. Figure 2.9 shows how ACL, KQML and Arcol manipulate agent construction.

1. **Mental versus social agency** — Mental agency is related to an agent's mental state (typically reports its beliefs and intentions). On the other hand social agency assumes agents as social creatures that interact with one another. Once again and as illustrated in Figure 2.9 Arcol and KQML promote mental agency. Mental states can include: beliefs, goals, desires, intentions.

2. **Design autonomy** — The design autonomy allows to minimize the requirements of agent builders and because of that promoting heterogeneity and also allows the proliferation of a wide range of practical systems. Arcol and KQML (Figure 2.9) present low design autonomy (agent implementation uses mental concepts).

3. **Execution autonomy** — Execution autonomy is basically the freedom that an agent has in order to take its own actions. This execution is, in the ACL language, limited because it requires the agents to be sincere, cooperative, benevolent and so on. In this particular point KQML better preserves execution autonomy mainly because of its nature: this language was design for interoperation while the ACL language was designed as a proprietary language for a specific system.

2.9.3.3 Ontologies

As mentioned before, in the agents communication process the agents must share the same language and also the same vocabulary. However,not rarely, this situation is not verified. This problem could be solved with the use of a common ontology. An ontologies overview will made in next section.

2.10 MAS Learning

Learning techniques in MAS occupy an important role inside the DAI area. In this area the major researchers' motivations are:

1. Provide to the heterogeneous MAS the ability to learn and adjust its individual and collective behaviors in an automatic way;

2. Clearly understand the mechanism of natural learning groups of team processes.

Normally MAS are complex systems (with complex status and features). Because of that agents development with learning capacities occupy an important role in MAS.

2.10.1 Different Learning Types in MAS

The learning process in MAS will be influenced by information exchanges, share assumptions and similar viewpoints about the environment, social convention, among others [WG96]. This process could be divided in two groups [WG96]:

1. **Interactive Learning** — Situation in which agents in a collective perspective try to achieve its common goals using learning process;

2. **Individual Learning** — Situation where each agent tries to achieve its own learning goals. The learning process is affected by the other agents, its knowledge, beliefs, intention among others.

2.10.2 Credit or Blame Determination

One of the major issues in the multi agent learning process is to determine the credit or blame of a change (positive or negative) in the global behavior of the system. According to Weiss and Gerhard [WG96] this task can be divided into two sub problems:

1. Attribute credit or blame of the changes to each external agent action. However this attribution can be very complex due to the fact that this change can be influenced by various agents actions;

2. Attribute credit or blame of the changes to internal decisions. This problem is similar to the one that is presented in the individual learning and basically consists in identifying the decisions that led to the actions implementation that directly influence the system behaviors.

This conceptual decomposition (in subproblems) it not very clear in the majority of the problems [SW99]. Because of that most of the MAS approaches solely focus on one of them (considerably simplifying the other) [SW99].

2.10.3 Important Features in Learning

According to Sen and Weiss [SW99] learning MAS approaches can be classified using a set of auxiliary features:

1. **Decentralized Level** — Related to the distribution and parallelism learning process level;

2. **Learning Shape** — Regarding the availability of data used in the learning process. The learning methods can be divided classified in: online (when the learning process occurs during the normal agent operation) and off-line(when the learning process occurs after the normal agent operation analyzing the storage data);

3. **Learning Environment** — The environment can possess many different features. In consequence of that it is extremely important to analyze environment features such as dynamics, uncertainty, accessibility, determinism and complexity. The learning process will be more complex in dynamic, continuous,non-deterministic and inaccessible environments;

4. **Interaction specific properties** — These properties include: the interaction levels (observation, negotiation among others), interaction persistence (short or long term), interaction frequency, interaction variability etc. In consequence of that there are some situations where the learning process can be extremely fast and simple (e.g. observing another agent in a short period of time) or extremely complex and time consuming (e.g. a negotiation process involving a large number of agents);

5. **Involvement specific properties** — This involvement can be characterized by its relevance (essential or dispensable in the collective learning process) and by the role played by the agent (generalist or specialist in a certain activity);

6. **Goals specific properties** — These properties can include goals compatibility between various agents (complementary or conflicting) and the type of improvements regarding the learning process (individual or collective);

7. **Learning Method** — It is possible to highlight the following methods:

   (a) **Direct Learning** — Direct knowledge implementation without any complementary influence of the agent;

   (b) **Instruction or Advice Learning** — Fusion between the previously acquired agent capacities and its current knowledge;

   (c) **Learning by Example** — Extraction and reinforcement learning through positive and negative examples;

   (d) **Learning by Analogy** — Knowledge transformation used by solving previous problems into the new ones resolution;

   (e) **Learning by Discovery** —Collects new knowledge through observation experiences, hypothesis thesis and, experimental results analysis;

8. **Learning by Return Type** — The return learning indicates the improvements obtained up until a specific moment. It can be provided by the environment, calculated by the agent or provided by other agents. Normally the return learning methods can be divided in these classes:

   (a) **Supervised Learning** — The return specifies the desire agent (apprentice) activity and the final goal is to obtain a closest return value;

   (b) **Reinforcement Learning** — The return specifies the utility of the actual agent (apprentice) activity. The goal consists in maximizing that utility value;

   (c) **Unsupervised Learning** — In this type of learning there is no return. So the goal is to find useful actions and activities having as a base error trial system;

9. **Time to obtain return** — The way to obtain return could be:

   (a) **Immediately** — All the actions obtain immediate feedback allowing the quality of agents evaluation of their actions;

   (b) **On Request** — In each moment, the agent (apprentice) can request an return of an action or a set of actions;

   (c) **Delayed** — The return is sent for the agents but not in an immediate way. This delay is not controlled by the agents.

2.10.4 Learning Algorithms

Throughout the years many are the learning algorithms that have emerged. Examples of thoset are: Statistical Methods (Unsupervised Learning), Case Base Reasoning, Decision Trees, Reinforcement Learning, Inductive Logical Programming, Genetic Algorithms (and other evolutionary algorithms) and Neural Networks. In this section a brief introduction about the different machine learning algorithms used in this thesis, will be exposed.

2.10.4.1 Multivariate Adaptive Regression Splines (MARS)

Friedman's 1991 Multiple Adaptive Regression Splines (MARS) model [Fri91][SK92] employs recursive partitioning to locate product spline basis functions of an adjustable degree, rather than constants. This results in smooth adaptive function approximation as opposed to the crude steps or plateaus provided by regression trees. The method also takes into consideration splines involving interactions between previously selected variables so it can orient its basis functions other than on the original data axis. To aid interpretation, model terms are collected according to their inputs and their influence is reported in an ANOVA manner. The effects of individual variables and pairs of variables are collected together and graphically presented as function plots. MARS also employs cross-validation, prunes terms after over-growing, and can handle categorical variables.

MARS builds its models according to Equation 2.1 where the aim is to add together the weight of basis functions $Bi(\mathrm{x})$ (ci are constant terms).

$$\hat{f} = \sum_{i=1}^{k} c_i B_i(x) \tag{2.1}$$

The construction of the MARS models is divided into two distinct phases: the forward and the backward passes. In the forward pass the algorithm starts with a model which only consists of the intercept term. After that, and being a greedy algorithm, it will include in the model the basis functions pairs that give the maximum reduction for the sum-of-squares residual error. Each new basis function consists of a term, already in the model, multiplied by a new hinge function which is defined by a variable and a knot. This addition continues until the maximum number of terms is reached or if the change in the residual error is negligible. In order to generalize the model produced in this phase, the backward pass consists of pruning the model. It will remove terms one by one until the best submodel is reached and this is measured by the GCV (Generalized Cross Validation) variable.

The result of MARS is an interpretable equation. The features used in the equation are the ones that are selected to explain the target variable. Since the MARS algorithm uses the recursive partitioning algorithm [BFOS84], this approach to selecting features is similar to the one described in [Car93].

2.10.4.2 RreliefF

RELIEF [KR92] is considered one of the most successful algorithms due to its simplicity and effectiveness [Die97]. In the beginning, this algorithm was only used in classification problems but through the years its scope has been expanded to the regression problems (with its version RreliefF). Recently it has been proved that this algorithm solves a convex optimization problem aimed at maximizing a margin-based objective function [SL06, SW08].

In this research project the RreliefF algorithm proposed by [RSK03] is used. Generically, this algorithm calculates the weights for each one of the input variables. These weights can be used to obtain a feature weighted distance, guaranteeing that the similarity is differently measured weighing the attributes, according to their relevance to the output variable.

Doing a more deep analysis of the method proposed by Robnik-Sikonja [RSK03], at the beginning of the calculation process it is important to define three variables (Algorithm 1): V is the number of input variables of the data set, K - represents the number of K-nearest neighbors and T - represents the number of iterations.

Following the advices of Robnik-Sikonja (to decrease the computation costs and to increase the stability of the weight estimations) in this research project the T and K values were defined in 50 and 10 respectively . In order to calculate the distance between the variables, the researchers have used the equation 2.2, where *teq* and *tdiff* are 5% and 10% of the length of the input variable's value interval (as suggested by the authors), and d represents the absolute difference of the input variable A for the two examples, I1 and I2.

Algorithm 1 The RreliefF algorithm

Require: $R_i : i = 1,2,...M, the \quad M \quad input \quad vectors \quad from \quad the \quad training \quad set$
Require: $\tau, the \quad correspond \quad M \quad target \quad values$
Require: $t, the \quad number \quad of \quad iterations$
Require: $k, the \quad number \quad of \quad nearest \quad examples$
  $setall N_{dC}, N_{dA}[A], N_{dC\&dA}[A], W[A] to 0, where A = 1,2,...,a$
  for $i := 1$ to t **do**
    $randomly \quad select \quad example \quad R_i$
    $select \quad k \quad examples \quad I_j \quad nearest \quad to \quad R_i$
    for $j = 1$ to k **do**
      $N_{dC} := N_{dC} + diff(\tau(.), R_i, I_j) \times d(i,j)$
      for $A = 1$ to V **do**
        $N_{dA}[A] := N_{dA}[A] + diff(A, R_i, I_j) \times d(i,j)$
        $N_{dC\&dA}[A] := N_{dC\&dA}[A] + diff(\tau(.), R_i, I_j) \times diff(A, R_i, I_j \times d(i,j)$
      end for
    end for
  end for
  for $A = 1$ to V **do**
    $W[A] := N_{dC\&dA}[A]/N_{dC} - (N_{dA}[A] - N_{dC\&dA}[A]) \times (t - N_{dC})$
  end for
  return W

$$diff(A, I1, I2) = \begin{cases} 0 & : d \le teq \\ 1 & : d > tdiff \\ \frac{d-teq}{tdiff-teq} & : teq \le tdiff \end{cases} \tag{2.2}$$

One final note is concerning with the feature selection. In this research, the filter criteria used was suggested by [Hal00] which selected the features that presents a RreliefF weight larger than 0.01.

2.10.4.3 K-Means Algorithm

Data clustering or Q-analysis or Typology or Clumping or even Taxonomy (depending on the applied field [JD88]) appeared for the first time in 1954 [Jai10]. One of the possible definitions of cluster analysis can be found at Webster [Jai10] "a statistical classification technique for discovering whether the individuals of a population fall into different groups by making quantitative comparisons of multiple characteristics".

Clustering algorithms can be divided in two groups: Hierarchical and Partitional [Jai10]. The Hierarchical algorithm works in two ways: it starts by agglomerating all of the data in the same cluster and recursively dividing the cluster into small clusters (divisive mode) or it starts to put each data point in respective clusters and merges the most similar clusters in a hierarchical cluster (agglomerative mode). On the other hand, the Partitional algorithm does not impose a hierarchical structure and finds all the clusters simultaneously through the partitioning of the data.

Normally, due to the nature of the available data, the Partitional algorithms were preferentially selected. One of the most famous/adopted Cluster Partitional, due to its simplicity, efficiency and empirical success, is the KMeans algorithm. Over the years it has been used in several areas. The first work dates back to the1950s [Ste56]. It does not use a target field and this algorithm tries to uncover patterns in the set of input fields. It tries to group data in different sets (designated as clusters) according to their similarities.

2.10.4.4 Bagging

Created by Breiman, Bagging ("**b**ootstrap **agg**regat**ing**") votes classifiers generated by different samples (replicates). A bootstrap sample is generated from instances from a training set of size m, with replacement. From n Boostrap samples [ET94], k_1, k_2,...,k_n n classifiers C_i are built, each one from a different k_i. A final classifier C_b is built from C_1, C_2,..., C_n where the result is the most predicted class by its base-classiffiers (majority voting) with ties broken arbitrarily (for more details see [Bre96] [Bre98]).

2.10.4.5 Random Forest

Created by Breiman in 2001 [Bre01] Random Forest instantly became a commonly used method, mainly due to its simplicity (in terms of training and tuning) and performance [TTF09]. Similar to the previously analyzed algorithm, Random Forest constructs correlated trees. However, in this case, for each tree node v variables are randomly selected (considering $v <<$ the input variables) and the best split in these v variables is used to split the node. At the end, the "forest" picks the most voted class, over all the trees in the forest (for more details see [Bre01]).

2.10.4.6 Support Vector Machines (SVM)

Based on the concept of decision planes that define decision boundaries, SVMs were developed by Vapnik [Vap99], for binary classification. Basically, this technique tries to find the optimal separating hyperplane between two classes by maximizing the margin between the classes's closest points (Figure 2.10). The points lying on the boundaries are called support vectors and the middle of the margin is our optimal separating hyperplane. To construct an optimal hyperplane, SVM employs an iterative training algorithm which is used to minimize an error function. A more complete overview can be found in [Vap99][BGV92].

2.11 MAS coordination

Malone and Crowston [MC91] defined MAS coordination as an integration and harmonious adjustment of individual effort towards achieving a common goal. In what concerns to MAS implementation two major issues in agents coordination perspective can be emphasized [Woo02]:

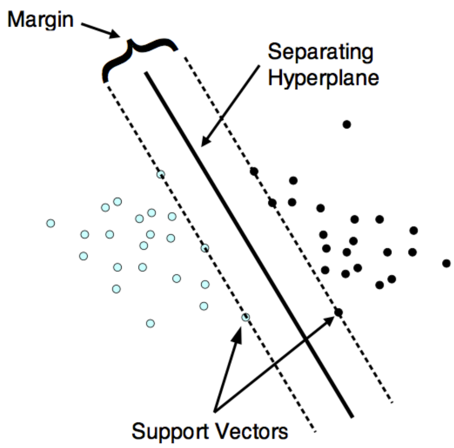


Figure 2.10: Classification (linear separable case)

1. Normally in MAS, the agents are implemented by different developers with distinct goals. In this way the agents will not share common goals and because of that will need to exist a negotiation process to obtain a good coordination process;

2. Due to the autonomous agent characteristics, they can take their decisions in a real time frame and because of that they also need to dynamically coordinate their activities with the other agents (which does not constitute an easy task);

2.11.1 The need to coordinate agents

Sichman and Demazeau [SD95] defined four basic relations of dependency between agents:

1. **Independence** — There are no agent dependencies;

2. **Unilateral** — An agent depends of another agent but there is no inverse relationship;

3. **Mutual** — Both agents depend on each other to fulfill their own goal;

4. **Mutual Dependence** — An agent (first agent) depends on another agent (second agent) to achieve a specific goal. Then the second agent depends on the first one to achieves a specific goal (which is not necessarily the same).

There is a substantial extensive research work that identifies different types of dependencies. For a complete overview please consult Malone and Crowston [MC94]. Other researchers also defend four reasons to exist coordination: to prevent chaos, to meet global constraints, to work together to solve problems and to prevent conflicts [GHN+97] .

2.11.2 Cooperative versus Competitive MAS

In the construction of a MAS two approaches can be detected: cooperative or competitive (self-interested) approaches. In the first approach the goal of the agent behavior consists in increasing the global system utility and not the personal one. Also improving the global system performance constitutes a important role. The second approach is the opposite compared to the first one, where the agent goal is to achieve individual goals (more realistic approach in scenarios like internet and e-commerce).

2.12 Conclusions

The agent research area suffered many transformations in the past years. Realities constituted by a set of autonomous communities (multi agent systems) capable to work together and sometimes compete to achieved individual goals were without any doubt an interesting area for research and industry communities. These interactions presuppose that each agent knows the other system agents. The agents coordination is a complex problem and sometimes it can lead to disorganized environments and even a chaotic situation. To avoid these situations, it is necessary that all agents speak the same language.

In the next two chapters a brief summarization of the ontologies development process in terms of language used as well as tools and methodologies will be exposed. In chapter 4 a brief summary of all problems that are inherent to all of the RoboCup challenges will also be executed. communication between other agents or humans.

This chapter does not have the intention to do an exhaustive autonomous agent and MAS literature review but only to present the basic concepts presented in these two areas. For a deeper review consult the following research works [Wei99] [Woo02] [Rei03].

Chapter 3

Ontologies

Over the years and, similar to the agents concept, many were the authors that tried to create a definition for the ontology term. One of the most accepted ontologies definitions was presented by Noy and McGuiness [NM01]. For these researchers an ontology is an explicit and formal concept definition of a discourse domain where the proprieties of each concept describe the characteristics, the attributes concepts and the attributes constraints.

At the beginning the ontologies principles and methods were developed in the AI area allowing the reuse and sharing knowledge. Since the 90's the use of ontologies is generalized through other research areas such as: knowledge representation, natural language processing and knowledge engineering. Nowadays, other research areas demonstrate interest for this research topic such as: information retrieval, knowledge management and web semantic.

The main reason for the use of ontologies is the promise of a shared and common understanding about a specific domain that can be communicated between people and application systems [DSW+00]. Fensel [Fen01] claims that ontologies can be used to represent explicitly the semantics of structured and semi-structured information enabling sophisticated automatic support for acquiring, maintaining and accessing information. In consequence, ontologies can constitute a key tool in the agents interaction, in a complex distributed system environment.

In this chapter a brief summary of the fundamental concepts concerning ontologies will be presented including ontologies general definition, main objectives, components and types. Then several ontologies languages will be presented and divided in two distinct groups: Classical and Markup languages. After that, several ontologies tools are presented which include tools for commercial and research purposes. Finally, some methodologies for ontologies development will be exposed.

3.1 Fundamental Concepts about Ontologies

In this section the fundamental characteristics of ontologies domains are highlighted such as their definitions, objects, types and components. Also a comparison between the ontology term and other terms presented in literature is presented. At the end it is also stressed the ontologies role in the semantic web.

3.1.1 Ontologies Definition

Throughout the years, many were the ontologies definitions that emerged. In what concerns the philosophy area, the ontology term appeared in the XVIII century where *Onto* means being and *Logos* means treaty [Pun07]. At the end of the XX century ontologies appeared more related to the computational science area.

Doing a more deeper literature review about the evolution of the ontology definition:

1. Neches et. al. [NFF+91] defined an ontology as basic terms definitions and vocabulary relations in a specific area. Also these authors claimed that ontologies provide the definition of rules, relating terms for vocabulary extensions;

2. Gruber [Gru93] claimed that an ontology is an explicit specification of a conceptualization. This conceptualization is basically the domain idea where an individual or a set of individuals are inserted;

3. Guarino and Giaretta [GG95] defended that an ontology can be a philosophical discipline, an informal conceptual system, a formal semantic description, a specification of a conceptualization, a vocabulary used in logical theory;

4. Grüninger [GF95] claimed that an ontology is a formal description of entities, properties, constraints and behaviors;

5. Borst [Bor97] uses as a base the definition presented by Gruber [Gru93]. This author defends that ontologies are a formal specification of a shared conceptualization. In this context the word shared means that an ontology must reflect a conceptual knowledge accepted by a group.

6. Studer et. al. [SBF98] merged previous ontologies definitions presented by Gruber [Gru93] and Borst [Bor97]. For this author an ontology is a formal and explicit specification of a shared conceptualization. Also for this author a conceptualization is an abstract model of a world phenomenon (after identifying the concepts of that phenomenon);

7. Noy and McGuiness [NM01] presented one of the most consensual ontologies definition. For these researchers an ontology is an explicit and formal concept definition of a discourse domain where the proprieties of each concept describe the characteristics, the attributes concepts and the attributes constraints.

Doing a generic comparison of the different ontologies definitions previously presented, it is possible to conclud that the aim of the ontologies is to capture and to create knowledge models that can be reused and shared between applications or group of peoples.

3.1.2 Ontologies Main Objectives

Throughout the years ontologies have been used in distinct research areas such as knowledge representation or software engineering. Similar to the ontologies definition there are also different proposals for the use of ontologies. For Gruber [Gru93] ontologies can be developed to:

1. Enable a machine to use knowledge in a specific application;
2. Enable knowledge sharing between machines;
3. Help humans understand more about a specific domain area;
4. Help human researchers reach a consensus on a certain area of knowledge.

On the other hand Noy and McGuiness [NM01] proposed the following use for the ontologies:

1. Share a common comprehension of an information structure between humans and softwares;
2. To enable the reuse of knowledge domains;
3. To elaborate assumptions of an explicit domain;
4. To separate the knowledge domain of an operational knowledge;
5. To analyze the domain knowledge.

Finally Grüninger and Lee [GL02] tried to summarize the use of ontologies in three groups:

1. For communication: Between computational implemented systems, between humans and between humans and computational systems;
2. For knowledge (and organization) reuse: For structured libraries or plans repositories and domain information;
3. For computational deduction: For internal plans representation, for internal structure analysis, inputs and outputs of conceptual implemented systems.

3.1.3 Ontologies Componentes

The ontologies components vary according to the used ontologies language however, all of them share some common elements. Regarding the formalism logic of the modulation techniques, ontologies components can be divided in two groups: the first one is based on frames and first order logic. This group identifies five components: classes, relations, functions, formal axioms and instances. The second group is based on Descriptive Logic (DL).

Although the two perspectives represent the same knowledge in different levels of formality and granularity both of them have the same basic components:

1. **Classes** to model tasks or domain concepts. These classes can be abstract or concrete (may have direct instances) and normally are organized as taxonomies;

2. **Attributes** to represent concepts characteristics. These attributes can also be named as slots or proprieties;

3. **Relations** to model association types between concepts. Binary relations are sometimes used to express concept attributes however, the relations' scope are different in comparison to the attributes scope;

4. **Intances** to represent specific elements. Normally these elements are specific entities of a certain class. New instances can be created and values can be assigned to attributes and their relationships;

5. **Functions** to represent special relations;

6. **Axioms** are used to validate the ontology's consistency or knowledge stored consistency.

3.1.4 Ontologies Types

To characterize an ontology there are a huge number of parameters that can be used. This characterization can be done through the conceptualization subject, the information that the ontology needs to expresses the richness of its internal structure and the dependency level of a specific task or point of view.

The majority of the proposals for defining ontologies' types are related to its functions, the level of its vocabulary formalism, its application and also its structure and concepts content. For Uschold and Grüninger [UG96] ontologies differ in this formalism degree in which are expressed the terms and their different meanings. These authors classified ontologies' types in four groups:

1. **Highly Informal** when the ontologies are expressed in natural languages. This type of ontology maybe ambiguous due to the underlying ambiguity of the natural language;

2. **Semi-Informal** when the ontology is expressed in a restrict way and structured in a natural language. Comparing this type of ontology with the previous type analyzed; this new type provides greater clarity on the concepts and reduces the ambiguity;

3. **Semi-Formal** when the ontology is expressed through artificial languages that are formaly defined;

4. **Strictly-Formal** when the ontologies are defined through formal semantics, theorems and confirmed proprieties.

On the other hand Van Heijst et. al. [VHSW97] classified ontologies following two perspectives: regarding the type of conceptual framework and regarding the conceptualization subject. For the first perspective the author distinguished three categories:

1. **Terminological Ontologies** that are constituted by lexicons that specify the terminology that is used to represent knowledge in the discourse field;

2. **Information Ontologies** that defined a DB structure;

3. **Knowledge Ontologies Modeling** that specify the knowledge conceptualization.

In what concerns to conceptualization subject, these authors distinguish four categories (according to the type of knowledge that is modeled):

1. **Application Ontologies** that define the knowledge model of a certain application. This kind of ontologies is difficult to reuse;

2. **Domain Ontologies** that define concepts and relations in a specific domain. These ontologies are reusable in specific domains such as science, engineering, medicine among others;

3. **Generic Ontologies** that define independent concepts in a general domain. These ontologies define concepts like events, time, space, cause, behavior, function, among others.

4. **Ontologies Representation** capture the representation principles used to formalize knowledge on a knowledge paradigm. These ontologies provide formal definitions of representation principles used in frame languages and allow the construction of other ontologies through the convention meaning based on phrases.

Guarino [Gua98] classified the ontologies through the generality level (Figure 3.1 illustrates these relationships between the ontologies types):

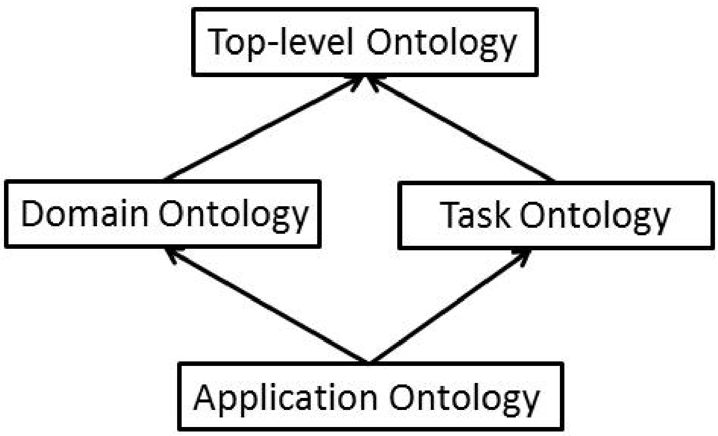


Figure 3.1: Relations between the Ontologies Types regarding their generality (adapted from [Gua98])

1. **Top Level Ontologies** that describe general concepts and, in consequence, they are independent of the problem or domain. However, this type of ontologies is more applicable in large communities of users;

2. **Domain Ontologies** provide a vocabulary about a generic domain. These ontologies are reusable in a given domain;

3. **Task Ontologies** describe the vocabulary related to a task or generic activity by specializing the terms introduced in Top Level Ontologies;

4. **Application Ontologies** describe concepts related to a certain domain or task, which are often specialization of both related ontologies.

Finally Lassila and McGuiness [LM01] classified the ontologies based on the richness of their internal structure. The main categories are:

1. **Controlled Vocabulary** is the most simplest ontologies notion. Normally this vocabulary is constituted by a finite list of terms. A typical example is a catalogue;

2. **Glossary** is the list of words or meanings that normally are expressed by phrases in natural languages;

3. **Thesaurus** provide additional semantic between terms however these thesaurus cannot provide a hierarchical explicit structure;

4. **Informal *is-a* hierarchies** that contains a generic notion of generalization and provide specialization but not as a strict hierarchy subclass;

5. **Formal *is-a* hierarchies** that organize the concepts according to a strict hierarchy subclass;

6. **Frames** that include classes and their proprieties that can be inherited from lower level classes of a *is-a* taxonomy;

7. **Value Restriction** that allows the restriction applications about the proprieties' associated values;

8. **Generic Logical Condition** that are normally written in very expressive representation ontologies languages allowing the specification of first order logic conditions on concepts and their properties.

Table 3.1 summarizes the differences presented in the different approaches. In conclusion and, in spite of ontologies not presenting the same structure, there are some common characteristics.

3.1.5 Ontologies versus Other Similar Notions

Sometimes the ontologies terms are used as a synonym of other terms such as controlled vocabulary, taxonomy, thesaurus or knowledge bases. This process is the result of some functions overlap that are common in these concepts. Figure 3.2 illustrates the differences between these concepts.

The differences between taxonomy and controlled vocabulary consists in the terms classification. The taxonomy uses for that a hierarchy genealogical tree. Sometimes Thesaurus are an

Perspective	Classification
Formality	Highly Informal Ontologies
	Semi-Informal Ontologies
	Semi-Formal Ontologies
	Strictly Formal Ontologies
Graularity	Terminological Ontologies
	Information Ontologies
	Knowledge Modeling Ontologies
Subject	Application Ontologies
	Domain Ontologies
	Generic Ontologies
	Representation Ontologies
Generality	Top-Level Ontologies
	Domain Ontologies
	Task Ontologies
	Application Ontologies
Richness of Internal Structure	Controled Vocabulary
	Glossary
	Thesaurus
	Informal is-a Hierarchical
	Forma is-a Hierarchical
	Frames
	Value Restriction
	Generic Logical Condition

Table 3.1: Comparison between the different types of ontology

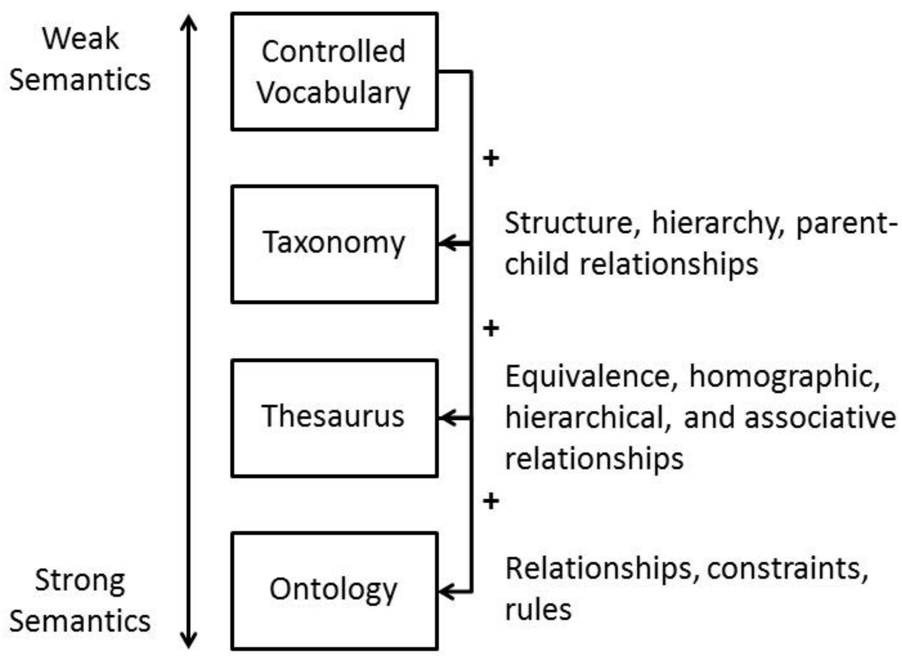


Figure 3.2: Semantic Spectrum (adapted from [CFLGP07])

extension of a taxonomy and allow to elaborate phrases as well as relationships between terms. A thesaurus can be converted in a taxonomy or in a controlled vocabulary however, with this conversion it loses its expressiveness and its semantic level. The ontologies are similar to the taxonomies, however the first ones present a richer relationship between attributes and terms in a semantic point of view. A common functionality between ontologies and thesaurus is that both of them contain a set of special terms (providing clear understanding of their meaning) related to a specific domain. A thesaurus deals with terms while an ontology deals with concepts [NF03]. In taxonomies there is only a entity relation (more precisely an *is-a* relation). For this reason, if an ontology has the aim to represent concepts using only an *is-a* relation it will be similar to a taxonomy.

In addition the ontologies' community makes a distinction between ontologies that present many taxonomies similarities and ontologies that model a domain (that can be divided in two groups):

1. **Light Ontologies** include concepts, relations between concepts and proprieties that describe those concepts;

2. **Heavy Ontologies** adding to the previous definition of axioms and constraints.

Finally the ontology term can be confused with the DB term. For the AI community a DB is constituted by two parts: a terminological box (T-Box) and an assertive Box (A-Box). A T-Box is constituted by a set of concepts and their definitions. Normally this kind of box includes a taxonomy of terms interrelated and an instance of concepts. Both ontologies and DB represent knowledge however, a DB provides additional instances for which knowledge is applied and inferred.

3.1.6 Ontologies and Semantic Web

The semantic web is an effective infrastructure that improves the knowledge visibility of the web. For many authors the semantic web is the next generation of the world wide web (WWW). For Berners-Lee et.al. [BLHL01] semantic web is the actual web extension and where is given a clear information understanding allowing their interpretation to a human or a machine. For the same authors the biggest new web challenge is to structure all available data in order to be processed not only by humans but also by machines. Table 3.2 illustrates the evolution the WWW. As can easily be seen, the semantic web is the next step of the evolution of the WWW. The web' semantic main goal is to developed an architecture that reinforces the web contents with formal semantics allowing the creation of data, information and knowledge processable by machines. Berners-Lee et.al. [BLHL01] believe that ontologies applications will increase the semantic web developments. For the same authors the semantic web presents different layers. The ontology vocabulary layer is the main research topic for web semantic (Figure 3.3). Technologies like Extensible Markup Language (XML), Resource Description Framework (RDF), RDF(S) represent the base for the ontology language. The ontologies are also integrated in the logic layer because the majority of the ontologies need logic axioms.

In conclusion ontologies are very important in the semantic web developments and because of that semantic web promotes that development.

	Static	**Dynamic**	**Syntax**	**Semantic**
Encoding	HTML	+ RDBMS	+ XML	+ RDF/OWL
Creation	Manually	Generated by server-side applications	Generated by applications based on schema	Generated by applications based on models
Users	Humans	Humans	Humans and applications	Human and applications
Paradigm	Browse	Create/Query/ Update	Integrate	Interoperate
Applications	Browsers	Browsers	Process Integration, EAI, BPMS, Workflows	Intelligent agents, Semantic engines

1995 → 2000 → 2005

Table 3.2: Evolution of the World Wide Web (adapted from [CFLGP07])

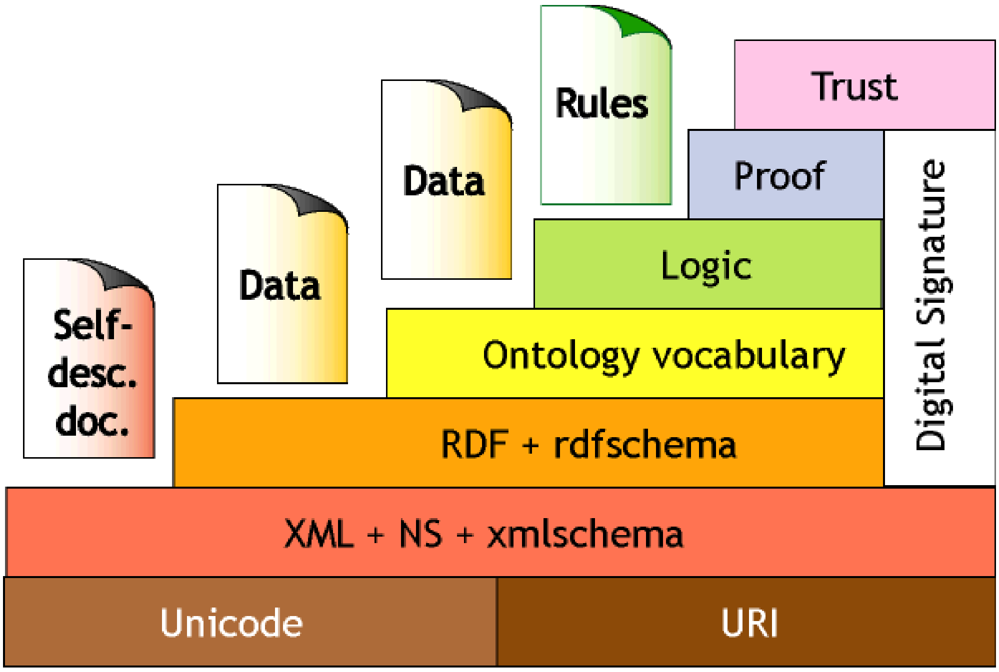


Figure 3.3: Semantic Web Architecture [BL00]

3.2 Ontologies Languages

In this section a general perspective about the languages used in the ontologies representation is performed. First a historical perspective is performed and after that the most significant language is presented with a special emphasis to the OWL(Web Ontology Language) language.

3.2.1 Evolution of the Ontologies Language Representation

These kinds of languages are used to developed ontologies, allow the knowledge codification of specific domains and frequently include reasoning rules that support knowledge processing.

The first language that appeared was derived from the knowledge representation such as descriptive logic, first order logic and frames. With the appearance of the WWW, the new languages became the Markup type, mixed with some language formalism derived from knowledge representation.

The research community divided these languages in two groups:

1. **Classical Languages** which include descriptive logic languages, frames and first order logic. Some examples of those languages are: LOOM, Cycl, Ontolingua, OCML, F-Logic, Kif, OKBC;

2. **Markup Languages** which include web standard languages. Some examples of those languages are: SHOE, XOL, RDF, OIL, DAML+OIL,OWL.

3.2.2 Classical Languages

As mentioned before, the classical languages include the unused languages and also languages maintained exclusively be research groups. The most relevant languages within this group are:

1. **Cycl** (Cyc Language) [LG90] is a formal language based on frames and first order logic. The Cycl vocabulary is constituted by terms that can be divided as constants, non atomic terms and variables. Also these terms are combined with significant language expressions allowing the formulation of assertions in the knowledge base Cyc2. Cyc is a large knowledge DB based on common sense and created by Microelectronics and computer technology corporation;

2. **F-Logic** [KLW95] language was developed by the computational science of the New York State University and basically integrates features derived from object-oriented programming and from knowledge representation based on frames and first order logic;

3. **Kif**(Knowledge Interchange Format) [GF92] is designed to knowledge change between systems (created by different developers, in different times and using different programming languages). Also, Kif is a first order logic language with a simple syntax and some extensions that support the reasoning about relations. This language is classified as a low level ontologies representation language.

4. **Ontolingua** [FFR96] is based on the previously analyzed language and it is used in the ontologies construction through Ontolingua Server. This server provides support for the ontological construction modules that can be grouped, extended and redefined by other ontologies;

5. **OCML** (Operational Conceptual Modeling Language) [DMG99] is based on frames with a similar syntax to the Lisp family. In consequence, OCML provides primitives for class definition, relations. functions, axioms and instances. It is also possible to define rules and attach procedures.

6. **LOOM** [Mac91] is used in the development of intelligent environments. LOOM is composed by a knowledge representation system used to provide deductive support to the declarative language part. This declarative knowledge consists in defining rules and facts.

7. **OKBC** (Open Knowledge Base Connectivity Protocol for Knowledge Base Interoperation) [CFF+98] provides an uniform model for a knowledge system representation based on a common conceptualization of multiple components. This language is defined by an independent programming language and this protocol supports network connection.

Figure 3.4 represents the relationship between classical languages (previously analyzed). Doing a simple analysis it is important to note that Ontolingua language is based on KIF language

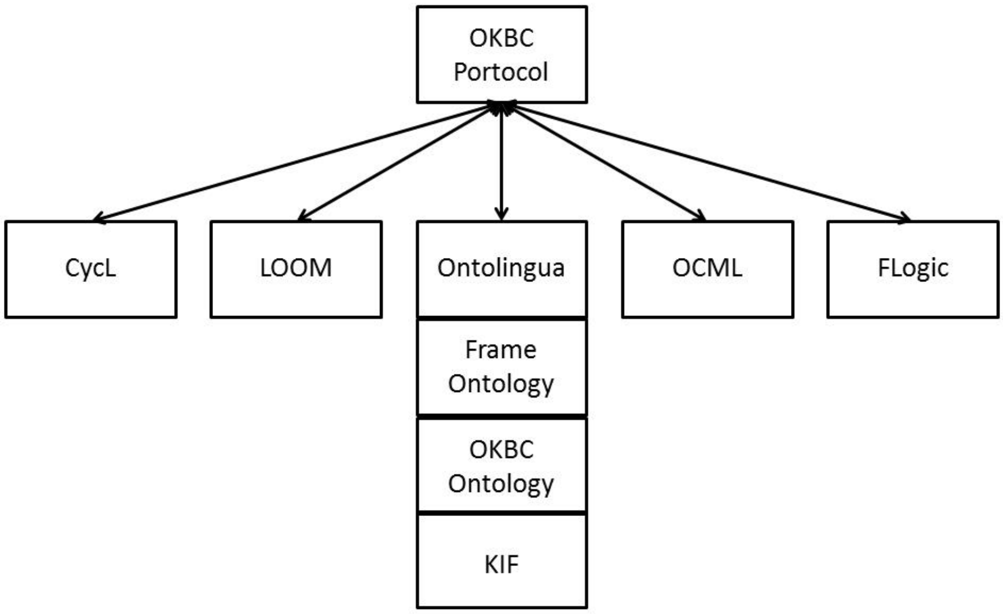


Figure 3.4: Classical languages for ontologies representation (adapted from [CFLGP07])

integrated with frame and OKBC. Also this protocol was developed to allow the interoperability between the Cycl, LOOM and Ontolingua languages.

3.2.3 Markup Languages

Since the 90's researchers focused their efforts in the languages development that can be used in the WWW. The result was the creation of various languages based on HyperText Markup Language (HTML) or XML and also based on frames. A list of those languages is presented:

1. **SHOE** (Simple HTML Ontology Extension) [LH00] is an extension of the HTML languages which aims to incorporate semantic knowledge in processable documents and provides tags to represent ontologies;

2. **XML** (eXtensible Markup Language) [BPSM+04] is a subtype of the SGML (Standard Generalized Markup Language) language.The aim of the XML language is to simplify the implementation and interoperability between HTML and SGML languages;

3. **XOL** (Ontology Exchange Language) [KCT99] was created to facilitate the creation of shared ontologies. Normally XOL is used as intermediate language for ontologies to transfer between DB systems, ontologies development tools or application programs.

4. **RDF** (Resource Description Framework) [LS98] was developed by the W3C(world wide web consortium) for processing metadata. RDF offers interoperability between applications and emphasizes facilities to enable automated processing of web resources.

5. **RDF Schema** [BG04] was developed by the W3C and it was specified through the basic information RDF model;

6. **RDF(s)** [BG04] is the combination between RDF+RDF Schema. The RDF(s) is highly expressive and allows the representation of concepts, taxonomy of concepts and binary relations;

7. **OIL**(Ontology Inference Layer) [FHH+00] is compatible with the RDF Schema and includes semantic for the description of the meaning terms;

8. **DAML**(Darpa Agent Markup Language) is an extension of XML and RDF languages;

9. **DAML+OIL** [HPSH02] is an updated version of the DAML language and provides a huge constructor set for ontologies creation and the generated information can be read and interpreted by computers.

10. **OWL** (Web Ontology language) [MvH04] is a W3C recommendation used when the encapsulated documents information need to be automatically processed by applications and humans. This language can also be used to represent the meaning of terms and their relationships. This language is more expressive comparing to XML, RDF and RDF(s) and some researchers defend that OWL is a language review of DAML+OIL language.

Figure 3.5 presents a schema of the markup languages analyzed for ontologies representation. It is important to note that SHOE is started to be a HTML extension and after some time it is adopted as a XML extension (XHTML=SHOE+XOL). The XOL language is a XML extension and finally the OIL, DAML+OIL and OWL languages are developments of derived RDF languages.

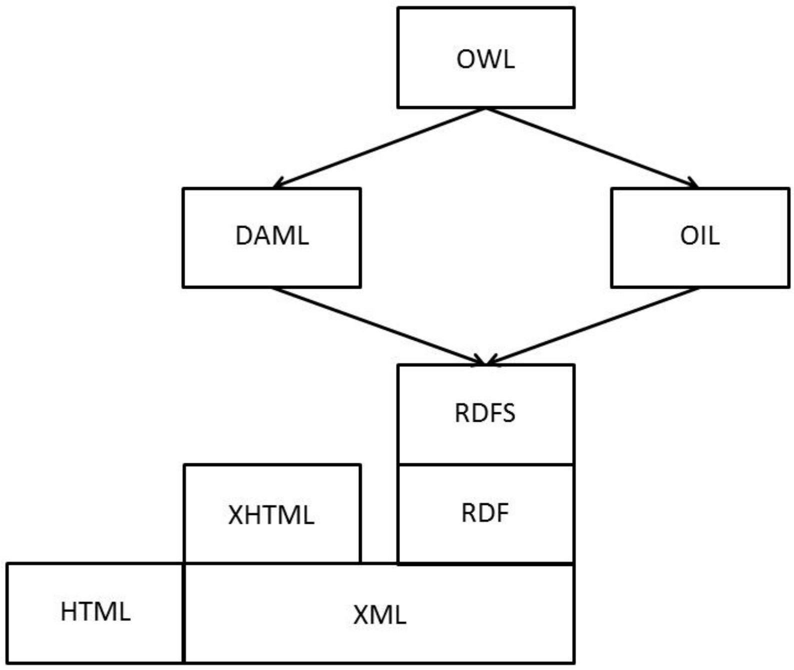


Figure 3.5: Markup languages for ontologies representation (adapted from [Cor05])

3.2.4 Languages Comparison

The XML provides a syntax for structured documents but does not define semantic constraints about the document meaning. The RDF language is a data model for objects representation and their relationships provide simple semantics that can be represented in XML syntax. The RDF(s) language is a vocabulary for proprieties and classes' resources descriptions of the RDF language with semantics for hierarchical generalization of their proprieties and classes. The OWL is an evolution of the RDF providing more vocabulary to describe proprieties and classes including their relations, proprieties characteristics and enumerate classes.

While the Ontolingua and Shoe allow the creation of multiple binary relations, in the other languages these relations are represented by decomposition. Functions can be defined in Ontolingua, LOOM, OCL, F-Logic, Kif, OIL, DAML+OIL and OWL languages. Formal axioms can be defined by Ontolingua, LOOM, OCML and F-Logic languages. Rules can be defined in LOOM, OCML and SHOE and finally procedures can be defined in Cycl, Ontolingua, LOOM, OCML and Kif.

3.2.5 OWL Language and its extensions

In this section the OWL will be subject to a deeper analysis (since later it will be used in an ontology implementation). This language was originally design for the applications that need to process the information content. This language provides a greater interoperability of web content compared to other languages such as XML, RDF and RDF(s) through the addition of formal semantics. Also OWL supports more powerful reasoning techniques. The document of this language became an formal W3C recommendation in February 2004. A new recommendation of it was launched in June 2009. This new language [MGH+09] provides new OWL extensions with the addition of a set of new user features.

3.2.5.1 OWL SubLanguage

The OWL provides three incremental expressive sublanguages: OWL Lite, OWL DL and OWL Full. The OWL Lite sublanguages support the user types that at the beginning of the process only need features related to a classify hierarchy and also simple constraints. The OWL DL (Descriptive Logic) sublanguage includes all language constructors but with some constraints. Finally the OWL Full is used to users that want maximum expressiveness and syntactic freedom of the RDF without any computational guarantees. Table 3.3 illustrates a comparison between these three sublanguages. In a brief conclusion it is easy to note that despite fact that OWL Lite

	OWL Lite	**OWL DL**	**OWL Full**
Usage	Classification hierarchy, simple constraints	Maximun expressiveness, high reasoning ability	Maximun expressiveness, free syntax, unwarranted reasoning
Representation Language	Fundamental part, subset of OWL DL	All, but used under certain constraints, based on description logic	All, us free without constraints, extension of RDF
Reasoning	High efficiency	High efficiency	No warrantee

Table 3.3: Comparison between the three OWL sublanguages

is a fundamental part of the OWL language it presents some features limitation compared to the other two languages.

The most important OWL concepts are classes, proprieties, instances of classes and their relations. An explanation of those concepts is done in the next section.

3.2.5.2 OWL Class

The OWL classes provide an abstract mechanism of a group of resources that have similar characteristics. These classes can be used to represent different concepts and their ontology hierarchies. In OWL, classes are described through a description or an axiom. A class description can contain:

1. A class identifier;
2. An exhaustive class enumeration;
3. A property restriction;

4. An interception of two or more class instances;

5. An union of two or more classes description;

6. A complement of a class description.

Table 3.4 presents a categorization between the language constructors with different class descriptions.

<table>
<tr><th colspan="2">Description Type</th><th>Language Constructs</th></tr>
<tr><td colspan="2" rowspan="2">Class Identifier</td><td>owl:class</td></tr>
<tr><td>rdfs:subClassOf</td></tr>
<tr><td colspan="2">Enumeration</td><td>owl:oneOf</td></tr>
<tr><td rowspan="6">Property restrictions</td><td rowspan="3">Value constraints</td><td>owl:allValuesFrom</td></tr>
<tr><td>owl:someValuesFrom</td></tr>
<tr><td>owl:hasValue</td></tr>
<tr><td rowspan="3">Cardinality constraints</td><td>owl:maxCardinality</td></tr>
<tr><td>owl:minCardinality</td></tr>
<tr><td>owl:Cardinality</td></tr>
<tr><td colspan="2" rowspan="3">Intersection, union, complement</td><td>owl:intersectionOf</td></tr>
<tr><td>owl:unionOf</td></tr>
<tr><td>owl:complementOf</td></tr>
</table>

Table 3.4: OWL classes descriptions and their constructors

3.2.5.3 OWL Propreties

A propriety is a bynary relation. There are two different proprieties categories in OWL language:

1. **Object Proprieties** that define relations between instances of two classes;

2. **Data Type Proprieties** that define relations between class instances and data types.

A property axiom defines the characteristics of a property. In Table 3.5 it is characterized the proprieties axiom constructors supported by the OWL language.

3.2.5.4 SWRL(Semantic Web Rule Language)

SWRL [HPSB+04] is an expressive rules language based on OWL language. This language combines OWL and RuleML (Rule Markup Language) providing the user with the possibility to create rules that can be expressed in the set of OWL axioms. The SWRL also provides a built-in function library allowing comparison, mathematical operation and string manipulation in the formulating rules.

3.2.5.5 SQWRL(Semantic Query-enhanced Web Rule Language)

SQWRL is a query language based on SWRL that can be used to question the OWL ontologies. Similar to SQL (Structured Query Language) this language also provides many operations regarding the knowledge extraction.

Property Characteristics	Language constructs
RDF Schema constructs	rdfs:subPropertyOf
	rdfs:domain
	rdfs:range
Relations to other properties	owl:equivalentProperty
	owl:inverseOf
Global cardinality constraints	owl:FunctionalProperty
	owl:InverseFunctionalProperty
Logical property characteristics	owl:SymmetricProperty
	owl:TransitiveProperty

Table 3.5: OWL proprieties characteristics and their constructors

3.3 Ontology Tools

Nowadays, there is a huge number of tools for ontologies development, domain modulation, knowledge systems construction, ontologies visualization, projects management or others modulation tasks.

The aim of this section is to present an overview about the ontologies tools existent on the market.

3.3.1 Features of Ontologies Tools

Using the ontologies features as a filter the Onto Web Consortium (OWC) [Con02] divided the ontologies tools in five groups:

1. **Development Ontologies Tools**: This group includes the tools and the environments that are capable of constructing an ontology from scratch or reuse an existent one. Beyond the conventional features like edition and navigation, these tools include some other features such as documentation, graphic visualization, libraries and inference engines attached;

2. **Fusion and Integration Ontologies Tools**: These tools have emerged to solve an integration problem involving many ontologies in the same domain. Normally this problem arises when there is a merge of two companies or when is necessary to improve the quality of the existent ontologies;

3. **Evaluation Ontologies Tools**: These tools ensure the quality in the used ontologies and the respective technology. This insurance is very important to avoid integration problems.

4. **Annotation Tools base on Ontologies**: The aim of these tools is to allow the users the insertion and semi automatic markup maintenance based on ontologies. The emergence of these tools took place along with the emergence of semantic web;

5. **Storage and Querying Tools**: The aim of these tools is to allow a simple query to the ontology.

It is important to note that in this research work only development ontologies tools were used. For Escorcio and Cardoso [EC07] the most important features to choose a development ontology tool are: Robust and ready to be used, Free and Open Source, Provides support to most of the activities involved in the ontology development process and ontology practice, Supports RDF, Supports RDF Schema and OWL, Offers a collaborative environment, Offers server-based environment with consistency checking support, Offers easy-to-use functionality for visual creation and editing, Offers a query builder, Supports a methodology (explained later), Supports editing formal axioms and rules, Supports the growth of large scale ontology, Supports versioning, Promotes interoperability, Has a reasoner, Has a graphical viewer, Promotes easy and fast navigation between concepts, Offers Plugins.

Depending of the nature of the work some of the previously analyzed features can occupy a major role comparing to other features.

3.3.2 Development Ontologies Tools

These tools help the users in the ontologies creation and edition process. Next, the authors will present a list of the most used development tools:

1. **Ontolingua** [FFR96][1]:Created by the Knowledge System Labs in Stanford University this system is constituted by a server and a representation language where the ontologies are the base of the language. This server is capable to export and import the ontology for several languages (DAML+OIL, Kif, OKBC, LOOM among other). The Ontolingua has no inference engine. The users of this system must have basic notions about Kif and Ontolingua to use this tool [EC07].

2. **WebODE** [CFlGpV02][2]: This tool was developed by the ontologies engineering group of the AI Department of the Technical University (Madrid). The WebODE contains an editor, a system of knowledge management based on ontologies, a web annotation tool among others. The ontologies can be accessed through a Java API (Application Programming Interface). This tool allows the ontology generation in XML, RDF(s), OIL, DAML+OIL, OWL, F-Logic and other languages and also supports many concurrent users and the Menthontology methodology (explained in the next section)

3. **Swoop** [KPS^{+}05][3]:It is an open source and java tool that supports a navigator and an ontology editor. First developed by Mind Lab of the Maryland University, Swoop is a web tool for creation, edition and ontologies processing. This tool supports:OWL validation, ontologies partitions among other features. This tool also provides a collaborative annotation feature.

[1] http://www.ksl.stanford.edu/software/ontolingua/
[2] http://webode.dia.fi.upm.es/WebODEWeb/index.html
[3] http://code.google.com/p/swoop

4. **Knoodl** [Lan07][4]:It is a tool that at the same time is an ontologies editor, a wiki and an ontologies register. Knoodl is also a knowledge based developed by Revelytix [5] company. This tool supports the creation of communities allowing their members to import, create, discuss, document and published ontologies. Also this tool supports OWL, RDF and RDF(s).

5. **Altova Semantic Works**[6]: It is a RDF/OWL graphical editor for the development of web semantic applications. This tool supports features for the creation and edition of a graphical configurable display where it is possible to print RDF and OWL graphical representations. Altova Semantic Works is capable to generates the following type of files:N-Triplex XML, OWL, RDF and RDF(s);

6. **TopBraid Composer**[7]: It is a modulation and development ontologies environment based on eclipse platform[8] commercialized by TopQuadrant[9]. This tool supports knowledge bases, management configuration of knowledge models and their instances among others. RDF, RDF(s) and OWL are the languages supported by this tool that also have integrated a deduction engine, a SWRL editor and SPARQL queries (it is a query language for RDF).

7. **OntoStudio**[10]: It is a environment for creation and ontologies maintenance most widely traded. This tool is the sucessor of the Ontoedit tool created by Applied Informatics and Formal Description Methods Institute of the Karlsruhe University. OntoStudio supports a mapping tool in which heterogeneous structure can be mapped to another one (in a quick and intuitively process), a graphical rules editor and an integrated test environment.

8. **Neon Toolkit** [HLSE08][11]: Based on the OntoStudio this tool supports many features such as: basic edition, visualization, browsing, import and export in many languages formats. Also this toolkit contains plugins for visualization and management, reuse, learning and ontologies mapping;

9. **Doddle-OWL**(Domain Ontology rapiD Development) [MFIY06][12]: It is an interactive environment for development of domain ontologies. Written in java this tool contains five distinct models:ontologies selection, visualization,construction, refinement and translation.

10. **Protege**[13]: It is an ontologies editor and a knowledge based framework maintained by the Stanford University. Protege is an open source tool that supports a knowledge modulation structure, visualization and ontologies manipulation in different representation formats.

[4] http://www.knodl.com
[5] http://www.revelytix.com/
[6] http://www.altova.com/semanticworks.html
[7] http://www.topquadrant.com/products/TB_Composer.html
[8] http://www.eclipse.org/
[9] http://www.topquadrant.com/
[10] http://www.ontoprise.de/en/home/products/ontostudio/
[11] http://neon-toolkit.org/wiki/Main_Page
[12] http://doodle.owl.sourceforge.net/
[13] http://protege.stanford.edu/

3.3.3 Tools Comparison

The choose of a toll will always depends of the user needs. A valid approach consists in identify each tool characteristics (description, architecture, interoperability, representation, inference, services) and choose the one that is more appropriated for a specific project [EC07]. Between the previous analyzed tools, Protege is the most well known in the research community normally used in domain modulation and for the construction of knowledge based systems that promote the interoperability. The Ontolingua was also constructed to facilitate the ontologies development based on forms in a web interface. The Altova Semantic Works is a commercial visual editor that has a visual intuitive interface and supports drag and drop features. The WebODE tool is a web applicative that supports navigation, edition, documentation, fusion, reasoning and other activities involved in the ontologies development process. The Swoop is an editor based on the web that contains validation and various syntactic views. From the selected tools, some of them have commercial concerns (Altova Semantic Works, TopBraid Composer and OntoStudio), others need a specific learning of a language (Ontolingua), others are more web applications (WebODE, Swoop, Knoodl) and others follow a specific methodology (WebODE). However, these functionalities are not sufficient enough to choose a tool. In the development of our ontology (explained in chapter 6) and comparing the Protege tool with the OntoStudio, the Protege presents a major advantage of being an open source tool. This characteristic is also presented in the Neon toolkit, however the number of features is very limited compared to the previous tool mentioned.

3.3.4 Protege Tool

The Protege platform supports two main modules for ontologies modulation:

1. **Protege-Frames**: it is an editor that allows the users to develop ontologies based in frames following the OKBC protocol;

2. **Protege-OWL**: it is an editor that allows the users developed ontologies for semantic web.

This tool also supports plugins developed by the users community. Two of those plugins are:

1. **SWRL Tab**: it is a development environment that allows the user to work with the Semantic Web Rule Language (SWRL). This environment also supports a mechanism that provides the interoperability between various rules engines.

2. **SWRL Jess Tab**:this plugin supports the rules execution in SWRL using the Jess[14] rule engine.

[14] Moreinformationsavailableathttp://www.jessrules.com/

3.4 Methodologies for Ontologies Development

This section presents some methodologies used in the ontologies development process. Before that, it is important to clarify some criteria regarding ontologies definition. For Gruber [Gru93] an example of those criteria are:

1. **Clarity**: An ontology must be capable to communicate its meanings to their users;

2. **Consistency**: An ontology must support inferences that are consistent with their definitions;

3. **Extensibility**: An ontology must be capable to define new terms based on previous definition;

4. **Minimized Coding**: The conceptualization must be specified in the knowledge level without relying on any symbol or coding language;

5. **Minimal Ontological Commitment**: An ontology cannot be restricted to the modulated domain.

In the next section there will be presented four different development ontologies' methodologies.

3.4.1 Methontology

This methodology was proposed by Fernadez et. al. [FGPJ97] in 1997. Normally this methodology is used in the ontologies construction process from scratch, taking as a base an existent ontology or in a reengineering process. Methontology is constituted by three phases:conceptualization specification, formalization and integration, implementation and maintenance. Its life cycle is based in prototypes' evolution and specific technics for each activity. Other activities like quality control, knowledge acquisition, evaluation and documentation are simultaneously performed with the activities related to the ontology development. Figure 3.6 summarizes the different phases of the ontologies life cycle. Each phase consists in activities that are going through a finite number of states:

1. **Planification**: it is important to plan all development processes like tasks, time, resources allocation among others;

2. **Specification**: At the beginning of the ontology development it is important to stablish its purposes and its scope;

3. **Acquisition of Existing Knowledge**:This phase is crucial for knowledge acquisition. This type of procedure also ensures a degree of consensus in the field

4. **Conceptualization**: After the Knowledge Acquisition phase, it is necessary to conceptualize knowledge using some knowledge modulation techniques;

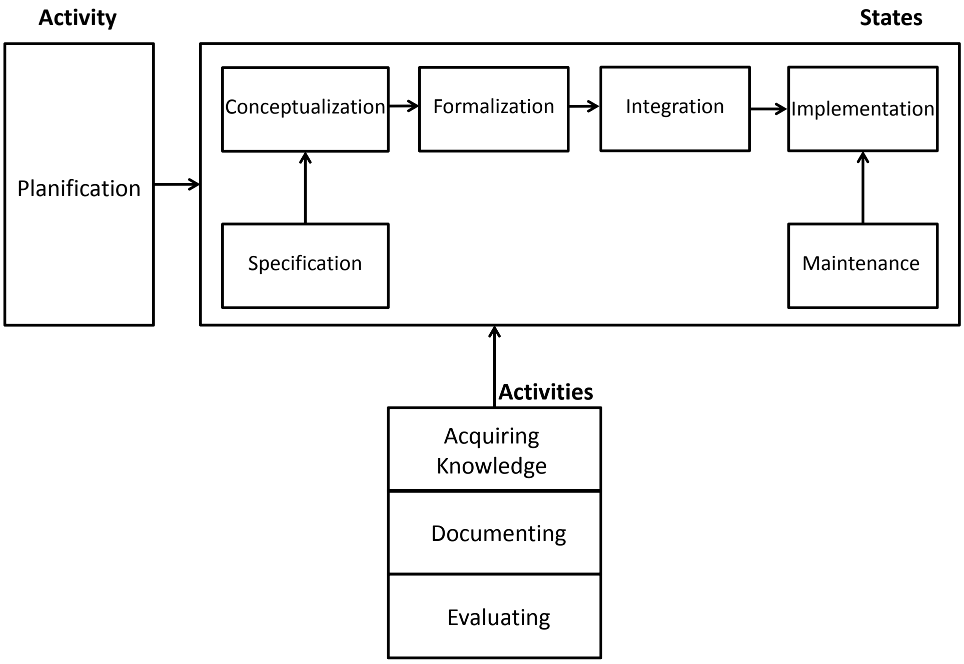


Figure 3.6: Different Ontologies life cycle phases (adapted from [FGPJ97])

5. **Formalization**: This phase is responsable for transforming a conceptual model into a formal one using a frame based system or descriptive logic representation;
6. **Integration**: Ontologies are designed to be reused;
7. **Processing**: For an ontology being processed by a computer it is important to select a processable implementation of a formal language;
8. **Evaluation**: It is important to evaluate the developed ontology allowing the detection of wrong definitions;
9. **Documentation**: After the previous step it is recommended to document the ontology promoting a simple reuse and ontology modification;
10. **Maintenance**: Finally it is important to note that after having finished an ontology it is also important to ensure its maintenance.

In the conceptualization phase this methodology recommends to structure the knowledge domain in a conceptual model. The activities involved in this process are:

1. Construct a complete glossary of terms. These terms include names, concepts, instances, proprieties among others;
2. Cluster the terms gathered in the glossary of terms as concepts or verbs;
3. For each set of related concepts, it is important the construction of classified trees of concepts;

4. It is important to establish a data dictionary for the collection of all concepts and their definitions. It is also important to construct: an attribute table of instances to provide information about the attributes and their values, an attribute table of class to capture the concepts but not the instances; an instance table to capture the instances and multiple decisions trees that graphically present attributes and constants;

5. In a similar way it is important to construct a diagram of verbs that includes the dictionary of verbs to express the verbs meaning in a declarative way; a formula and a rule table for formula and rule description.

3.4.2 101 Methodology

This methodology was created by Noy and McGuiness [NM01] in 2001. These authors proposed its use in an iterative and refinement process. In order to help the ontologies developers the same authors established three fundamental rules:

1. There is no unique way to module a domain (there are always many alternatives);

2. The development of an ontology is an interative process;

3. The ontologies' concepts should be approximate to objects.

Also the same authors established seven main activities:

1. Identify the domain and the ontology scope;

2. Reuse existent ontologies;

3. List the main ontology terms;

4. Classes definition and their hierarchies;

5. Classes proprieties definition;

6. Restrictions definition;

7. Instances creation.

3.4.3 Upon Methodology

The UPON (Unified Process for Ontology buildiNg) methodology was proposed by Nicola et. al. [NMN05] in 2005. This methodology is based on the IBM Rational Unified process and user Unified Modeling Language (UML). In UPON there are cycles, phases, interactions and workflows (Figure 3.7). This methodology supports features like:

1. **Use Oriented Cases**: The first input is the creation of scenarios as well as the creation of the use of cases of the discourse domain;

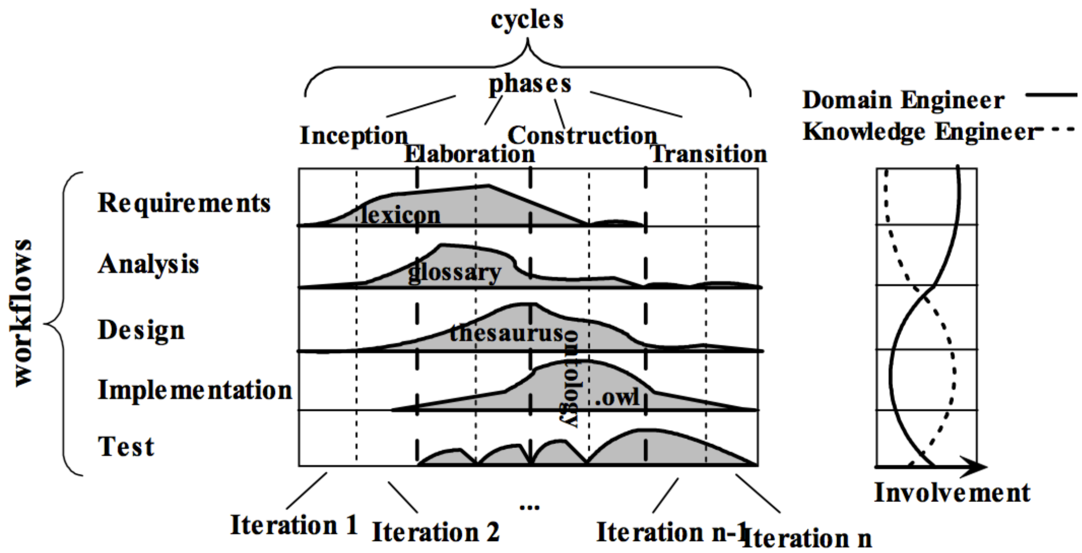


Figure 3.7: Different Ontologies phases using UPON Methodology [NMN05]

2. **Interactive**: The different methodologies phases of the development are followed interactively, starting with the details (in a very beginning state) and refining them until reach the specific domain aspects;

3. **Incremental**: The ontology can incrementally became more flexible accommodating new information gathered from new scenarios.

This methodology follows the unified process and has four distinct phases:

1. **Absorption** in which the concepts capture and modeling use cases is required;

2. **Elaboration** where an analysis of requirements is made and the fundamental concepts are identified;

3. **Construction** where a generic ontology structured must be designed;

4. **Transaction** where the ontology is validated through strict tests.

3.4.4 O4IS Methodology

The O4IS (ontologies for information systems) was proposed by Kabilan [Kab07] in 2007. This methodology is oriented for the conceptualization, conception and ontologies development having as a target developers with little experience. The O4IS distinct phases are:

1. Establish the ontology scope;

2. Establish the users profile, the applications and the functional requirements;

3. Choose the ontology architecture (physical and logical);

4. Choose the ontology development approach;
5. Choose the level of ontology representation;
6. Choose the methods and tools for knowledge acquisition;
7. Knowledge representation: implemente the domain ontology;
8. Evaluate and verify the domain ontologies;
9. Use, maintain and manage the domain ontology.

3.4.5 Brief Methodologies Comparison

The Methontology was the first methodology that recommends the use of conceptual models in the ontologies design and, because of that it becomes a reference in the ontologies development area. The 101 Methodology is more a tutorial oriented using the Protege tool as the development environment. However, the proposal steps are sufficiently generic to be adopted in other tool. The UPON methodology combines the rational unified process with the explicit specification of an engineering process of an ontology which must be processed. Also it elucidates about the domain specialist and developer role in the life cycle of an ontology development. The O4IS methodology is the most recent one and uses many concepts from other methodologies. However, it presents an important disadvantage: this methodology is oriented only for domain ontologies.

3.5 Conclusions

It is easy to note that ontologies can be a solution to create a standard and understandable language for a specific application domain. Over the years many were the research groups who have bet their research in development of ontologies and today many development tools and languages are available. The selection of those tools and languages must attempt to the ontologies application context and the features that this application will support.

In chapter 6 it will be presented a soccer ontology developed in this thesis. The main goal of the development of this ontology was to clearly understand the relations between the concepts involved in this particular domain and create some new soccer concepts related, for instance, with the tactical team detection.

Chapter 4

Domain Application Analysis: Human and Robotic Soccer

Included in the Collective Sport Games (CSG) context, soccer is nowadays the most played sport [Rei96]. However, this sport quickly became one of the largest business of the planet and it is not rare that the broadcasting rights for 4 years have been sold in England for approximately US $1 billion, a typical sponsor contract for a top European team is valued at US $6 million (yearly) and the annual salary of a top striker is rumored to be US $6 million [Kon00].

In the competition context, the tactical modelling appears as one of the main research areas focusing on the players and coach activities. Based especially on the game organization many authors have tried to understand the game logic [Duf93] [Gar97a]. In consequence of that, the tactical modelling has been used to identify relations between game events using in this detection many individual and collective success factors. For Hughes [HB02] the soccer tactical performance reflects the importance of team work, of movement in the field of play which constitutes an importany factor in the detection of positive and negative players technical aspects.

Over the years in this area many studies have emerged concerning the teams and players actions [YHL93] [Gar97a] [BJM02]. Methodologies and sophisticated tools, with an enormous capacity to collect and record data, are indispensable means for researchers and also for coaches. For the analysis of tactical solutions five specific game phases are typically considered [Gre89] [Bat98] [Gar97a]: offensive phase, defensive phase, attack-defense transition, defense-attack transition and normalized/pattern plays (free-kicks, corners, throw-ins etc.). Also for this analysis it is determinant to know the history of soccer and consequently its entire tactical development over the years.

In this Chapter the history of the CSG is presented as well some different systematization perspectives. Also the soccer history is presented specially regarding to the tactical and game rules evolution. As mentioned before as one of the most played sports on earth, over the years soccer has inspired several research projects. One of the most prestigious projects that uses soccer as the base its the RoboCup international competition. A description of this competition is also presented describing its leagues, its goals and also the tools that support the 2D simulation league, more precisely, the soccer server and its technical issues.

4.1 Collective Sports Games

The CSG appeared, for the first time, in primitive societies where it was common to see a large number of games that uses, as their main object, the ball. In this period there were many CSG around the world, such as the *tsu-Chu* in China (where the ball should be launched in the middle of two sticks buried in the ground), *koura* in the Arabia countries, *skinny* in North America, *pok ta pok* (precursor of the basket game) in the inca civilization(VII century a.C.) among others. In Japan, in the X century a.C., the population played a game called *kemari* where a small ball is kicked by the participants and some researchers believe that this game was the precursor of the soccer game.

Throughout the years, many are the authors that tried to define "what is a game?". In the book *A Terminologia da Educação Física e do Desporto(1973)* the game is " a complex activity predominantly emotional and physical, spontaneously developed following pre-defined rules, with the aim of competition and at the same time to adapt to the society's reality". In this same publication the Sport Game is presented as a "set of physical exercises in the form of game with a specific object (ball,disc etc.) used by two teams or two opponents following established rules". Using this definition as a base, it is possible claim that the CSG are specific situations that include two opponent teams which have a common initial goal: recover the ball. After that, and avoiding the opponents' players, they try to reach the target/goal. The nature of the teams participation in these games, follows established rules that define: the number of participant players, the game duration, the number of points/goals reached in specific movements, the play field characteristics, the players uniform characteristics and the object used to play the game, the referees number and their functions in the game [Gar01b].

Over the years, many authors tried to identify similar aspects between all of CSG. One of the more accepted approaches by the research community was presented by L. Teodorescu [Teo75]. In his approach, Teodorescu presented nine similarities between the CSG:

1. The existence of a circular object game (ball, oval ball, disc, etc.);
2. A complex character of the competition in individual and collective terms;
3. Pre-Established rules defined by an international board group;
4. The definition of a game duration (regular and extra time);
5. The definition of assessment criteria (points, goals, etc.);
6. The technical and tactical characteristics of the game players;

7. A well defined competitive system;
8. The beauty of the different sport events;
9. The definition of the game preparation correlated with the training sessions methodologies.

In recent years, other authors tried to characterized the CSG. In 2001, Garcia [Gar01a] based on Waldrop [Wal92], defined the CSG as complex adaptive systems that present six distinct characteristics:

1. The game involves many agents with distinct individual and collective goals, acting without any control of their actions;
2. The agents, normally designated as players, produce interactions with the other players (that could be teammates or opponent players) in a specific environment. These agents also act and react through the game actions, produced by the other game actors. These actions provoke constant changes in the game environment;
3. The agents organize and reorganize in complex structures, according to the increase of external game changes and, through the agent learning and adaptive process;
4. The agents are capable of predicting the future according to internal models (internal logic), that they execute following pre-stablished rules. This knowledge will produce specific behaviors in the agents' interpretation through the game;
5. The agents do not have capacity to optimize their behavior due to the large number of possibilities that the environment supports. For that reason the only possibility that the agent has is to change its capacity through the analysis of the behavior of its opponents;
6. There are well-defined behaviors sets that can be attached to a task or/and space. However when the context changes, new behaviors emerge as well as new opportunities;

The first attempt to systematize the CSG was created by Buhler which defends that the emerging of the CSG was related to the level of societies development, with their mentalities and, also more important, with the competitive (individual or collective) level of the teams. After that, the systematization that obtained more acceptance was created by Dobler [DST89] which divided the CSG in 4 groups:

1. Sports Games with Target (goal, basket or hole). In these particular games some technical and tactical analysis could be made regarding the characterization of the attempt to reach the target and, on the other hand, analyze how to prevent the opponent from reaching the target. In this context a more deep division can be made:
   (a) Physical Contact Allowed: Examples of these games are American Football (USA), Ice Hockey, Handball, Hurling (ireland), *La Crosse* (Canada and USA), Rugby, Water Polo, among others.

   (b) Physical Contact is not Allowed: Basketball, Cicloball, Roller Hockey;

2. Return Games (implies launching a ball over a line, a net, etc.): These games could be played individually (1x1), in pairs (2x2), triples (3x3) or in larger groups. These could be divided in:

   (a) Played individually and/or in Pairs: Badminton, Table Tennis, Squash, Indiaca (played in South America);

   (b) Triple Teams or more Players: Beach Volley, Indoor Volley;

3. Games with ball Percussion which involve catching the ball after it is launched. Examples of these games are: Baseball(played in USA, Japan,Cuba), Softball(USA, Japan,Cuba), Cricket(Great Britain, Australia, India), Oina(Romania);

4. Games involving sending the ball to a target with percussion, impulse and stroke. Examples of these games are: Golf, Boccia (Italy, Swiss), Gorodki (Russia).

In 1992, a new classification for the CSG emerged [RE92] (Figure 4.1).

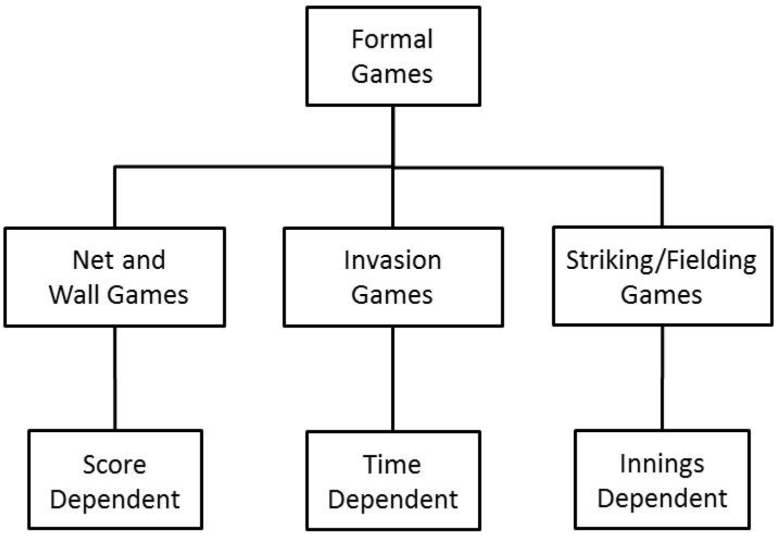


Figure 4.1: Read and Edwards Sistematization (adapted from [RE92])

However, this systematization was complete only in 2002 by Hughes and Bartlett [HB02] when they divided it in several subsections. They started by dividing the Games with net and wall in two distinct categories. Also they divided the net games in three distinct groups as represented in Figure (4.2)

In this category, the existence or not of Volley or the existence or not of the bounce, frames the volley in this section.

In the category of wall games and despite the fact that the games identified had the same success factors, the authors did the following distinction (Figure 4.2).

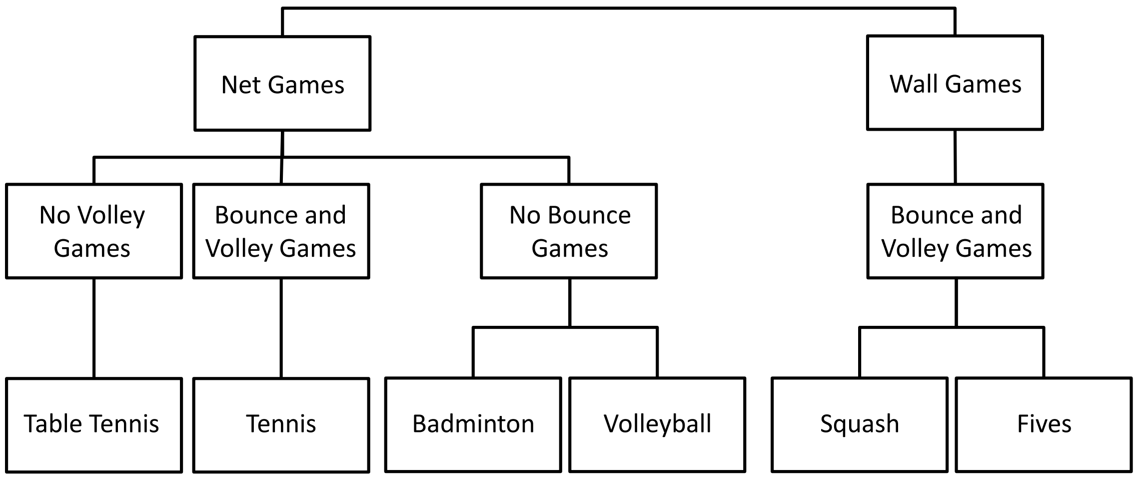


Figure 4.2: Sub-Categorization of Net Games and Wall Games with examples (adapted from [HB02])

The invasion games category was also divided in three groups: Goal Throwing Games, Try Scoring Games, Goal Striking Games (Figure 4.3).

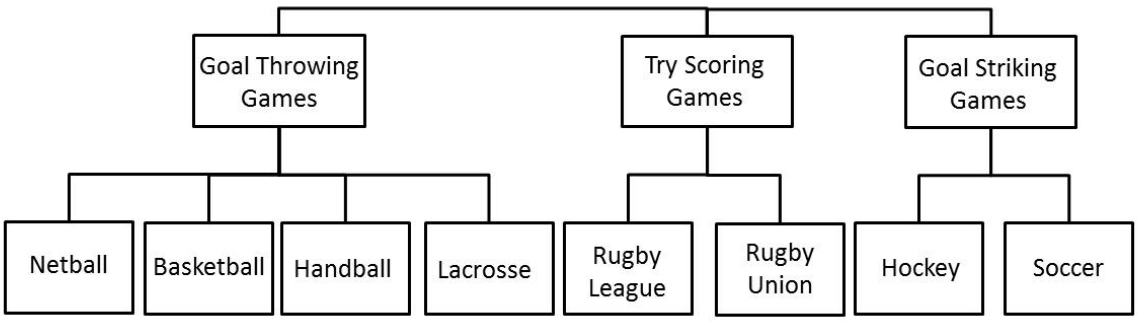


Figure 4.3: Sub-Categorization of Invasion Games with examples (adapted from [HB02])

The authors also divided Base games category into two groups: Wicket Games and Base Running Games (Figure 4.4).

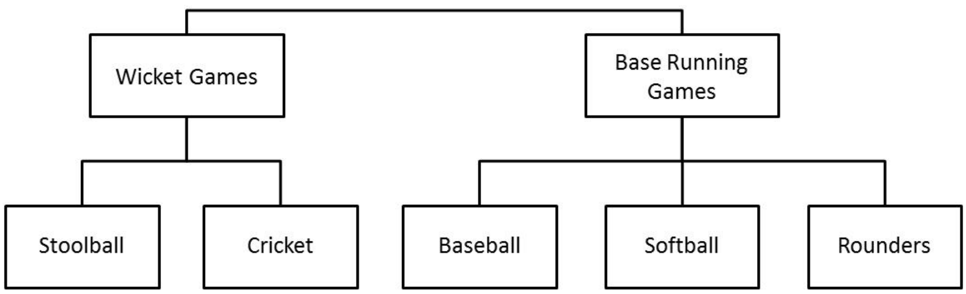


Figure 4.4: Sub-Categorization of Base Games with examples (adapted from [HB02])

In the next section a brief introduction about a specific CSG (soccer) in terms of its history and evolution throughout the ages is performed. The robotic soccer was also analyzed in terms of its history as well as its most important competition (the RoboCup competition).

4.2 Human Soccer

For many authors, soccer is the most played sport on Earth [Rei96]. In this sport, included in the CSG universe, two teams try to score more goals than its opponent (the main goal of the game). In this context Castelo [Cas03] defends that "The game possesses a specific dynamic, a context that defines its essence. This essence, included in the game rules, gives rise to a series of attitudes and technical/tactical behavior patterns. More specifically the requirements that are imposed on the players are determined by the game profile". In consequence of that, in 1982, Kacani divided players in three categories:

1. **Universal Players** – Players are capable of fulfilling with the same performance (typically high), many tasks, in the offensive and defensive field;

2. **Semi-Universal Players** – Players are capable of fulfilling with high performance only one of the game play moments (offensive or defensive);

3. **Specialists** – Players with a defined expertise, able to perform effectively the duties of a phase of play in a particular sector of the field (e.g. the goalkeeper.)

After this previous contextualization, a history of human soccer will be presented in the next section. Also the most famous robotic competition based on human soccer will be exposed (RoboCup) as well as its leagues. At the end of this chapter, the server that is used in the RobCup 2D simulation league will also be presented.

4.2.1 Human Soccer History

Soccer is a CSG where two teams try to score at least one more goal than its opponent. Concerning its history, some researchers concluded that soccer is the result of a slow evolution of different ball games throughout the years, from rudimentary forms until reaching the tactical, technical and physical complexity that it presents today. Morelli [Mor86] claims that there were facts that prove that this sport began to be practiced in the prehistory era (considered by some authors as a *pre-industrial* soccer era [Wal00]) where men were running after a round object (similar to a soccer ball). Other references have emerged in literature such as in Homero's Odyssey when the character Nausica was found playing a ball game with her mates.

During those times, according to some authors [PC02], many game versions around the globe have emerged: *Kemari* (praticed in China and Japan), *Epyskiros* (Ancient Greece), *Haspartum* (Ancient Rome), *Soule or Choule* (France) and *Calcio* (Italy):

- *Kemari* (Japan) - Around 2.600 b.C, the Chinese Yang-Tse created a game called *Kermari* for military purposes. This game was practiced in open spaces (outdoor) with a square shape and where the ball could not get out of the field improving the technique of shooting. Through two big fans of the game (the emperors Engi and Terei) this one had a rapid diffusion in the middle east region [Sou97]. Despite of this game has been played in Japan, in China the game quickly became a profitable activity.

- North, Center and South America - Normally it is common in studies that involve soccer history to appear references related to Japan and also Europe. However, some researchers affirm that when Colombo arrived to America, in his second trip, he observed some local natives playing a game similar to soccer. Also, this practice would have more than a thousand years and was practiced by the Mayan and Aztec civilizations [Sou97].

- *Epyskiros* (Ancient Greece) - In 884 b.C held the first Olympic Games in Greece without any game involving a ball (Figure 4.5). However, the religious practice of exercise for

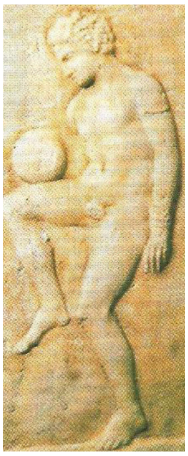


Figure 4.5: Picture carved in stone dating from ancient Greek

  body and mind for all kinds of social classes was already practiced. At this point a game called *Epyskiros* emerged where a ball (bladder of an ox) was kicked by a group of players. However the rules and the number of participants are unknown [Dua97].

- *Haspartum* (Ancient Rome) - Through the centuries the Roman people spent the majority of their time in activities related to science, philosophy and art. In consequence of that, the practice of sport was somehow put aside. Over the years the sports were slowly introduced and the first reference for a ball game consists in a game played in a field with three lines (two in the wings and one in the center) and with a rectangular shape. The main goal was through the consecutive passes between teammates, to reach the opponent's bottom line, winning points. The teams consisted of defenders (the more slowly ones) which are called *lócus stadium*, the midfield players which are the *medicurrens* and played for both teams and finally the players with more offensive actions which are the *pilae praetervolantis et supriactae*

- *Soule or Choule* (France) - Some researchers present some doubts related to the origins of the *Soule* game (France or England). Some support the English origins with an episode in the Viking era. In the VIII century, the vikings attacked England without success and their chief was murdered and decapitated. To commemorate this achievement, the villagers kicked the chief's head of the chief through the village. Other researchers believe that the game appeared in France, through the romans (with the emperor Julius Caesar), in the 58 and 51 b.C. This game was inspired by the *Harpastum* Roman and its designation was changed from region to region (sometimes the designation changes for *Choule*). In the beginning of the Middle Age, *Soule* faced some religious and social problems and became a dangerous and violent game. Also the authorities at the time, banned it to be played in public places. However, the game had many fans specially in the royalty such as Henrique II (king of England) [VGaR06]. At this time more civilized games practiced specially in aristocrats schools have emerged such as the *field game* (created in 1520 in the Eton college which is a mix between soccer and rugby) and in Oxford and Cambridge the soccer gained enormous popularity in the 16th and 17th centuries respectively.

- *Calcio* (Italy) - The origin of *Giocco del Calcio* remotes to 17th of February of 1530 when in the city of Florença two political factions measured forces in a ball game (Figure 4.6). Each team has computed by 27 players. The ball could be played using the hands and/or feet

Figure 4.6: Picture representative of the ball fill for the *Giocco del Calcio*

and the goal was to put the ball in the opponent's gorge (similar to a goal) and designated as *caccia*. This violent battle lasted over two hours and typically processed in *Piazza di Santa Croce*. The diffusion of this game was even greater in the Renaissance of the Medici family (XV to XVIII century) and had many important supporters like Leonardo da Vinci or Maquiavel.

- *Foot-Ball* (England) - At the beginning of the 19th century, the successive government bans and the violent nature of the game decreased the number of practitioners of the primitive ball games. The rural depopulation and the almost uninterrupted industrial work also contributed for that decrease. In Public Schools, the students' little interest for the academic formation and the disrespect for their teachers were two main problems. In 1830 this reality suffered a radical following of new values such as the importance of the competition and fair play. In this environment the game was seen as a tool capable of creating an young elite with a physical and intellectual capacity, combined with a strict moral code and an unshakeable belief in action. In 1840 due to the construction of the train lines, the spreading of the game throughout England was enormous and because of that the first reformulation of the game's rules appeared. At this point, two separate practices were established: one that allows touching the ball with hands (rugby) and other prohibiting such use (soccer). Until 1871 there was no a Goal Keeper Player in the soccer game (because no one had the authority to touch the ball with his hands) and also just one year before the referee figure appeared. In 1863 the Football Association was founded whose major goals were to organize the matches between team members and definitely establish the game rules (because in that time the rules varied according to the country area where the game was played). During the industrial era, some authors believed that soccer operated as an outlet for the hard work [Hop06], others think that the increased participants number results was due to the free working times [Wal00]. The increased importance given to leisure and sports allowed the creation of the official competitions. In 1888 the first England Soccer League competition took place (Figure 4.7). Soccer quickly became a mass sport (as evidence by the number of

Figure 4.7: Picture representative of a match in the England Soccer League Competition

spectators at each game). The average of match assistance in 1888 was about 4600 people and, 10 years later, more than duplicated that number. In 1872, in the city of Glasgow, the first national match took place between England and a national team constituted by Scottish people who lived in London. Since 1870 until 1890 the soccer game has spread to all European countries, and in 1904 FiFA (F'ed'eration Internationale de Football Association) was created. This association is responsible for organizing the World Cup competition where 31 national teams all over the world compete to achieve the first place at the final podium. This competition takes place 4 in 4 years (with the exception of the years of the second world war).

4.2.2 Soccer Game Evolution Trough the Years

In 1848 the first standard set of soccer rules appeared. This set was strongly influenced by the Rugby rules but present some particularities such as the prohibition of executing a pass to a teammate that was positioned in the front of the ball possession player, among others [Tad92b] [Cas96]. These restrictions will influence directly the way a soccer team plays the game. If a player executes a pass to a teammate that was at his front (in the attacking field perspective) the referee will consider that as an offside situation. In consequence of that, the majority of game situations were characterized by individual actions executed by the players (it is a common situation for a player that has the ball trying to pass through the opponents and if he is successful, scoring a goal). On the opposite way an opponent player will try to stop the attacking play, turning the game situations very violent [Tad92a]. This era was classified by some researchers as the dribbler era [Cas96] (an example of a team formation used in that era is represented in Figure 4.8).

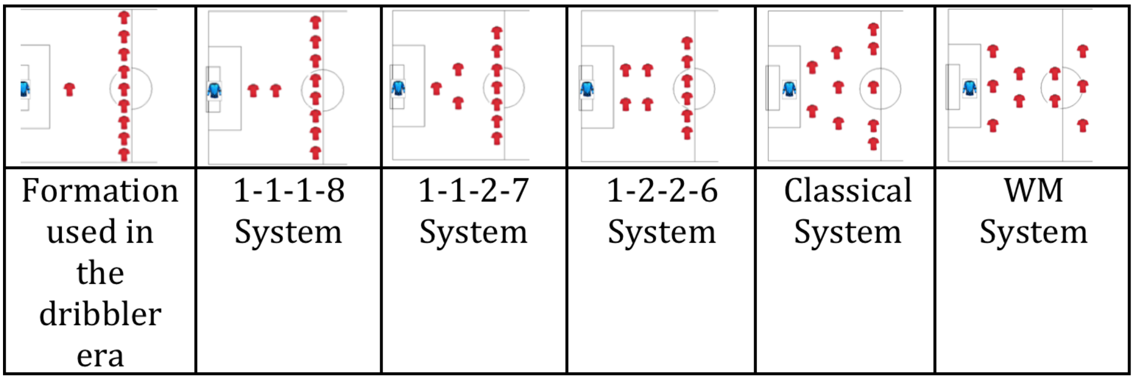


Figure 4.8: Formation Systems Evolution before 1931

In 1863, with the creation of the Football Association, their creators considered that Rugby and Soccer could not be considered as identical sports. In this way, the soccer game suffered a huge development and in 1866 a set of distinct rules have emerged. The rule which changed more was the offside. After that changed, a player could execute a pass to a teammate that is in front of him in the soccer field unless, the receiver player has less than 3 opponent players between him and the opponent goal line. This simple change among others, transformed the game, turning it a more collective one (with more collective actions specially offensive ones). These actions became a major factor in the final game result [Gre92]. In consequence of that the soccer teams progressively assumed a more defensive perspective and the evolution concerning to the increase number of defenders and midfield players [Cas96] (This evolution had a duration of twenty years approximately). An example of these systems evolution is illustrated in Figure 4.8.

For many researchers, the first soccer system only appeared in 1884 and was designated as *Classical System* or *Pyramid System* (Figure 4.8). Basically this system was organized with the following structure: (1 Goal Keeper, 2 Defenders, 3 Midfielders and 5 Strikers). It is important to note that this system presents a concern regarding the position of the players in the field versus the occupation of the three field sectors (defense, midfield and attack) [Cas96].

The classical system was used for more than 50 years. However, in 1925 the offside rule was changed again decreasing the minimum number of opponent players allowed between the receiver ball player and their goal line (decrease from 3 to 2 players). This change will influence the evolution of the soccer systems. In 1930 a soccer coach named H. Chapman appeared (Arsenal Head Coach at the time) that created a revolutionary system called the "WM" system (Figure 4.8).

Nowadays, this system was considered for many researchers as the mother of all soccer formations and the father of all soccer systems [Tad92b] [DF98] [Lob07]. The novelty of this system consists in including a numeric balance between defenders and strikers and consequently appeared the individual defense with collective concerns [Seb96]. Also this characteristic provoked a more rational occupation of field regions and consequently a more collective game. During that time, the tactical component together with the technical one determined the game quality. In the Chapman era two more quality soccer coaches emerged: Pozzo and Meist, that will change completely the way of playing the game [Lob07]. The first one, V. Pozzo, created a new concept for the game *The Counter Attack*. Pozzo based his collective Idea in a passive game with strong defenders and with a simple main goal: after recovering the ball, passes it, in a quickly process, to a striker which, using his technical skills, will create a score opportunity. The combination of counter attack situations and the constantly changes in the players position allow Pozzo to obtain success in his coach career [Seb96] [RDM98].

The second coach was H. Meisl that during his times as Austrian coach created a team called *The Wonder Team* between 1931-1935. His game idea is based on technical players characteristics and their soccer field positions. Meisl uses the classical system with some changes: with the balance between all of the field sectors, Meisl was capable to defend with more players (comparing to his opponent team) [Mar79] and the attacking players have a simple goal to achieve: not let the opponent players play. This new concept was designated as *Pressing* [Seb96] [Lob07]. Figure 4.9 illustrates the Meisl WunderTeam system.

One of the similarities that characterize these three coaches was that all of them had in their teams players with good technical skills. In this era all the players were specialists (concept previously explained) different from the present time where normally a player could occupy different positions in a field during a match [Lob07].

The WM system was used for more than 50 years. However, in 1953 England's national team was defeated by Hungry's national team. Until that time this system suffered some changes. However at the beginning of the 50's G. Sebes (Hungry's national head coach) presented a new

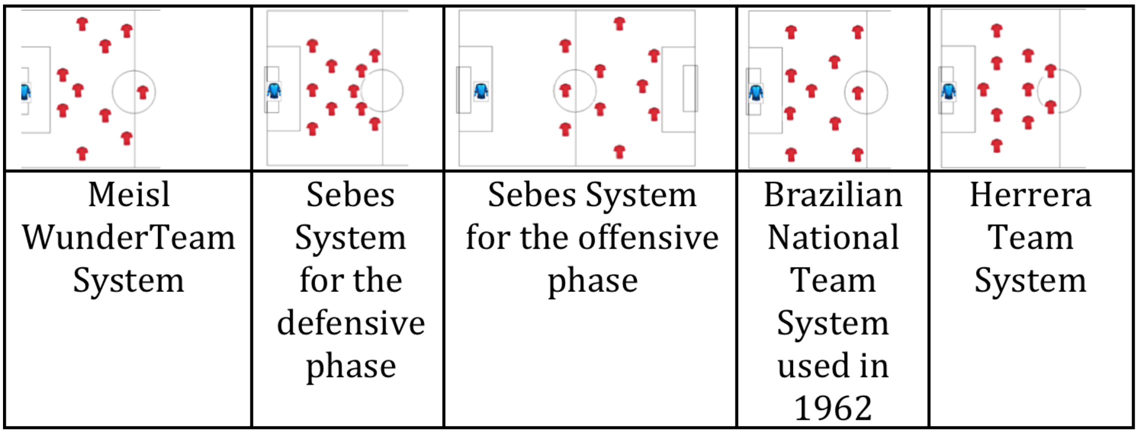


Figure 4.9: Formation Systems Evolution after 1931

player dynamic specially in the offensive sector. His team behavior changes whether they are attacking or defending (Figure 4.9 and some authors believe that this was the beginning of a new soccer system 1-4-2-2 constituted by 1 Goal Keeper, 4 Defenders, 2 Midfielders and 4 Strikers [Men79] [Tad92b] [Lob07].

In 1958, the Brazilian national team presented that system in the World Cup and four years later presented a new system 1-4-3-3 (1 GoalKeeper, 4 Defenders, 3 Midfielders, 3 Strikers)(Figure 4.9). Some authors defend that this change occurs between competitions due to player characteristics.

In the 60's a new soccer system emerged called *Catenaccio* created by the Internazionale Milano Head Coach H. Herrera. This system was characterized by the creation of a new player position called *Libero* which consists in positioning a player behind his defense line (very useful for the situation where a teammate is being overtaken by an opponent player) (Figure 4.9). In terms of offensive actions this system was based in the counter-attack concept (previously explained).

Resuming the 50's and 60's, the game was substantially improved due to the emerging of new soccer systems (based on the tactical evolution and the emerging of players with great technical skills). Also the collective actions became even more important influencing the final game result [Cas96].

Until 1970, the evolution of the game was based in the merge of two dimensions: Tactical and Technical. Tactical dimension related to the systems evolution and Technical related to the appearance of players with great technical skills. In the last years of the 60 decade, the Physical dimension emerged as an answer for the different dynamic the game presents. Since the 70's two new concepts appeared in soccer:the tactical.physic concept and the tactical-technical concept. The first one was in the english teams over the 70,80 and 90's they had big fans and basically consists in supporting the tactical dimension exclusively in the physical dimension. Typically these teams present some tactical stiffness. The other new concept (tactical-technical) is based in the tactical and technical dimensions and, of course, the physical dimension is present a supports for the two other major dimensions (in the opposite direction of the other concept where the major

Idea of all process is the physical dimension). R. Michels (one of the most important European Coach of all times and considered by the FiFa the coach of the 20th century) proved that although the physical dimension could be used in soccer as team strategy, the other concept presents better results in the majority of the games. This conclusion is also supported by other coaches like A. Sacchi which training the A.C Milan team in the early 90's. This Italian coach proved that the tactical-technical concept needed to be adopted in order to win titles.

In conclusion, and doing a brief summary about soccer history, the first evolution consisted in transforming the game in a more collective one. The second transformation occurs with the game's system evolution and with the emergence of players with excellent Technical Skills. Finally, the third evolution consists in using the better soccer concept between tactical-physical and tactical-technical dimension.

In terms of the future it is very difficult to predict the next evolution however, some authors believe that the teams and specially the players' skills will improve in the nearest times and because of that, soccer coaches will need to invest in new strategies that include new ways of training, new systems and new players' dynamics [Val01] [Cru02].

4.3 Robotic Soccer

The year of 1997 is a turning point in the history of AI and robotics. In May 1997, a computer (Deep Blue) defeats the human world chess champion solving a research problem with more than 40 years. Also in this year NASA's pathfinder mission made a successful landing and the first autonomous robotics system, Sojourner, was deployed on the surface of Mars. Finally, the RoboCup project appears with an ambitious goal: to develop a robotic soccer team capable of defeating the human World Cup champion team.

4.3.1 RoboCup History

RoboCup [KAK+95] [KTS+97] is an international research and educational project whose main objective is the promotion of Artificial Intelligence (AI) and Intelligent Robotics. Basically, the research problem behind this project is the Robotic Soccer, where a number of distinct technologies are needed to construct a real or virtual Robotic team capable of playing a soccer game with a set of distinct rules. The original idea of Robotic Soccer was introduced in 1992 by Alan Mackworth [Mac93]. In parallel, in the city of Tokyo, a group of Japanese researchers promoted a workshop related to the use of soccer for the research community especially in AI areas. Also, some Soccer Robotic prototypes and a simulator project were defined. The result was the creation of a Robotic League called Robotic J-League (inspired in the name of the professional Human Soccer League). After its huge success, this project became an international project with the name of Robotic World Cup Initiative - RoboCup.

In a parallel way, many researchers have already used robotic soccer as their research domain. Itsuki Noda (together with other researchers) in the ElectroTechnical Laboratory (ETL) developed a simulator for virtual soccer games which later gave rise to the soccerserver [CFH+03] (explained later).

In order to promote the investigation in this field, a long term objective was proposed: "by the year 2050 a humanoid Robotic team will be capable of defeating the world champion Human team in a soccer match according to FIFA rules" [KAK+97]. Although this objective seems to be a bit unrealistic today, others are that can be established and can constitute a base for future projects, such as the creation of Robotic Soccer teams with identical style and play behaviors, when compared to Human teams, or the creation of teams capable of playing a soccer match against a Human team (not necessarily the world champion team).

The competition that anticipated RoboCup was held during the IROS96 with eight simulated teams and a demonstration of medium robots. The first RoboCup competition was held in Nagoya 1997 with over 40 teams. The organization estimates that more than 5 million spectators assisted the games which turned the RoboCup into one of the biggest events ever.

Since the first edition, RoboCup became one of the most massive annual research events (Figure 4.10 illustrates the last RoboCup event) and because of that huge public/researchers affluence, the RoboCup board, decided to create parallel events (before the main RoboCup event) which allow the researchers to have an opportunity to test their teams. Normally these events have the designation of Opens like for instance the German or USA Open.

Figure 4.10: General aspect of RoboCup 2009 - Graz

4.3.2 RoboCup Federation

The RoboCup Federation is an international organization, registered in Switzerland, whose the main goal is to organize the international efforts in order to promote science and technology using for that robots, soccer games and software agents [Fed01] .

The main RoboCup federation functions are connected to the annual organization of the world championship and coordinating the overall efforts of all researchers in the field. As the number of participants in the RoboCup is very large and spread over the world, the RoboCup federation decided to create regional committees promoting the RoboCup research in different geographical areas.

This federation is composed by a chairman, a trustees set of researchers, an executive committee, an adviser committee and a technical committee (one member for each league).

The technical committees have the goal to established the competition rules as well as to establish the future of the leagues. At the beginning these committees were constituted by members designated by the execute committee. However, since 2001 this method was changed. In consequence of that the executive committee elects four elements and the other three elements are democately elected by the research RoboCup community.

4.4 RoboCup Leagues

The challenge proposed by the RoboCup organization to the AI and Robotic researchers includes 5 groups (although the main role is occupied by the RoboCup Soccer)(Figure 4.11):

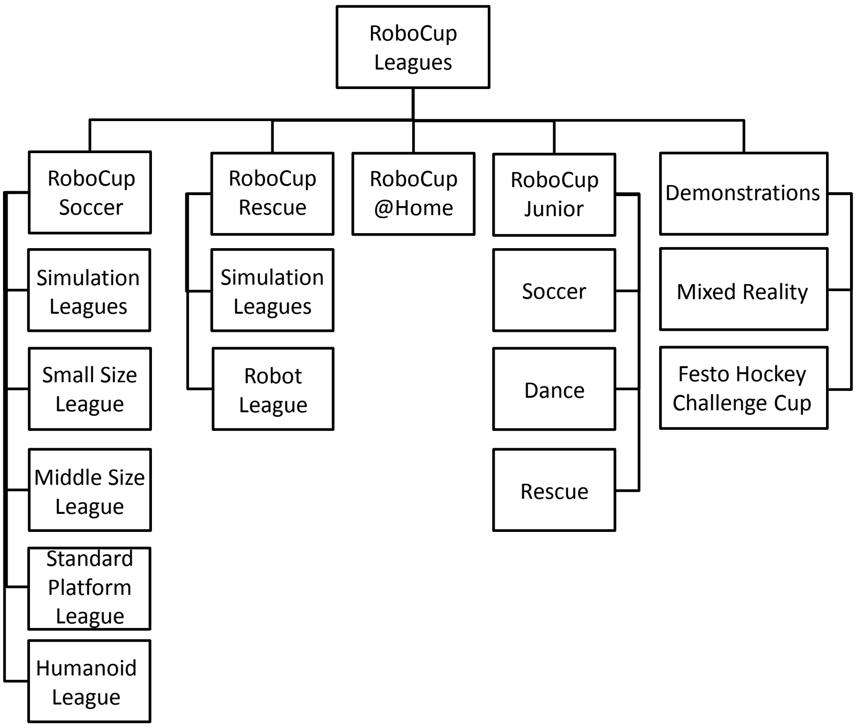


Figure 4.11: RoboCup Soccer Leagues

1. **RoboCup Soccer** – Robotic soccer including two simulation leagues and four main robotic leagues with distinct rules;

2. **RoboCup Rescue** – Research application in the search and rescue in large disasters domain. This category is divided in 3 groups: a simulation league, virtual league and a robotic league;

3. **RoboCup@Home** – Competition focused on real world and human-machine interaction with autonomous robots. The main goal is to create robotic applications that can assist humans in everyday life;

4. **RoboCup Junior** – The use of RoboCup in an educational environment. Using a simple infrastrutue, children and young people can create teams of robots to play soccer (reduced situation like 2x2), dancing or rescuing victims;

5. **Demonstrations** –The demonstrations show new ideas and concepts for future competitions.

The robotic soccer is the main event of the RoboCup. This competition is divided into five leagues:

1. **Simulation Leagues** – Using the ssmspark, two virtual teams constituted by eleven players simulate a soccer game;

2. **Small Size League** – Using a small field, two teams composed by 5 small sized robots with centralized control and vision each simulate a soccer game;

3. **Middle Size League** – Teams composed by 5 or 6 autonomous robots compete in a 18x12 meters field;

4. **Standard Platform League** – Teams composed by 3 humanoid Aldebaran Nao robots compete in a reduced sizesoccer field;

5. **Humanoid League** – Autonomous robots, freely constructed by participating teams, with a human-like body and human-like senses play soccer against each other.

Beyond these Leagues, others have emerged with different competitions:

1. **Coach Competition** – A game analysis system that should be able to change the behavior of a soccer team in the middle of a match (real time decision), considering the opponent team behavior;

2. **Intelligent Sports Commentator** – A virtual commentator developed with the intention of commenting each event of a soccer match;

3. **3D Visualizers** – Construct three-dimensional game viewers, including realistic animation and realistic sounds.

Every year the RoboCup federation organizes its main event (RoboCup) measuring the scientific progress of various research groups. This main event is composed by different leagues and challenges, demonstrations and a scientific congress [LSST07] [VROD08] [IMWZ09] [BLN10]. There are also throughout the years many competitions related to RoboCup such as the European Championship, the German Open, Portuguese Robotic Open (included only middle size league), the Japan Open, the China Open, the USA Open among others.

4.4.1 Simulation Leagues

Unlike the other leagues, in the simulation leagues there is only virtual robots and their focus is in artificial intelligence and team strategy. These leagues are composed by two main sub-leagues: 2D and 3D. The 2D simulation league is based in the soccerserver platform [NNMH97] [CFH+03]. This system simulates a 2D soccer game composed by a soccer field and two teams. Each team is composed by eleven players and eventually by a soccer coach which connect to the simulator using a client-server architecture and UDP sockets. This simulation accepts low level player commands executing them imperfectly and sends perception information (also imperfect) to the players.

The 3D league robot is a simulation of the NAO robot used in the Standard Platform League. The usage of this simulated robot not only shifted the aim of the 3D simulation competition, from the design of strategic behaviors for playing soccer, towards the low level control of humanoid robots and the creation of basic behaviors like walking, kicking, turning and standing up, but also provided a test platform for the teams of the Standard Platform League (explained later).

In the next sections all the RoboCup leagues will be subject of a deeper analysis (with major focus in the 2D simulation league).

4.4.2 Small Size Robot League

The Small Size Robot League is also known as F-180 league. In this league two teams composed by five robots play a soccer game with an orange golf ball in a small sized field (slightly bigger than a ping pong table). During the games, the robots control, receives information sent by a camera (located 4 meters above the playing surface). After that, the robot control will process that information, determining the positions, orientation and robots velocities, choose the better command for each robot and finally through a radio communication the robot control sends the appropriate command for each agent (this variation is by far the most common one in this challenge).

The soccer surface is made up of a green carpet and bordered by white walls (Figure 4.12). The official dimensions for each robot is 180 mm diameter circle and must be no higher than 15 cm (unless for the local vision variation where the robot has its own board and because of that can be greater). The game duration is 10 minutes and during that no human interference is allowed.

At the beginning of the 21st century, this league has earned the reputation of engineering league due to the need of high speed and precise control [SP01]. The main improvements in this league have been at the level of electromechanical and control systems, digital electronics and wireless communications [SP01].

Figure 4.12: Small Size Robot League

4.4.3 Middle Size Robot League

In the Middle Size Robot League many research topics are presented such as agents autonomy or local vision processing. In this new scenario two teams constituted by six robots in a 18x12 meters play a soccer game without any human intervention (except substitutions) and the duration of each half of the game is 10 minutes (Figure 4.13).

Figure 4.13: Middle Size Robot League

Initially this league was projected to present a perception and a locomotion simple task but also at the same time constitutes an exciting problem for its participants [Vel00] [Fed01].

Over the years many improvements have been detected. An example of that is the 2010 competition in which all teams were able to play with net goals only (eliminating the color coding yellow and blue respectively - previously used to identify the goals). The ball is the only object that is still color-marked. Also all teams are capable of establishing inter-team cooperations and receive all referee commands. Some teams like Cambada, were even capable of passing, dynamic role exchange and to execute simple SetPlays [LLCF09] [LLFC09].

4.4.4 Humanoid League

In the Humanoid League autonomous mobile robots with a human like appearance play soccer against each other (Figure 4.14). The competition is held in 3 classes: kidsize, teensize and

Figure 4.14: Humanoid League

adultsize. The number of players can vary (between 2-3) as well as its size, the size of the playing field, the ball size, etc. The match lasts two equal periods of 10 minutes (with an interval of 5 minutes).

The action takes place on a green rectangular carpet field which contains two goals, field lines and two landmark poles. Generically, the rules are similar to the human soccer with some restrictions (for example the number of players previously mentioned).

The major research topics presented in this league are dynamic balance, kicking the ball and visual recognition of the scene objects.

4.4.5 Standard Platform League

Until 2004 this league was called Sony Four-Legged League and in that challenge each team was constituted by 4 robots (AIBOERS210) constructed by Sony. No hardware modification was allowed and because of that the team with the most appropriated software normally wins the competition (Figure 4.15).

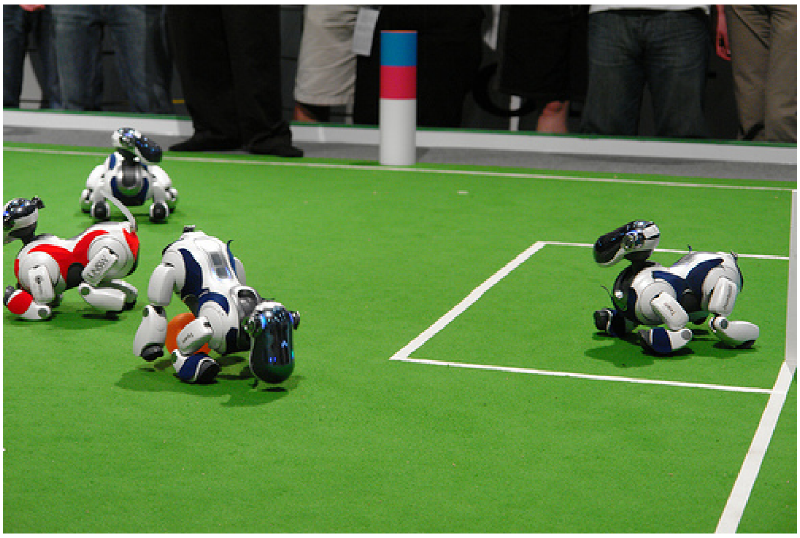


Figure 4.15: Sony Four-Legged League

These robots (with a dog like appearance) played in a 4x3 meters field and until 2002 the remote communication, global vision and wireless communication were not allowed.

Similar to the previous league, the duration of the Sony Four-Legged games was 20 minutes (10 minutes for each period). In the playing field 9 colors were used: the pink and green color for the markers, yellow and light blue for the markers and the goals, red and dark blue for the robots equipments, orange for the ball, light green for the field and for its lines. In this league the major research problems are: the vision problem due to the limited view angle and the oscillation effect caused by the legs locomotion of the robots; the localization problem due to the necessity of having image processing algorithms able to handle images created by cameras with three degrees of freedom.

Between 2004 and 2007 the AIBO remained the challenge platform. However as the league was no longer organized by sony, changed its name to Four-Legged League. After RoboCup 2007 the humanoid Aldebaran Nao became the platform used in the competition (Figure 4.16).

4.4.6 The RoboCup Rescue

The search and rescue in large scale disaster situations is one of the more relevant social problems. The effects of the Hurricane Katrina in New Orleans city (2005) or the effects of the torrential rains in Madeira Island (2010) are two examples of major natural disasters of our time.

The RoboCup Rescue project emerged in 1999 promoting research and development in many areas like: multi agent teams coordination, physical agent in search and rescue environments, information infrastructures and a realistic simulator. This application's domain was chosen due to its similarities with the soccer domain (dynamic environment, incomplete information with noise,heterogenous agents and the implementation of complete tasks. Some authors believe that due to these similarities some methodologies defined in soccer leagues can also be used in the rescue league [KTN+99] [CLR07a]. The main goals of the RoboCup Rescue are [SBK01]:

Figure 4.16: Standard Platform League

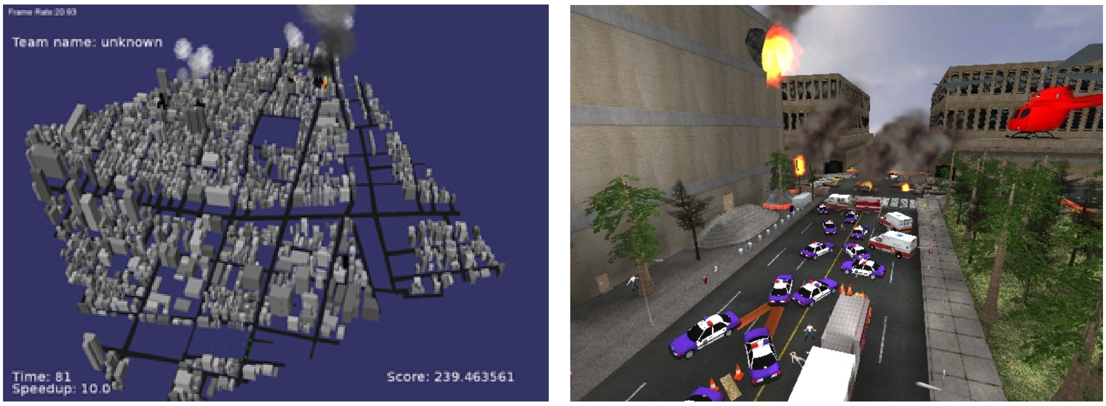
1. Development and application of advanced intelligent robotics and AI techniques in search and rescue situations;
2. Introduce new problems with social significance as robotics and AI challenges indicating new research directions;
3. Proposed infrastructures for further systems based on robotics and AI;
4. Promote new developments in this area through the RoboCup competition.

This league was constituted by two major projects: The Rescue Simulation League and Rescue Robot League. The simulation league is splited into two sub-leagues: Agent Simulation with the aim of develop simulators that form the infrastructure of the simulation system and emulate realistic phenomena predominant in disaster and Virtual Robots where a group of virtual intelligent and heterogenous agents like fire fighters, commanders, victims, volunteers conduct search and rescue activities in a virtual disaster world (Figure 4.17(a), Figure 4.17(b)). This league involves many AI/robotics research topics such as behavior strategy (multi agent planning, realtime/any time planning, agent heterogeneity, robust planning, mixed initiative planning).

The second project is the Rescue Robot League whose aim is to promote research and development in the search and rescue domain involving multi agent team work coordination, physical robotic agents for search and rescue, personal digital assistants, a standard simulator and decision support systems (Figure 4.18(a),4.18(b)).

4.4.7 The RoboCup@Home

The aim of this competition is to promote the development of research applications in the robotic domain that can assist Humans in everyday life (Figure 4.19). The scenario chosen for this compe-



(a) Agent Simulation (b) Virtual Robots

Figure 4.17: Rescue Simulation League (Agent Simulation - (a) and Virtual Robots -(b))

(a) Example of a Rescue Competition environment (b) Example of a Rescue Robot

Figure 4.18: Rescue Competition environment (a), Rescue Robot(b)

Figure 4.19: Robocup@Home

tition is the real world, more precisely the living room and the kitchen. However in the future this competition can cover other areas like a garden/park area, a shop, a street or other public areas.

4.4.8 The RoboCup Junior

RoboCup Junior is a project-oriented educational initiative for students up to the age of 19. Due to the RoboCup ambitions it is extremely important that children (young researchers) have contact with this reality. Unlike other leagues that were designed to produce research at the highest level by the respectively research labs, the goal in the RoboCup Junior is to present a suitable environment where a children using a generic robot hardware can easily program a robot and have a nice experience in the robotic reality. This league presented five different challenges:

1. **Dance Challenge** where young researchers create and develop dancing movements for the robots. These robots are dressed with dance costumes and move in creative harmony to the music (Figure 4.20);

Figure 4.20: Dance Challenge

2. **Soccer Challenge** where the young researchers must establish a strategy for autonomous soccer-playing robots (Figure 4.21);

Figure 4.21: Soccer Challenge

3. **Rescue Challenge** where a young researcher develop autonomous robots to rescue victims in simple disaster scenarios (Figure 4.22);

4. **CoSpace Demo Challenge** offers an opportunity to its participants to explore robotics technology, digital media and the CoSpace concept. It also provides a platform for researchers that have interests in animation and gaming. This challenge is the bridge between RoboCup

Figure 4.22: Rescue Challenge

Junior and RoboCup major Simulation and comprises two sub-leagues: CoSpace Adventure Challenge and CoSpace Dance Challenge (Figure 4.23(a), 4.23(b));

(a) CoSpace Adventure Challenge (b) CoSpace Dance Challenge

Figure 4.23: CoSpace Adventure Challenge (a), CoSpace Dance Challenge(b)

5. **RoboDemos Mini-Workshop** encourage young researchers to present new robotic projects developed at schools, clubs, community centers and other places offering new education perspectives.

4.4.9 The RoboCup Demonstrations

RoboCup demonstrations shows new ideas and concepts for future robotic competitions and research fields. These demonstrations included, in the last few years, among others:

1. **Festo Logistics Competition (FLC)** using a standard mobile robot platform *Robotino* with open interfaces. The focus of this competition is that, using a 6x6 meters arena, the autonomous guided vehicles must attempt to manufacture and deliver the maximum possible number of finished products. Competition takes place as follows: Teams first leave to discover the unknown functions of the 10 machines as quickly as possible, store their location in the production hall and, communicate this information to the other team members. The opponents will block the paths between the unfinished parts store and outgoing goods(Figure 4.24);

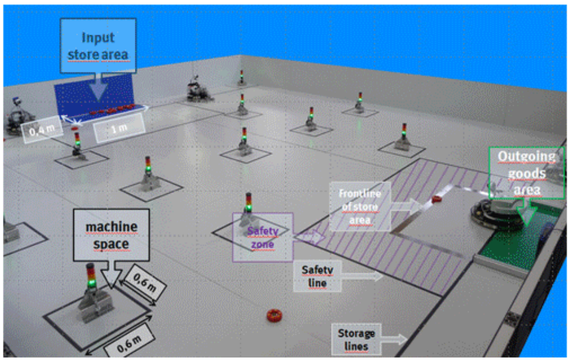


Figure 4.24: Festo Logistics Competition

2. **Mixed Reality Competition** is based on a soccer tournament and presents a mixed reality constituted by 2 cm tall robots with a virtual ball and environment. Te teams are constituted by five players each and the game duration is 20 minute (10 minutes per period) (Figure 4.25).

Figure 4.25: Mixed Reality Competition

4.4.10 Other Associated Challenges

In the next section three non active RoboCup challenges will be presented. The justification for choosing these challenges is related to the work presented in this thesis.

4.4.10.1 Coach Competition

In the simulation league, a coach is a privileged agent with the goal of helping its players through the game. For this competition two different types of coaches are available:

1. **Online coach** which is used to send reliable information to the players during the game. However its capacities were very restricted. Because of that this agent only communicates with its players when the game is stopped or during the game (using coded messages that come to players with delay). The main goal of the coach is to detect the play pattern(s) of

the fixed-opponent in each game and report it. The opponent's strategy is not the same in every game and to avoid random reporting the RoboCup organization stablished a maximum reports number (which corresponds to the number of strategies used for the opponent in each game). Before the beginning of the competition teams can submitted patterns, however the patterns that are not predictable and exploitable will be rejected by the organization.

2. **Offline coach** is used only to prepare a team for a specific opponent (its use in competition is not allowed). This coach can control the game mode, assign speeds and guidance to the players which is very useful for testing certain situations during practice.

Both of them received all the messages sent by the referee and the players and global environment informations without any noise. Also this type of agent is capable to send audible feedback to its players. However some restrictions are imposed by the server.

To avoid misinterpretation from both the players and the coach, a standard language is necessary which presents a clear semantics and easy to use. The development of such language constitutes a new research area.

The first language created was the Coach Unilang [RL02]. This language was based in common robotic and human soccer concepts like: regions, time periods, situations, tactics, formation, player behaviors, etc. However the RoboCup Federation adopted a more low-level language called CLang [CFH+03] that initially imported the basic Coach Unilang concepts. Throughout the years, the substantial concepts difference presented by both languages disappeared, as Clang approached the high-level nature of Coach Unilang.

4.4.10.2 3D Monitors

Over the years, the development of 3D viewers for many RoboCup competitions constitutes a stimulating research challenge. The preferential domain for this kind of applications is the robotic soccer. However, in the last years specially after the rescue competition emerged, the development of 3D viewers has been generalized. Some development viewers will be further analyzed:

1. **The Virtual RoboCup viewer** was developed by Jung et. al. in 1999 [JOH00]. It was the first three-dimensional viewer to be implemented for the simulated robotic soccer, running on SGI (Silicon Graphics, Inc.) graphic stations supported by Irix6 operating system (Figure 4.26).

   The researchers main difficulty consisted in creating a tridimensional animation through the two-dimensional data provided by the soccer server, maintaining the players animation more realistic as possible when they kick the ball. To achieve this goal the researchers have created many intermediate stages in the animation process using the keyframe technique.

2. **The Magic Box viewer** was developed in OpenGL [XLL+02] [LHSC03]. The animation of this viewer was based on the players actions and, to turning it more realistic, an extra parameter was added in the ball animation (height). This viewer supports different camera

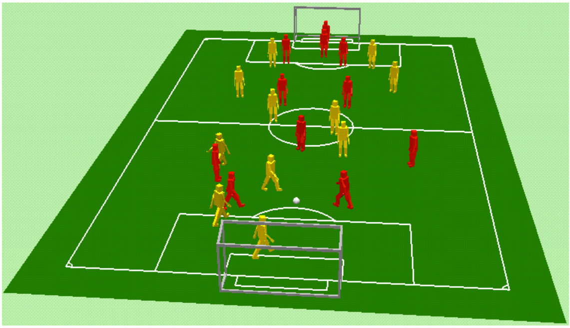
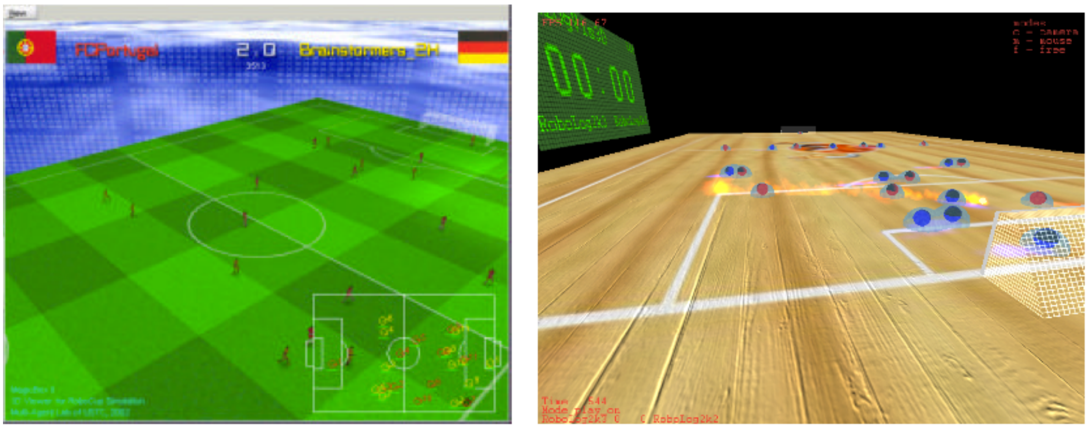


Figure 4.26: Virtual RoboCup Viewer of the Bielefed, Germany University

angles of the game and was capable of calculating simple final game statistics. Finally an intelligent commentator system is also available in this viewer, able to report all actions that are happening in the game (Figure 4.27(a)).

3. **The Robolog viewer** used a generic tridimensional framework *Tool.In* to represent three-dimensional simulations that typically use a graphics architecture scene, modeling the objects aspects in the virtual world and its interactions with the environment directly [OR02]. This viewer supports an automatic or a interactive, camera control mode. In the second control mode a user can use the keyboard to move the camera to a specific action point. In the first control mode the camera is always focusing on the entire game play [BAOR03] (Figure 4.27(b)).



(a) Magic Box Viewer (b) RoboLog Viewer

Figure 4.27: Magic Box Viewer (a) and RoboLog Viewer(b)

4. **The Venue viewer** allows the RoboCup visualization in three different modes: monitor, Workbench and Cave mode [SRG+00] [SRG+99b] [SRG+99a].

   The monitor mode allows a tridimensional visualization of a soccer match. This mode uses multiple cameras positioned on the playing field with the selection of the view camera (an algorithm based approach) follows the game action (basically consists in tracking the ball).

The Workbench mode covers the simulated game through a 3D perspective with just a single fixed camera. Although in terms of size, the perspective presented by this mode is smaller than the previous analyzed mode, the viewer has a full perspective about the movements of the different objects (ball and players) in the game. Finally the Cave mode (the most innovative) allows the user to be immersed (camera linked to a player) in the game and interact with it. This mode uses the same information and communication as the original 2D monitor viewer, but now views the state of play in a CAVE virtual scenario. The animation of the players is based on the OpenGL utility where their movements are carried out through a set of basic points that are interpolated by splines in order to smooth the animation. The possible moves of the players are standing, walking and running, which are adjusted linearly interpolating between the three modes. The Cave mode also allows the user interacts with the soccer server. Using three devices (glasses, joystick, device attached to the foot of the human player) it is possible for a human player to simulate a running, turning or even a kicking action and sending this command to the server [RBGS00]. At this time, despite of the fact that the idea was vey interesting, the performance of the system was not so good. In recent times a new project that uses the same idea as the base was the Wi Project (that will not be addressed in this thesis) (Figure 4.28).

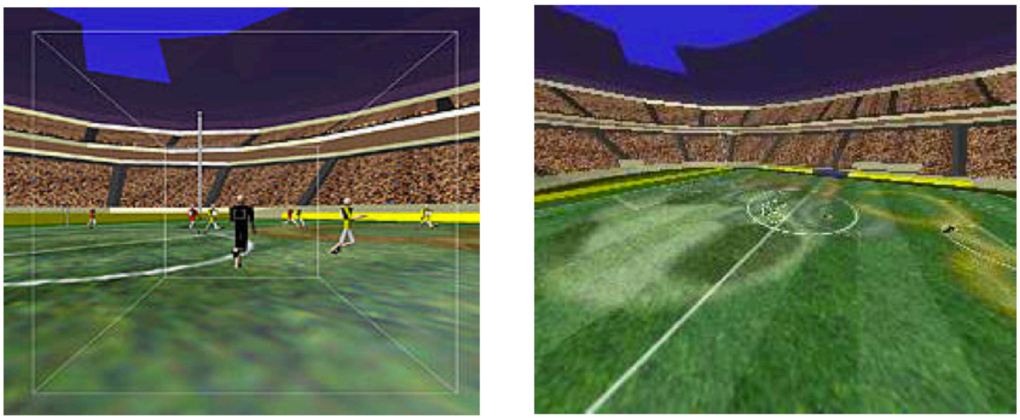


Figure 4.28: Venue Viewer

5. **Caspian viewer** was divided into three modules: 3D viewer, commentator and soccer analysis tool [SGM+03]. The main goal of the viewer is to develop a system that resembles reality in what concerns to viewing a soccer game.

   Similar to other analyzed viewers, Caspian uses a set of cameras spread through the soccer field and two control modes: an automatic and a manual mode. For the first mode the better perspective of the game is selected in two steps: in the first step the user (director) selects the camera through a heuristic based on the game mode, ball and players position. On the second step the agent decides which is the camera status (based on position and ball velocity) (Figure 4.29).

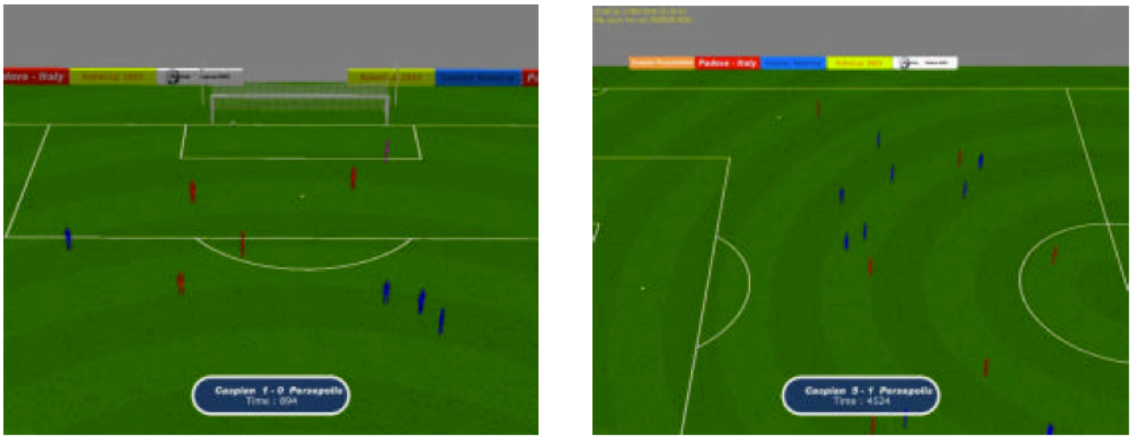


Figure 4.29: Caspian Viewer

6. **RA3DM viewer** (RoboCup Advanced 3D Monitor) is a a tridimensional application that allows the visualization of a soccer game. The aim of this tool is to give the users a realistic experience of a soccer match in three dimensions while maintaining a healthy balance through a realistic animation and special effects [PPaNC04].

   The RA3DM system is composed by three main animations executed by the players: walk, run and kick. The animation process is simplified using the keyframes interpolation technique. Similar to other tools, RA3DM presents an automatic and manual camera control (Figure 4.30(a)).

7. **Avan** is an integrated system composed by three modules: a soccerplayer, a coach and a viewer. The viewer is composed by two modules: tridimensional viewer and analysis tools / debugging (Figure 4.30(b)).

(a) RA3DM Viewer

(b) Avan Viewer

Figure 4.30: RA3DM Viewer (a) and Avan Viewer(b)

The graphical system was developed in OpenGL and their modules were developed in 3D Studio Max. The 3D viewer features are similar to the ones previously analyzed.

8. **MAICC–Muti Agent Intelligent Camera Control** provides a virtual 3D viewer with a pronounced three-dimensional realism of the animations and sounds, plus a Multi-Agent System for intelligent control of the camara which allows the viewing of games in real time or deferred [Lou04]. This system allows automatic control and intelligent filmmaker and a set of cameras (pre-positioned in the scenario) to give the viewer the best image to illustrate the scenario as if it were a television coverage. Thus, the objectives of MAICC focus on the implementation of a decentralized autonomous prepared for treatment of a film with characteristics of adaptability during the soccer game that is being simulated (Figure 4.31). The results confirm a better quality of viewing in relation to traditional viewers, allowing

Figure 4.31: MAICC Viewer

the conclusion that the coordination between agents plays a key role in the filming of the regions of interest to the scenario attached to a synchronization of plans for filming

4.4.10.3 Intelligent commentators

The development of game analysis system and intelligent commentators (using natural language) constituted an interesting research challenge.

Using a commentator system, it is possible to generate real-time reports for arbitrary matches of the RoboCup simulation league. In this particular point, three approaches will be analyzed – Byrne, MIKE and Rocco [ABTi$^+$00]. All three approaches use as input data the same information the RoboCup Soccer Simulator Monitor receives for updating its visualization, such as:

- Player – location (Cartesian coordinates within the field), orientation (body and head orientation), energy parameters (stamina, effort, recovery) and viewing capabilities (width and quality), among others;
- Ball – location (Cartesian coordinates);
- Game – current simulation cycle, game state (throw-in, offside, corner, goal kick and others), team names and current score.

Looking more throughly at each approach, the Byrne system generates appropriate affective speech and facial expressions, based not only on game analysis data, but also on the character's personality, emotional state, commentator's nationality or the team it supports, among other types of information, and uses a face model as an additional means of communication [BLB98].

The MIKE (Multi-agent Interactions Knowledgeably Explained) system [TINF+98] is a real-time commentator system that supports three distinct languages: English, Japanese and French. The main capability of this system is to identify interactions between players in order to classify team behaviors and to generate predictions concerning the short-term evolution of a given situation.

The Rocco (RoboCup Commentator) is a continuation of a research project that appeared in the 1980's called Soccer [AHR88], which, using natural language, tried to interpret a scene in a restrict domain. Rocco is a TV-style live commentator which uses the RoboCup Soccer Simulator Monitor combined with an emotional spoken description of a specific scene.

A comparison between these three systems is illustrated in Table 4.1.

Commentor System	Analysis	Natural Language Generation	Output
Byrne	Obervers' recognize events and states	Templates, marked up for expression and interruption in real-time	Expressive speech and facial animation
MIKE	Events and states (bigrams, voronoi)	Templates, interruption, abbreviation using importance scores	Expressive speech
Rocco	Recognition automata	Parameterized template selection + real-time nominal-phrase generation	Expressive speech

Table 4.1: Comparison Between Commentator Systems (adapted from [ABTi+00])

4.5 The RoboCup Soccer Simulator

The RoboCup simulation league is based on the soccerserver [NNMH97] which simulates a virtual soccer game played by two teams with eleven players each. The simulator was built as a multi agent simulation environment and with real time features allowing the competition between two virtual teams. There is only one type of object in this reality *GameObject* which is part of the game (Figure 4.32). This object can be one of two types: A line of the field (*Line*) or a *Field Object* that is divided in *Mobile Objects* like the ball or the players and *Stationary Objects* like markers that are in the play field.

In this section all the information regarding the implementation of the soccer environment through the soccer server will be presented. It is important to note that information presented in this section is mostly based on the last official server documentation [CFH+03].

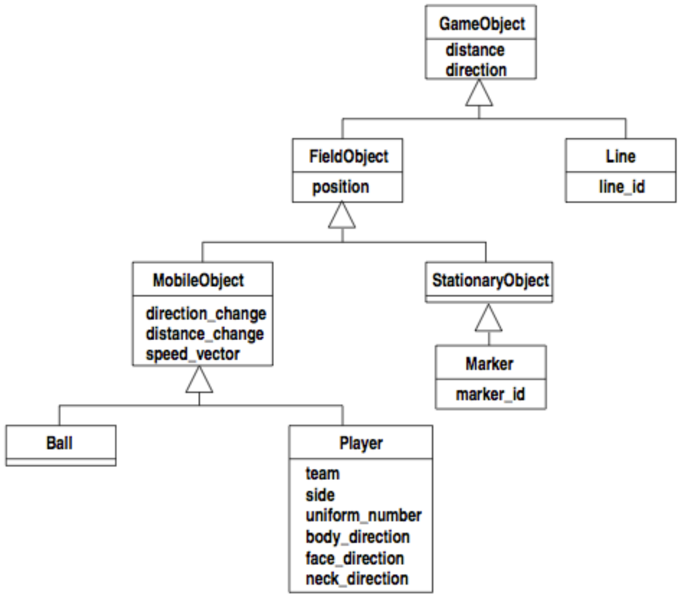


Figure 4.32: UML diagram of the objects in the simulation [CFH+03]

4.5.1 Simulation System

The soccer simulator system is composed by 3 main modules: a server (soccer server), a visualizer (soccer monitor) and a video (logplayer). Using a client/server architecture the simulator is responsible of executing the game simulation. Using pre-stablished rules, this simulator provides a virtual soccer field and simulates all the mobile objects movement (ball and players). The clients (players) connect to the simulator through a UDP socket connection. The simulator receives the clients' commands, executes them (simulating the respective objects movement on the game field) and sends sensorial information to the clients.

The monitor is a tool that allows the visualization of virtual games. This tool communicates with the simulator through UDP sockets receiving all the players and ball positions. After that displays these information in a graphical mode allowing their visualization by the human.

The log player allows the game visualization at any time through a log file. This file is generated by the server at the end of each soccer game.

4.5.1.1 The Soccer Server

The soccer server allows the execution of a virtual soccer game played by two teams of virtual autonomous agents. As this tool is based on a client/server architecture there is no restriction in what concerns the development language or operation systems involved in team development process (the only restriction is that each team must support a UDP/IP communication) (Figure 4.33).

Each team is constituted by 11 players (clients) and eventually a coach (client with special capacities and privileges). Each client separately connects to the server using a specific port. For playing a game, the client sends a message through the port 6000 to the server and as an answer

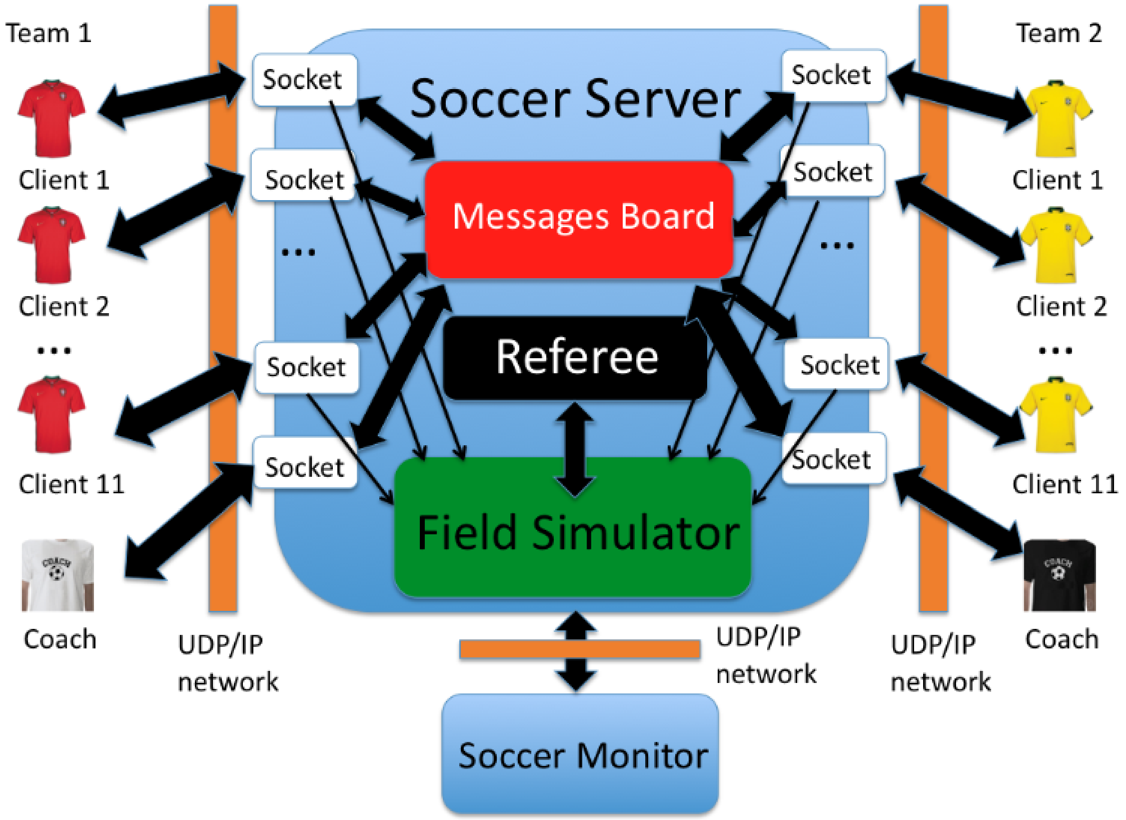


Figure 4.33: Soccer Server Architecture

the server must accept the connection attributing a specific connection port to the client (after this moment the client and server will always change their messages through this port).

The server works with discrete times or cycles (1 cycle corresponds to 100 milliseconds) and at the end of each cycle (after exchanging information with its clients) the server updates its environment and virtual field.

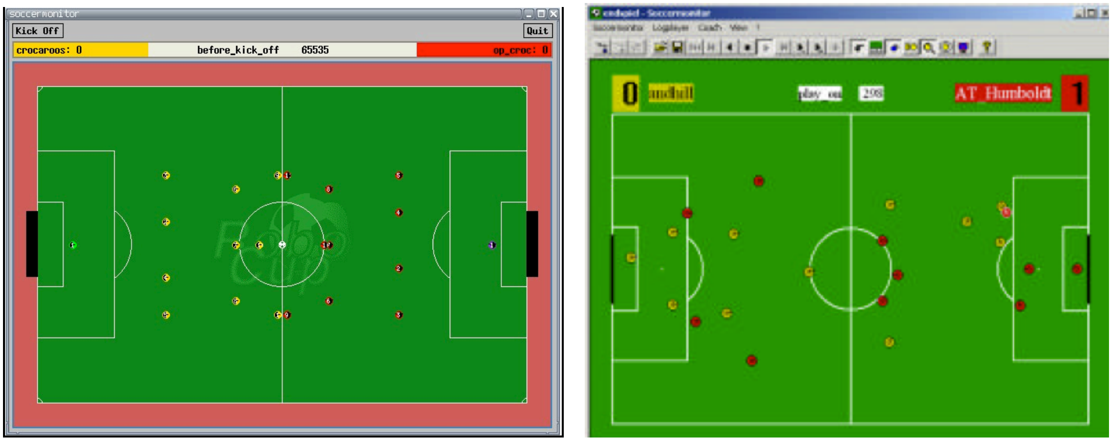
4.5.1.2 The Soccer Monitor

The soccer monitor is a graphical tool that through the communication with the server (using sockets with UDP protocol), is capable of representing all the mobile objects movement existing in the virtual environment. Through the simulator modular architecture it is possible at the same time to have more than 1 monitor connected to the server (that can be for instance a monitor and a game analysis system).

The traditional soccer monitor supports many objects configuration such as changing the ball or players color and also allows the configuration of the visualized message regarding their size and sources (Figure 4.34(a)).

Similar to traditional soccer monitor (in terms of supported features), and one of the first monitors for the windows operating system, was the Klaus Doer monitor [Dor00] (Figure 4.34(b)).

Through the years this monitor was being used by many universities in several research projects related to the RoboCup [CCYL02].



(a) Traditional Soccer Monitor (b) Klaus Doer Monitor

Figure 4.34: Soccer Monitors

At the beginning of RoboCup competition the traditional monitor was used in official games. However, since 2001 another monitor is used – the Frame View. This new tool developed by the Kalsruhe University supports new features such as different enlargements in field regions, information about player stamina, information related to the vision of a particular player, among others giving the user a better analysis about the game (Figure 4.35).

Figure 4.35: Frame View Monitor

In the chapter 7 of this thesis other visualization tools capable of calculating higher level game information (like final game statistics) will be presented. Because of that these tools were excluded from this section.

4.5.1.3 The Log Player

The log player is an application that allows a visualization of a recorded game. Similar to a conventional video record, this tool uses the information storage in a file (log file) and represents it on a monitor allowing the user a better analysis about the strengths and weakness of the different participant teams in the RoboCup challenge.

The log player supports many features such as:advance actions for a given cycle, forward, rewind and stop actions among others.

4.5.2 The Simulated World and the Game Rules

In this section a description about the execution of the robotic simulated game including the game rules and the world simulated dynamics will be performed. In order to better understand better the exposed concepts it is necessary to possesses a basic knowledge about the human soccer and its rules.

4.5.2.1 The Game Field and its Objects

The virtual robotic soccer field has the official human soccer field dimensions (105*68 m) containing several lines to help players in their localization tasks. Figure 4.36 illustrates the virtual field with its lines and localization marks. The *(goal r)* and *(goal l)* flags correspond to the center

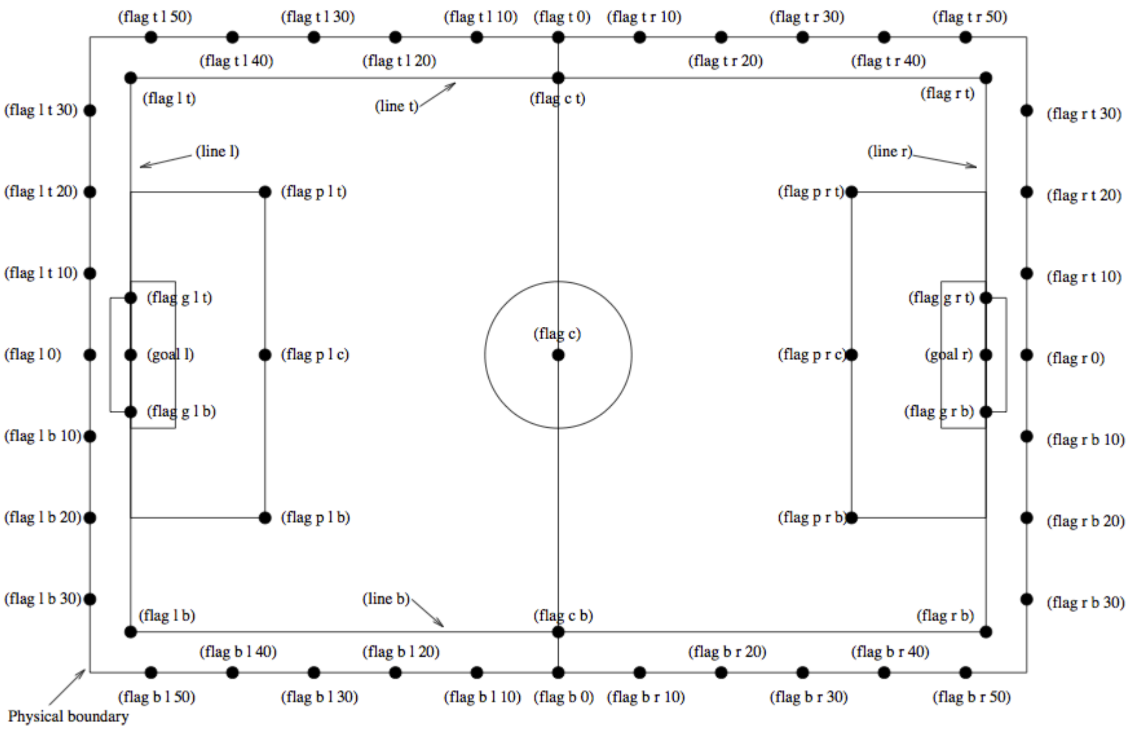


Figure 4.36: The flags and lines in the simulation ([CFH+03])

of the left and right goal line, the *(flag g l b)*, *(flag g l t)*, *(flag g r t)* correspond to the two virtual goal posts. *(flag c)* corresponds to the virtual flag located in the middle of the field, etc. Each flags has its own label that is used in its identification in the communication process between player and simulator.

The duration of the simulated robotic game is 10 minutes (6000 cycles) divided in two identical periods of 3000 cycles. In case of a draw situation in the regular game time (in a decision match like for instance in the quart finals of a tournament) and similar to the human soccer, an extra time period begins.

4.5.2.2 The Artificial and Human Referees

In simulated robotic games a virtual referee is responsible for carrying out a set of predefined rules. Although the virtual referee does not need the assistance of a human referee, in obstruction and anti game situations its task becomes very difficult. To solve that problem the soccerserver provides an interface that allows a human referee to execute one of two actions: mark a foul to a particular team at a given field point or run a dropped ball to the ground at a specific point field.

During the years, the researchers improved the virtual referee features turning this agent in a more autonomous agent. Today, this referee is capable to mark free kicks in situations where a player passes the ball to himself or throws the ball to the ground when a team spent too much time to put the ball back on the pitch (for instance in a throw in situation). More informations about the virtual referee can be found at the soccer server manual [CFH+03].

4.5.3 Server Communication Protocols

To play a simulated game, the players (clients) must possess certain capabilities to follow a set of communication server protocols. In this section an overview of the connection, action and perception protocols will be performed.

4.5.3.1 Connection Client Protocols

The connection protocol allows the client to connect, reconnect or disconnect to the server (Table 4.2). It is important to note that the soccer server supports several clients versions varying the parameters used in its messages.

From client to server	**From server to client**
(initi TeamName [version VerNum)] [(goalie)]) TeamName :: = (-|_|a-z|A-Z|0-9)+ VerNum :: = the protocol version (e.G. 7.0)	**(init Side Unum PlayMode)** Side :: = l | r Unum :: = 1~11 PlayMode :: = one of play modes **(error no-more-team-or-player-or-goalia)**
(reconnect TeamName Unum) TeamName :: = (-|_|a-z|A-Z|0-9)+	**(reconnect Side PlayMode)** Side :: = l | r Unum :: = 1~11 PlayMode :: = one of play modes **(error no-more-team-or-player) (error reconnect)**
(bye)	

Table 4.2: Client Command Protocol ([CFH+03])

4.5.3.2 Client Perception Protocol

The client perception is divided into three groups: hear perception, seen and sense body. The client sensor protocol is very simple due to the fact that it is not expected that a client sends a message to the server after receiving sensorial information.

4.5.3.3 Client Action Protocol

The client action protocol defines the command messages syntax that the clients are allowed to send to the server and the possible server answers (Table 4.3). Normally the server executes the

<table>
<tr><th>From Client to Server</th></tr>
<tr><td>(dash<Power> <Dir>) <Power>:: = [-100, 100] <Dir>:: = [-180, 180]</td></tr>
<tr><td>(turn <Angle>) <Angle> ::= [-180, 180]</td></tr>
<tr><td>(move<PosX><PosY>) <PosX> ::= [-52.5, 52,5] <PosY> ::= [-34.0, 34.0]</td></tr>
<tr><td>(kick<Power><Direction>) <Power> ::= [-100, 100] <Direction> ::=[-180, 180]</td></tr>
<tr><td>(tackle<Power>) <Power> ::= [0, 100]</td></tr>
<tr><td>(catch <Direction>) <Direction> ::= [-180, 180]</td></tr>
<tr><td>(turn_neck<Direction>) <Direction> ::= [-180, 180]</td></tr>
<tr><td>(change_view <ViewWidth>) <ViewWidth> ::= Narrow | Normal | Wide</td></tr>
<tr><td>(attentionto<TEAM><UNUM>) | (attentionto off) <Team> ::= opp | our | l | r | left | right |
<TEAM_NAME> <UNUM> :: = TEAM_MEMBER_ID</td></tr>
<tr><td>(say <Message>) <Message> ::= Text</td></tr>
<tr><td>(pointto <Distance><Direction> | (pointo off) <Distance> ::= Integer <Direction> ::= [-180, 180]</td></tr>
</table>

Table 4.3: Client Action Protocol ([CFH+03])

action sent by the clients without any answer. The server only sends a client answer in three situations:

1. When the client sends a message with an unknown command or with illegal parameters (in this case the server will answer with an error message);

2. When the client sends the command **score** - in this situation the server will answer with the time and the game result;

3. When the client sends the command **sense_body** - in this situation the server responds with physical sensorial agent information.

4.5.4 Agent Perception

As previously mentioned in simulated soccer, the agents have three types of sensors:

1. **Hear Sensors** that detect the messages sent by the referee, coach, teammates and opponents;

2. **Visual Sensors** that detect visual information available that include distances and directions of objects and players who are in the agent field of view;

3. **Physical Sensors** that detect the agent's state including its energy, velocity and neck angle.

Using these three types of information, the agent is capable to construct a clear image about the game events.

4.5.4.1 Visual Information

The visual information occupies a major role in simulated soccer. With this information the agent is capable of calculating the distance and directions of the objects that are in its field of view in a certain moment. The agent can choose two types of mode : synchronous and asynchronous (almost unused). In the synchronous mode, the low view quality cannot be used and three view width are available (Table 4.4): In all view modes, rcssserver send "see messages" from the beginning of the

Mode	View Width (degree)	See Frequency
Narrow	60	every cycle
Normal	120	every 2 cycles
Wide	180	every 3 cycles

Table 4.4: View Widths available in the visual information

cycle. Each view width can be calculated as follows (4.1):

$$\begin{aligned} &\textbf{narrow:} = 60 = visible\_angle(90) * send\_step(150)/sim\_step(100) \\ &\textbf{normal:} = 120 = narrow * 2 \\ &\textbf{wide:} = 130 = narrow * 3 \end{aligned} \quad (4.1)$$

Figure 4.37 illustrates the visual perception of the agents. The agent is represented using two cir-

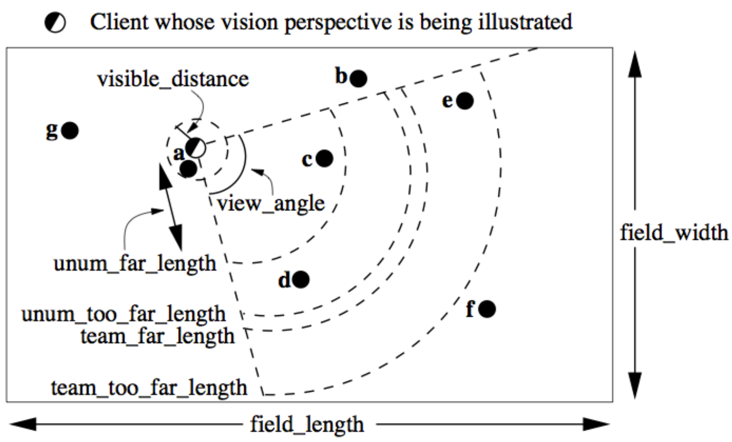


Figure 4.37: The visible range of an individual agent in the soccer server ([Sto98])

cles (the most lighter one corresponds to the agent's front). The dark circle corresponds to the other field agents. Only the objects that are inside of the vision angle (view_angle) or a certain distance (visible_distance) are visible. The other parameters (unum_far_length, unum_too_far_length, team_far_length and team_too_far_length) affect the precision and the visualization type information sent to the agent.

The information sent by the server can be summarized as follows:

1. **Static Objects** (flags, lines and goals) – Object name, relative distance and direction;
2. **Players** – The information depends on the distance the observer has to a player;
3. **Ball** – Similar to the player - The information depends on the distance the observer has to player.

4.5.4.2 Aural Information

The aural information has great importance because all referees send messages of this type. Also the players can use these messages with their teammates and coach (if there is one). The format of these messages is:

(**hear** <Time> <Direction> <Team> [UNUM] <Message>) or (**hear** <Time> <Team> [UNUM])

where: <Time>::integer <Direction>::real <TeamName>::string [UNUM]::self | referee | online coach left | online coach right | <Direction> <Message>::=string which <Time> indicates the current game time [UNUM] can be self, referee, online_coach_left or online_coach_right and <Direction> the relative direction of the player who sent the message.

The simulator possesses many different kinds of parameters. The main ones allow the configuration of the maximum communication distance (audio_cut_distance), the maximum message number that is possible to hear in a specific cycle interval (hear_max, hear_inc and hear_decay) and the maximum message size (say_msg_size).

Beyond the audition capacity limitation, the server also implements a spatial message range limitation. In consequence, in order to hear the message, players must be at a maximum distance (audio_cut_dist) from the emitter.

4.5.4.3 Physical Information

Beyond the visual and auditory perception the simulator sends to the agents physical perceptions (related to the agent's state). This information is automatically sent by the players in each sense_body_step. The message format for the physical information is:

(**Sensebody Message**

(arm (movable <MOVABLE>) (expires <EXPIRES>) (target <DIST> <DIR>) (count <COUNT>))

(tackle (expires <EXPIRES>) (count <COUNT>)) (focus (target none) (count <COUNT>))

(focus (target l | r <UNUM>) (count <COUNT>) (collision none | [(ball)][(player)][(post)])

(stamina Stamina Effort) (speed AmountOfSpeed DirectionOfSpeed) (head angle HeadAngle) (kick KickCount) (dash DashCount) (turn TurnCount) (say SayCount) (turn neck TurnNeck-Count) (catch CatchCount) (move MoveCount) (change view ChangeViewCount))

Where:

1. **<MOVABLE>** is the number of cycles till the arm is movable;
2. **<EXPIRES>** is the number of cycles till the arm stops pointing;
3. **<DIST> and <DIR>** are the distance and direction of the point that the player is pointing to, relative to the players location, orientation and neck angle;
4. **<COUNT>** is the number of times that the command has been successfully executed by the player.

4.5.5 Agents Action

In the RoboCup simulation league, agents have many parameterized commands turning the list of actions available to the agent almost infinite. A set of these commands is listed in Table 4.3. These actions can be divided in 4 types: movement (dash, turn and move); interactions with the ball (kick, tackle and catch), perception control (turn_neck, attentionto, change_view) and communication (say, pointto).

4.5.5.1 Movement – Move

The simulator allows 3 movement commands:

(dash<Power><Dir>)

(turn<Dir>)

(move<PosX><PosY>)

in which <Power> $\subset$ [-100, 100] and <Dir> $\subset$ [-180, 180], <PosX> $\subset$ [-52.5, 52.5] and <PosY> $\subset$ [-34.0, 34.0]. The movement command can be used to position a team in a field or to allow the goalkeeper to move to a specific position within its penalty area (after catching the ball).

4.5.5.2 Movement – Acceleration and Energy

The dash command is used to accelerate the player in any direction (with a certain power). The simulator does not allow the players to move always with the maximum speed (player_speed_max) and because of that the simulator gives the player limited energy (stamina). When player's stamina is recovered during the game, his stamina capacity is also consumed. If the player's stamina capacity becomes 0, his stamina is never recovered and he can use only his extra stamina.

4.5.5.3 Movement – Rotation

The **turn** command is used to change the direction of the player's body. To turn this concept more realistic the inertia definition was added to the simulator. This way, the player rotation angle is not equal to the moment but depends on the player's velocity in a given moment. When a player moves with greater speed it is more difficult to accomplish the rotation due to inertia effort (equation 4.2):

$$\begin{aligned} Real\_angle &= Moment * (1.0 + Noise)/(1.0 + inertia\_moment * |v\_t|) \\ Noise &= Random(-player\_rand, player\_rand) \end{aligned} \tag{4.2}$$

where the inertia_moment is the simulator parameter, Noise is a random value taken from an uniform distribution [-player_rand, player_rand] * $|v\_t|$ corresponds to the absolute value of the players velocity. As the player cannot execute a dash and a turn command in the same cycle, its maximum velocity when executing a turn command will be player_speed_max * player_decay. The introduction of the previous restrictions limitations like the noise addition, limited energy and inertia changes completely the problem's complexity.

4.5.5.4 Ball Control – Kick

The simulator supports three types of ball control commands:

(kick <Power> <Direction>)

(catch <Direction>)

(tackle <Power>)

A player uses the **kick** command to kick the ball with a certain power in a specific direction. The **catch** command can be only performed by the goalkeeper to catch the ball in a certain direction. Finally the **tackle** command is used when a player wants to lay down and kick the ball with a given power in his body direction.

When a player sends a **kick** command he most provide two parameters:

1. The kick power which will determine the ball acceleration between [minpower, maxpower];
2. The kick direction which must be given in degrees.

When a kick arrives to the server, it can only be executed if the ball is closer to the player (related to the maximum kick distance). This distance is defined as ball_size + player_size + kickable_margin. In other words, if the distance between the ball outside and the player outside is less than kickable_margin. Heterogeneous players (described below) will have different kick capacities in terms of kickable_margin. The performance of this command execution is related to the relative ball and player position. If the ball is near and in front of the players, this command will be executed with more efficiency (equation 4.3):

$$FinalPow = Power * kickpowerrate * (1 - 0,25 * BallDir/180 - 0.25 * BallDist/kickablemargin) \tag{4.3}$$

in which Ball_Dir and and Ball_Dist are the ball direction and ball distance related to the player's body respectively. Also there is a parameter kick_rand that adds noise to the the player's kick. For normal players the value of this parameter is 0 however, for heterogeneous players this value varies according to their type.

4.5.5.5 Ball Control – Catch

The goalkeeper is the only player with the capacity to execute a **catch** command in order to catch the ball. The only parameter of this command is its direction. When the ball is in the catchable area the goalkeeper catch probability will substantial increased. Figure 4.38 represented that area.

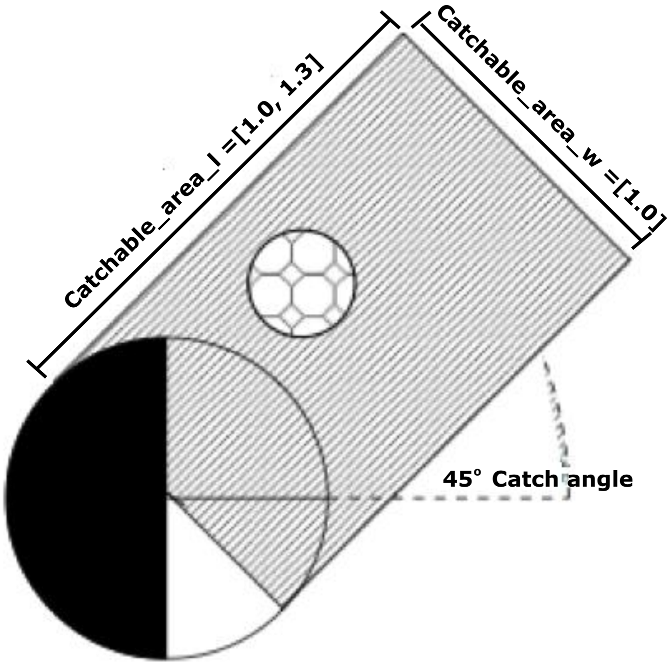


Figure 4.38: Example of the Goalkeeper Catchable Area (adapted from [CFH+03])

4.5.5.6 Ball Control – Tackle

The main goal of the introduction in the competition of this command is the improvement of realism in the 2D simulation league. The command format is:

(tackle <Power>)

The tackle effects correspond to kicking the ball in the player's direction with a certain power specified in the command. However, the success of the execution of this command is not guaranteed and the success probability increases with the decrease of distance between the executed tackle player and the ball. After executing this command, a player freezes for 10 cycles.

4.5.5.7 Perception Control – Flexible Neck Rotation

Using the **turn_neck** command the players can rotate their neck obtaining an important control about their perception. The command syntax is:

(turn_neck <Angle>)

This command modifies the player's neck direction and consequently its vision angle relatively to its body angle. If the angles passed in the command parameter is not valid, the command will be adapted to allow neck direction relatively to its body (remain within limits). Also there is no noise associated with this command.

4.5.5.8 Perception Control – Vision Configuration

The agent's visual perception can be obtained through the **change_view** command with the following syntax:

(change_view <Width>)

Similar to previously analyzed commands the cone vision opening can take the following values [narrow, normal or wide]. The players can use this command to improve the information sending by the server.

4.5.5.9 Perception Control – Audio Configuration

The command **attentionto** introduced new challenges in the competition. Through this command it is possible to choose which are the agents that have precedence to listen to information in each instance. The command syntax is:

(attentionto <Team> <Unum>) | attentionto off)

where <Team> :: Opp | our | l | r | left | right |; <Team_Name> allows the identification of the team that owns the player and <Unum> is the identification number. Each attentionto command overrides the previous one.

4.5.5.10 Communication Say

The communication model of the soccer server is based on the possibility of sending messages to the environment which are distributed by the closest agents. An agent can send a message to other agent using the say message with the following format:

(say <Message>)

The unique parameter is related to the message that the player wants to send.

4.5.5.11 Communication Pointto

The player has the capacity to visually communicate to other players using the **pointto** command with the syntax:

(pointto <Distance><Direction>) | (pointto off)

A player after pointing to a certain position remains pointing for a period between 5 to 20 cycles. The other player receives the information in the form of:

(p<Team><Players>)<Distance><Direction><VarDist><VarDir><BodyDir><HeadDir><PointDir>
or

(p<Team>)<Distance><Direction><PointDir>

In this command there is noise and it varies according to the distance between the player that is pointing and the one that is receiving the information.

4.5.6 Heterogeneous Agents

To stimulate the researchers in other domains such as the dynamic resources allocation, the heterogeneous agents (there are 18 different types randomly selected at the beginning of each game) were created and possess different characteristics (in comparison with the normal/standard players) regarding velocity acceleration, energy recover and distinct inertia moments. Table 4.5 summarizes some of those characteristics.

Parameter	Value	Range	Description
player_speed_max	1.0	[1.0,1.2]	Player with more speed
stamina_inc_max	45.0	[25.0,45.0]	Player who recovers less energy in each cycle
player_decay	0.4	[0.4,0.6]	Player whose speed decreases more slowly
inertia_moment	5.0	[5.0,10.0]	Player that turns with more difficulty in movement
dash_power_rate	0.006	[0.006, 0.008]	Player who accelerates quickly
player_size	0.3	[0.1, 0.3]	Player with a smaller body
kickable_margin	0.7	[0.7,0.9]	Player with a largest kick area
kick_rand	0.0	[0.0,0.1]	Player with less kick precision
extra_stamina	0.0	[0.0,100.0]	Player with some extra energy
effort_max	1.0	[0.8,1.0]	Player that comes more easily in an effort situation
effort_min	0.6	[0.4,0.6]	Player who comes out more easily from an effort situation

Table 4.5: Heterogeneous Players Parameters ([CFH+03])

4.6 Conclusions

Without any doubts the RoboCup has stimulated the research in DAI and intelligent Robotic areas. The use of a well known research domain (soccer) led to even more researchers to participate in this competition. The simulation league (analyzed more deeply in this section) includes many research tasks like MAS communication, location and perception issues among other. Some of this domain characteristics includes [RL01]:

1. Real-Time Simulation – The simulator updates the world into discrete time intervals (cycles of simulation), each with a duration of 100ms. During this time, agents receive different types of sensory information and have to send requests for enforcement action to the simulator;

2. Realistic Energy model – In order to represent real time features, the simulator presents a realist energetic player model that prevents the player from running the whole game. As previously mentioned, when a player executes a dash command spends energy however, the player has capacity to recovers energy;

3. A Huge set of low level skills – An agent possesses two types of actions: primary (kick, dash, turn, tackle, catch and move) and secondary (turn_neck, change_view, attentionto,pointto and say). In each cycle the agent can only execute one of the primary commands;

4. Heterogeneous Agents– The introduction of heterogeneous players constitutes the addition of new important challenges. Before the beginning of the game, each team has the possibility to choose the type of each of its players. This selection will definitely influence the strategy of the team during the game;

The main goal of this chapter is to present an overview about the domain application used in this thesis. In what concerns to the human soccer a brief historical description was done regarding the evolution of the game through the years. On the other hand, in the Robotic reality an overview about the RoboCup competition was performed including its leagues (those who were conducted at latest edition of RoboCup and some leagues already extinct) as well as some technical details about the soccer server. In following chapters soccer server is exposed as supported of this thesis specially related to the experiences between robotic soccer teams.

Chapter 5

Tracking Systems

Soccer is one of the CSG with more participants and supporters all over the world (it is played by over 240 million players in 1.4 million teams and 300 thousand clubs around the world) [AYA+08]. One of the game's characteristics is that the players have the possibility to make real time choices, having the restrictions imposed by the rules, defined in a training session or in a professional competition, always in mind. This point is an element of freedom and consequently creativity. This reality should be managed by another individual called Coach whose purpose is to train the team for matches. In literature many are the authors that tried to define Coaching. For Hughes [HF01] coaching a soccer team is mainly a task of enhancing performance by providing feedback about the performance of the athletes and team.

During a soccer match, the coach can become the recipient of a great amount of information. As a result, he might not be able to evaluate and objectively exploit all the technical and tactical elements that may come along [FM91]. Emotional factors, such as stress and anger, or even more subjective aspects such as prejudice, can also lead to a decrease in concentration, and consequentialy misinterpretation of the game reality [CWR07]. In consequence of that, many are the coaches that tried to collect all this information through automatic performance analysis systems.

Two different types of analysis can be identified in a soccer match-Notational and Motion analysis. Notational analysis is based on event collection, either during the game by using a post-game analysis process, for classification and performance evaluation [Fra96]. Motion analysis is focused on raw features of an individual's activity and movement during a soccer match, without attempting any qualitative evaluation [CWR07]. Analysis can focus on four main categories-behavioral (mental factors that can be assessed from body language and other action), physical (movement and biometric data can be measured to improve training), tactical (includes choosing the appropriate strategy and tactics to be used against a specific opponent) and technical (skills such as passing or shooting can be assessed to improve training and performance) [CRW09].

More than offline tools, the use of real time soccer systems provides the coach with an opportunity to change some tactical or even technical aspects in a training session or in a real game situation. The majority of tools that exist in the market are very expensive and in some cases still present some technical issues like occlusion problems.

At the beginning of this thesis it was necessary to make a choice about the type of data to be used. This data could be one of two types: robotic 2d simulation league log files or human soccer data. The first type of data was available in public domain and because of that it is very easy to obtain. However the work visibility would be smaller. To obtain the second type of data it is necessary to contact the official soccer entities such as FiFA, UEFA or some European top clubs in order to ascertain whether they had availability to provide such data or, create our own data through the construction of a soccer analysis system. After some time without any response from official soccer entities we chose to create a simple analysis system based on a player tracking system. So, in this chapter a platform that could be used for a coach in a training session or even a match will be presented. This tool is capable to automatically calculate the full path of a player in the field in a specific interval of time, the most populated zone in the field during a game allowing the coach to review the performance of his team. Using a Wi-Fi network and a positioning engine on top of it, this system provides a visualization tool for such data on a real time basis. This information includes fully scalable concentration grids, a vision inference assuming that the tracked entities are associated with soccer players. In addition to what was exposed the system also works as a statistical collector meaning that it is possible to use data mining techniques predicting and categorizing typical player paths and also detecting their behavior patterns all over the game.

In this Chapter the current literature regarding the most relevant approaches concerning CSG specially related with soccer is presented, highlighting the tracking systems that exist in the market. After that a full description of the developed system's global architecture is presented [AVM+10] as well as a description of its most relevant modules. Finally the results will be exposed and conclusions is presented.

5.1 Literature Review

Nowadays the key factor in a soccer club's life is the game results. They determine-represent the success of the club and in many cases the coach's future. Because of that club coaches need to have maximum technical-tactics information about the game events and the way that it was played by the players [ABC+03].

Currently many are the computer systems that support coaching decisions before and after the game [LH01]. These type of systems may be divided in three distinct groups: performance evaluation, strategy development and real time competent assistant system. These last are the most complex ones because they involve some particular features like real time objects tracking, identification and classification of player movements and game events detection. In order to build an indispensable tool for a coach, these systems should be automatically capable of recognizing intentional activities in a multiagent environment with continually acting agents.

In the next subsections a group of generic off-the-shelf and academic tracking systems are presented.

5.1.1 Generic Tracking Systems

In literature there are many generic tracking systems that emerged over the past few years. These solutions are divided in two distinct groups: image based and non-image based.

5.1.1.1 Non-Image Based Systems

The Global Positioning System (GPS) is a satellite based solution that began being used by the U.S military forces for the planning of their operations particularly in arid and mountainous terrains. Since the 80's this technology became available for general public use and today it is normally used to do real time tracking analysis of different types of vehicles and as a base to analyze their motion [Yu05] [NKS05].

The Radio Frequency Identification (RFID) is an automatically wireless identification method that is capable of tracking objects and even people using radio waves. In terms of the required hardware, this technology uses a receiver and a set of tags which could be classified in: passive tags that are only detectable within a range lower than 13 meters of the receiver active tags that can be found within 40 meters of the receiver but need to have their own internal source power. Although the use of this technology could be an interesting solution for some areas, the high cost of the receiver and the active tag's average unit price is still an issue [Cha07].

Wi-Fi is the name given for a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections. Having this technology as a base it is easy to create a wireless data network that could be used in historical urban environments, academic campus etc. This technology could also be used for designing a tracking system. By reusing the wireless data network it is possible to create a tracking system on top of this infrastructure. Another advantage of this approach is the possibility of tracking an object using only a single access point, though in this particular situation the precision will diminish due to the lack of signal triangulation.

By comparing this last with other technologies the risks of occlusion and signal loss in this kind of approach can be considered very low mainly in environments that present low levels of metal concentration [Min06].

Bluetooth is a wireless protocol available in almost all mobile phones in the market. Although this protocol could be used in a tracking system, the high battery consumptions [JJ07], the short area coverage and the non-transparent connection establishment process make this approach inadequate for an efficient tracking system.

5.1.1.2 Image Based Systems

Thermal Signature is one of the most expensive tracking technologies existing in the market. It consists in detecting thermal signature of the objects tracked. The main purpose of these solutions is the reconnaissance and processing of thermal images. Although these systems present good results in some environments like oceans (within the objective of tracking living entities) [Rai03] the high cost of the equipment and in some cases the inexistency of a detectable thermal signature restricts the use of these kind of systems to a very controlled set of situations. Multi-Camera video surveillance is the most popular tracking technology. In terms of hardware required this technology uses a set of cameras spread in a trackable environment and a particular network. Despite being used in distinct scenarios some important issues still remain. The need to have high resolution equipment, dedicated network and the computational demands are still major problems that researchers have tried to optimize by using overlapping camera views [JRSS03] [KZ99].

In terms of generic tracking solutions and as a conclusion, all approaches have their strengths and weakness. Having in mind the characteristics of the CSG the best alternative seems to be a Wi-Fi based system. The competitive tag cost (that could be put for instance in a player shirt) and the high level of accuracy (in average less than 3 meters) compose important advantages. Another huge advantage is the fact that this technology is almost immune to the majority of occlusion problems that affects other approaches.

Another technology that could compose a good solution is the RFID based one. Despite of the fact that this technology requires standardization and consequently the cost of equipment will fall out (specially the receiver), the use of active tags could allow the increase of coverable area and accuracy levels have already reached good values. In spite of this, occlusion issues, related with liquids and metal still persist.

Multi camera surveillance systems are also quite common. In this kind of system some problems still remain like the camera's cost, the computational demands and the occlusion problems that constitute a very important factor that ought to be optimized. The other approaches like Bluetooth, GPS and thermal signature are not applicable in the CSG universe.

5.1.2 Sports Video Analysis

One of the major research areas in the CSG is the sports video analysis. In football/soccer domain researchers focused their work in problems like shot classification [GSC+95], scene reconstruction [YYYL95], structure analysis [XXC+04] [XXC01], event extraction [BJ03] [NKFH98] and rule-based semantic classification [TQ01]. These approaches used the image transmitted by the television and recorded them for posterior processing (after the match ended).

These kinds of systems are categorized by Ekin [ETM03] in two main groups: cinematic and object-based. The object based uses algorithms to detect objects in a video while the cinematic uses features from video composition and produce rules.

5.1.2.1 Cinematic Approaches

Xu et. al [XXC01] present a cinematic approach using for it the feature dominant color ratio to segment video. They defend that video reports should focus on playing field to extract game situations.

Xie et. al [XXC+04] used a Hidden Markov Models approach to detect two restrict events: play and break, in a video game. The complexity of this process is higher than in other sports like tennis or volley because, for instance, in soccer it is hard to determine if the game is stopped by a decision of the referee or by other highlights of the game-goal, corner, kick, shot, etc.

Other works like Ren and Jose [RJ05] tried to expand Xie's work and detect more game events like focus and replay in order to define new features/structures that they called Attack. Finally Qi et al. [QLD04] presented a multi-player tracking algorithm for using with low resolution video of various sports. Although this algorithm presented good results in occlusion situations involving members of opposite teams in the complementary situation: the players similar appearance turned the differentiation process very difficult.

5.1.3 Object Base Approaches

The object base approach demands more computational resources but it allows more high-level domain analysis. In order to detect a large number of game events the work developed by Gong et. al [GSC+95] analyzes the ball's trajectory and the relationship between the players' moves over the match. In literature there are also many works that tried new approaches like merging audio and video information [BJ03]. Although this kind of approach could constitute higher level domain analysis one big issue is the asynchronies between audio and video queues.

5.1.3.1 Real Time Tracking Systems

Over the past few years, new approaches appeared that use a multi camera tracking system to track players which promote new kinds of features like a near real time analysis. In this section only solutions developed for outdoor environments will be analyzed. By comparing to classic approaches analyzed in the previous subsections, these systems use a fix number of stationary video cameras placed in a traceable environment. This type of approach increases the overall field of view reducing the dynamic occlusion problems and improves the accuracy and robustness of the information collected.

Cai and Aggarwal [CA96] and Khan [KJRS01] track the object using the best view camera and if the trackable object leads the field of view they change it to a neighbored camera. Other authors like Stein [Ste98] and Black [BER02] assume that all trackable objects are in the same plane and compute the homography transformation between the coordinates of two overlapping images captured with uncalibrated cameras.

In Xu et. al (Xu 2004) work, eight cameras were used and were calibrated in a ground plane coordinated system using Tsai's algorithm [Tsa86]. Unfortunately this work presents some technical difficulties like problems with sparse landmarks in the coverable area that decrease the accurate calibration and data association and situations involving more than two players grouped in the same game region.

A main limitation detected in this type of tracking systems based on cameras is the inability to accurately track more than one player [IS04] [OMN+02]. The accuracy decrease specially in player congestion situations (like for instance a corner situation) [BMM+07]. On the other hand players who spend long periods of time isolated from the game have a very high accuracy (like goal keepers). Throughout the years many attempts have been tried to change this situation like increasing the cameras number or changing their placement [BMM+07]. In some situations where the player cannot be tracked the camera operator can correct the error in the player's path manually [THF96]. Another possibility is the creation of systems that use moving cameras. However, as shown by Koichi et al. [KMI+02] and Araki et al. [AMTY98] even in this field some improvements need to be made as automatic occlusion detection of players and the robustness of the detecting of change of player's position.

Summarizing, in CSG and more specifically in soccer the unique tracking systems that already exist in real environments are camera based. As demonstrated previously these systems still have to optimize some features like occlusion problems, computational demands, material cost and lack of portability.

5.1.4 Global Comparison

All the technologies presented have their optimal usage scenarios, and with the purpose of choosing the best suited for this project, a comparison using different parameters has been made. For this matter six different parameters have been defined: cost involved which comprises the unit price per tag and the receiver's cost; accuracy that concerns the location error involved and the coverage area defined as the maximum area that an approach is capable of covering within acceptable values of the previous parameters. Energy consumption is also an evaluation parameter and it is especially relevant in technologies, where an external power source is required. Finally, the response time is the time interval that goes from the acknowledgement of the last known good location of a tag and current one. Legal issues concern the existence of legal aspects on system's implementation. Also in this evaluation a scale with four distinct values (low, medium, high and very high) and two initials NA (not applicable) and A (applicable) are used. By analyzing Table 5.1 one can conclude that Wi-Fi technology is the better option: it presents high levels of accuracy, in average less than 2 meters, it does not have any legal issues involved in its use and presents a very competitive hardware price. The possible reuse of an existing network infrastructure is another advantage to be taken into account since it has direct impact on the involved cost, despite the eventual need for a network strengthening to enhance triangulation possibilities. In addition to what was stated one could also mention that Wi-Fi is relatively immune to most of the occlusion that arise with other approaches.

RFID is the second best tracking solution analyzed. The use of active tags increases the coverable area-forty square meters using a single receiver-and increases accuracy levels. Despite of this obvious advantage, RFID still lacks standardization which naturally emphasizes the necessary integration process and supplier evaluation. Occlusion problems are not transparently solved and so there is some notorious interference in spaces with liquids and metal structures.

GPS's overwhelming worldwide coverage area together with high levels of accuracy seamed to make this technology appropriate. Despite its advantages, its main purpose is for outdoor domains,

	Costs		Accuracy	Coverage Area	Energy Consumption	Response Time	Legal Issues
	Tag	Receptor					
GPS	Medium	NA	Low	Very High	Medium	Low	A
RFID Passive Tag	Low	High	Low	Low	NA	Low	A
RFID Active Tag	Medium	High	Medium	Medium	Low	Low	A
Wi-Fi	Low	Low	High	High	Low	Low	NA
Bluetooth	Low	Low	Medium	Low	High	Low	A
Thermal Signature	NA	NA	Medium	Medium	NA	Low	NA
Infrared Sensors	NA	Low	Medium	Low	Low	Low	NA
Multiple Cameras	NA	Medium	Medium	High	Low	Medium	A

Table 5.1: Technology Comparison

instead of confined spaces where its relative accuracy is considerably lower. This fact associated with its significant tag cost relegates this technology as a third choice for indoor real-time tracking systems.

Although Bluetooth and infrared sensors represent good overall solutions in terms of cost, the coverage area is somewhat confined and the medium levels of accuracy invalidate the application of any of these systems in this project.

5.1.5 Commercial Available Vision-Based Analysis Systems

In this section some of the most used vision-based analysis systems will be presented, exposing their strengths and weaknesses. Also at the the end, a comparison between them will be discussed. It is important to note that as soccer is an outdoor sport, vision-based analysis systems for indoor environments are not considered in this analysis.

5.1.5.1 Semi-Automatic/Online Systems

Nowadays there are many sports performance analysis systems capable to present different player tracking capacity features [Nee03].

ProZone (West Yorkshire, England) is a soccer analysis system that uses a set of cameras (normally 8-12 cameras) spread through the soccer field capturing the match events. After that the player positions are manually marked in the video (this process takes longer than 34 hours for the 22 players and the ball) [Set03].

The information produced by this system can be used to coach staff to evaluate team and players performance [Set03]. This software presents a 2D match view without supporting the video image at the same time and also in this software individual (Figure 5.1) and collective performance (Figure 5.2).

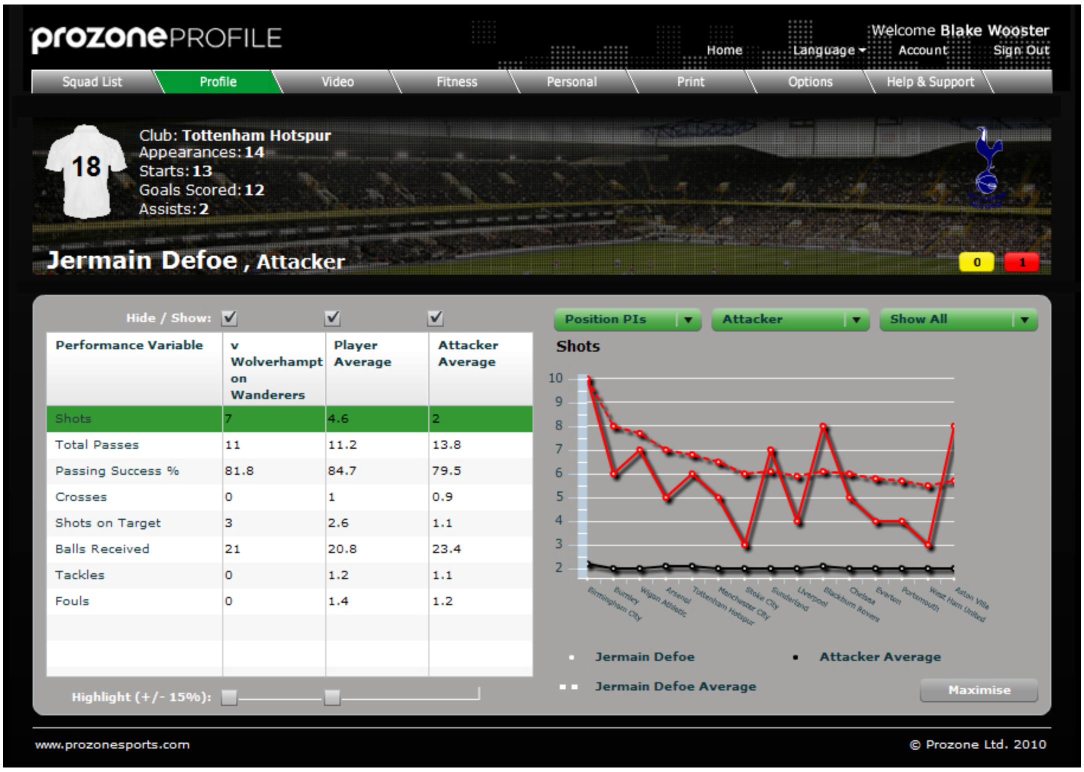


Figure 5.1: An Individual statistical treatment of a player using ProZone System

ProZone was the first analysis system implemented in the Premiership (the major soccer league in the U.K) [Por98] and in consequence of that the teams that have implemented it, obtained a pioneer advantage. However this system still presents two major issues: the Installation cost of the system (around 120 thousand euros) as well as the cost of the analysis of a particular game (which is not included in the installation cost). These two issues lead to many clubs being unable to pay for the use of these services. Another big disadvantage of this system is that it is not portable, which means that after the tools expensive installations process, a team can only be sure to get its home games analyzed.

The main advantage of this system is that it provides a huge number of match informations. However, to process it, in a short period of time (normally few days), the club must have at least one full time employer working in this task [Bal02].

Amisco (Sport Universal, Nice, France) is a soccer analysis software that through the images captured by cameras, spread in the soccer field, is capable of calculating reliable informations for a soccer coach. This software supports many different phases in the information treatment:

1. **Image Capture**: Using 6-8 cameras, spread by the soccer field, this system is capable to capture many match images, including the soccer game events (Figure 5.3);

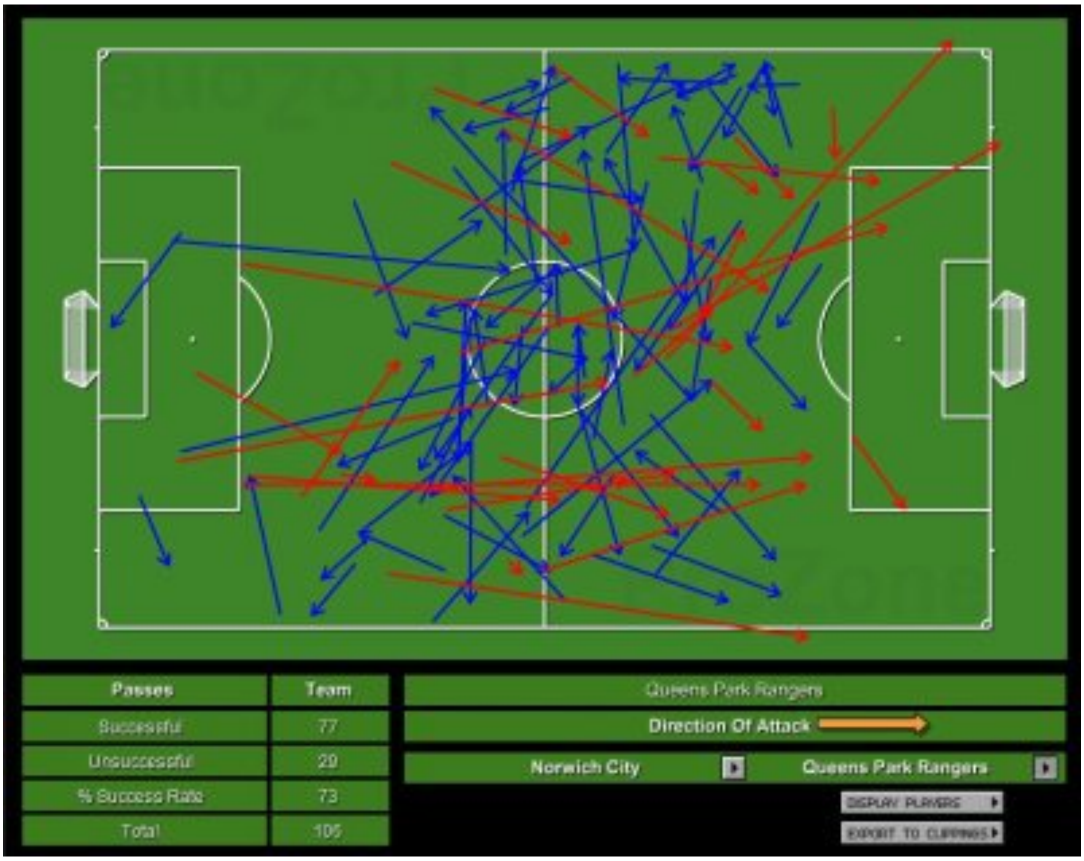


Figure 5.2: Team Statistical Treatment during a soccer game using ProZone System

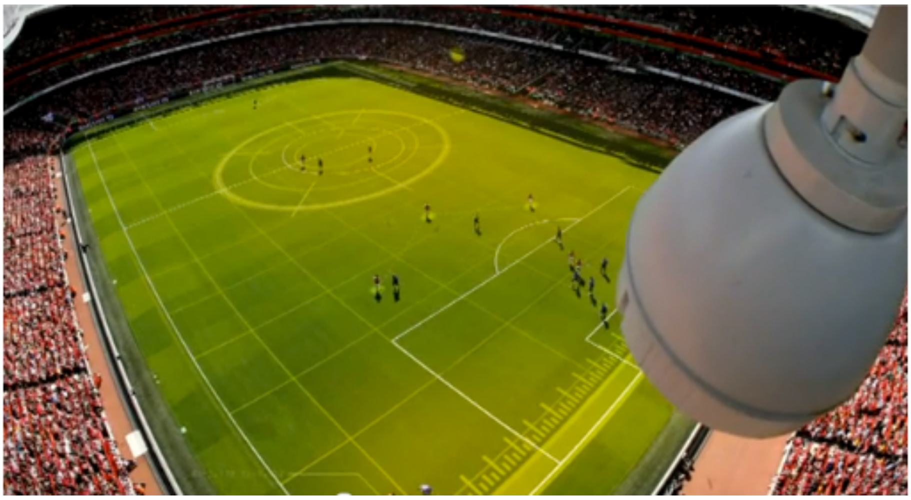
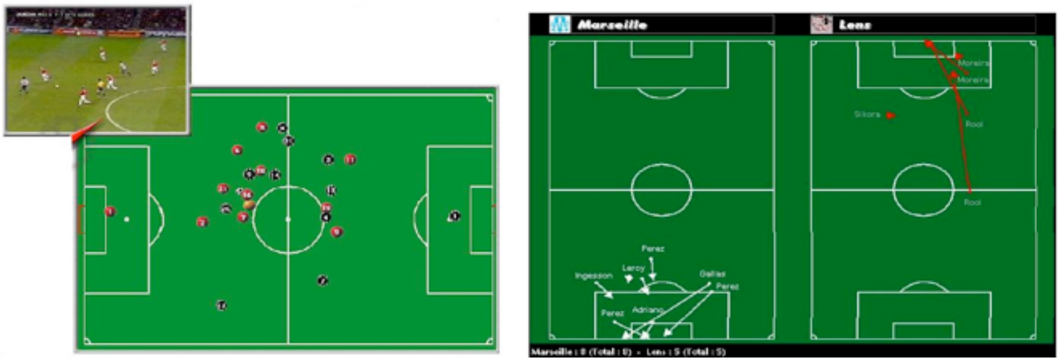


Figure 5.3: Events Capture Through Video Cameras in Amisco System

2. **Image Processing**: The images are manually processed identifying many match features like the positioning of the players and the ball, among others, allowing, the 2D game representation (Figure 5.4(a)) as well as the identification of the player pattern movements

(Figure 5.4(b)). However, this software presents a major issue: it needs full time operators in order to manually identify some game events such as fouls, off sides and cautions that occur in the game [Set03].



(a) Amisco 2d Viewer

(b) Amisco Individual Player movements

Figure 5.4: Example of a 2D representation of an image captured during a soccer match (a) and also a representation of players movements during a match using Amisco System (b)

eAnalyze (eSport, USA) is another video-based system that provides player tracking information from games in real-time. Similary to the other analyzed systems, also in eAnalyze many operators are needed in order to manually track player positions frame by frame (Figure 5.5). After this process, the software is capable of producing information regarding distances travelled,

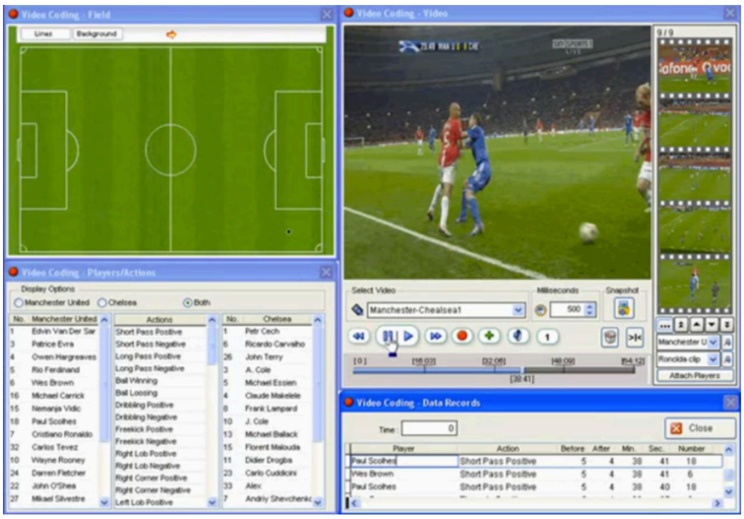


Figure 5.5: Example of a Event Detection using eSport System

speed breakdown of individual players among others. Table 5.2 represents a comparison study between the analysis software existing in the market. It is important to note that in this analysis softwares such as Dartfish, Digital Soccer, Game Breaker and Utilius VS were excluded because of their nature (these softwares only create video records to create movie presentations).

Name	Strengths	Weaknesses
ProZone	Statistical treatment on individual and collective level	The need for manual tretment of the images after acquisition; Very expansive solution;inability to show real video feed together with 2D analysis
Amisco	Statistical treatment on collective level	The need for manual treatment of the images after acquisition; non-existence of player and team modeling
eAnalyze	The simplicity of the interfaces	The need for manual treatment of the images after acquisition
Ascensio Match Expert	3D game viewer where the user can see the soccer match (from previously collected images)	Poor Statistical treatment when compared to the previous software analysed
Match Vision Studio	Rich statistical treatment	No model construction of the players and teams despite the statistical data alowwing it

Table 5.2: Professional Performance Tools Comparison

5.2 Project Description

In this section, the undertaken project is described in detail regarding its several components and analysis perspectives.

Having this in mind, the electrical infrastructure is detailed and after that the system's global architecture is depicted, in order to have an overall glimpse. The database model is further explained and the final two subsections are dedicated to the tools' individual description.

5.2.1 Electrical Infrastructure

Most of professional soccer coaches state that the training session should have the same length as a conventional soccer match-ninety minutes. Consequently any training support system should stay active for all of this period. To fulfill this goal an electronic system was designed. In this approach a conventional 45A car battery is used directly connected to a 600w UPS. The UPS battery is also connected to the car battery in order to increase the autonomy of the system. This electrical infrastructure (Figure 5.6) is capable of providing power for more than 120 minutes.

In order to increase the WI-FI network's density, a star topology approach is used. A router is connected directly to the battery's electrical extension and it is placed behind the goal. The access points (APs) were placed in specific points of the penalty box as shown in Figure 5.6.

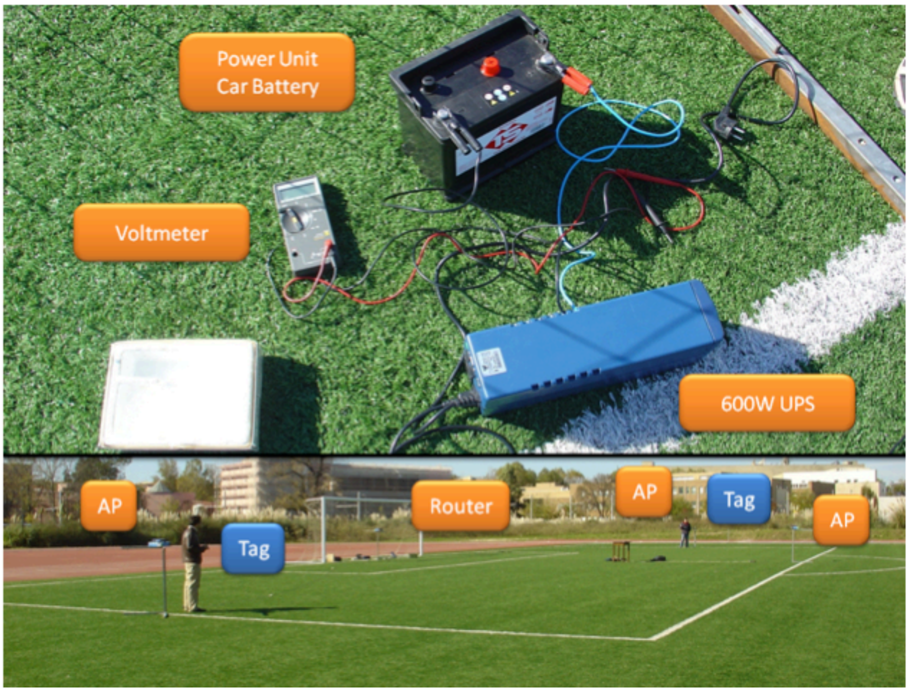


Figure 5.6: Electrical Infrastructure and WI-FI Network

5.2.2 Global Architecture

In this subsection the system's global architecture will be illustrated as well as its modules and how they interact and therefore extract not only system components dependencies but also information flow analysis. All of the system's components and their relationships are exposed in Figure 5.7.

Having the above mentioned in mind, and paying closer attention to the numbers in figure, one is able to identify the system's modules as follows: Offline map editor; Wi-Fi enabled localization tag; Position Engine; Database for data integration and storage; Real-time monitoring application and Web enabled real-time and historical business intelligence.

Although most of these elements are object of further explanation in the next subsections, one ought to undertake a brief description of those whose nature is not obvious and in order to clarify their interaction.

The first action, that ought to be conducted, in offline mode, consists in conducting a complete map creation-edition. The user shall specify, amongst other details, depicted in subsection Map Editor, the image file representing the soccer field layout and the used scale. This information is compiled in a XML file for both the position engine and real-time monitoring tool and submitted to the mentioned database for the historical BI application.

The Wi-Fi tag consists in an active 802.11 a/b/g board with a couple of power batteries attached. These are configured to connect to a specific Wi-Fi network-security, DHCP but another network configurations are also possible-in order to directly communicate with the position engine. By using this kind of wireless technology, it is possible to partially or totally reuse the spot's

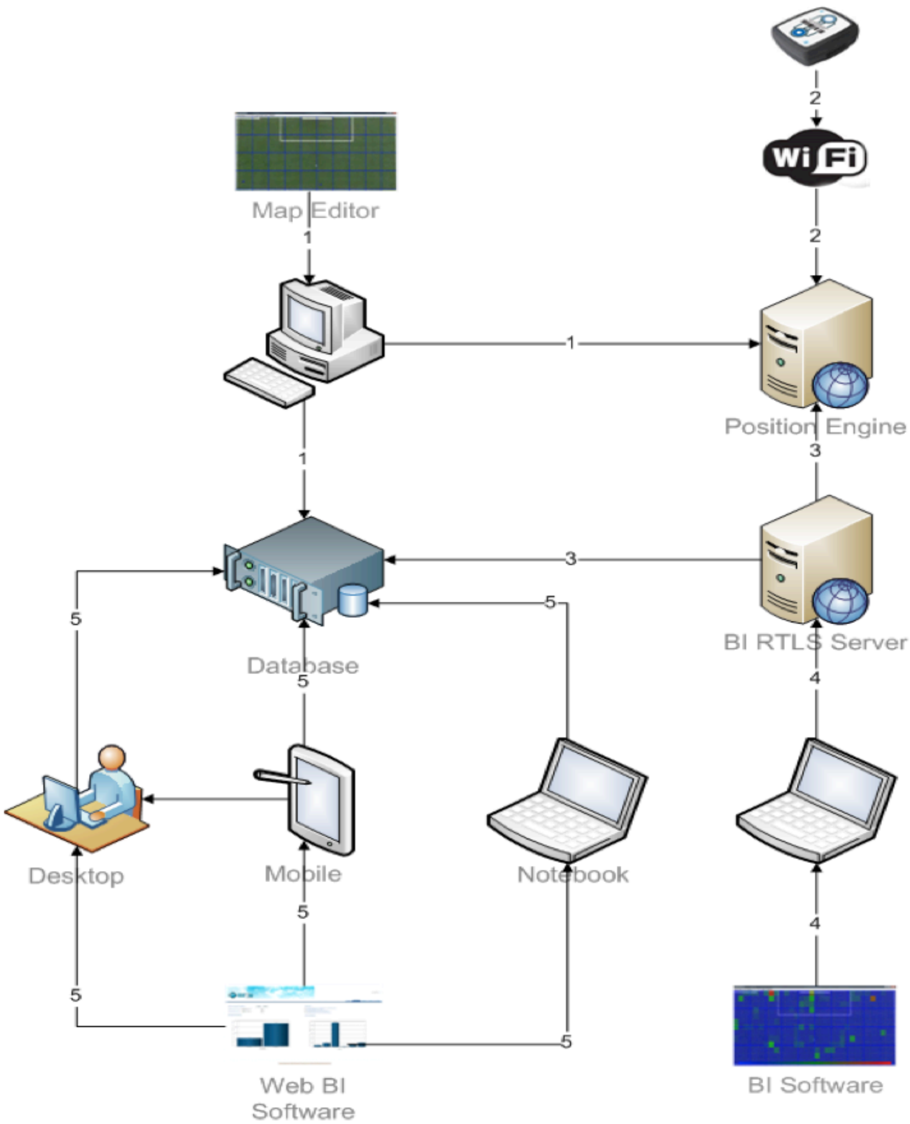


Figure 5.7: System's Global Architecture

arena network infrastructure, having only, for special requirement, a high density of access points as the accuracy naturally increases with this factor.

The position engine used, periodically collects data from the tags and updates their position against a pre-loaded localization model. This model requires a previous offline site survey for measuring Wi-Fi signal strength and for network items-routers and access points-precise localization. The engine is also web-enabled and supports a HTTP/XML API so that third-party applications can interact with it, therefore accessing localization and status information regarding each individual registered tag.

Using this communication protocol, the developed real-time monitoring server is responsible for gathering, at a specific periodicity that typically equals to the position engine frequency-every tag's valid location data. With this information, this module is directly responsible for updating the database and caching the session's data for the real-time monitoring application.

Having the continuous up-to-date database as a solid information reference, it was possible to enable both real-time and historical business intelligence applications. For real-time knowledge extraction, it was only used data referring to active sessions. For historical analysis, and delegating all the process effort to the database engine, specific and dynamic time windows were used to filter data. Despite the additional explanations that are given in subsection Real-Time and Historical BI Application, the versatility of such application must be referred as it congregates both web-enabled features and zero data process as it is all delegated to the database engine and allocated in a dedicated server- enabling its usage in a wide range of devices, including PDAs and mobile phones, alongside with traditional notebooks and desktop computers.

As a synopsis, one might refer the system's architecture as fairly distributed, where offline information regarding soccer field layout and wireless network definitions team up with a real-time web-enabled position engine, which enables third-party applications to collect and store data, so that diverse specific end-users can access both real-time and historical knowledge in a wide range of equipments, therefore enhancing coaching efficiency levels.

5.2.3 Database Model

Having into consideration the specific reported system's application in the soccer domain-usually characterized for multiple player movements all over the field combined with the project's idiosyncrasies-specifically in what concerns the localization tracking frequency-the database model paradigm followed consists in a hybrid form of a data warehouse star architecture with a slight normalization flavor, as illustrated in Figure 5.8).

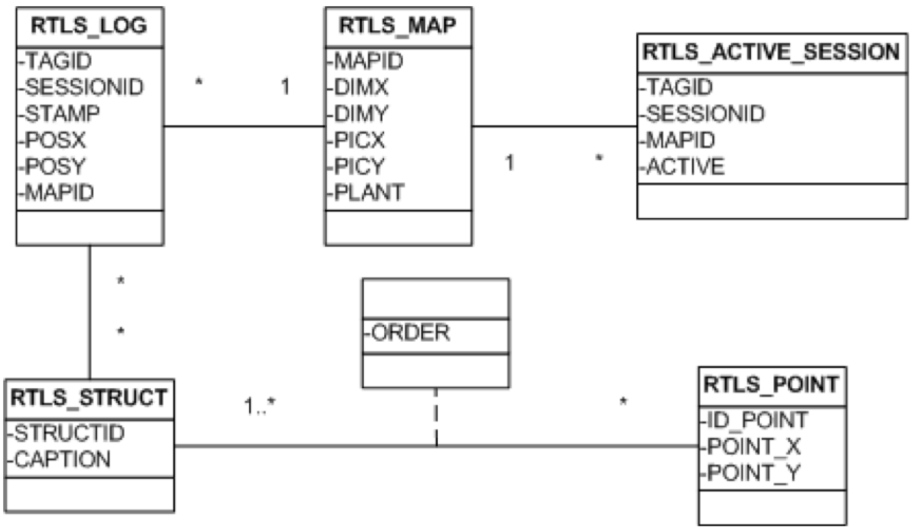


Figure 5.8: Database Model

Regarding the strong star model, it is supported for the high data production levels, and perhaps most important, the fact that all data insertions are machine responsibility, as depicted in the above subsection, therefore preventing human error. It is also vindicated by historical analysis that it may cover hundreds of thousands and even millions of records.

On the other hand, some database normalization was introduced in order to cope with real-time requirements that would not be compatible with computing hundreds of records out of a table with millions of records, in a continuously systematic way. Another argument in favor of database normalization resides in the soccer's field layout.

Referring to specific database model items, it is important to note the central relevance of rtls log as central table responsible for storing all localization data. For each pair of tag/session identification, a particular position is recorded in a given layout with a specific timestamp. The concept of session may be different in each training session according to coach's decision. A new session could be related to three distinct situations: a player substitution (when a player is substituted by a colleague), a player out of the field (for instance to receiving medical assistance) or other situation when the player is out of the limits of the region that was defined by the coach for a specific situation in training session.

In order to achieve real-time requirements, some redundancy has been introduced concerning active session identification, so that active players identification could be easily, and most importantly, efficiently retrieved.

5.2.4 Map Editor

Map Editor is a traditional, network enabled, desktop application responsible for complete soccer field layout definition. The soccer coach shall open an image file and provide the interface with the drawing scale, in order to convert pixels to meters and vice-versa. Afterwards, the tool offers the possibility to pinpoint and draw, over the original layout, spawn areas-concept that will allow the detection of new sections.

Once the layout is completely defined, the coach is able to save map characterization in a XML file in any available location and/or commit it to a specified database-with the previously described database model implemented.

The XML file will be an input for both the position engine and the real-time monitoring server, and, on the other hand, the committed database information is ground for historical computation and analysis.

Summarizing, the Map Editor constitutes itself as an auxiliary tool, vital for system's setup and dynamic enough to cover all the analyzed soccer field. Its dual output enables a flexible usage for several system components and, simultaneously, due to XML openness, enables third-party development and integration.

5.2.5 Real-Time and Historical BI Application

In order to extract significant business intelligence knowledge based on historical data and not only real-time information, the authors decided to make an immediate use of the raw position data stored in the database. Taking advantage of using Oracle as equally laboratory and production database, Oracle's Application Express was used to generate a web application responsible for processing data and, most importantly to aggregate information in an understandable way.

As depicted in Figure 5.9, the Apex's engine is directly embedded in the database, thus directly dealing with coach's web requests. With this architecture, several systems can easily access BI application as all heavy processing is the database server's responsibility, leaving the coach device with only chart rendering computation.

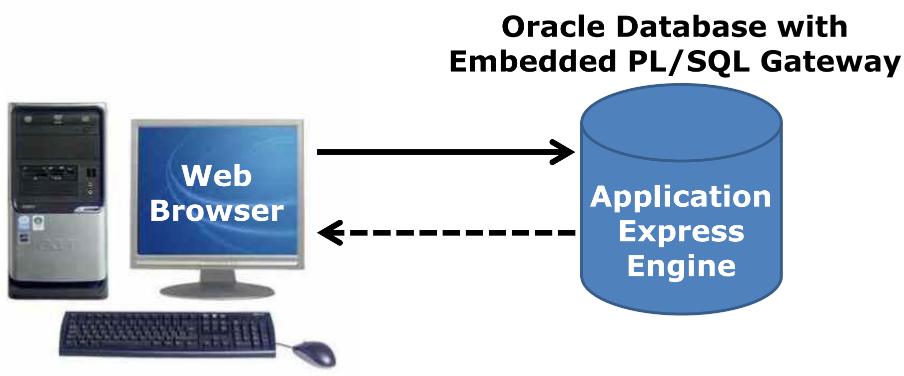


Figure 5.9: The Apex's engine

5.3 Results

The results exposed in this section concern to the data gathered over a training exercise conducted with four human players in a real soccer field's penalty box with its dimensions as well as the goal's as recommended by the Federation Internationale de Football Association (FIFA). The exercise's purpose was to train a player's shot accuracy after receiving a pass from a winger. For that matter a goalkeeper, two wingers and a striker participated in this experience having each of them a Wi-Fi tag attached to their shirts.

The penalty box was also divided in a 10*4 grid for calibration purposes and also to guide the site's surveying process. The following picture (Figure 5.10) shows how the exercise was conducted.

To clarify the Wi-Fi network's density one ought to first specify the access points' positioning. A router was placed behind the goal as well as the batteries and the entire electrical infrastructure described in the previous section. The remaining three access points were also used and positioned over the center of the remaining lines that define the penalty box (excluding the one which contains the goal line). To maximize the signal's strength all the Wi-Fi devices emitting a signal were put on top of a structure that allowed them to gain 1.20 meters of height. They were also put twenty centimeters away from the real lines so that the players' moves were not affected by their presence. Figure 5.11 shows the signal's strength and noise levels on this particular scenario.

Since this is an outdoor environment the authors believe that the gathered noise values are the main cause for the error on the player detection because they are not being compensated by

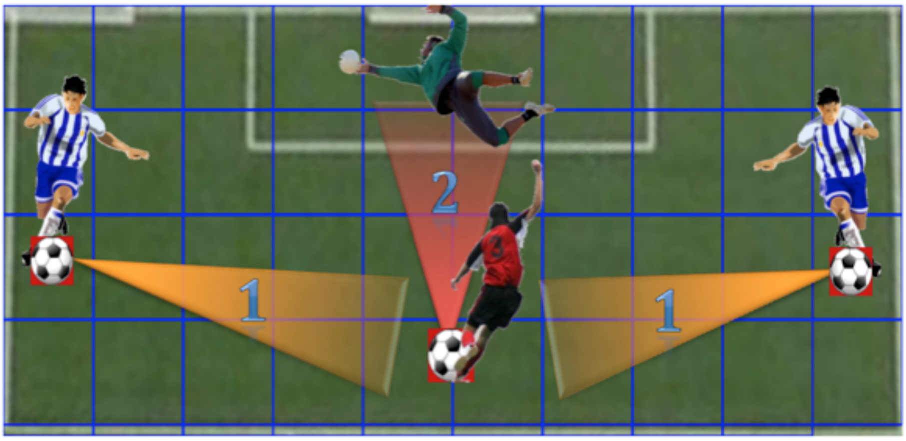


Figure 5.10: Soccer Exercise Conducted

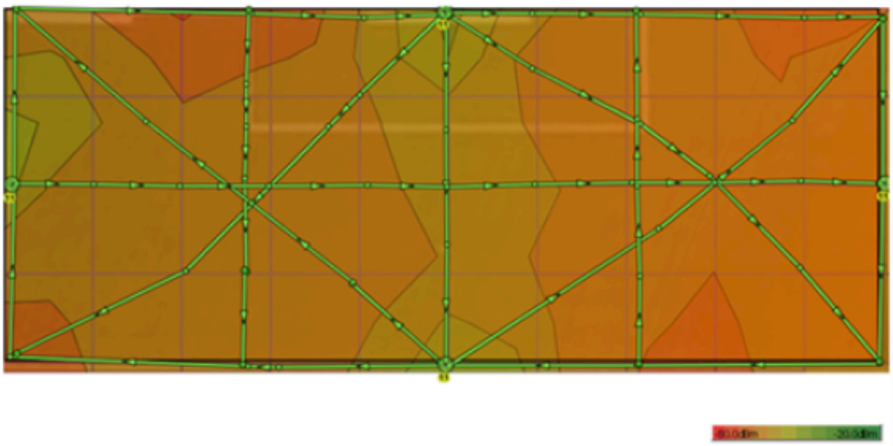


Figure 5.11: Signal Strength and Noise for WI-FI Network

refraction and reflection phenomena which are typical in indoor environments. One ought to point out that this test was conducted with high-end devices and so there is a high probability of diminishing the noise's impact just by changing the hardware to high-end artifacts, as their value mostly differs on the applied power on signal emission.

Even so, the next figures clearly demonstrate that the system was able to track the players during this exercise which lasted about thirty minutes. For instance, on Figure 5.12, showing the box's density over the entire exercise with the scale depicted at the bottom of the picture, one can observe a red cell on the goal area which undoubtedly corresponds to the goalkeepers' presence waiting for the striker's shots. The neighbor cells are also highlighted as the goal keeper moved a bit during the exercise in order to better cover the striker's shots on goal. The other highlighted cells demonstrate how the other three players moved during this training session.

Figure 5.13 shows a real time screenshot of the player density where one can observe the wingers' position after having one of them pass the ball.

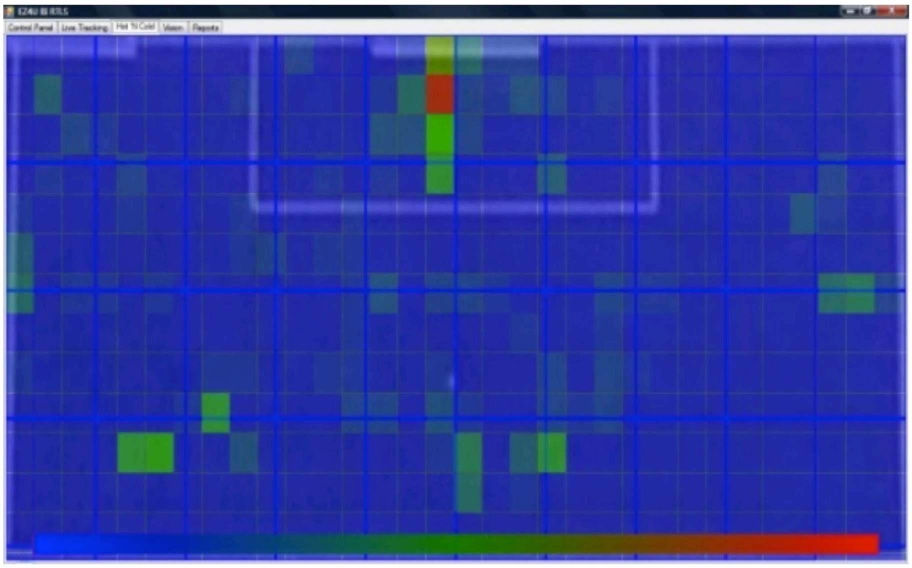


Figure 5.12: Box density over an exercise

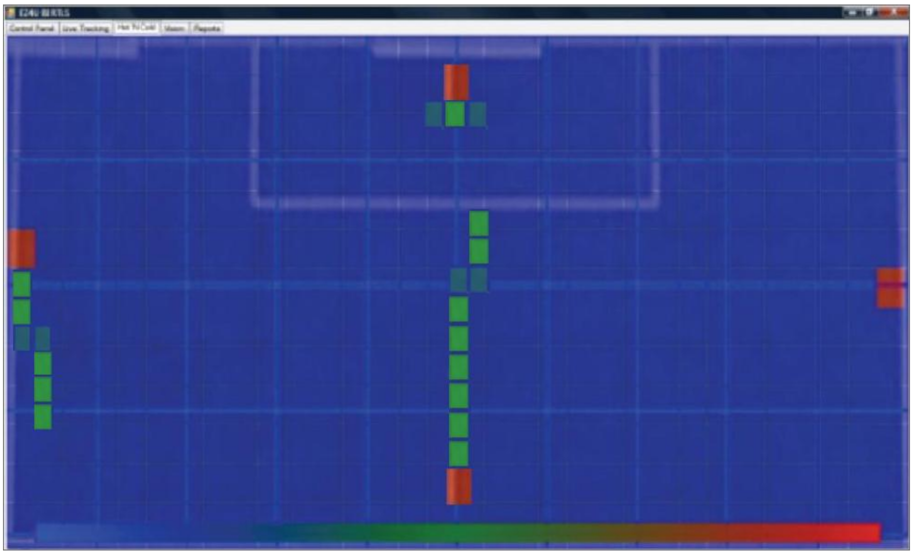


Figure 5.13: Player density in game field

Finally on Figure 5.14, one can observe the left winger and striker's position during a pass. On this particular figure the players are represented as blue dots over the field. In this case the error between the obtained position and the real one did not exceed two meters for each player, which also justifies the fading green cells on the box's corner (shown on Figure 5.12) as the wingers could decide from where they wanted to perform the pass as long as their distance to the box's limits did not overcome three meters.

Overall, the system remained stable during the whole training session thus confirming its robustness and applicability as a tool for scientific soccer analysis.

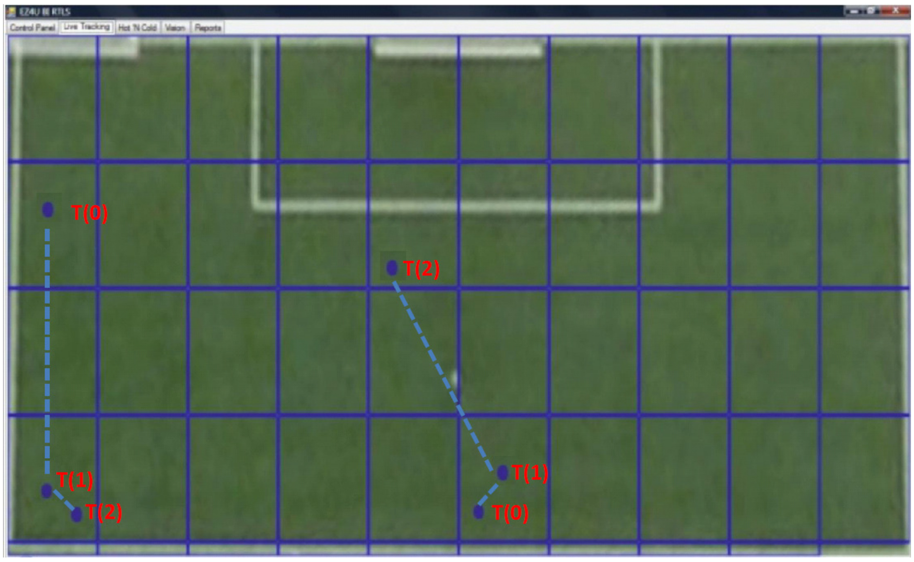


Figure 5.14: Striker Position during a pass in three consecutive instances (T(0), T(1), T(2))

5.4 Conclusions

In this section, the tracking system project's main conclusions are presented as well as major future work areas and potential collateral applications are depicted.

In this chapter a new tracking system tested in a penalty box area in the universe of soccer was presented. As mentioned in the section Project Description, with the construction of a portable and little expensive system that includes basic wireless network, a car battery and UPS it was possible to track players on a real time basis all over a penalty box in a soccer field. This project shows that a Wi-Fi based technology could constitute an excellent solution for soccer. Unlike other location systems in this approach the occlusion problems are reduced to a residual level, the signal's strength al does not degenerate over the period of the training session and the accuracy levels are quite satisfactory- in average less than 3 meters - using low cost equipments - router and access points. With this little expensive tracking solution any team's coach has detailed reports about the performance of a specific player or all team in a training session or even in a soccer mach. The possibility of having real time player positions in a specific situation and historical player paths composes an important tactical indicator for any soccer coach. In this particular item the Oracle's Apex Technology proved to be a solid solution. It allows multiple simultaneous accesses and, consequently, dramatically enhancer analysis empowerment, while, at the same time, eliminates heavy data computation from end-users terminals. These characteristics allow accesses from unconventional systems such as PDAs, smartphones, notebooks and desktop computers through their web-based interface. This particular feature has a great importance for technical staff that, for instance, is spread through the soccer stadium in a match.

Summarizing, it is fair to state that the project's initial ambitions were satisfactory met and that the cooperation with an important university in the sports area was extremely important for better measuring the system's positive impact. The technology's transparency, allied with the future work areas, is believed to greatly improve potential applications, thus significantly widening the project's initial horizons

Chapter 6

SocOn - A Soccer Ontology

The concept of ontology appeared in the 80's, but only in the **90's** did it become popular among researchers [Gru93] (as mentioned in chapter 3). Nowadays, ontologies are present not only in academic environments, but also in the business world, where they represent an important role in many online applications, such as e-commerce (Amazon and eBay), search (Yahoo and Lycos) and others [McG03].

Over the past years, in the soccer area, two distinct ontologies have emerged, proposed by Ranwez and Moller, respectively. Ranwez's research work is focused on construct narratives abstraction, based on a set of related events [CVR98]. In 2002, Ranwez et al. defined a soccer ontology (Ranwez Soccer Ontology[1]) that represents a set of soccer concepts like rules, actions or player attributes (name, nationality, and so on). This language is used to support video annotations in a soccer match. In 2004, a new soccer ontology is proposed by Moller (SWAN Soccer Ontology[2]) to be integrated in the SWAN (Semantic Web Annotator) project, which consists in automated extraction of metadata from natural language web content, and presents many additional concepts representing agents within the game as well as a number of concepts surrounding a soccer match. This project uses the KIM platform (Knowledge and Information Management)[PKK+03], which is based on GATE (General Architecture for Text Engineering)[3] [CMBT02].

Robot Soccer Ontology developed by Stanton and Williams [SW04] is an ontology that describes some robotic soccer concepts for the RoboCup SONY 4-Legged League (previously analyzed). The main goal of this ontology is to transform the sensorial and symbolic information collected by the robot through the game in concepts and physical objects reporting the information collected for knowledge structures and inference rules.

[1] Ranwez, S. Ranwez Soccer Ontology. DAML Ontologies Library Web site, submitted in 2002, available online at http://www.daml.org/ontologies/273

[2] Moller, K. SWAN Soccer Ontology, submitted in 2004, available online at http://sw.deri.org/~knud/swan/ontologies/soccer

[3] More information available online at http://gate.ac.uk/

In order to better understand the soccer concepts in this thesis a soccer ontology was developed [AFRG10], named SocOn, using the Protégé framework and the *OWL* language. The main goal of its development is to better characterize the concepts that are included in a soccer match. In the next sections all of the ontology development process will be described and finally the results will be presented.

6.1 Languages used in Team Training

COACH UNILANG is a language developed by Reis and Lau [RL02]. In this new language the concepts are grouped through conceptual areas: Field Regions where it enable the representation of field areas with different shapes, Time Periods where the game periods are represented by its duration or time interval, Tactics which allow the high level configuration of the teams (including offensive and defensive characteristics), formations, player types, situations, among others.

Clang [CFH+03] is a standard coach language developed for reducing the communication mis understanding between coach and his players. It was based on COACH UNILANG proposed earlier [RL02]. With this language it is also possible to define conditions, actions and regions. The conditions are constructed through the logic connectives from descriptive propositions about the atomic state.These propositions are related to: players positions, ball position and the game mode. The actions are regarding to the orders that may be sent to players and finally regions which are connected to select areas in the field using polygonal shapes, among others. In conclusion Clang is based on directives and the main goal is transmit to the players a certain behavior which it is desired that they have throughout the game. However these concepts are very targeted to the competition where this language is inserted.

Doing a brief comparison between these two languages (Clang and COACH UNILANG) it is possible to note that the first one represents more low-level concepts (more related to players as individual entities) while COACH UNILANG represent more high-level concepts (included more team concepts and less oriented to a player).

6.2 Soccer Concepts Comparison

Table 6.1 illustrates a comparison between the analyzed approaches. Doing a brief analysis it is easy to note that the first two ontologies analyzed presented very low results. This fact is regarding to the goal of these two approaches which is concerned to information annotation in the web. The other ontology analyzed is confined to a very specific environment (robotic soccer) and because of that it did not present good results in this comparative. In spite of that this ontology presents a good game model and also good objects, time periods and events representation. Regarding Clang and COACH UNILANG the main difference concerns the nature of the represented concepts. While the first one represents low-level concepts and because of that presents in the three initial criteria good results, however in the other criteria more related to higher level concepts the COACH UNILANG obtained better results.

	Objects	Actions	Players Orders	Players Atributes	Players Position	Team Formation	Team Tactic
Ranwez Ontology	-	-	-	-	-	-	-
Swan Ontology	-	-	-	-	-	-	-
Robot Ontology	+	+/-	-	-	-	-	-
Clang	+	+	+	-	-	-	-
Coach Unilang	+	+/-	+	+/-	+	+/-	+

Table 6.1: Soccer concepts comparison between the analyzed approaches.

As a summary for this subsection, it is relevant to note that there is no ontology or language that includes at the same time players and team concepts.

6.3 Structural Ontology Definition

In this section an overview about the concepts represented in the soccer ontology as well as their relationships will be presented.

6.3.1 Main Class

Normally in the development of an ontology using *OWL* language a main class is usually set up. In this soccer ontology the main class was the Soccer Game. In a soccer domain there are many concepts that are interrelated. In our ontology these concepts are divided in different conceptual areas:

1. **Objects** defining the objects to be modeled. In this case the objects are the soccer field, the ball, players and teams;

2. **Regions** where is proposed a division of the playing field in various regions. These divisions are extremely important to simplify the definition of some concepts such as the identification of team's formation;

3. **Periods** defining the different game periods;

4. **Events** where actions are identified, its conditions and results. Some of this area concepts are passes, shots, ball possession among others;

5. **Situations** where all game situations are identified like attacks, throw-ins, corners among others;

6. **Players** where the attributes, behaviors and players positions are classified through the game;

7. **Teams** where the concepts related to team strategies and formation are ranked;

Figure 6.1 represents the class hierarchy of the main concepts developed in the soccer ontology.

Figure 6.1: Class Hierarchy of the Main Developed Concepts.

6.3.2 Main Conceptual Relations

Using the *OWL* language in the ontology development, the conceptual relations are described through Properties-Object. So, defining those properties it is possible to obtain concepts' relations. In Figure 6.2 are presented the Soccer Ontology Properties-Object. As mentioned before the main

Figure 6.2: Main Proprieties-Object presented in the Soccer Ontology.

goal of Properties-Object is to relate concepts. An example of those relationships is illustrated in Figure 6.3. Doing a brief analysis it is possible to verify that Objects are represented by attributes,

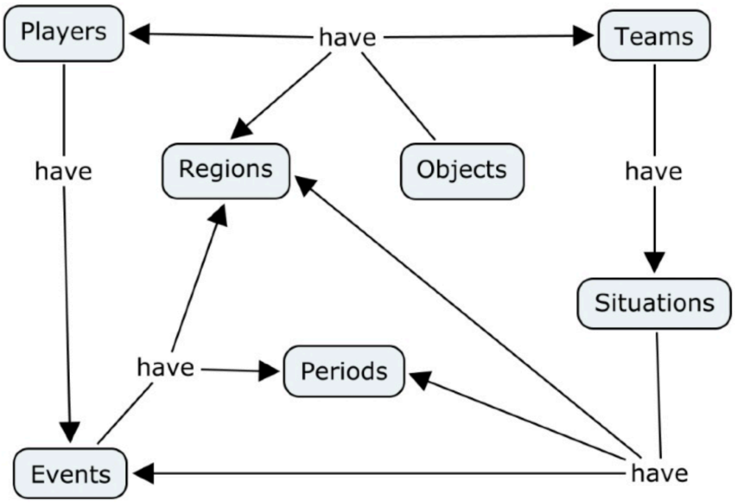


Figure 6.3: Main Concepts between the Ontology Conceptual Areas.

positions and players behaviors, the players characteristics are represented through Events among others.

6.3.3 Data Types

In this soccer ontology the used input data is represented in Figure 6.4. Basically, this data consists

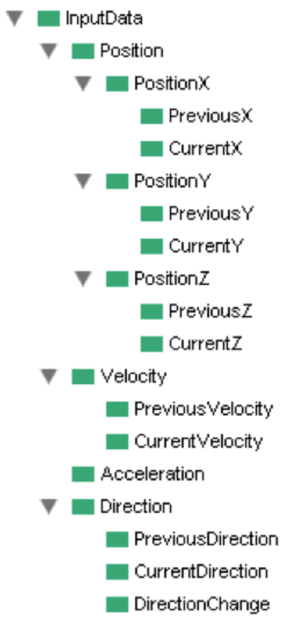


Figure 6.4: Ontology Input Data.

in the cartesian coordination, velocity, acceleration and direction of players and the ball.

6.3.4 Representation of Game Space

In the ontology development process it is important to defined the origin point of the coordinates in the soccer field. The upper left corner was chosen as the point of origin of the coordinates due to a main reason concerning the value of the variables (Figure 6.5) (always positives inside the soccer field). Also it is defined the attack direction for both teams (Our_Team attacks Left to the

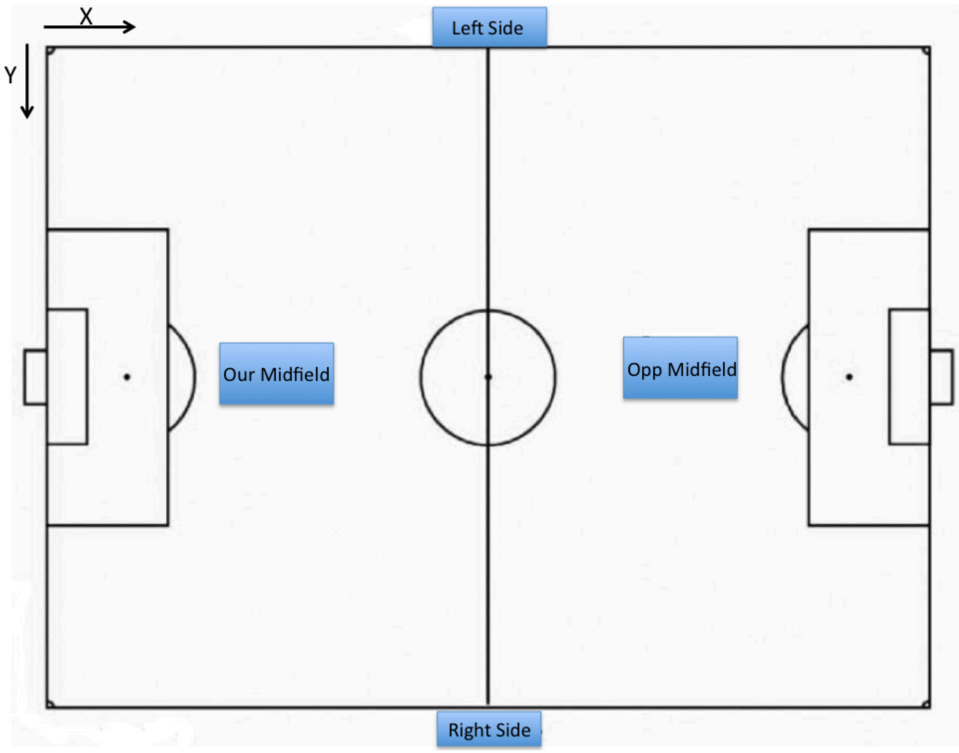


Figure 6.5: Representation of Game Space.

Right directions and the Opp_Team in the opposite direction changing at the Halftime). Finally the Right and Left Side of the soccer field are also defined.

6.4 Objects Definition

In this soccer ontology two types of objects are considered: concrete and abstract. The concrete objects are divided in two subgroups: mobile and static (Figure 6.6 illustrates that definition):

1. **Ball** is a mobile object that is the attention center of the players;

2. **Players** are 22 mobile objects spread in the soccer field. In this ontology only the goalkeepers have distinct properties concerning the other players;

3. **Game Field** is the unique static object in this environment. It is represented though two properties: length and width;

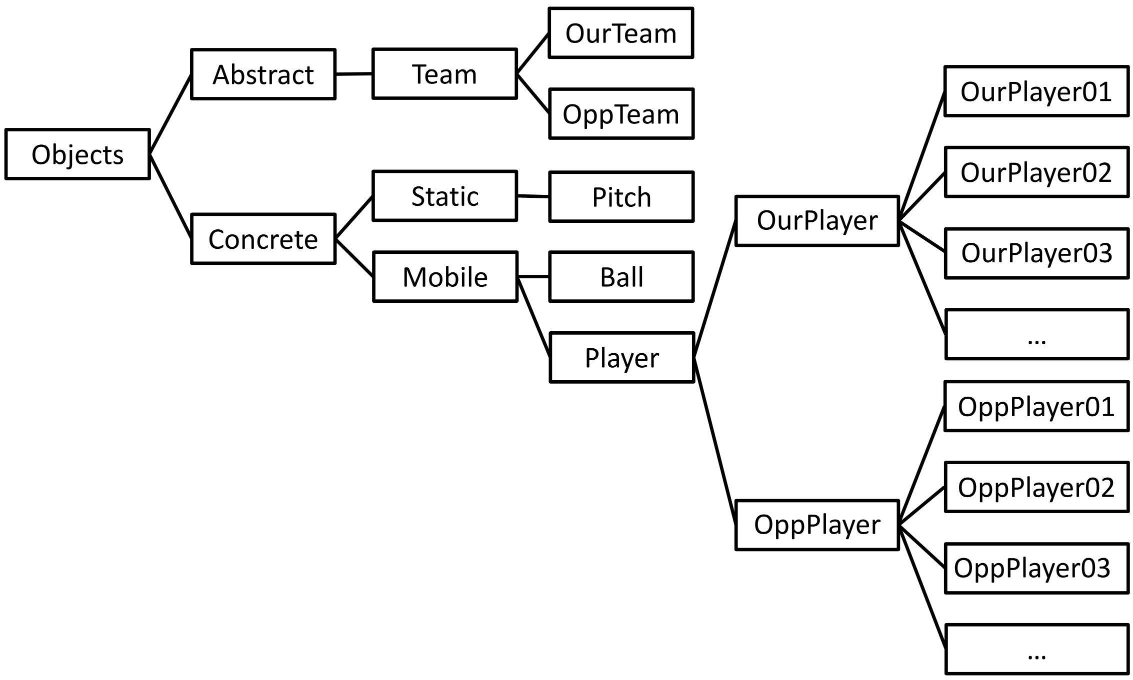


Figure 6.6: Hierarchical Representation of Object Types.

4. **Teams** are abstract concepts composed by eleven players each. To characterize a team during a match it is important to analyze the most rearmost and advanced player as well as the player who is more left and rightmost (in terms of field perspective) in order to characterize the team in terms of length and width.

6.5 Regions Definition

In this ontology the region concept is related to the definition of a sub area in the soccer field. In some game periods this division allowed the increase of the characterization of some match actions. Having this in mind in this work three distinct regions were defined according to [RL02]:

1. **Field Regions** where the soccer field is divided in N areas according to the granularity of the performed study goals;

2. **Variable Regions** where some new areas are defined through the field regions. These new regions are very important to characterize some game situations;

3. **Team Regions** where new areas are defined according to team occupation area in the soccer field. These regions are very useful to characterize the team formation.

6.5.1 Field Region

As mentioned before the field region definition is directly connected to the study goals that an ontology user wants to perform. Because of that two different field regions are executed trying to present two different approaches designed to cover a huge set of study types (Figure 6.7 and Figure 6.8). In order to define those regions four variables were used according to its limitations

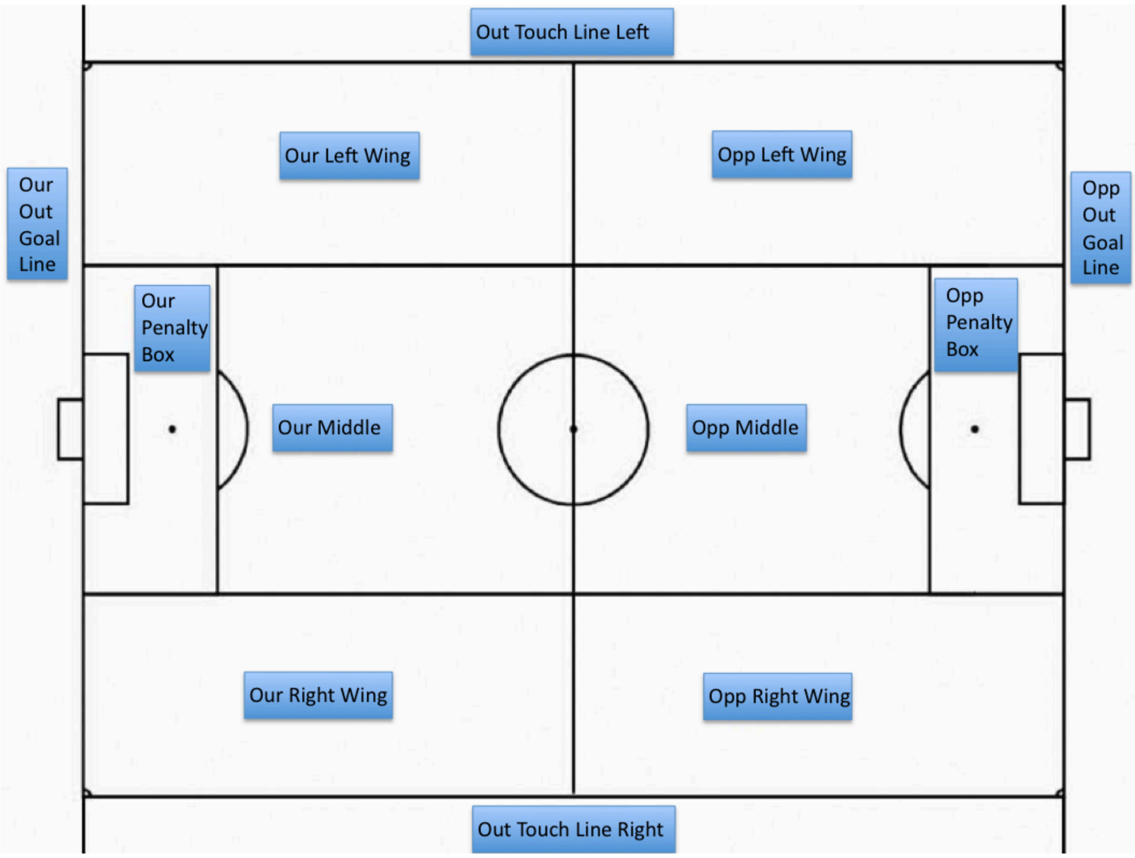


Figure 6.7: Field region definition using 14 distinct areas.

(6.2).

$$\begin{aligned} MinX &= ((FieldHeight \times MinXMultiplier) + MinXAdder) - MinXSubtractor \\ MaxX &= ((FieldHeight \times MaxXMultiplier) + MaxXAdder) - MaxXSubtractor \\ MinY &= ((FieldWidth \times MinYMultiplier) + MinYAdder) - MinYSubtractor \\ MaxY &= ((FieldWidth \times MaxYMultiplier) + MaxYAdder) - MaxYSubtractor \end{aligned} \tag{6.1}$$

To better understand these formulas an example using the Figure 6.8 will be executed. Using the official soccer field dimensions (105x68) (Figure 6.9) and the cartesian axis defined in Figure 6.5 we want to define the OurLeftFlankBack. The *MinX* and *MinY* have "0" as value. To calculate the other two variables we used the pre defined formulas:

$$\begin{aligned} MaxX &= ((105 \times 0.5) + 0) - ((105 \times 0.5) - 16.5) \\ MaxY &= ((64 \times 0.5) + 0) - (11 + 5.5 + 3.65) \end{aligned} \tag{6.2}$$

It is important to note that the 3.65 value is concerning to the goal line dimension ÷ 2.

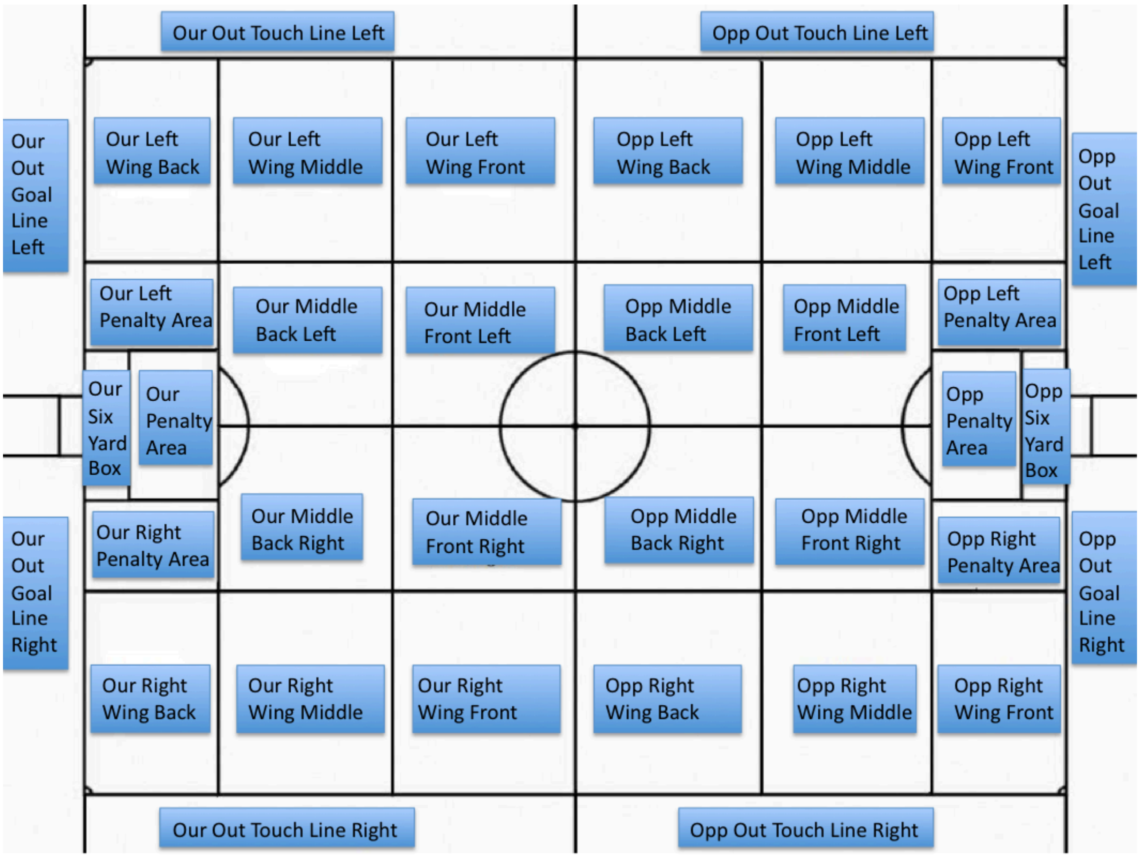


Figure 6.8: Field region definition using 36 distinct areas.

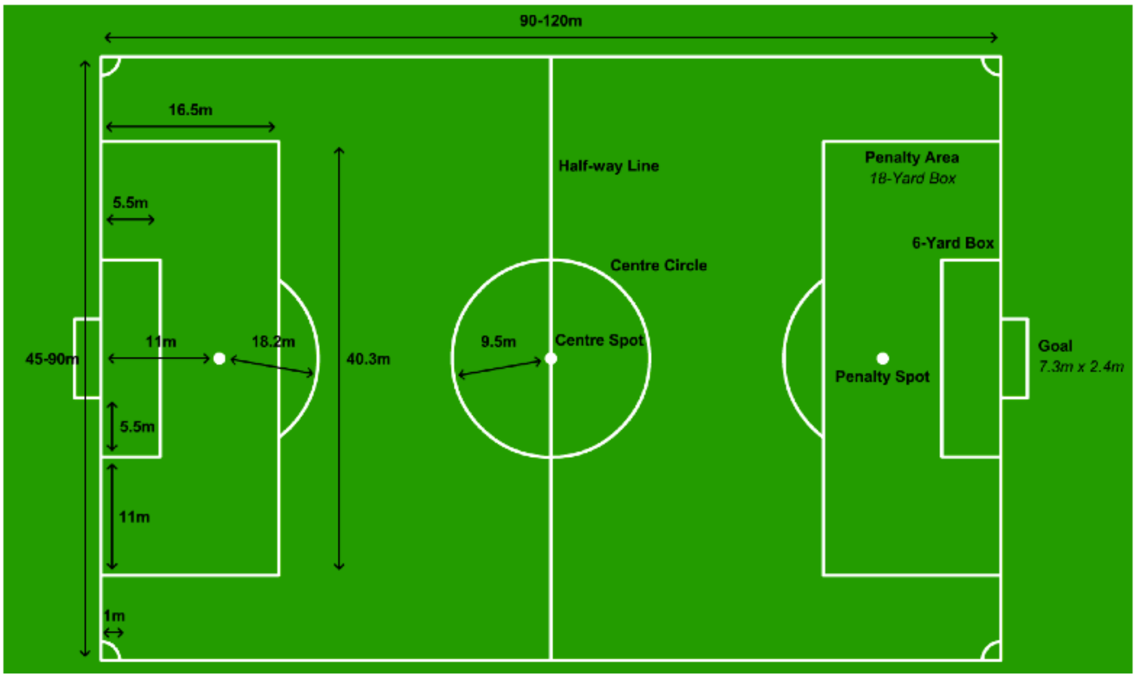


Figure 6.9: Soccer field official dimensions.

6.5.2 Variable Regions

Variable Regions correspond to small areas in the soccer field that help to characterize some game situations like penalty, corner, goal kick and kickoff situation (in the midfield region).

6.5.3 Team Regions

Normally to characterize a team formation some systems used non flexible concepts admitting that a team used a rigid formation and player positioning through the game. However, the reality is a slightly different. Because of that in this ontology a team region concept is defined. Excluded the goal keeper position (because it is a position that does not change much through the game), for the other players a super region is defined according to the Figure 6.10. This region is divided in

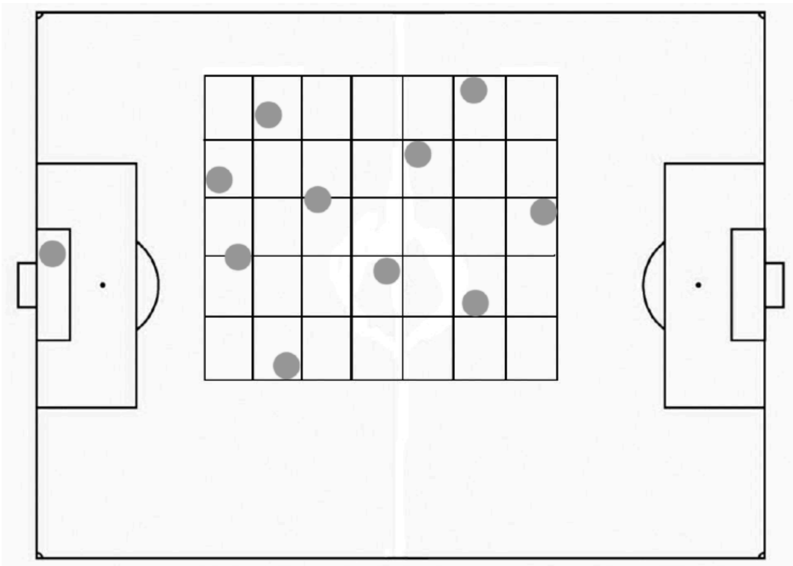


Figure 6.10: A Team Region.

a 7x5 matrix to simplify the classification of payer position and, can change in each game period.

6.5.4 Periods Definition

To better characterize a game situation or an event it is extremely important to create a period concept. This concept is directly related to a specific game time frame. In this ontology a period is divided in 3 distinct groups: Normal, Extra time and Penalties. The Normal and Extra time periods are also divided in First and Second Half and depending on the ontology use it is also possible to divided these periods in sets of 15 minutes . To characterize a period the ontology needs to obtain informations regarding to the Start and End time for each period. After that to evaluate a period the ontology only needs a real number included in a specific range as an input .

6.5.5 Events Definition

An event is composed by conditions and actions which are aggregated to players and the ball through the game. The conditions are divided in three groups [RL02]:

1. **Distance Conditions** which verify the relative distances between players and between players and the ball. Because of that this distance is divided in:
   (a) **Ball-Player** which is concerning to verify if the ball has a player at its range (which basically consists in verifying if a ball has a player in a distance less than one meter). If this condition is true it is possible to claim that a ball has a player at its range;
   (b) **Player-Ball** is the opposite condition (comparing to the previous analyzed) This condition will verify if a player has a ball at his range;
   (c) **Player-Player** which verifies if a player has another player (can be a teammate or a opponent player) at his range;
2. **Ball Conditions** which indicates the ball's state. This state can be one of six:
   (a) **Ball Touch** is when the ball changes Its direction without increasing its velocity;
   (b) **Ball Kick** is when the ball increases its velocity;
   (c) **Ball Possession** is verified in the period before a ball touch or between ball kicks. These kicks must be executed by the same player or between teammates;
   (d) **Air Ball** when the ball presents a z coordinate value higher than 0.0001 meters;
   (e) **Ball Out of the Field** when the ball is out of the playing field. In this situation a ball can go out through the side line or by the goal line;
   (f) **Stop Ball** when the velocity value of the ball is closest to zero;
3. **Player Conditions** define the player's state related to different player actions in the game. These actions can be divided in the following groups:
   (a) **Ball Possession** verified when a player executes a kick or a ball touch;
   (b) **Marking**. A player executes a marking action when he is the closest man regarding an opponent (independent to the opponent having the ball);
   (c) **Pressing**. A pressing situation is similar to marking one, however, in this case the opponent player must have the ball;
   (d) **Catch the Ball**. This situation occurs when a player catches the ball (following the game rules only the goal keeper has the ability/permission to catch it);
   (e) **Behind the Ball Line**. This situation occurs when a player is behind the ball line according to its coordinate (in his team attack perspective);
   (f) **Offside Position**. This situation occurs when a player in his team attack process is ahead of the last opponent defender. In this study the positional offside was not considered.

Regarding the Actions, they are the result of a previous condition (previously analyzed) successful validation process. So, in order to proceed an action, a set of initial and end conditions must occur. In this ontology a set of actions was defined (Figure 6.11). In the following list it is

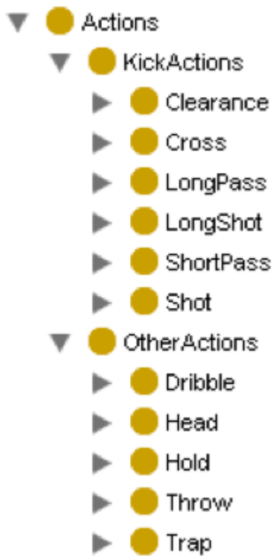


Figure 6.11: Hierarchical Representation of the Actions Classes.

executed a brief description of each actions defined in the ontology:

1. **Shot**. To classify an action as a shot a set of conditions must occur. the first one is concerning to an existence of a kick. After that, the direction of the ball (the ball needs to take the goal direction and its direction and velocity cannot allow any teammate to capture it otherwise this situation is classified as a pass) is validated as well as the field region where the kick occurs (if the kick happens more than 25 meters from goal this action will not be considered as a shot);

2. **Shooting distance**. Similar to shot, this action occurs when a player kicks the ball in a goal distance between 25-35 meters;

3. **Heading**. A heading is a particular case of the shot and it is execute when the ball has a positive z coordinate and a player increases its velocity;

4. **Short Pass**. A pass occurs when a payer kicks the ball in a teammate direction. This pass can be considered successful if the teammate, after some time, receives the ball (otherwise unsuccessful). A short pass is a pass executed in a less than 25 meters distance [Gar97b].

5. **Long Pass**. This action is a pass executed over a distance exceeding 25 meters;

6. **Crossing**. This action is a particular pass case. Normally a crossing is executed in the lateral regions with the goal of executing a pass to a teammate inside the penalty area.

7. **Throw-in** is an action that occurs after the ball goes out through the side line;

8. **Dribble**. This action occurs when a player increases the ball velocity several times in a row, overcoming his direct opponents (which are implementing successive markings);

These actions were classified according one of two values: Successful when it is verified the initial and end condition for a specific action or Unsuccessful in the opposite situation. This last situation can occur when for instance an opponent player intercepts a pass or when the ball is out of field.

6.5.6 Situations Definition

A situation can be interpreted as a setpiece that occurs during a game. Figure 6.12 illustrates the classes that are included in this concept. In a soccer game there are 6 different setpieces:

Figure 6.12: Hierarchical Representation of the Situation Classes.

1. **Kick off** situation appears at the beginning of each game period and after a goal is scored. The kick off involves two teammates in the midfield zone. After the ball leaves that zone this situation ends;

2. **Goal Kick** situation starts after a ball leaves the field through the goal line kicked by an attacking player. After that a ball kick is executed by a player and the situation ends;

3. **Penalty Kick** situation occurs when the ball is stopped in a certain time period in the variable region (inside the penalty area) and the closest player only has one opponent player in front of him–normally the goal keeper. This situation ends after the ball leaves the penalty area;

4. **Throw-in** starts after a ball leaves the field through a side zone and ends after the ball is inside again in the play field;

5. **Corner** situation occurs when the ball leaves the field through the goal line touched by a team defender;

6. **Free Kick** situation is probably the most complex one to detect. In this ontology a free kick occurs when the ball is stopped in a specific field region and the opponent team players (comparing to the closest ball player) are at a minimum ball distance of 9 meters.

Finally it is important to note that in the setpiece concept there is no differentiation concerning the direct or indirect free kick. Also the "Jump ball" situations were not addressed in this ontology.

6.5.7 Players Definition

In this section concepts related to the players positions, characteristics and behaviors are presented.

6.5.8 Players Position

At the beginning of a soccer game each player is assigned a field position. Normally these positions are related to the three soccer field sectors (defensive, mid and offensive) depending on the type of formation used by the team. The player position classification is based on team regions (previously defined) and during the game the players will be classified according to their position in the 7*5 matrix (Table 6.2).

<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>1</td><td rowspan="2">Right Back (RB)</td><td rowspan="2" colspan="3">Centre Back (CB)</td><td rowspan="2">Left Back (LB)</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td>Right Wing Back (RWB)</td><td colspan="3">Defensive Midfielder Centre (DMC)</td><td>Left Wing Back (LWB)</td></tr>
<tr><td>4</td><td>Right Midfielder (RM)</td><td colspan="3">Midfielder Centre (MC)</td><td>Left Midfielder (LM)</td></tr>
<tr><td>5</td><td>Right Att. Mid. (RAM)</td><td colspan="3">Attacking Midfielder Centre (AMC)</td><td>Left Att. Mid. (LAM)</td></tr>
<tr><td>6</td><td rowspan="2">Right Wing (RW)</td><td colspan="3">Striker (SS)</td><td rowspan="2">Left Wing Forward (LWF)</td></tr>
<tr><td>7</td><td colspan="3">Centre Forward (CF)</td></tr>
</table>

Table 6.2: Players Position Classification Through the Team Regions.

An example of that classification is illustrated in (Figure 6.13). Once again it is important to note that for this classification and, attending that a goal keeper is the most positional place in a soccer team, this is excluded from the classification. For the other players initials are used to classify their positions. Finally the ontology user can define if he classifies the players position through the game or only in some game situations excluding corner situation where there is usually a large cluster of players in the same region - making it difficult to classify their formation.

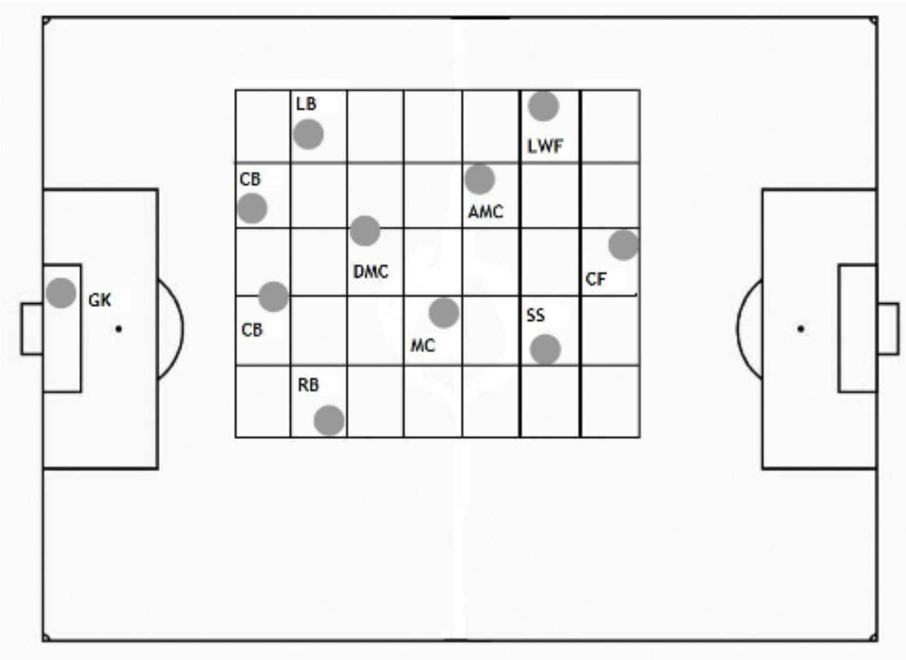


Figure 6.13: A Player Classification by Matrix Position.

6.5.9 Players Attributes

The players attributes used in this ontology are statistical results related to the actions executed by players through the game. This ontology provides only the detection of these actions, then the calculation of the attributes must be performed by another system.

The attributes are spread through four categories:

1. **Precision** which represents the players' action precision. This precision is the result of the actions player count (successful and unsuccessful) through the game;

2. **Capacity** represents the player's ability to execute a specific action;

3. **Velocity** is related to the speed actions implementation by the player. An example of that is the shot action, long pass action among others;

4. **Tendency** represents a behavior pattern of the player during the game. This attribute represents which is/are the actions that are performed more frequently by the player.

6.5.9.1 Precision

The action player precision is the percentage ratio between the successful and the unsuccessful actions. These actions can be dribbles, headings, shots, passes, throw-ins among others.

For this type of attribute we use a five value scale – Bad, Weak, Reasonable, Good, Excellent distributed in the following percentage:

1. **Bad** when less than 20% of the actions are successful;

2. **Weak** when 20%–40% of the actions are successful;

3. **Reasonable** when 40%–60% of the actions are successful;

4. **Good** when 60%–80% of the actions are successful;

5. **Excellent** when more than 80% of the actions are successful.

6.5.9.2 Capacity

The capacity represents the player ability to perform an action. Similar to the previous attribute, this feature is classified according to the same five value scale. The capacity capability is evaluated in five different situations:

1. **Reception Capacity** is related to the player's ball reception capacity. This evaluation will be higher depending on the number of the opponent players that the reception player has in his range in the moment that receives the ball. So the classification will be Bad if the player in the moment of receiving the ball did not have any opponent player at his range, Weak if the player has one opponent at his range, Reasonable if the player has two opponents at his range and so on until reaching the Excellent (the player has at least four opponent players at his range);

2. **Dribble Capacity** is concerning to the ability of the player to overcome a number of opponents in consecutive dribbles. The classification is identical to the one previously analyzed. One overcome opponent will be classified as Weak until he reaches the Excellent classification;

3. **Interception Capacity** is concerning to the player's capacity to intercept a pass or a shot (the classification process is similar to the previously analyzed);

4. **Tackle Capacity** is related to the player's capacity of recovering the ball, when it is controlled by an opponent player;

5. **Capacity to Catch the Ball** consists in evaluating the percentage ratio that the goal keeper has in terms of goalie catch.

6.5.9.3 Velocity

In a soccer environment it is important to classify the actions' velocity. However, to characterize the velocity it is important to calculate the player's maximum velocity with and without the ball and also the ball velocity in actions like short or long pass, shot or crossing. The same scale was used to characterized these two velocities (player and ball) – Very Slow, Slow, Normal, Fast and Very Fast. For the player these values are between 15 and 30 Km/h and for the ball the values are between 60 and 105 Km/h.

6.5.9.4 Tendency

The tendency of a player is regarding to the frequency that he executes an action in comparison with other actions. In this ontology some actions are selected in order to identify which is the most common one in a player perspective. These actions are shots, short or long passes, crossing, dribbles, runs.

6.5.10 Teams Definition

At the beginning of a soccer game, a specific team always presents a strategy which is composed by a formation, tactics among other things.

6.5.10.1 Team Formation

Normally a team formation is a number combination of 11 numbers that can vary between 11-0-0-0 until 0-0-0-11. In this ontology in order to classify the team formation is used the previously analyzed player position classification which simplifies this task. An example of a formation classification is a team that presents 4 defenders, 3 strikers and 3 midfielders. For this case the formation used for the team is 1-4-3-3 however, this formation can suffer some changes with the increase of the analysis granularity regarding for instance the midfielders position. One of the midfielders can be more offensive than the others and in that case the formation classification will be 1-4-2-1-3. Also as the goal keeper is the a statical position in the field, normally the first digit in the formation formula is omitted.

6.5.10.2 Teams Tactics

Similar to player attributes in order to process the team tactics the ontology needs discrete data. However as mentioned previously the main goal of this work is to understand the relations between all the concepts that included a soccer match. To better characterize the team tactic 7 different groups were defined (Figure 6.14):

1. **Pass Style (Short, Mixed and Long)** is concerning to the percentage ration of each type of pass that a team most executes. If a team presents at least 65% of total passes are short this means that the team pass style is a pass short . On the opposite way if at least 65% of total passes are long the pass style will be the long pass and the style will be classified as mixed;

2. **Game Rhythm (Very Slow, Slow, Normal, Fast, Very Fast)** is regarding the rhythm of the offensive actions. In other words this tactic consists in determinating the time spent by a soccer team in its offensive movements.

3. **Field Use (Right Flank, Left Flank, Both Flanks, Middle, Mixed)** is related to the field region that a team most uses in its attacks movements;

4. **Risk Degree (Very Safe, Safe, Normal, Risky, Very Risky)** is related to the number of players that a team puts ahead of the ball line in a defensive situation. If a team uses a high

Figure 6.14: Different Tactical Parameters for a Soccer Team.

risk tactic also the probability of conceding a goal will be increased. In consequence of that a Very Safe tactic is when all the team players are behind the ball line, Safe 10 players behind the ball line, Normal between 8 and 9 players behind the ball line up to Very Risky that less than 7 players are behind the ball line;

5. **Mentality (Very Defensive, Defensive, Normal, Offensive, Very Offensive)** is concerning to the tendency that a certain team has to attack or defends a game result. In the ontology this definition is related to the number of players that a team puts ahead of the ball line in an attack situation;

6. **Defensive Line (Retreated, Normal, Advanced)** is related to the position of a team in its defensive moment. If the defensive line of that team is retreated it means that the last defensive man (before the goal keeper) is at a distance of 20 meters of his goal line. For a Normal defensive line the last man is between 20-40 meters and for the last case the last man is more than 40 meters of distance regarding his goal line;

7. **Team Width (Close, Normal, Large)** is regarding to the position of the players in a offensive situation. If in the Y coordination the distance of the opposite teammates is less than 40 meters the Team Width is close, otherwise if the distance is between 40-55 meters the Team Width will be classified as Normal; more than 55 meters will be classified as Large.

6.6 Results

In this section, a set example of tests performed during the development of the ontology will be presented as well as the validation of Its concepts.

6.6.1 Ontology Development Process

During the development of the ontology many tests were performed based on ontology queries. Those queries were executed using the SQWRL language (similar to SQL language for a database).

6.6.1.1 Object Players Queries

In this section some tests related to the players attributes are illustrated. Table 6.3 represents an example of some ontology objects as well as their cartesian coordinates. Together with this information also the field dimensions (105*68) were inserted in the ontology.

Object	Current X	Current Y
Ball	4	4
OurPlayer02	3.5	4.5
OurPlayer03	17	3
OppPlayer02	5.5	4
OppPlayer03	16.5	6

Table 6.3: Cartesian Coordinates Values for the Simulated Objects.

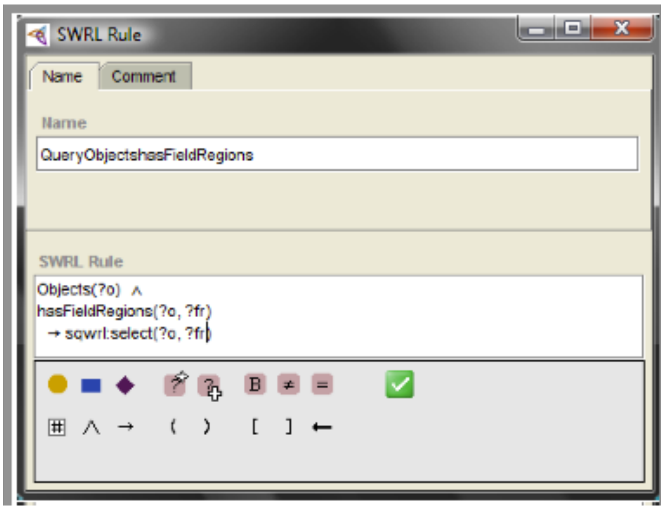
One of the most basic question is to ask the ontology about the region where each object is inserted (Figure 6.15(a)). Analyzing the objects positions it is easy to see that all the objects are in the same field region less the *OurPlayer03* (Figure 6.15(b)).

In order to validate the relative distances between objects two new questions were created (Figure 6.16(a)).. The first one is related to the distance conditions of the simulated objects (Figure 6.16(b)). and the second is related to the player's conditions that are applied to the distance conditions (Figure 6.16(c)). Analyzing the results presented in Figure 6.16(b), it is important to note that the ball only has in its action radius the *OurPlayer02* which means that this player is less than a meter of the ball. On the other hand the same player has in his action radius the ball and the *OppPlayer02* which means that he is less than three meters regarding the *OppPlayer02*. Finally the *OppPlayer03* has the *OurPlayer03* in his actions radius.

Regarding the Figure 6.16(c) the players' conditions are illustrated. The *OurPlayer02* is the BallOwner, the *OppPlayer03* is performing a marking action because he is part of the opposite team (in terms of ball perspective). The *OppPlayer02* is at the same time performing a marking and also a pressing action because he is closest to the *BallOwner* and finally the only player that is not behind the ball line is the *OurPlayer03*.

6.6.1.2 Object Team Queries

As mentioned before, in this ontology many team proprieties were defined (Figure 6.17). In order to test those properties a set of objects and its cartesian coordinates were created as illustrated in Figure 6.18. The teams are represented using the red (our team in ontology perspective) and blue color (opponent team). To validate some team properties a set of queries were executed as



(a) SWRL Question for Identified the Field Regions of the Simulated Objects

Object	Field Region
Ball	OurLeftFlankBack
OurPlayer02	OurLeftFlankBack
OurPlayer03	OurLeftFlankMiddle
OppPlayer02	OurLeftFlankBack
OppPlayer03	OurLeftFlankBack

(b) Field Regions for Simulated Objects

Figure 6.15: SWRL Question for Identified the Field Regions of the Simulated Objects (a), Field Regions for Simulated Objects (b)

illustrated in Table 6.4. Doing a brief analysis both teams presented the same width however the

Expression	Name	Result
TeamMinX(OurTeam,?minx)->sqwrl:select(OurTeam,?minx)	OurTeamMinx	40
TeamMaxX(OurTeam, ?maxx)->sqwrl:select(OurTeam, ?maxx)	OurTeamMaxX	87
TeamLength(OurTeam, ?leng)->sqwrl:select(OurTeam, ?leng)	OurTeamLength	47
TeamMinY(OurTeam, ?miny)->sqwrl:select(OurTeam, ?miny)	OurTeamMinY	13
TeamMaxY(OurTeam, ?maxy)->sqwrl:select(OurTeam, ?maxy)	OurTeamMaxY	51
TeamWidth(OurTeam, ?width)->sqwrl:select(OurTeam,?width)	OurTeamWidth	38
TeamMinX(OppTeam, ?minx)->sqwrl:select(OppTeam, ?minx)	OppTeamMinx	53
TeamMaxX(OppTeam, ?maxx)->sqwrl:select(OppTeam, ?maxx)	OppTeamMaxX	90
TeamLength(OppTeam, ?leng)->sqwrl:select(OppTeam, ?leng)	OppTeamLength	37
TeamMinY(OppTeam, ?leng)->sqwrl:select(OppTeam, ?miny)	OppTeamMinY	14
TeamMaxY(OppTeam, ?maxy)->sqwrl:select(OppTeam, ?maxy)	OppTeamMaxY	51
TeamWidth(OppTeam, ?width)->sqwrl:select(OppTeam, ?width)	OppTeamWidth	37

Table 6.4: Queries Representation of Teams Proprieties.

length of the team that is defending is lower compared to other team.

The values presented in the min and max properties are extracted from the coordinates of the

Name	Expression
QueryRangeConditions	hasRangeConditions(?obja,?objb)->sqwrl:select(?obja,?objb)
QueryPlayerConditions	hasPlayerConditions(?player,?cond)->sqwrl:select(?player, ?cond)

(a) SWRL Question for Identified the Field Regions of the Simulated Objects

SQWRLQueryTab | QueryRangeConditions | QueryPlayerConditions

?obja	?objb
OppPlayer02	Ball
OurPlayer02	Ball
Ball	OurPlayer02
OurPlayer03	OppPlayer03
OppPlayer03	OurPlayer03
OurPlayer02	OppPlayer02
OppPlayer02	OurPlayer02

(b) Distance Condition Results

SQWRLQueryTab | QueryRangeConditions | QueryPlayerConditions

?player	?cond
OppPlayer03	Marking
OurPlayer02	BallOwner
OppPlayer03	BehindBallLine
OppPlayer02	Marking
OppPlayer02	Pressing
OurPlayer02	BehindBallLine
OppPlayer02	BehindBallLine

(c) Players Condition Results

Figure 6.16: SWRL Question for Identified the Field Regions of the Simulated Objects (a), Distance Condition Results (b) and Players Condition Results (c)

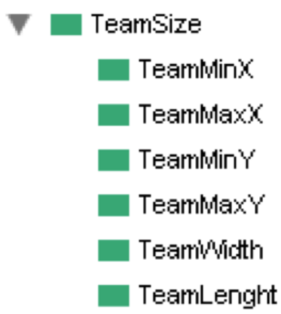


Figure 6.17: Representation of Team Proprieites.

players. Example: to determine the *OppTeamMinx* the all opponent team players x coordinates

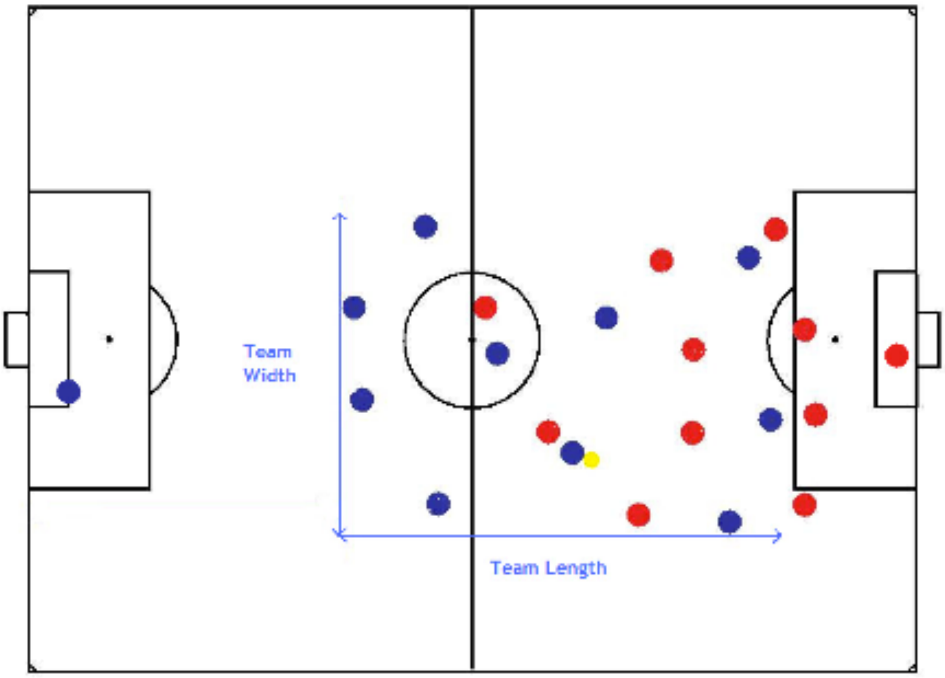


Figure 6.18: Representation of a Static Game Situation.

are compared. The smallest player value will be the *TeamMinX*. Another information that can be validated from the ontology is the team tactics. As mentioned before, in order to test these concepts a set of values must be inserted in the ontology: the number of players behind the goal line and the number of players that are positioned on the flank regions. Figure 6.19 illustrates the sqwrl expression to query the team's tactics and Figure 6.20 presents the results of that query.

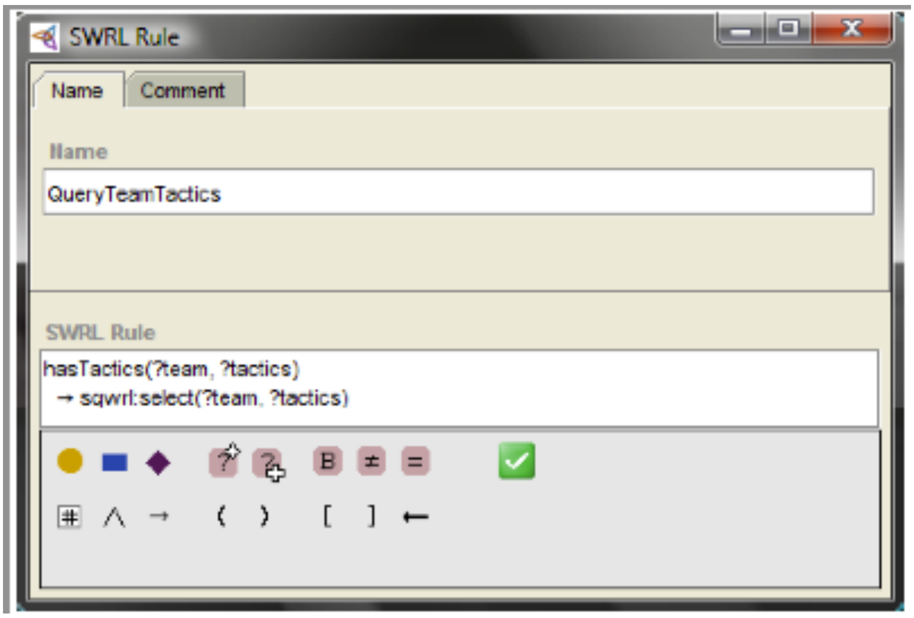
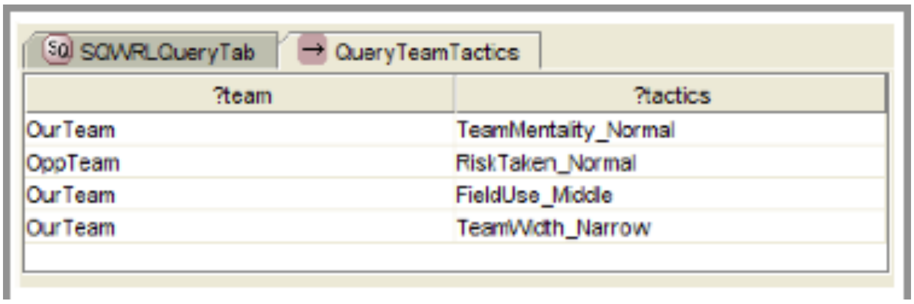


Figure 6.19: SWRL Question for Identified Team Tactics.



?team	?tactics
OurTeam	TeamMentality_Normal
OppTeam	RiskTaken_Normal
OurTeam	FieldUse_Middle
OurTeam	TeamWidth_Narrow

Figure 6.20: Team Tactical Results.

Analyzing Figure 6.20 the opponent team only presents the normal risk degree because the ball is not in adjacent ares (in the penalty area perspective) and this team puts 9 players behind the ball line. *Our Team* presents a normal mentality because 7 players are behind the ball line. The use of the field is in the centre because less than 3 players are in the flank region and finally the team width is narrow (less than 40 meters).

6.6.2 Concepts Validation

In order to validate the ontology concepts and their relations a survey was sent to soccer experts (a full view of this survey can be seen in the Apendix section). This experts' board is composed by Academic Professors as well as Master Degree students in Sport Science in the area of Soccer. The survey took about 30 minutes to answer and 60 experts have completed it.

6.6.2.1 Actions Types Analysis

The goal of this section is to validate the actions that are capable of representing the soccer game such as passes and shots and also some defensive actions like pressing movements.

Concerning the classification of the pass types: 50% of the experts considered the short pass as the pass that is executed for distances of less than 10 meters; 20% defended that this pass is executed between two consecutive regions and finally 20% of them answered that it is a move from "foot to foot".

Regarding the long pass, 40% stated it that is a pass executed from a distance of 30 meters, 30% defended that is a pass executed from a distance of 20 meters, 10% considered that it is a pass executed from a distance of 10 meters and finally 20% claimed that it is an executed pass between two non consecutive regions. Doing a comparison between the experts answers and the concepts implemented in the ontology it is relevant to note that more than 50% agree with the long pass concept. However in terms of short pass and due to the inexistence of a medium pass concept the majority of soccer experts claimed that a short pass is a pass that is executed for distances of less than 25 meters. As mentioned before this concept difference is eliminated with a medium pass concept creation.

In the survey a question about the zones that a player can execute a cross was elaborated. For those purposes Figure 6.21 was created to better clarify the concept. Regarding the situation

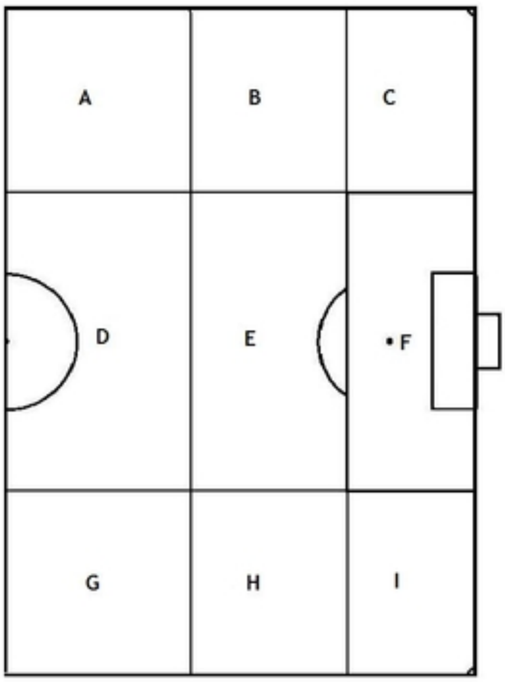


Figure 6.21: Midfield Regions Divisions to Validate the Crossing Concept.

represented in the Figure 6.21, 100% of the experts claimed that the executed passes in Zone C to Zone F should be considered as crosses; 63% of the same experts considered that the executed passes in Zone B to Zone F should also be considered as crosses. Other passe zones are considered to validate the crosses: Zone C to Zone E (38%), Zone B to E (25%) and Zone A to F (25%) and finally Zone C to I (13%).

The answers that obtained the highest percentage regarding the crossing definition meet those concepts defined in the ontology: an executed pass in the Zone C or B to Zone F should be considered a crossing.

To validate the defensive actions Figure 6.22 was created considering that the blue team was in defensive process. Using this situation as a base, the experts had to classify 14 sentences according to 4 values: Totally Disagree, Partially Disagree, Partially Agree and Totally Agree. Regarding the final results, 13 of the 14 situations were validated by the experts which composes excellent indicators in what concerns to ontology definitions.

6.6.2.2 Player Attributes Analysis

In order to evaluate the players' tendency in executing an action, 3 ranges were defined in the ontology (25%, 50% and 75% of the total actions). More than 75% of the experts chose the first range (25% of the total actions) as the value that is capable to indicate a player tendency.

Similar to the defensive actions, in the players attributes the experts can classify the sentences using one of four values. The results of that classification are illustrated in Table 6.5 and the answers that match the ontology concepts are represented using a bold font. Doing a brief analysis

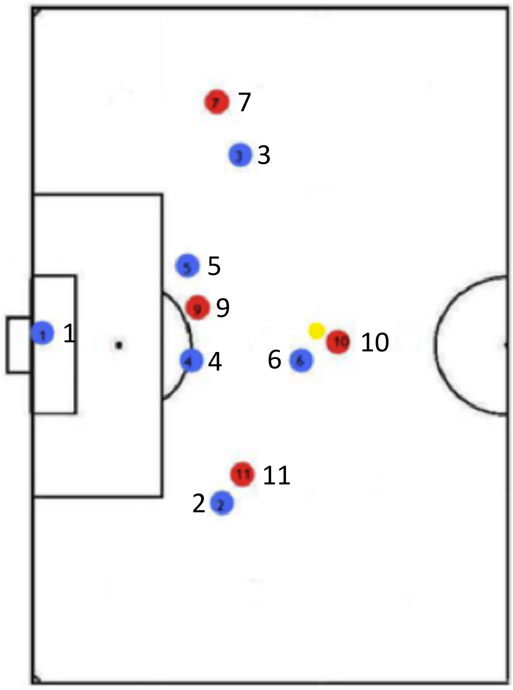


Figure 6.22: Defensive Actions for a Specific Game Situation.

	Totally Disagree	Partially Disagree	Partially Agree	Totally Disagree
A player who shots well inside the area, and also shots well far away	**100%**	0%	0%	0%
A player which is good in executes long shots, he is usually good to executes crosses.	**43,35%**	**16,56%**	40.09%	0%
A player with good dribbling, usually has good ball control	**0%**	**50,75%**	49,25%	0%
A player who makes good short passes, also he is a good player to make long passes.	**82.88%**	**17,12%**	0%	0%
A good player to mark corners, usually he is a good player in crosses execution.	17,65%	0%	**82,35%**	0%
A good player to make long passes is also good to make crosses.	**53,7%**	**20,2%**	26,1%	0%

Table 6.5: Evaluation of Players Characterization.

of the obtained results it is important to note that only the third sentence is inconclusive otherwise the other answers match with the ontology's concepts.

6.6.2.3 Strategy Team Analysis

One of the concepts that was addressed in the strategy of the team was risk degree of a team during a game. The first approach consists in the experts validating this concept according to the number of players that a team would use behind the ball line. The results are illustrated in Table 6.6 (using the bold font to indicate the match with the ontology concepts). Once again, the answers match the

Risk Degree	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Very Safe							100%
Safe				21,67%	18,33%	**60%**	
Normal			16,67%	**41,67%**	**41,67%**		
Risky		21,67%	**40%**	38,33%			
Very Risky	**45%**	**36,67%**	18,33%				

Table 6.6: Assessment of the risk degree of a team.

concepts represented in the ontology. Regarding this topic (player attributes) two more concepts were validated: Defensive line Style (Table 6.7) and Team-Width (Table 6.8).

Normal Defense Line	**Results**
Between 20 and 30 meters	37,67%
Between 20 and 40 meters	**32,45%**
Between 30 and 40 meters	29,88%

Table 6.7: Defensive Line Style Concept using as reference the distance between the last defender and his goal line.

Team-Width	**Results**
Between 30 and 45 meters	11,34%
Between 35 and 50 meters	19,34%
Between 40 and 55 meters	**53,43%**
Between 45 and 60 meters	15,88%

Table 6.8: Team-Width Concept.

Regarding the results obtained by these two concepts it is easy to note that the Team-Width concept was totally validated while the defensive line style presented inconclusive results.

6.6.2.4 Positioning and Formation Analysis

In order to validate the positioning and formation concepts the following images were presented to the experts (Figure 6.23 and 6.24). The obtained results for positioning and formation classification are represented in (Table 6.9 and 6.10). The positioning results showed that in spite of the fact that some times the comparison between the visualization concepts and cartesian coordinates

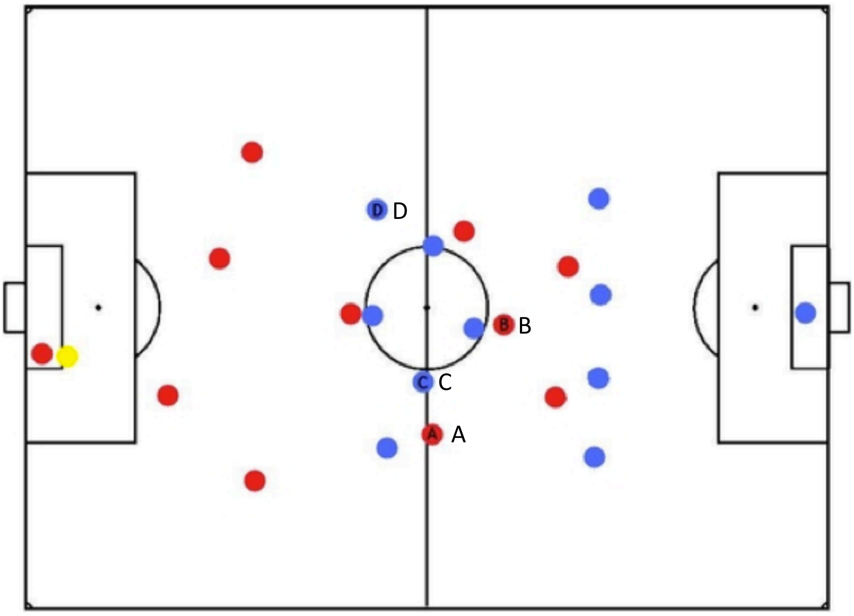


Figure 6.23: Game situation using two teams: blue and red team.

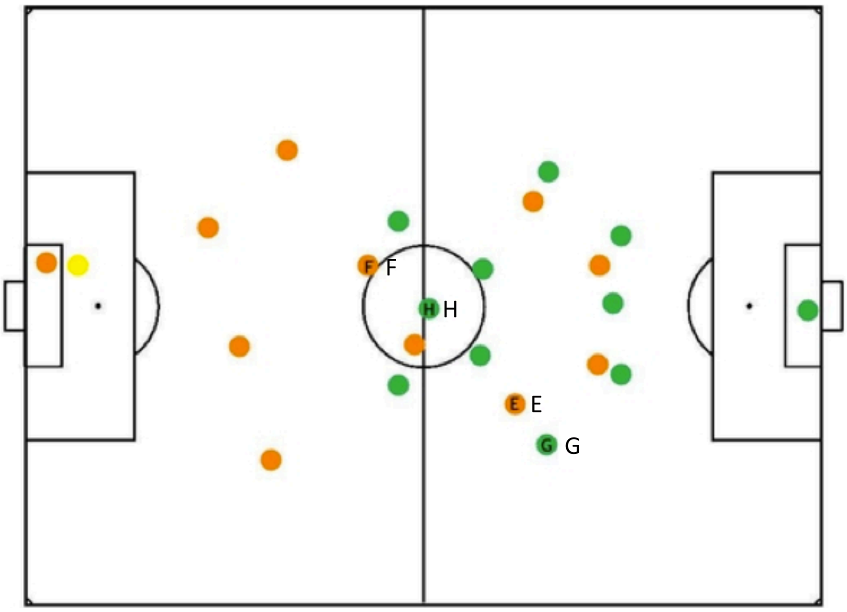


Figure 6.24: Game situation using two team: orange and green team.

concepts are different, in this case only one concept did not match the one proposed in the ontology. Concerning the team formation validation only one concept did not match the one presented in the ontology.

Players Position	**Results**		
Player A	**Right Midfielder-48,75%**		Right Back Midfielder-51,25%
Player B	**Attacking Midfielder Centre-100%**		
Player C	**Left Back Midfielder-100%**		
Player D	**Right Winger-100%**		
Player E	**Right Midfielder-51,25%**	Attacking Midfielder Centre-17,43%	Right Winger-31,32%
Player F	**Midfielder Centre-100%**		
Player G	**Left Back-67,87%**		Left Midfielder-32,13%
Player H	**Attacking Midfielder Centre-100%**		

Table 6.9: Positioning Results of the four teams.

Formation	**Red Team**	**Blue Team**	**Orange Team**	**Green Team**
4-4-2	**65,45%**		48,76%	
4-1-3-2	34,55%			
4-3-3		**82,65%**		
4-1-2-3		17,35%		
4-2-4			34,4%	
4-2-2-2			**16,84%**	
5-3-2				31,2%
5-2-1-2				**33,23%**
3-5-2				16,2%
3-4-1-2				19,37%

Table 6.10: Formation Results of the four teams.

6.7 Conclusions

In this section cartesian coordinates based tool capable to automatically calculate final game statistics and a soccer ontology capable to represent some high-level concepts were proposed. For the first case, the results presented (using cartesian coordinates extracted from 2D simulation league games) were very encouraging, showing that it is possible to extract reliable information from the soccer game only by using Cartesian coordinates.

Regarding the soccer ontology all the concepts represented were interactively validated in the development process and after that a validation survey method was used. Because of that a board composed by 60 soccer experts (composed by full academic professors and MSc students in sport science) were used to validate the ontology concepts. The majority of the concepts (80%) represented in the ontology were fully validated and consequently the goal of this work was totally achieved. One final note regarding the Protégé framework and *OWL* language that have proven to be mature enough for representing high-level concepts.

For this research work, as mentioned before, the goals were totally achieved however it will be interesting to test this ontology in an integrated soccer system capable to identify soccer actions, team formations and other reliable information.

Chapter 7

Statistics Generation

As mentioned previously, the choice of data to be used in this PhD thesis was not an easy task. To obtain data related to the human soccer many attempts have been performed. The first attempt consisted in communicating through email to the official soccer entities as FIFA and UEFA to at least obtain data regarding to a barely visible competition like the U17 World Cup or a pre-round of Europe League games (until today we did not obtain any feedback information). The second approach consists in contacting some reliable clubs like Manchester United asking for soccer training data [1]. The last attempt consisted in directly contacting companies who work on game analysis such as Deltatre [2] but unfortunately we were not successful. In consequence of that we tried to generate our own data using a tracking system based on Wi-Fi technology (explained in chapter 5). In spite of the satisfactory results to expand this approach covering all the soccer field including the ball tracking did not compose an easy task. So we decided to use for this thesis the log files provided by the RoboCup 2D simulation league which are accessed through a public domain.

In this chapter, two soccer statistics generator tools (Soccer Scientia and SSSET) will be presented capable to automatically calculate final game statistics using the RoboCup 2D simulation league logfiles. The first one consists in calculating statistics using only players and ball Cartesian Coordinates (obtained through the log files) [AMS+10] while the second approach based its knowledge in the information gathered by the soccer server [PARG10]. Some of the calculated statistics were based on the previous concepts defined in the SocOn ontology (presented in the previous chapter). Finally the results were presented and a comparison between the robotic and human soccer was performed [ACCa+10b].

[1] We contacted by email Professor Carlos Queiroz but unfortunately we did not receive any feedback

[2] More informations available at `http://www.deltatre.com/`

7.1 Soccer Analysis Tools

Over the past years, many systems were capable of calculating final game statistics in both realities: human (analyzed in chapter 5) and robotic soccer. In this section an overview of robotic soccer analysis tools will be executed.

With the emerging of RoboCup and more specificaly with the creation of leagues such as simulation 2D, the robotic developers found in the field of performance tools a valid solution for measuring and improving the performance of their teams in the competition. In consequence of that, through the years, many were the performance tools that emerged:

1. **SoccerScope2** (Electro-Communications Japan University) [soc04] is an open source analysis tool written in java. This tool is capable of automatically calculating a huge set of statistics and at the same time works as a monitor allowing the visualization of a log file information (only supports 2 and 3 log file format). Finally, SoccerScope2 contains a well-structured code which composes a good basis for future developments (Figure 7.1).

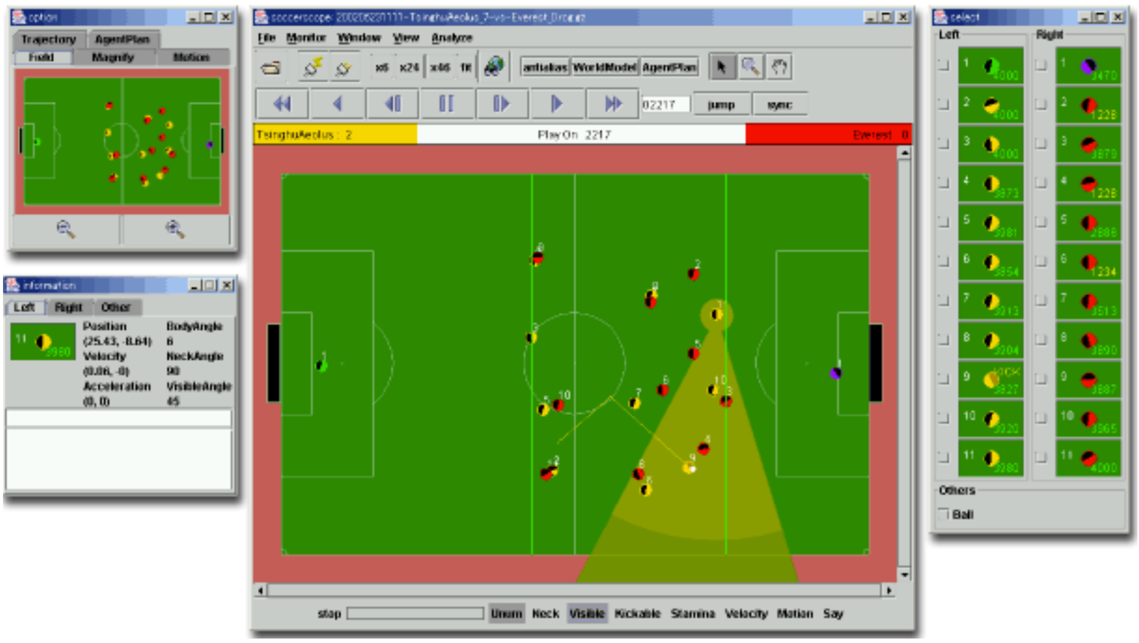


Figure 7.1: SoccerScope2 Interface.

2. **Team Assistant - SBC** (Shahid Beheshti, Iran University) [ZSS+03] is composed by three main modules: a logplayer that allows the user to visualize different players attributes such as their vision and typical movements. The graphical debugger allows the graphical representation of players movements and finally the game analyzer that through the information collected on the soccerserver is capable of calculating a huge set of statistics such as ball possession, passes, dribbles and shoots (Figure 7.2).

A comparison table between the analyzed tools can be seen in Table 7.1. It is important to note that none of the analyzed tools is capable to calculate high level statistics.

Figure 7.2: Team Assistant Interface.

Name	Strengths	Weakness
SoccerScope2	-Includes a highly developed graphic module; Capable of calculating a huge set of statistics	-Generated statistics are still very basic
Team Assistant	-Both individual and Collective Statistics; Realistic 3D Viewer	-Generated statistics are still very basic
SSIL Statistics	-Capable of calculating many different statistics; XML statistics output includes a XSLT for convenient HTML visualization	-The set of generated statistics is still incomplete and did not represent a good spectrum of team performance
Logalyzer	-Capable of calculating individual and collective statistics	-The set of generated statistics is still incomplete

Table 7.1: Comparison of the Robotic Soccer Analysis Tools.

7.2 Statistics Definition

In order to load the robotic log files, both developed statistics tools used a module of the Soccer-Scope 2[3] software. Written in Java, this software presents a well structured code with an extensive

[3]More information available online at http://ne.cs.uec.ac.jp/~koji/SoccerScope2/index.htm

and comprehensive documentation, that provides new users with a rapid learning curve. However, it supports only version 3 of the 2D Simulation League log files (2007 competition or earlier). In order to use the 2009 log files in this research work, which are created using version 4, a tool present in the RoboCup Soccer Simulator Log Player package[4] is used to convert them into version 3.

The majority of soccer events, with the exception of those related to game breaks, such as faults or forced breaks to provide assistance to an injured player, have similarities. At the origin of this kind of events is always an increase of the ball velocity or a change in the direction of ball's motions (named a *kick*), which can represent various events, like a pass, shot, and so on. Equation (7.1) shows this concept, where t_1 and t_0 are instants of time and V_{ball}, D_{ball} are ball velocity and direction respectively.

$$kick\,(t_0) \leftarrow (||\vec{V_{ball}}\,(t_0)|| < ||\vec{V_{ball}}\,(t_1)|| \vee D_{ball}\,(t_0) \neq D_{ball}\,(t_1)) \wedge t_1 = t_0 + 1 \tag{7.1}$$

For both developed tools, the soccer game information was organized in an array (Figure 7.3), each position representing a cycle in the game (in the robot soccer, a game consists of 6000 cycles, so in this particular situation the array will have 6000 positions), and containing information about the players and ball position, player energy and vision, among others.

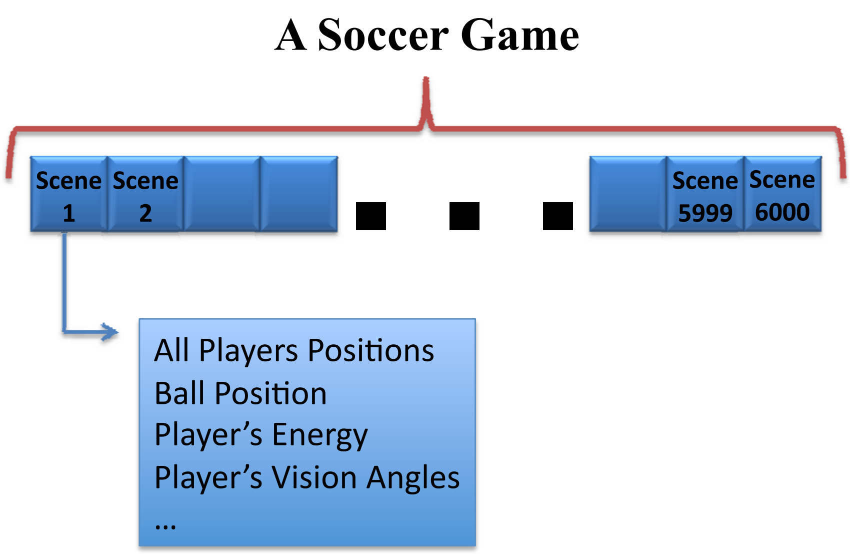


Figure 7.3: Soccer Game Structure.

To calculate final game statistics some classes were defined in order to establish the soccer field structure and ball dimension (class SoccerRegion) and also to define the calculation rules for such statistics. These rules are based in three conditions (Start Condition, Constraint and Final Condition, varying from event to event), where each is applied to two distinct Scenes (two different

[4]More information available online at http://sourceforge.net/apps/mediawiki/sserver/index.php?title=Download

positions in the game vector). The class Statistics is responsible for the calculation of all statistics (Process method) which analyzes the game vector according to the nature of the events and the three conditions mentioned before. When all statistics are calculated, the results are displayed in a cherry (Java Swing Domain-Specific Language) based interface.

Before the game analysis can be performed, the user has to specify which are the events he wants to be detected. After that, the sequential array analysis starts. In the first loop through the array, all *kicks* are detected and marked, representing the possible events that occurred in the game. In the second step, the positions following any given *kick* (and before the next detected *kick*) are analyzed according to second and third rules (constraint and final condition, respectively, which normally leads to a reduction in the number of events (many *kicks* correspond to dribble events, which are not detected by the tools developed) (Algorithm 2). Also, a class for each event type was created, as to allow a more flexible and modular internal programming architecture, which also enables the application to take advantage of parallel processing capabilities.

Algorithm 2 Generic Event Detection Algorithm

```
for Cycle i = 0 to max_Cycles − 1 do
  if (kick.start_condition(scene[i], scene[i+1])) then
    addKick(i)
  end if
end for
for all Event Class event do
  for all Cycle i in Kicks do
    for j = i+1 to nextKick do
      if (! event.constrain(scene[j])) then
        break
      end if
    end for
    if (event.final_condition(scene[j])) then
      addEvent(event, i, j)
    end if
  end for
end for
```

7.2.1 Soccer Field Regions

With the purpose of increasing the information quality of the statistics calculation and to better define some soccer concepts, several regions were defined as illustrated in Figure 7.4 having as a base the ones defined previously in the SocOn ontology (defined in the previous chapter).

For each midfield (defensive and attacking), a set of 16 regions were defined. All these regions are described using relative coordinates and suffer a 180 degree rotation for the second half of the game. Also, some global variables are defined, like ball size (relevant in goal detection situations) or margins of the field (important to detect events that occur when the ball exits the playable area).

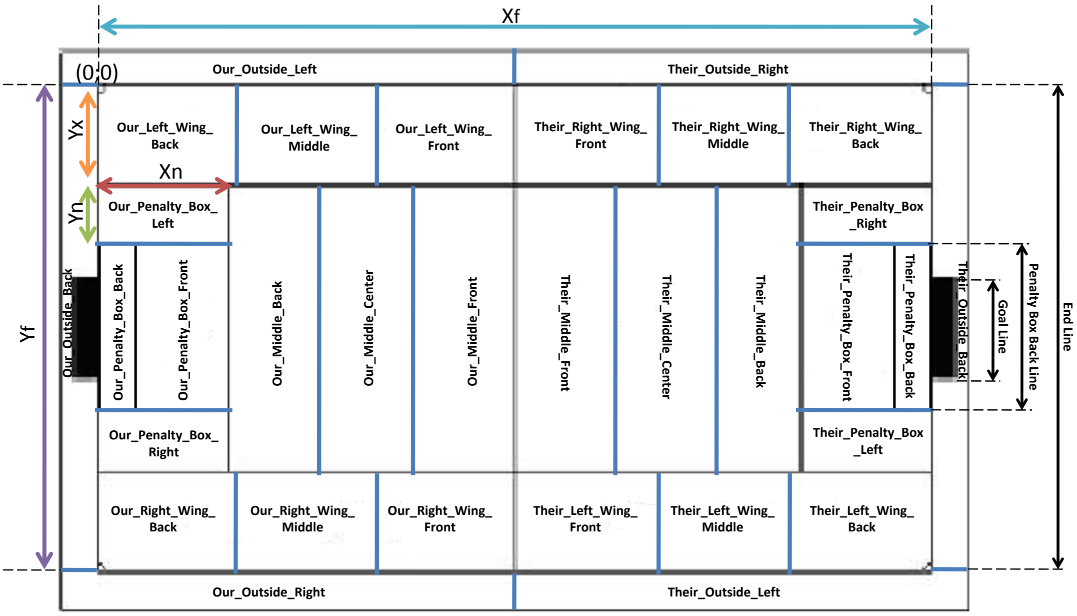


Figure 7.4: Soccer Field Regions Divisions.

In order to transform the process of specifying the soccer field division into a more flexible one, 5 dynamic variables (X_n, X_f, Y_x, Y_n, Y_f) were created, as illustrated in Figure 7.4, allowing the user to quickly change the previously establish divisions without loosing features in the soccer heuristics. X_f and Y_f represent the field dimensions, X_n the penalty box length, Y_n the penalty box wing width and Y_x the wing area width. Other areas, like the middle areas (back, center and front) or the wings (back, middle and front), are obtained using these variables as a basis.

7.2.2 Soccer Heuristics

In this section, all soccer events presented in the developed softwares are explained and justified.

7.2.2.1 Pass Information

Generically, in a soccer game, there are two types of passes – successfully executed ones, which means that a pass is well performed between two teammates, and missed passes (sometimes called wrong passes), which results in a ball recovery by the opponent team.

Having the *kick* as a basis, the detection of a successful pass consists in identifying two consecutive *kicks* performed by teammates, and detecting the moment (between the two *kicks*) when the second player receives the ball, as depicted in Equation (7.2), where P_0 and P_1 are the passer and receiver players, respectively, P is a generic player, t_0, t_1 and t_2 time instants, and b is the ball.

$$SuccessfulPass(P_0, P_1, t_0, t_1) \leftarrow KicksBall(P_0, t_0) \wedge KicksBall(P_1, t_2) \wedge SameTeam(P_0, P_1) \wedge$$
$$t_1 > t_0 \wedge t_2 > t_1 \wedge P_0 \neq P_1 \wedge (\neg\exists (P,t) : t > t_0 \wedge t < t_2 \wedge KicksBall(P,t)) \wedge$$
$$(\neg\exists (P,t) : t > t_1 \wedge t < t_2 \wedge P \neq P_1 \wedge dist(b,P) < dist(b,P_1)) \tag{7.2}$$

The main issue in this type of events is to detect when a player receives the ball. The initially adopted solution was based in a proximity algorithm, that after detecting the first *kick*, analyzes the area within a circle, centered on the ball (which reduces the number of possible players to test for possession), and detects the player closest to the ball (which will be considered the ball receiver). In spite of this solution being used in many research works (and often considered as a classical one), some issues still remain. The main problem consists, in some occasions, in the detection of the correct receiver when other players are near the ball's trajectory. This situation is the result of a wrong correlation between ball possession and ball proximity concepts, which have different meanings. In consequence of that, the initially used algorithm was adapted into a new approach, based in the detection of two consecutive *kicks* (Algorithm 3).

Algorithm 3 Pass Detection Algorithm

```
for all Cycle i in Kicks do
  cycle ← false
  originalKicker ← getPlayerKicking(i)
  secondKicker ← getPlayerKicking(nextKick(i))
  if originalKicker ≠ secondKicker ∧ sameTeam(originalKicker, secondKicker) then
    for j = nextKick − 1 to i + 1 do
      for all Player p in Players do
        if p ≠ secondKicker ∧ distance(p, Ball, j) < distance(secondKicker, Ball, j) then
          receiveCycle = j + 1
          cycle ← true
          break
        end if
      end for
      if cycle then
        break
      end if
    end for
    AddSuccessfulPass(originalKicker, secondKicker, i, receiveCycle)
  end if
end for
```

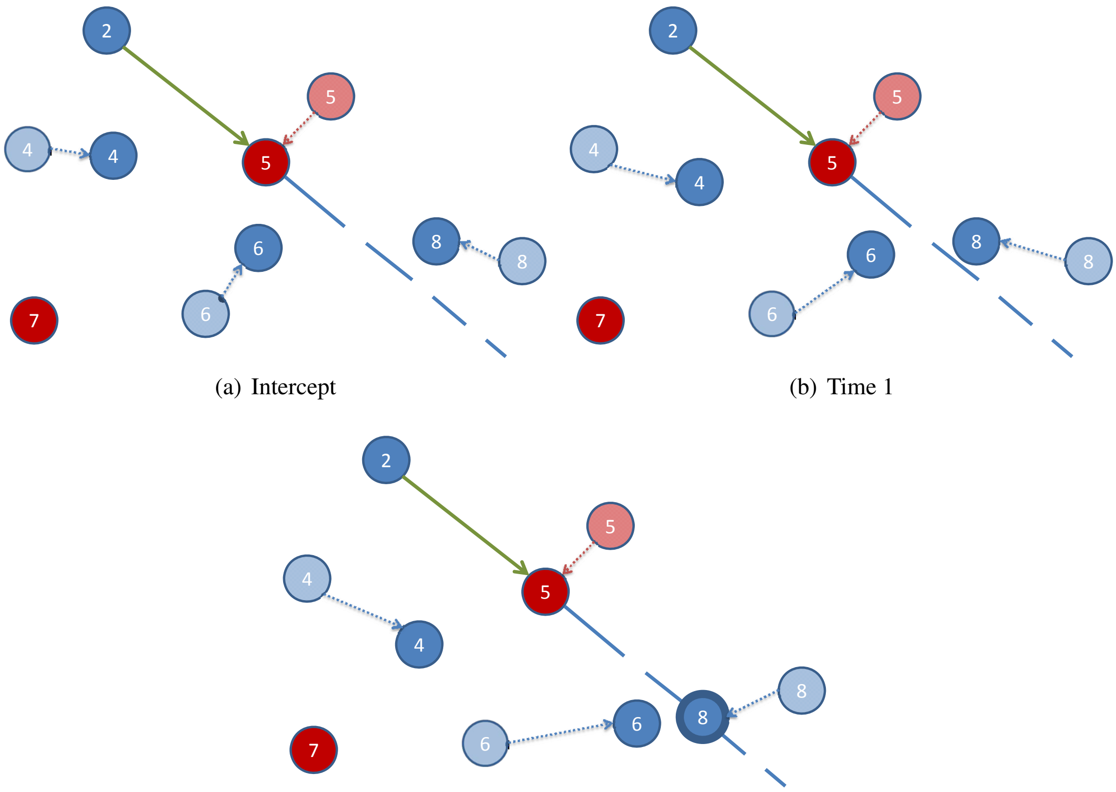
After receiving or intercepting the ball, the player, after some cycles, will execute a *kick* (which can be a pass, a shot, or even dribble). It's assumed that if a player does not touch the ball, he never had it under his control. So, to detect the existence of a pass, the analysis was centered in detecting a *kick* from the possible receiver of the ball. After detecting the second *kick*, the only issue remaining is to find when he gained possession of the ball, which can be calculated by performing a reverse temporal analysis and applying the proximity algorithm discussed above. If this player is the same that performed the first *kick*, the event is a dribble (not included in this analysis); if the player belongs to the same team as the one who performed the first *kick*, a successful pass is detected; otherwise (the player does not belong to the same team), a missed pass is detected, which can represent several distinct scenarios, explained below.

In the case of a missed pass being detected, in most situations (except for a shot in the direction of the goal line), it is important to determine which was the player that would most likely have received the ball. For those purposes, an algorithm was developed based on ball motion detection and prediction. The path traveled by the ball (between the first *kick* and the moment when either a second player receives the ball or it leaves the play field) is used to determine its direction and velocity, which are then used to simulate its path (from the moment the first *kick* was executed and beyond the receiver player or up to the play field outline, respectively). With this information, the distance between the ball and every player (of the same team as the one that performed the *kick*) is calculated for each cycle of the path–*BallPath* (Algorithm 4), and a probability of a player reaching the ball is associated. This probability is inversely proportional to the distance between player and ball, and includes a multiplying factor that decreases with the temporal distance from the initial *kick* (given that long passes are usually less likely to occur, especially in robot soccer). The players' positions considered in this algorithm are determined considering an ideal motion towards the ball, in an interception course. If only one player could actually intercept the ball, it is chosen as the most probable receiver for the ball; otherwise, the player with the highest probability of intercepting the ball is chosen.

Algorithm 4 Distance Calculation Algorithm

```
IniCycle ← Cycle of Initial Kick
for all Cycle c in BallPath do
   for all Player p in Teammates do
      dist ← distance(p, Ball, c, IniCycle)
      addToProbabilityVector(p, Probability(dist, c − IniCycle))
   end for
end for
return HighestProbability(ProbabilityVector).Player
```

A more graphical example of this algorithm is represented in Figure 7.5(a), 7.5(b) and 7.5(c). In this particular scenario, the blue player (number two) tries to execute a pass for a teammate. However, the ball was recovered by a red player (number five) (Figure 7.5(a)). In consequence of that, according to the ball movement, the algorithm traces a virtual path that would have been executed by the ball if it had not been intercepted by the red player. After that, and as easily seen in Figures 7.5(b) and 7.5(c), for each possible ball position, in each cycle, the distances between the ball and each player are calculated according not only to the previous player position, but also to an estimate for a new player position in a specific cycle. This situation is very interesting because if the algorithm was only based on the distance between the players and the ball, the teammate that is closest to the ball is player number six (in the first instance after his team lost the ball –



(a) Intercept (b) Time 1

(c) Time 2

Figure 7.5: Example of Distance Algorithm

Figure 7.5(b)). However, analyzing the ball movement, and doing an estimation for all player movement, the conclusion is that the player that had a higher probability to receive that pass was player number eight.

Another variation of pass which, was detected in the SSSET tool (explained later), is the "wing chain" and "pass chain". In order to detect the "wing chain event", the soccer field was split in the three equal regions/corridors: left, middle and right. If the event algorithm detects a successful pass between two teammates and if this pass occurs between the left and the right regions (or vice-versa), the algorithm will classify it as a successful pass and a "wing chain". The "pass chain event" consists in identifying the number of consecutive successful passes that a team is capable to do in a match.

For the same statistical generation tool two more features were developed "ball possession" and "temporal sequences". Regarding the first concept, it is considered that a team has the possession of the ball in a given interval of time whether, during that time, none of the opponent players intercept the ball and the ball does not leave the play field. The second new concept consists in evaluating the time a team spent to achieve the last third of the field without losing the ball. This information is extremely relevant in order to classify the offensive style that a team is using in a game. This classification is divided into four levels: slow, medium, fast or break (depending when the opponent team recovers the ball).

7.2.2.2 Shot Information

A shot occurs when a player shoots the ball, in his attacking field, in the general direction of the opponent's goal line, and with enough initial velocity that allows the ball to reach it. Equation (7.3) shows these conditions, where $\vec{A_{ball}}$ represents the acceleration of the ball, $Vel\,(b,t_0)$, $Pos\,(b,t_0)$ represents the ball velocity and position in an instance t_0 respectively and, P represents a player.

$$Shot\,(P,t_0) \leftarrow Belongs\,(P,team) \wedge KicksBall\,(P,t_0) \wedge InRegion\,(Pos\,(P,t_0)\,,AttackField\,(team)) \wedge \left(\exists\,(t) : Pos\,(b,t_0)_X + Vel\,(b,t_0)_X\,t + \frac{\vec{A_{ball}}_X t^2}{2} > X_f \wedge Pos\,(b,t)_Y > 0 \wedge Pos\,(b,t)_Y < Y_f\right) \tag{7.3}$$

Three distinct events are detected that can be considered a shot: a Shot on Target, an Intercepted Shot and a Shot. A Shot on Target occurs when a player kicks the ball in the goal direction and the kick has enough strength to make the ball reach the goal line adding tolerance (0.5 meters) around the goal for each side. On the other hand, if the ball does not have the goal direction as defined previously (after kicked by a player) but still leaves the field through the Penalty Box area (and is not in conditions to be considered a shot on target), this event will be marked as a shot. Finally if an opponent player intercepts the ball after the first player kicked the ball (with all the conditions to be classified as a shot target or a shot), this event will be classified as an intercepted shot.

7.2.2.3 Goal Scoring

To win a soccer match, one team needs to score at least one more goal than its opponent. A goal occurs when the ball, in its totality, passes through the goal line. Implementation constraints, however, make this condition insufficient by itself – the ball, after being kicked by a player, can occupy the same position it would after a goal, but crossing through the penalty box back line instead. So, the detection of the goal must also analyze if the ball actually intercepted the goal line before completely leaving the play field, using the path of the ball for that purpose (see Algorithm 5). It is important to note that, when a goal is detected, no other event is considered (namely the Shot on Target and the Outside events). Algorithm 5 demonstrates the differences between a Goal, a Shot on Target and a Shot.

Another concept presented in the SSSET tool is the "goal opportunities" which basically consists in the identification of the situations where an attacking player has a large probability to score a goal. For this matter, a goal opportunity is defined by creating a virtual triangle having the attacking player and the opponent team goal poles as vertexes (Figure 7.6) and checking if the number of the opposing team players inside that area are less than two.

Algorithm 5 Goal, Shot on Target and Shot Detection Algorithm

```
for all Cycle i in Kicks do
  kicker ← getPlayerKicking(i)
  for j = i+1 to nextKick(i) − 1 do
    BallInitialPosition = BallPosition(i);
    BallFinalPosition = BallPosition(j);
    if InRegion(BallPosition(j+1), OutsideBack) then
      if Intercepts(BallInitialPosition, BallFinalPosition, GoalLine) then
        AddGoal(kicker, i)
      else if Intercepts(BallInitialPosition, BallFinalPosition, GoalLine+0.5) then
        AddShotonTarget(kicker, i)
      else if Intercepts(BallInitialPosition, BallFinalPosition, PenaltyBoxBackLine) then
        AddShot(kicker, i)
      end if
    end if
  end for
end for
```

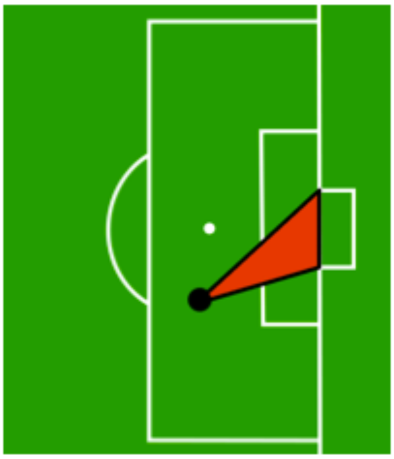


Figure 7.6: The Goal Opportunity Triangle Area – the black dot represents the attacking player

7.2.2.4 Outside Information

Nowadays, in both Human and Robotic Soccer, due to the competitiveness of the teams, the 'set pieces' compose an important role[Gar97a]. When a ball moves outside the play field, three situations can occur: throw-in, corner or goal kick. The only factor that distinguishes these situations is the region where the ball left the play field. In order to detect which team will have ball possession, the algorithm analyzes which player executed the last *kick* before the ball left the play field – Algorithm 6 (the *calculateOutsideType*() function determines the type of outside: throw-in, corner or goal kick).

Algorithm 6 Outside Detection Algorithm

for all Cycle i in $Kicks$ **do**
  $kicker \leftarrow getPlayerKicking(i)$
  for $j = i+1$ to $nextKick - 1$ **do**
    if $InRegion(BallPosition(j), Outside)$ **then**
      $outsideType = calculateOutsideType(BallPosition(j))$
      $AddOutside(kicker, i, outsideType)$
    end if
  end for
end for

7.2.2.5 Offside Information

The offside rule was established in 1924 and is probably the most difficult real-time situation to detect in a soccer match. In this work, an offside is defined in a similar way to a pass event. However, to detect this event, two conditions must be verified (Algorithm 7): a successful pass must be detected and the player that receives the ball must be in an invalid field position at the moment of the pass. A position is considered invalid to receive a pass if the receiver, at the moment of the pass, is the last player before the goalie, and the ball's motion has a component in the direction of the end line. Also, a player is only considered to be in an invalid position if he is in his attacking midfield.

During the game, if an offside situation occurs but the referee does not mark it (because an opponent player intercepts the pass), this type of event is classified as an Intercepted Offside (position validation rules are applied to the receiving player) – see Algorithm 7. This algorithm uses the Algorithm 4 to calculate which is the receiver that is more closest to the ball.

Algorithm 7 Offside Detection Algorithm

for all Cycle i in $Kicks$ **do**
  $originalKicker \leftarrow getPlayerKicking(i)$
  $secondKicker \leftarrow getPlayerKicking(nextKick(i))$
  if $!sameTeam(originalKicker, secondKicker)$ **then**
    $receiver = DetermineReceiver()$
    if $InvalidPosition(receiver, i)$ **then**
      $AddInterceptedOffside(receiver, i)$
    end if
  end if
  if $originalKicker \neq secondKicker \wedge sameTeam(originalKicker, secondKicker)$ **then**
    if $InvalidPosition(secondKicker, i)$ **then**
      $AddOffside(secondKicker, i)$
    end if
  end if
end for

7.3 Soccer Scientia Tool

In this section, the modules which compose the soccer scientia tool (the tool exclusively based in the players and ball Cartesian Coordinates) will be presented and also a results validation will be executed. It is important to note that the main goal of this work is to create a tool capable to automatically calculate final game statistics using only Cartesian Coordinates.

7.3.1 Soccer Scientia Modules

The Soccer Scientia software starts to ask the user which will be the input data. After that the user can choose which are the statistics that want the software to calculate. After that this tool will present three distinct modules: Animation, Chart and Table (Figure 7.7).

7.3.1.1 Animation Module

The animation module is composed by three distinct areas: the viewer area where the user can see all the game actions, the event detection area where it is possible to choose a specific event in a time frame and starts to analyze the game from that point and finally the video area where a user can jump to a specific cycle in the game, rewind the game etc (Figure 7.7 – marked with yellow lines). As mentioned before in the viewer area an user can see a faithful game representation.

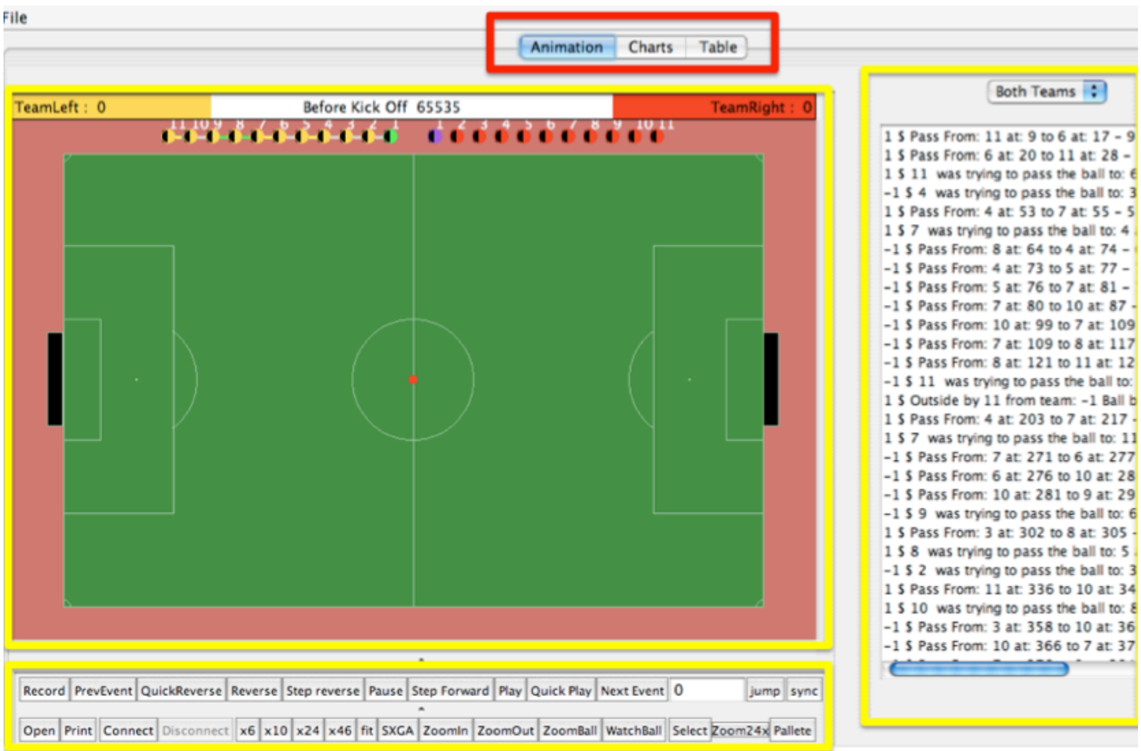


Figure 7.7: The Three Soccer Scientia Main Modules – marked with a red line

However, this software provides another type of high-level information like a force model or a historical of executed passes. The force model is regarding to the behavior of a team in a game more precisely to the pattern of passes between its players (Figure 7.8). In order to identify those

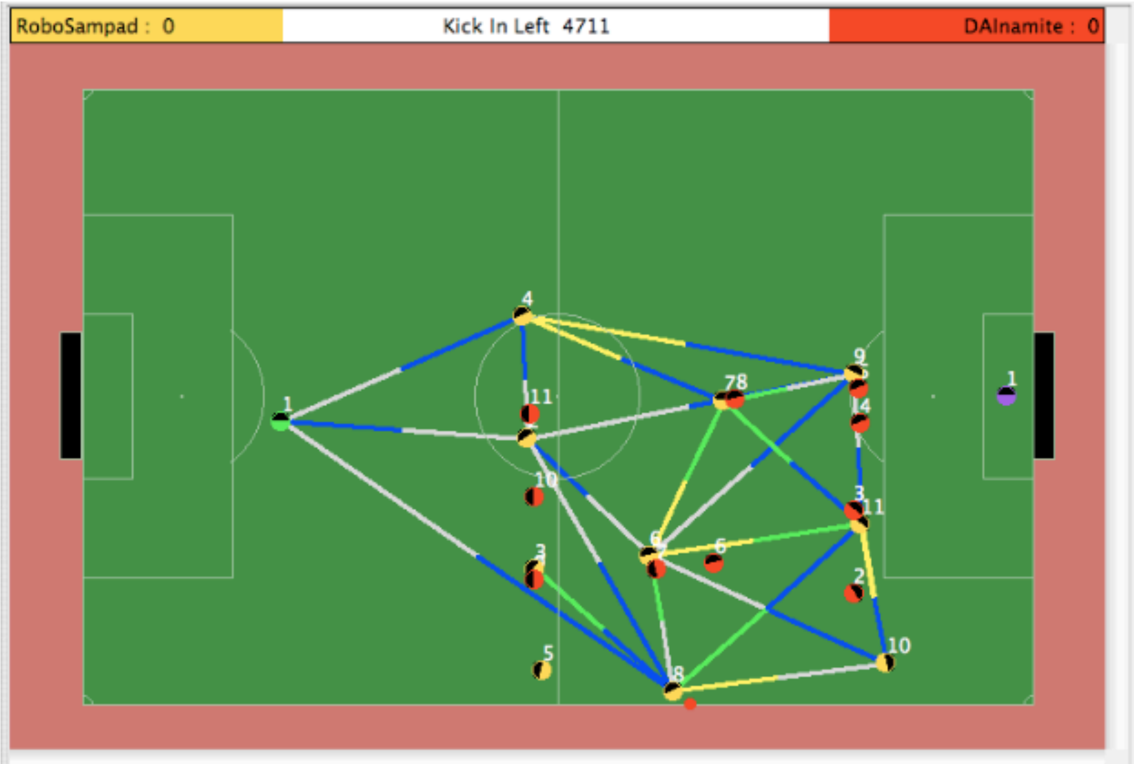


Figure 7.8: An example of a Force Model

patterns a gradient with a set of defined colors was used to identify the frequency of passes between teammates (the light grey means 0 passes, blue color is between 1 an 3 passes, green color is between 3 and 6 passes and yellow color more than 6 passes). It is important to note that in this work, this detection is bidirectional which means that it is possible to measure the pass frequency between each player of the team (also the color gradient used is totally configurable). For example in Figure 7.8 it is easy to note that player 8 executes more than 1 successful passes to his goal keeper however his teammate did not execute, until that moment, any successful pass to him. In what concerns the history of executed passes this tool provides a graphical representation of successful passes executed by the players in the respectively field zone (Figure 7.9). These two types of high-level information are extremely important for a soccer coach not only to analyze his next opponent but also to measure his team and players performance.

7.3.1.2 Chart Module

The chart module is related to the individual players performance. In this module a user/coach can easily analyze the performance of his players concerning to the final game statistics such as frequency of successful passes, shots on target, goals scored etc.(Figure 7.10).

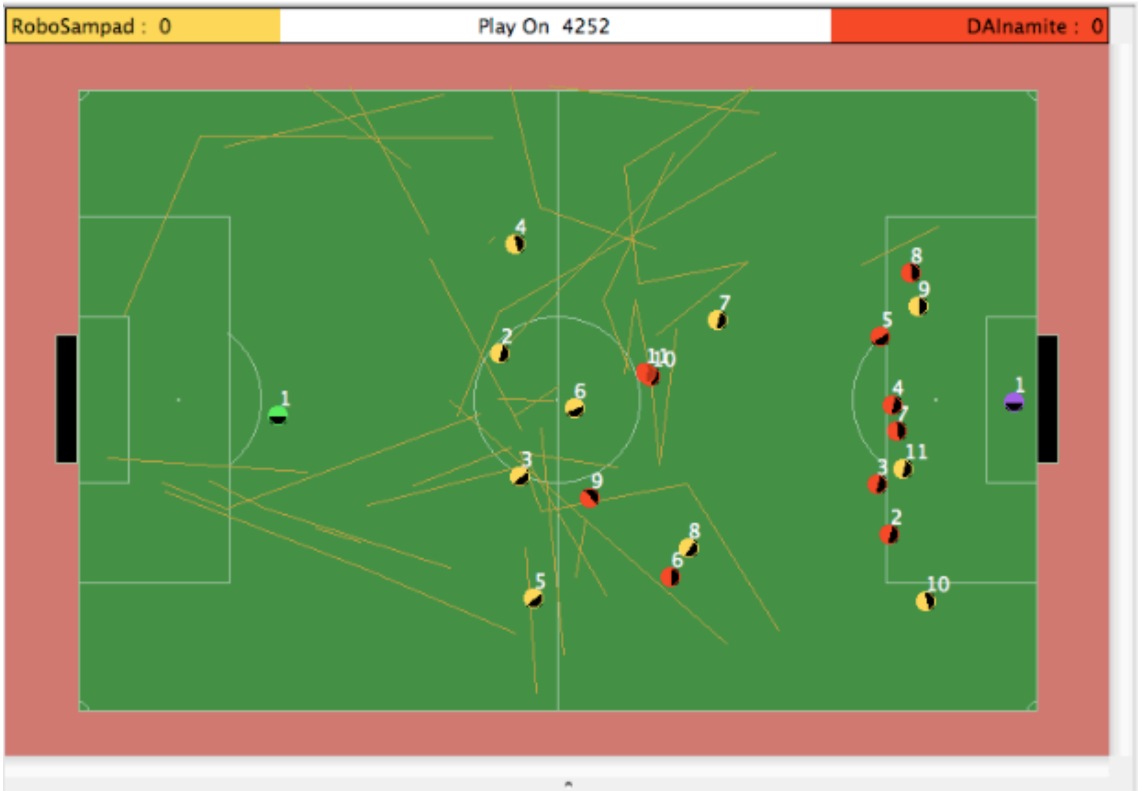


Figure 7.9: An example of a team successful pass representation through the soccer field

Figure 7.10: An example of the frequency of players successful passes

7.3.1.3 Table Module

Finally in the table module the soccer scientia software presents the collective team performance in final game statistics shape (Table 7.2).

Animation | Charts | Table

	Left Team	Right Team
Pass	64	155
PassMiss	44	43
Shoot	6	14
ShootIntercepted	0	5
ShootTarget	0	4
Goal	0	1
Outside	6	11
Offside	0	0
OffsideIntercept	0	0

Table 7.2: An example of a collective team performance

7.3.2 Results

In order to validate the soccer concepts described in the previous section, a set of twelve 2D Simulation League games (six from 2007 and six from the 2009 competition) were selected. The criteria for choosing these particular games were the competition phase where they occurred and the variety of results that were observed: high goal differences (four games), small goal differences (less than three goals – five games), and draws, with and without goals (one and two games, respectively). Also, the set of selected final game statistics was divided in four distinct groups: Pass, Shot, Offside, Goal and Outside. Validation was performed having a manual classification process as a basis – this classification was done by a board of soccer experts.

Starting with a global analysis of the events selected for detection during the analyzed matches, Table 7.3 shows that successful passes are the most common event in the games, followed by missed passes (with less than half of the occurrences of the former). All other events present a much lower frequency, with outsides standing out as the most frequent of these events (approximately 16 outsides per match).

	Successful Passes	Missed Passes	Shot	Intercepted Shot	Shot on Target	Goal	Outside	Intercepted Offside	Offside
High Goal Difference (4)	234	105.75	7.5	12	5.00	10.75	13.75	2.25	2.75
Small Goal Difference (5)	227	100.2	3.4	3.4	1.00	1.8	15.20	3.00	4.20
Draw (3)	155.67	75.67	3.0	4.33	3.33	0.67	23.00	0.67	2.67
Average (12 Games)	211.5	95.92	4.67	6.5	2.92	4.5	16.67	2.17	3.33
Standard Deviation	39.88	25.07	5.48	8.61	5.55	6.36	7.09	2.76	3.47

Table 7.3: Average Frequency of Events

Regarding pass results (both successful and missed passes), as can be seen in Table 7.4, more than 98% of the total successful passes and more than 96% of missed ones were correctly detected (with a standard deviation of 2.09% and 4.39% and presenting an average of 3,33 and 3,08 of passes observed but not detected per game named Difference in Table 7.4, respectively). It is important to note that validated missed passes include not only a correctly detected missed pass but also the most likely destination player. Thus, these results compose good indicators, especially regarding two complex factors in calculation, such as the detection of ball possesion and the calculus of the destination player in the missed pass.

	Successful Passes				Missed Passes			
	Percentage Detected	Average of Sucessful Passes Observed	Difference	False Positive	Percentage Detected	Average of Missed Passes Observed	Difference	False Positive
High Goal Difference (4)	99.25%	234	1.75	0.25	96.93%	105.75	3.25	0.25
Small Goal Difference (5)	98.15%	227	4.2	0.2	98.00%	100.2	2	0.2
Draw (3)	97.43%	155.67	4	0.33	93.83%	75.67	4.67	0.33
Average (12 Games)	98.34%	211.5	3.33	0.25	96.60%	95.92	3.08	0.25
Standard Deviation	2.09%	39.88	3.98	0.45	4.39%	25.07	3.53	0.45

Table 7.4: Accuracy Results for Successful and Missed Passes

Regarding the different groups of games, one can see that matches with a higher number of goals also have a higher number of passes, both successful and missed, with those matches that ended in a draw presenting a considerable difference in comparison to the other games. Also, the average number of total successful passes observed is approximately 2.1 times higher than missed ones.

In what concerns shots, Table 7.5 shows the results for End Line, Intercepted and Target Shots, where one can see that the average detection percentages are between 74 and 85%.

	Shot				Intercepted Shot				Shot on Target			
	Percentage Detected	Average of Shots Observed	Difference	False Positive	Percentage Detected	Average of Intercepted Shots Observed	Difference	False Positive	Percentage Detected	Average of Shots on Target Observed	Difference	False Positive
High Goal Difference (4)	70%	7.5	2.25	0.75	70.83%	12	3.5	0.5	85%	5	0.75	0.25
Small Goal Difference (5)	76.47%	3.4	0.8	0.4	82.35%	3.4	0.6	0.4	80%	1	0.2	0
Draw (3)	77.78%	3	0.67	0.33	76.92%	4.33	1	0.66	90%	3.33	0.33	0
Average (12 Games)	74.64%	4.67	1.25	0.5	77.16%	6.5	1.67	0.5	84.17%	2.92	0.42	0.083
Standard Deviation	21.69%	5.48	0.97	0.522	26.60%	8.61	2.67	0.522	24.69%	5.55	0.90	0.288

Table 7.5: Accuracy Results for End Line, Intercepted and Target Shots

These results can be explained by two main reasons, intrinsically linked to the robot soccer reality. The first reason, that explains the low number of shots that occur during matches, is related to robot soccer team strategies, that invariably attempt to score a goal by a combination of passes until the player is almost sure of reaching the objective of scoring. The second reason, that justifies the software's lower detection rate, is related to a somewhat rare situation that can occur during the match – two players can be at equal distance from the ball, when a *kick* is detected, both in a position to kick the ball, as depicted in Figure 7.11 (both players 2 and 7, of opposing teams, could have kicked the ball, resulting in completely different events being detected). This makes it extremely difficult for Cartesian coordinates based system to accurately detect all *kicks*. Given that shots are far less common than passes, these situations assume a higher percentage of occurrences, thus contributing to the presented results.

Table 7.6 presents the statistics related to goal and outside detection. All fifty-four goals scored in all twelve analyzed games were successfully detected by the software.

In relation to the outsides, more than 93% of all situations were also confirmed. As mentioned, outsides include three situations: throw-ins (when the ball leaves the field through one of the side

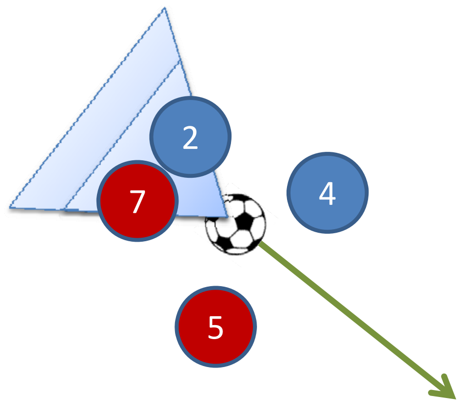


Figure 7.11: *Kick* Player Detection

lines), corner kicks and goal kicks (when the ball leaves the field through one of the end lines). The goal kick situation includes all End Line Shots and part of Target Shots (the other part being when an opponent player catches the ball in the Penalty Box Back Area).

It is important to note that the few situations the software was unable to automatically detect were all related to specific game situations that lead the RoboCup Soccer Simulation Server to mark an outside even before the ball crosses the field line. If the server detects that the ball's trajectory is leading it outside the play field, and that no player can reach it in time, an outside is marked, and the ball either moves to the destination place (corner, goal kick place, or the side line) or stops before leaving the field. The outside is announced through a server status variable, and the game continues, thus making it difficult for a Cartesian coordinate based system to correctly identify the event.

	Goal				Outside			
	Percentage Detected	Average of Goals Observed	Difference	False Positive	Percentage Detected	Average of Outsides Observed	Difference	False Positive
High Goal Difference (4)	100%	10.75	0	0	92.73%	13.75	1	0.25
Small Goal Difference (5)	100%	1.8	0	0	92.11%	15.2	1.2	0.4
Draw (3)	100%	0.67	0	0	95.65%	23	1	0.33
Average (12 Games)	100%	4.5	0	0	93.20%	16.67	1.08	0.33
Standard Deviation	0%	6.36	0	0	5.40%	7.09	0.90	0.49

Table 7.6: Accuracy Results for Goals and Outsides

Further analyzing the results, one can see that games with more even teams (draws and games with a small goal difference) have a higher number of outside situations. A higher number of outsides also leads to a lower percentage of useful play time, since a number of game cycles (approximately one hundred cycles) are allowed for each of these situations, and thus, games with a higher number of outsides usually have a lower number of other types of events, such as passes (the most common event in the game).

Regarding offsides (Table 7.7), more than 97% of the intercepted offsides and almost 93% of offsides were well detected, which composes good results in this particular scenario. All four situations where the software failed to correctly identify an offside are related to the situation depicted in Figure 7.11 and already described above.

	Intercepted Offside				Offside			
	Percentage Detected	Average of Intercepted Offsides Observed	Difference	False Positive	Percentage Detected	Average of Offsides Observed	Difference	False Positive
High Goal Difference (4)	100%	2.25	0	0	100%	2.75	0	0
Small Goal Difference (5)	93.33%	3	0.2	0.2	90%	4.2	0.4	0.2
Draw (3)	100%	0.67	0	0	88%	2.67	0.33	0
Average (12 Games)	97.22%	2.17	0.08	0.08	93%	3.33	0.25	0.08
Standard Deviation	3.21%	2.76	0.29	0.288	15%	3.47	0.62	0.29

Table 7.7: Accuracy Results for Offsides and Intercepted Offsides

Figure 7.12 presents an unified view on all statistics, and for all three types of games. It is relevant to note that only in three situations were the results of less than 92%: the three types of shots.

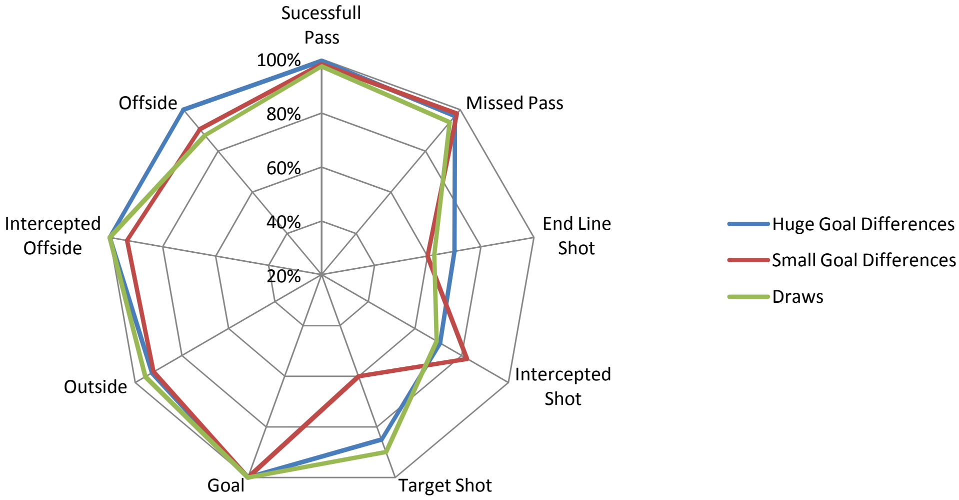


Figure 7.12: Comparison Between the Three Groups of Games Analyzed.

These results show that, even using only Cartesian coordinates as a basis for the detection of events, our approach proved to be efficient to solve this specific problem.

7.3.3 Conclusions

In this section a tool based exclusively in players and ball cartesian coordinated is presented. Based its statistical calculus on the detection of an event called *Kick*, which is related to the increase of the ball velocity, this tool is capable of automatically calculate a huge set os final game statistics. The results achieved in the validation process, and discussed in the previous section (in two thirds of the cases attaining over 92% of detection accuracy), show that the use of a sequential analysis process in the detection of soccer events composes a good approach to this kind of problems. Unlike other researches' works that base the identification of events in the game status, given by the RoboCup Soccer Simulation Server, the soccer scientia tool is capable of automatically identifying a larger set of game events (as is the case of an intercepted offside). Also, this method presents additional information, such as the identification of the player that would receive an intercepted pass (with an associated degree of certainty) and also allows the user to change, in a restricted form, some definitions in the heuristics, related to ball dimension, field regions and the rules for identifying events (pre- and post-conditions and constraints).

7.4 Soccer Server Statistical Extracting Tool (SSSET)

After we decided to use the RoboCup 2D simulation league log files exclusively as our research data, we developed a new statistical tool (SSSET) that was capable to automatically calculate final game statistics through information gathered by the soccer server. Doing a brief comparison between this approach and the soccer scientia software (previously analyzed) it is important to note that new types of statistics were added such as "Wing Chain", "Pass Chain", "Goal Opportunity", "Ball Possession" and "Temporal Sequence" (previously explained).

7.4.0.1 Method

Similar to the method used in Soccer Scientia software, in this new approach the statistics module required manual validation to confirm that the detected events exhibited the desired behavior. However, as the statistical calculus were based on the information provided by the soccerserver (explained in chapter 4), the validation process consists in randomly selected matches samples from RoboCup 2009 competition and execute a comparison between the event detection and the player(s) behavior(s).

Randomly selected samples – As previously stated the samples where randomly selected from the RoboCup 2009 competition matches. The selection results are as follows:

1. The match selected to detect the **Passes** was the final between WrightEagle and HELIOS2009. The time windows were 1000 cycles and the selected cycles were from 0 to 1000;
2. The **Ball Possession** was validated in the interval 746 to 1346 (600 cycles) in the Second Round match between UnKnown09 and HELIOS2009;
3. To validate the **Attacks**, a time frame of 600 cycles was analyzed in the match that opposed NemesisRC09 and HelliBASH in the time interval of 1805 to 2405;
4. Finally, the **Goal Opportunities** were validated using the log file from the match that opposed HfutEngine2D and opuCI. The time interval corresponded to $[4295, 5295]$ (1000 cycles).

7.4.1 Results

In this section the validation data process will be presented as well as the results generated by the statistical tool. To produce those results, the log files of the all participant teams in the 2009 2D RoboCup Simulation League were used.

7.4.1.1 Zone dominance

The zone dominance calculates the average of the ball possession in each field region per team. Figure 7.13 illustrates the zones for the three top teams in the RoboCup Soccer Simulation 2D 2009 league (Wright Eagle in red, Helios2009 in yellow and Oxsy in blue respectively). This zone is displayed as if those teams were attacking in the direction of the white arrow displayed in the Figure. As expected, all of these teams have great control over the opponent field. Doing a brief comparison between these teams it is important to note that WrightEagle team is the one that also retains more control in its own field followed by Helios2009 and Oxsy (exactly the same order as the competition final results).

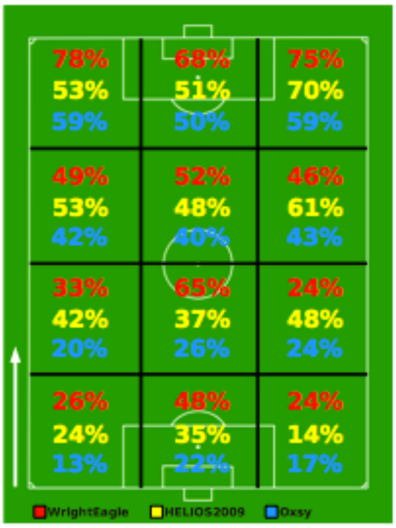


Figure 7.13: Zone Dominance for the Top Three Tournament Teams–The Direction of Team's attack is marked with white arrow

7.4.1.2 Passes

The ratio between the successful and missed passes is calculated and shown in Figure 7.14. Regarding that Figure, it is easy to note that even the team that presents the lowest ratio (successful versus missed pass) have a percentage of successful pass greater than 67% of the total executed passes. Doing a comparison between this data and the final tournament classification, it is interesting to see that the team that presents the best ratio is the champion team (Wright Eagle). However the Bahia 2D team, the worst team in the tournament presents only the sixth worst registry. Also it is important to note that Oxsy, which occupied the third position in the tournament, presented a bad ratio between successful and missed passed (sixth worst mark).

7.4.1.3 Wing variation

The number of wing variations in a soccer match, expresses a collective idea of play executed by a soccer team. Analyzing figure 7.15 the Helios2009, WrightEagle andOxBlue09 teams presented

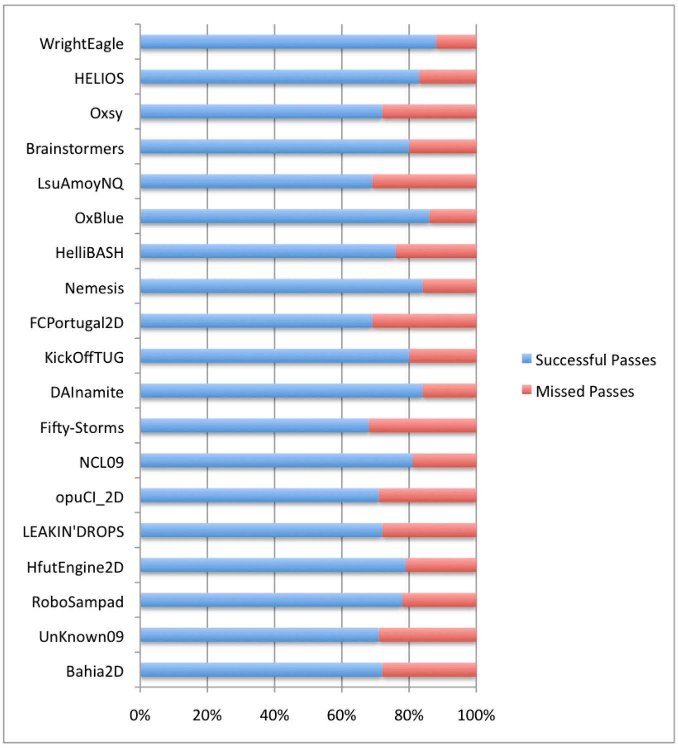


Figure 7.14: Successful and Missed Passes for all the Teams in the RoboCup 2D Simulation League

the largest number of wing variations in the competition, which is not surprising specially in what concerns to the Helios2009 and WrightEagle due to their final tournament position.

7.4.1.4 Temporal Sequence

A temporal sequence is detected when a team has the ball possession and without losing it, advances in the field with the direction of the opponent goal until achieving a specific zone called "the last third of the field". This sequence is essentially an attack and its classification is done according to the difference between the time that the team recovered the ball and and how long it took this team to reach the last third of the field zone. If for some reason a team loses the ball the sequence will be classified as Broken.

Figure 7.16 shows the calculus of time analysis for the top four teams of the tournament. It is important to note that the main difference between the top three teams is the number of break sequences over the competition. Also it is interesting to note that the team in the fourth place Brainstormers presents the lowest value of Medium Sequence which is a peculiar characterization of its game.

7.4.1.5 Goal opportunities

The winner of a soccer match is determined according to the final goal difference (scored - conceded). The team that presents the higher goal difference will be acclaimed as the winner of the

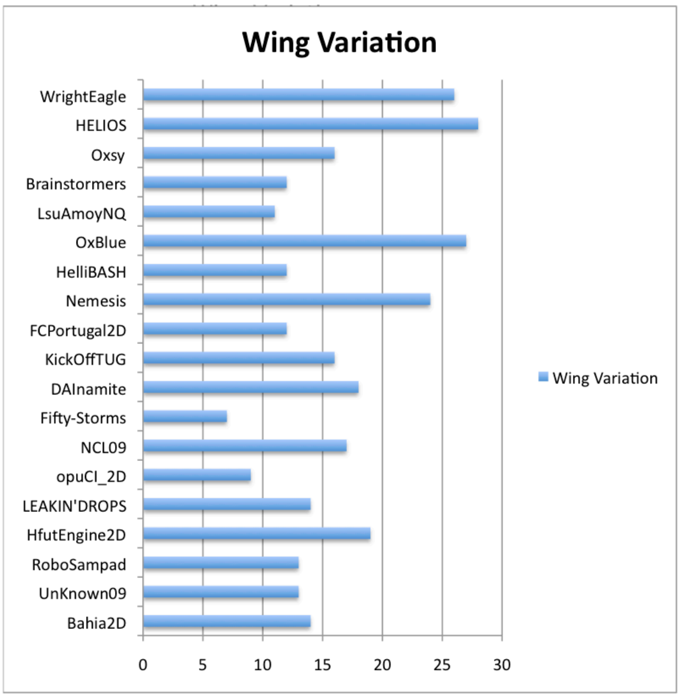


Figure 7.15: Average of Wing Variation per Game

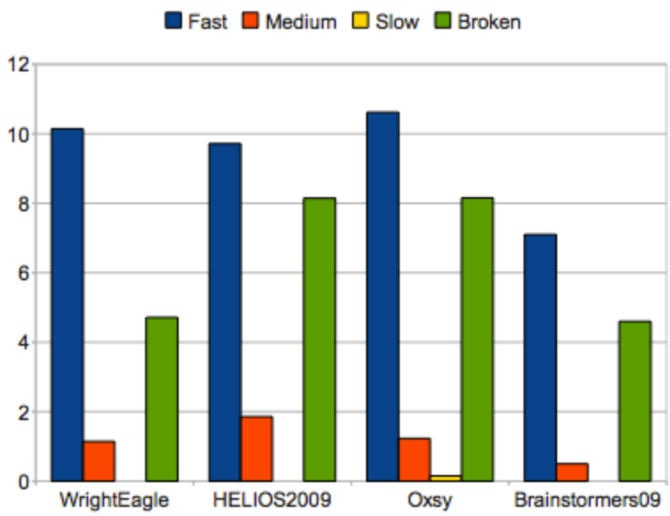


Figure 7.16: Average of Attacks Variation per Game

match. In order to achieve that primary goal, the creation of goal opportunities composes a good indicator for a coach to measure his team performance. Observing Figure 7.17 one can check that three of the four teams that have a large number of goal opportunities, were the three finalist of the tournament and two of them present the higher $(NGoals)/(NOpportunities)$ ratio (up to 40%) in Figure 7.18. Analyzing those Figures it is curious to note that the team that presents the best goal success ratio was ranked with the ninth place in tournament – FiftyStorm team which means that in spite of having a higher success rate, the number of goal opportunities is still small when compared to other teams.

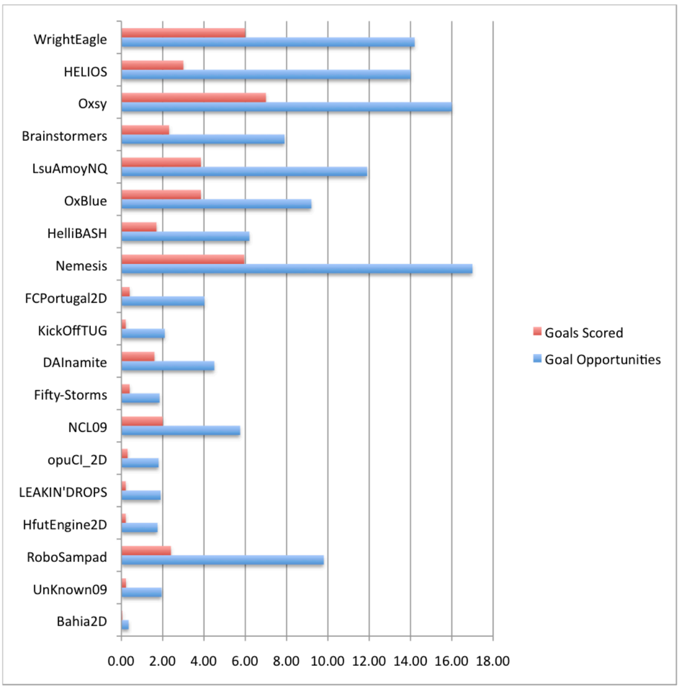


Figure 7.17: Average of Goals Scored and Goal Opportunities per Team

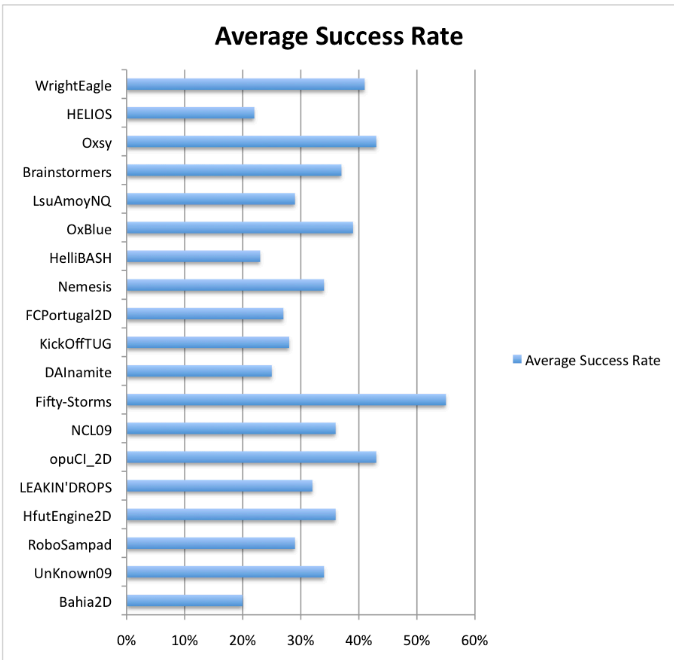


Figure 7.18: Ratio Between Goals Scored and Goal Opportunities per Team

7.4.2 Conclusions

In this section a tool capable of automatically calculating final game statistics using the information provided by the soccer server is presented (SSSET). The set of statistics were defined by a soccer experts board and the data used was the 2009 RoboCup Simulation 2D league log files. Regarding the results achieved the developed tool proved to be efficient in the calculus of those statistics and also it is important to note that not in all of the calculated statistics, the best teams in the tournament proved to be the best. This fact is influenced by the adopted team strategy in the tournament and which can sometimes vary according to its next opponent.

7.5 Comparison Study Between Robotic Simulation 2D League versus Human Soccer

The RoboCup competition has an ambitious main goal "In the Year of 2050 a robotic team defeat the world soccer champion team at the time". Having this goal in mind and, using the SSSET tool developed and, explained previously, a comparison study between the human soccer and the robotic simulation 2D league was performed [ACCa+10a]. In the next sections all the details of that study will be exposed and finally the results will be presented.

7.5.1 Method

In this comparison study one of the most important decisions is to decide which data (human and robotic) will be used. In a soccer competition such as an European or a World Championship only the best teams, with specific characteristics, could reach the final game competition. In this research project, eighty-two games, corresponding to different tournament finals were selected in order to compare Robotic and Human Soccer. Regarding to this goal, three Human Soccer Games (European 2004, World Championship 2006 and European 2008 finals) and seventy-nine Robotic Soccer Games (Robocup 2006, 2007, 2008 and 2009 competitions) were chosen (only including games of the two finalist teams in the final phase and final double elimination tournament (thus ensuring that only games between good teams were used). In the case of Human Soccer, the games were recorded in DVD format while in the Robotic Soccer a log file format was used.

Regarding the calculus of the statistics, most concepts presented in the SSSET tool were used and some new concepts were defined. A brief overview about the new concepts definition will be presented.

Regarding the offensive style used by the robotic team, four different types were defined: Organized Offense, Counter-Attack, Set Pieces and Long Passes. The Counter-Attack is defined as a collective move, in which a soccer team recovers the ball and reaches the last third of the field (their attacking field) in a short period of time, which is dynamically defined depending on the position where the team recovered the ball (similar to the definition of attack previously presented). A Set Pieces occurs in the game when, after a Throw-in, Goal Kick or a Corner situation being detected, a pass combination between teammates (always involving four players or less) is performed and the time spent by the team to achieve the penalty area is relatively short (depending on the field area). On the other hand, if a team, in its attacking movement, performs a pass combination among several teammates and the duration of this process is higher than the estimated by the Counter Attack Situation, this event is classified as an Organized Offense movement. It is important to note that these new concepts are similar to the temporal sequence concept defined previously however, in this study, the concept name was changed due to the familiarization of the soccer experts with concepts like Counter-Attack among others.

Finally, a Long Pass situation occurs when a player executes a pass to a teammate at a distance larger than 30 meters (based on the definition presented and validated in the SocOn ontology presented in chapter 6). In terms of movement that precedes the goal, in this research work, four distinct situations were defined: Combination Play, Individual Action, Direct Shot and Own Goal. In the Direct Shot situation, a player recovers the ball and instantly shoots the ball in the direction of the goal. On the other hand, in the Individual Action, a player recovers the ball but before shooting, he executes many individual actions (for instance, a slalom movement, which consists in, without losing the ball, passing it through a number of opponent players). The Combination Play is a movement that encompasses many team players (at least two players) and finally the Own Goal, as the name implies, is when a player scores a goal in his own goal. Finally, it is important to note that the number of faults is neglected, because, in the RoboCup Soccer simulation competition this number is in average less than two faults per game.

Regarding the detection of this set of events, two tools were used according to the different realities. In what concerns the robotic reality the SSSET tool previously presented was used (using the log files of the RoboCup 2D competition). For the human soccer reality a spreadsheet was created in order to classify the different events of the games using a method of observation. The main features that this tool supports are: identify all players that participated in the match, the different set of events, the duration of the event and finally, filter all events by time. This spreadsheet is also capable of displaying, at the end of the monitoring process, the final game statistics.

7.5.2 Data Analysis

All data was analyzed using R Software version 2.4.1 [5] . To perform the comparison between quantitative variables and, in the absence of sufficient power of test that proves the normality of the data used, we chose to use a non-parametric test instead for instance a T Test. Follow the guidelines presented by different authors ([Sal97] [Dem06]), a non-parametric test, Wilcoxon test was used (using W as the value of test statistic and P as the p-value for the test) and, the level of statistical significance was set at $P<0.05$ as recommended by Dietterich [Die98].

[5] More information's about R software available at http://www.r-project.org

7.5.3 Results

Analyzing the Human Soccer data, it is relevant to note that three consecutive finals were played by six different European teams and only one game needed extra time and penalty shot to find the winner of the match (2006 final). It is also interesting to note that the goal average (per match) in these matches is less than two goals, which could be explained by the high pressure that normally is presented in these games and by the value similarities between the two teams in this competition phase. When the final game statistics (Table 7.8) are evaluated, in what concerns the mean number of passes per game (W=2 P=0.000354 analyzing the total number of passes for a data set of 79 robotic games and 3 human soccer games). Thus it may be claimed that the mean number of passes per games is statistically distinct in human and robotic soccer games.

		Human Soccer			Robotic Soccer		
Groups	Final Game Statistics	Average	Standard Deviation	Percentage	Average	Standard Deviation	Percentage
Pass	Successful Pass	272	67,02	77,89%	96,12	36,91	66,97%
	Missed Pass	77,16	12,82	22,11%	47,38	12,19	33,03%
Shot	Shot on Target	8,66	4,5	71,25%	1,05	1,41	32,27%
	Shot	1,33	0,81	10,95%	1,405	1,71	43,27%
	Intercept Shot	2,16	2,13	17,8%	0,791	1,14	24,36%
Offside	Offside	3,16	1,47	100%	1,61	2,23	67,82%
	Intercept Offside	0	0	0	0,76	1,33	32,18%
Not Applicable	Faults	31	2,9	100%	1,32	0,2	100%
Outside	Goal Kick	8,66	2,16	26,39%	1,66	1,98	19,51%
	Corner	4,5	3,14	13,7%	1,2	1,95	14,09%
	Throw-in	19,66	3,2	59,91%	5,66	3,36	66,4%

Table 7.8: Generic Comparison between Human versus Robotic Soccer

In the Human Soccer the successful passes represent almost 78% of the total passes executed in the game (with 22,11% of missed passes). Comparing to the Robotic passes reality, where the number of successful passes is lower than 67% and the missed ones are higher than in Human Soccer (more than 33%). However, the successful passes are the most frequent event in both realities.

In the number of shots, there were also statistical differences between these two realities (W=64 P=0.000085641 analyzing the total number of the three types of shots for a data set of 79 robotic games and 3 human soccer games) with the predominance of the shot on target for the Human environment (71,25%) and shot for the Robotic environment (43,27%).

In the outsides group (Goal Kick, Corner and Throw-In) the most common event is the Throw-in. This particular situation is more frequent in Robotic soccer, but the other two considered events occur predominantly in Human Games (W=0 P=0.00048965 analyzing the total number of outsides situation for a data set of 79 robotic games and 3 human soccer games).

7.5.4 Goal Events

Doing a more careful examination of the goal event, and starting with the analysis of the goal scoring in a time-basis of half games: 75% of the goals in human scenario were scored in the first half of the match and only 25% were scored in the second half of the game. The other analyzed scenario presents a higher percentage of goal scored in the second half (52,01%) in comparison to the first half (47,89%) (Figure 7.19) (W=522 P=24.5 analyzing the total number of scored goals for a data set of 79 robotic games and 3 human soccer games). Thus there is not enough evidence to claim that the two realities are statistically different in what concerns goal scoring in the two game halves.

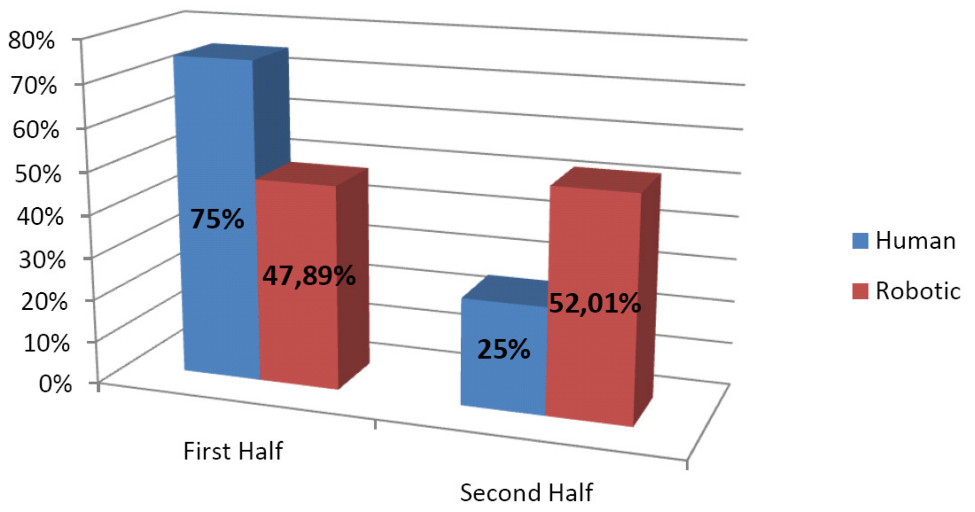


Figure 7.19: Frequency of scoring by period of time (first and second half) and by reality

Observing the type of offense during the scored goals (Figure 7.20) it was detected that the Human and Robotic worlds have antagonistic behaviors (W=571,5 P=31.632 analyzing the total number of offensive style for a data set of 79 robotic games and 3 human soccer games). In the Robotic scenario, the offense style that more often allowed to score a goal was the counter-attack situation (58% of the total), while in the Human reality the predominant style was the Set Pieces (75% of the total situations). In the three Human Soccer Games, no goal was scored in a counter attack situation. The other offense situation corresponds to 25% and 18,1% of the total scored goals in Human and Robotic realities respectively.

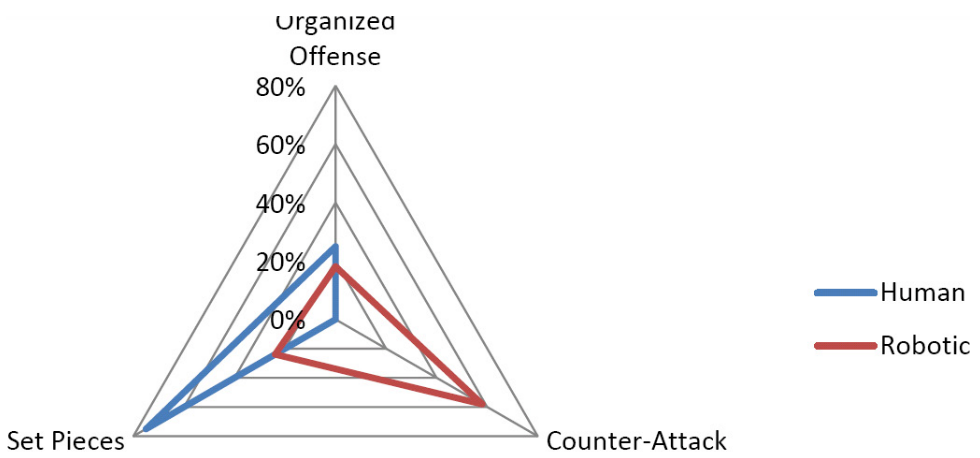


Figure 7.20: Type of offense style used to score a goal by reality

In the Robotic competition, the majority of the scored goals resulted from Combination Play (86,5%), followed by Individual Action (11,25%), Long Pass(1,5%) and Own Goal(0,75%) (Figure 7.21). In the Human scenario the results were a little bit different: the scored goals resulted from Direct Shot (25%), Combination Play (25%) and Long Pass (50%). These differences, however are not statistically significant (W=525 P=11.845 analyzing the preceded goal actions for a data set of 79 robotic games and 3 human soccer games).

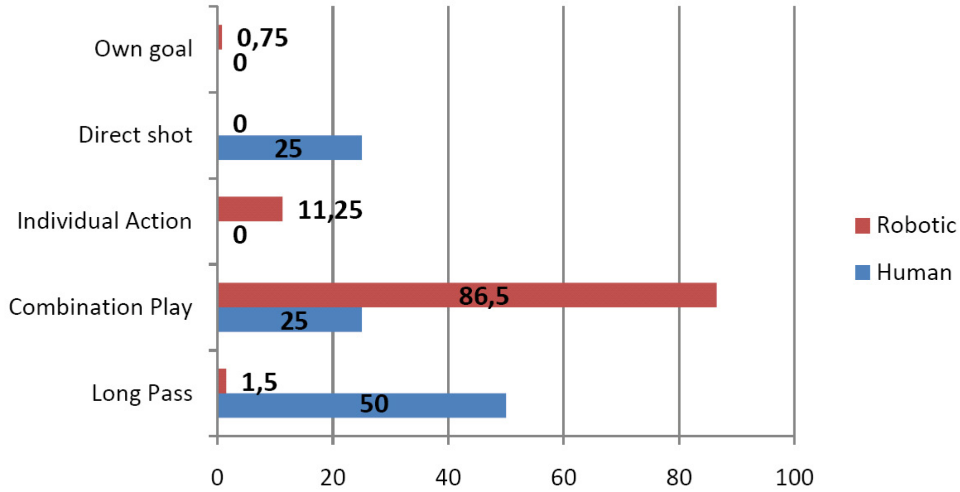


Figure 7.21: Action prior to goal by reality

These actions prior to the goal interfere drastically with the frequency of the Set Pieces observed in the two realities. In Robotic Soccer only the Throw-in was observed (in terms of outsides situations that precede the goal) and it composes 21,5% of the total scored goals. In Human Soccer, 53,3% of the goals were preceded by a Corner-Kick and 26,6% by a Penalty situation (Figure 7.22).

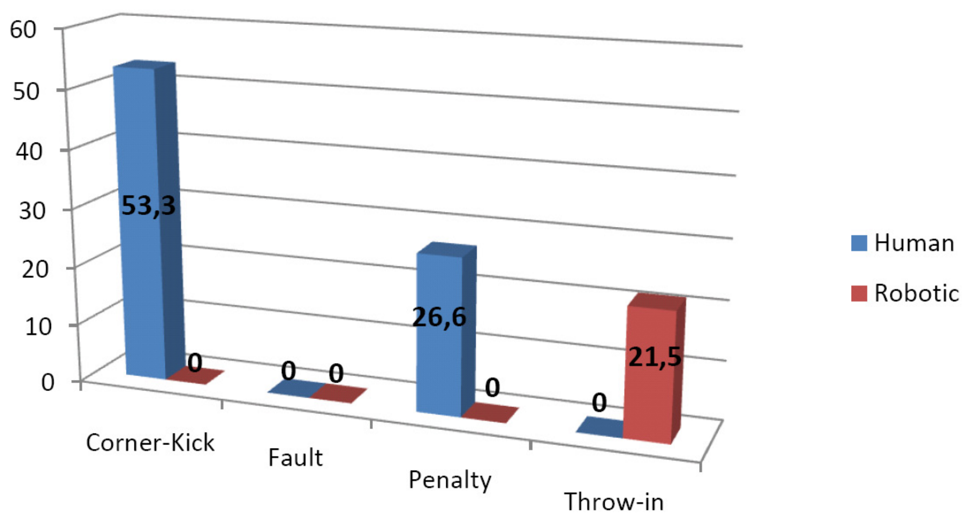


Figure 7.22: Frequency of Set Pieces types

The area of the field from where the offensive attempt was materialized was recorded for the two realities (W=539 P=42.85 analyzing the goal scored area for a data set of 79 robotic games and 3 human soccer games). In the Robotic world, almost 74% of the goals were scored from the Penalty Area, and in the Human Soccer, all scored goals were from inside the Penalty Area (50% from the Penalty Area and the other 50% from the Goal Area) (Figure 7.23).

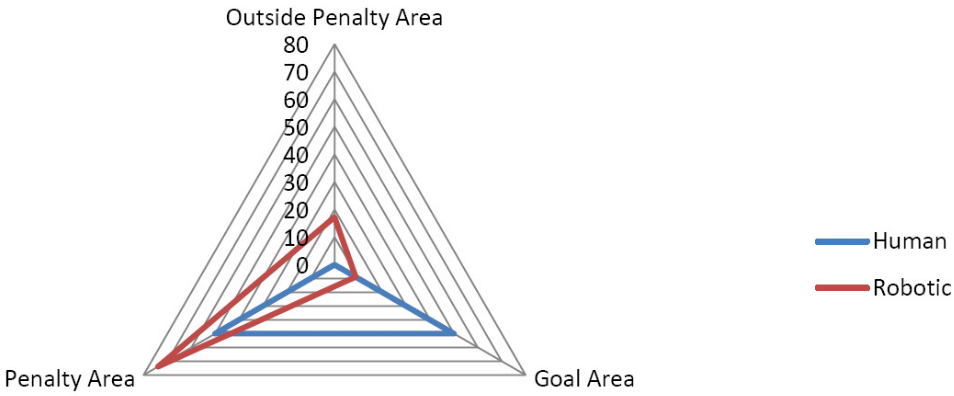


Figure 7.23: Area from which the goals were scored

7.5.5 Extra Time Comparison

Another similarity detected, refers to the games that need extra time to find a match winner (Table 7.9)). As the dataset includes only three games that needed extra time (1 Human and 2 Robotic), and since one Robotic game ended at the beginning of extra time due to the golden goal rule, the values used in Table 2 pertain to one Human game and one Robotic game.

		Human Soccer		Robotic Soccer	
Groups	**Final Game Statistics**	**Oberved Number**	**Expected Average Value**	**Oberved Number**	**Expected Average Value**
Pass	Sucessful Pass	236	191	27	10
	Missed Pass	57	49	20	9
Shot	Shot on target	3	5	0	4
	Shot	1	1	0	8
	Intercept Shot	1	2	0	6
Not Applicable	Faults	4	11	0	1
Outside	Goal Kick	4	6	0	4
	Corner	2	3	2	5
	Throw-in	11	13	6	7

Table 7.9: Extra Time Comparison

Normally, in the extra time period (which has a time span of 1/3 of the total game time in both realities) neither of the teams likes to take risks. Instead, they prefer to execute a more defensive game and, if a goal situation occurs, they will try to exploit it; otherwise, they will wait until the end of the period and go to a penalty session. In consequence of this, the teams' statistics usually drop in this period. Table 7.9 shows that the reference values calculated empirically (taking the games that ended in regular time as an indicative measure and using a proportional rule) do not have many similarities comparing to the regular time. Analyzing the Robotic and Human data together, the only extra time values that are superior than the reference values are the number of shots (successful and miss) however, the other statistics experienced a dramatic decline. When the two realities are separately evaluated the results show that there are not differences between them in terms of pass (W=6,0 P=14.62 comparing the total number of the passes in the extra time value to the reference value for a data set of 2 robotic games and 1 human soccer games), shot (W=2,5 P=18.2 comparing the total number of the three types of shots in the extra time value to the reference value for a data set of 2 robotic games and 1 human soccer games) and outsides (W=7 P=25.4 comparing the total number of the outside situations in the extra time to the reference value for a data set of 2 robotic games and 1 human soccer games).

7.5.6 Final Games View

Conducting now a more detailed analysis of each game, we started by analyzing the Human games. We observed that two of them ended in the regular time period, with only a goal scored (2004 and 2008 finals) and only one game (2006 final) ended in the penalties session. In what concerns the executed passes in the regular time of the matches it is easy to note that, all over the years, the frequency of successful passes increased and the opposite was verified in the missed passes (Figure 7.24). Having only the passes as comparison criteria, we concluded that the most balanced final match was the 2006 final and, curiously, this game was also the only one that needed extra time to find the winner. In the 2004 final, the winning team (Greece Team) had less successful passes executed (171 versus 216) and in 2008 the winning team (Spanish Team) also presents less successful passes executed, when compared to their opponent. However, in this final case, the successful passes of the Spanish Team (the winning team) composed more than 84% of the total executed against 83% of their opponent, unlike the 2004 final game.

If the analysis focuses on other statistics such as Shot (Figure 7.25), in the two first games the results show that the winning teams (Greece, Italian and Spanish Teams) had fewer shots against their opponents but still won the games. However, in the third analyzed game, the winning team (Spanish team) had almost four times more shots on target events against Germany and, in the other type of shots only on the intercepted shots did these two teams present equivalent values.

Making the same analysis for the RoboCup, and starting with the pass reality (Figure 7.26 – illustrates the pass median of the robotic competition years), it is interesting to note that through the years, the median of missed passes has decreased. This fact could be explained by the increase of the competitiveness of the best teams. Regarding the number of successful passes, they have increased, especially between the 2006 and 2008 competition.

Analyzing only the robotic finals, in all of them, except in one (2007), and unlike human soccer, the winner team always presents a higher number of successful passes and a lower number of missed passes than his opponent (the data showed in Figure 7.27 is filtered by year and Winning

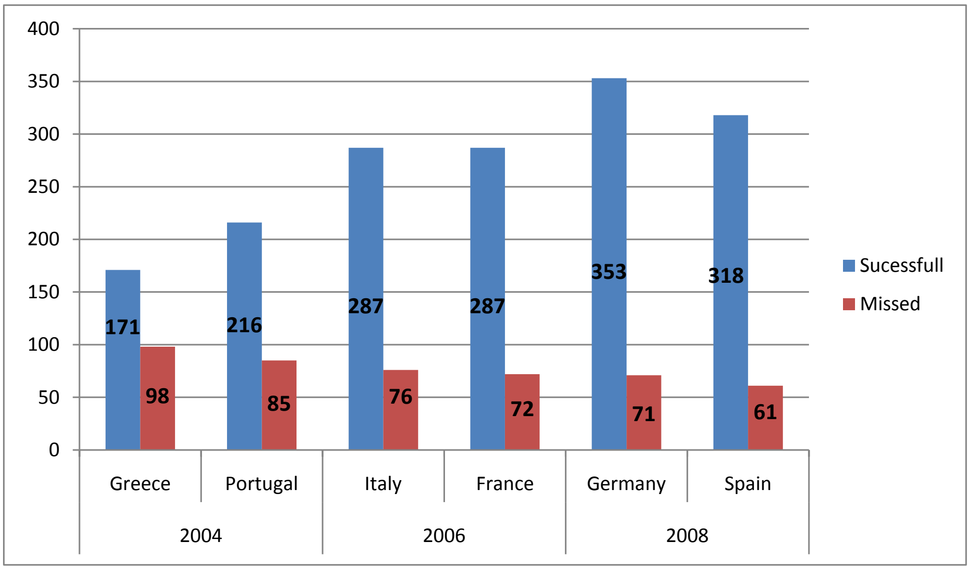


Figure 7.24: Pass Statistic in Human game finals by country

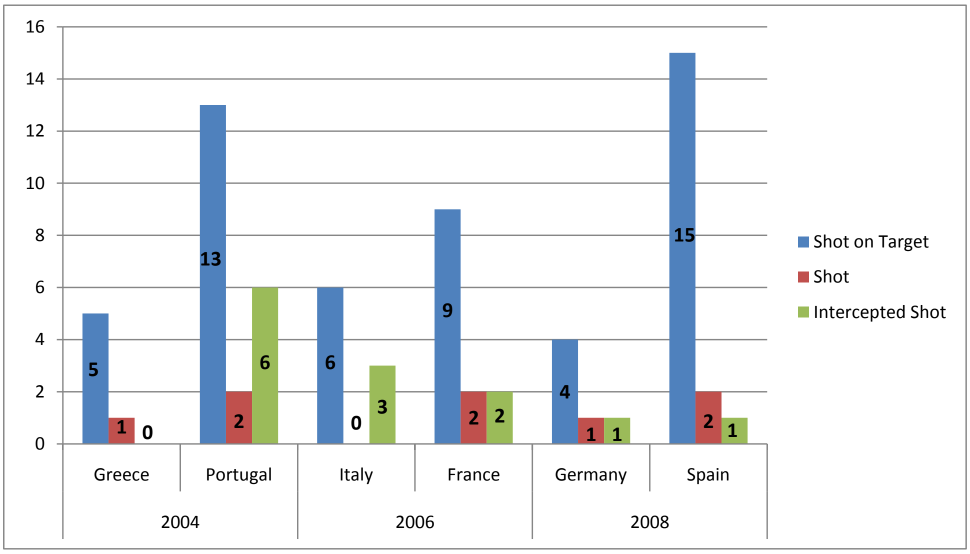


Figure 7.25: Different Types of Shots in Human Game Finals

team). Also, it is important to note that, in four finals, two ended in extra time and the total scored goals in the regular time of the game was eight goals.

In what concerns to Robotic finals, although the number of shots was in average similar in the two compared realities, the other two types of shots show very different frequencies ((Figure 7.28)).

7.5.7 Discussion

Over the past years, many researchers have tried to evaluate the development of Human Soccer, focusing their studies in goal characteristics [YHL93], [GMB97]. In order to carry out a

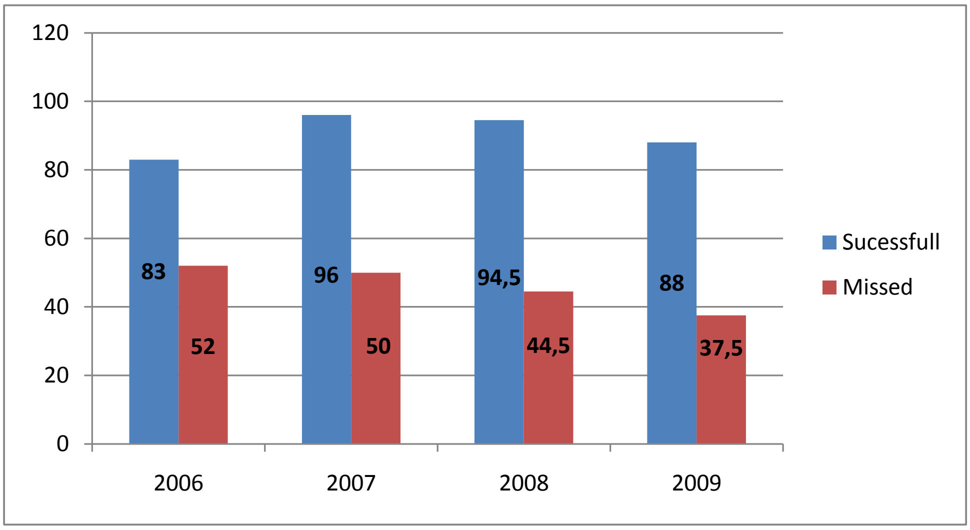


Figure 7.26: Median of Pass Statistic for RoboCup games between 2006-2009

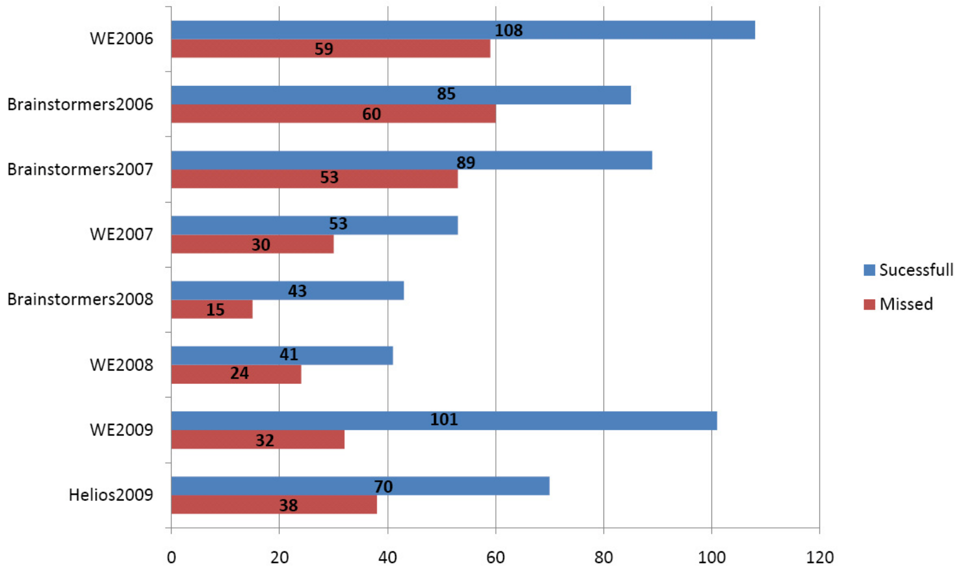


Figure 7.27: Pass Statistic for RoboCup Finals by Team

higher level comparison between the two treated realities a parallel with some research works will be produced. In 1968 Reep and Benjamin [RB68] analyzed more than 3000 matches and concluded that approximately 80% of all goals resulted from a sequence of three passes or less and a goal is scored every 10 shots. Although this study seems to be dated, in the past few years many were the studies that confirm these findings using different FIFA World Cup finals [FGM83] [HRN88] [PF89a] [PF89b] [FPN90] [Gre99] [HF05]. In our research games, the teams scored 400 goals and 62, 32% of them resulted from a sequence of three passes or less. For the second finding from Reep and Benjamin, none of the games confirm the theory of the need of ten shots to score a goal. Even if the definition of shot used by Reep only covered the shot and shot on target or only the shot on target (used in our research project), the extracted results would still

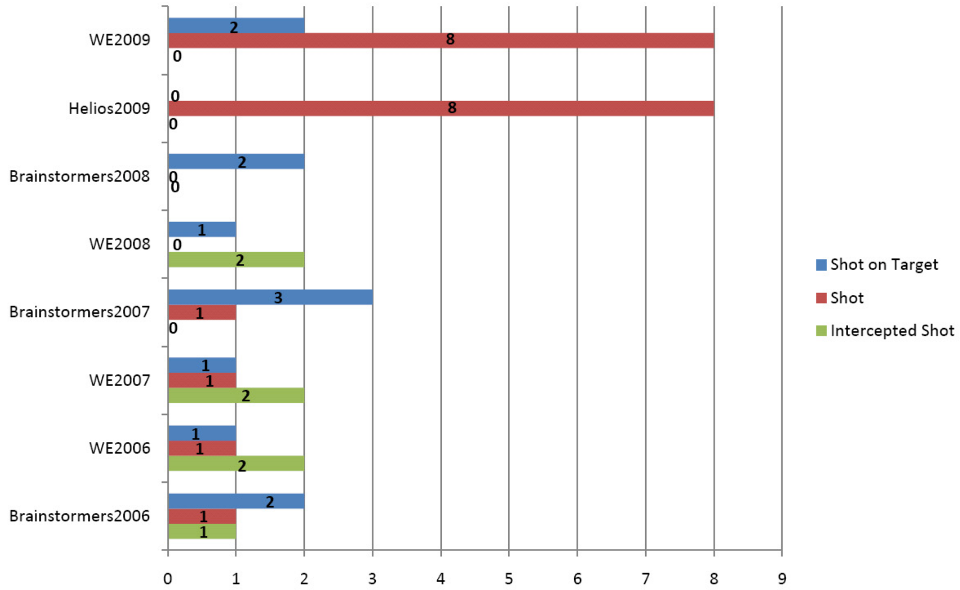


Figure 7.28: Shot Statistic in Robocup Finals by Type

not confirm his theory.

Continuing analyzing the goals and its characteristics, over the years many were the studies that aggregated the goal scoring in reference to time of accomplishment (per half time or even per periods of 15 min each) supporting that the frequency of goal scoring is time dependent [SL92] [ADM02] [BLS+05] [SMT05].

The results produced in our research work show that the highest percentage of goal scoring, in human reality, happened in the first half of the game. However, in the other analyzed environment this fact did not occupy an important role (47,89%, 52,01% for the first and second half respectively).

Regarding the type of offense during goals, Piecniczk [Pie83] concluded that 27% of the goals were resultant of a quick offense and only 28% were through organized offensive actions. After some years Dufour [Duf93] concluded that this trend had inverted: 88% came from organized offense and 12% from strike offense. More recent studies note that, in modern soccer, 16,9% of counter-attacks lead to a goal and only 11,1% of organized offenses are successful [AYAS05].

Nowadays, the execution of set pieces composes an import part of a team's tactics as well as in the outcome of a game. Like our study, many others claim that more than 1/3 of the scored goals in many competitions resulted from set pieces [Gar97a] [Ols98] [BLS+05]. On the other hand, we observed that, in the Robotic world, the most used movement to score goals was the counter attack (58% of the goals).

Examining the actions that lead to the goal, our findings show that in Robotic world the majority of the process resulted from combination plays (86,5%), contrarily to Human environment where the long passes represent the highest percentage of cases (50%). These findings were similar to the results obtained in the Human game finals by [YA04] which demonstrates that the long passes are the most frequent action that lead to the goal.

Analyzing the more common set pieces before the goal and, in consequence of the results previously explained; in the Robotic environment, only 21,5% of the scored goals were preceded by a set play (Throw-in situation). In the other analyzed reality, the most common set play was the Corner Kick (more than 53,3%), followed by the Penalty situation (more than 26%). However, the achieved results produced by some research works [JXYM93] [Pap02] indicating that, in spite of the corner Kick having a good percentage in terms of goals scored (27% and 24, 4% respectively, the most relevant set play was the free-kick, which represents 37% and 39%, respectively. Another recent study [YA04] showed that the major set play observed before the goals scored was the corner situation (40% followed by the free kick situation, with 30%). Although the comparison of these studies provides, to a certain point, dissimilar results, even in the Human final scenario, it is clear that corner kick situations occupy a higher percentage of the produced goals. Also it is relevant to note that, as previously referred, in the Robotic environment, as fault situations were not very common (less than 2 faults per match), the most common and unique set piece before the goal was the throw in.

In what concerns the area where the final effort was materialized, the findings of our research indicate that the majority of the Robotic goals were scored from inside the penalty box (82,74%). However, in the other observed reality the frequency was divided between the penalty area (50%) and the goal area (50%). In literature, the results achieved were very similar to our study. In the 2002-03 Champion's League season, Michailidis et al. [MMPP04] concluded that more than 64% of the goals were scored from inside the penalty area and 36, 5% from the goal area. Other studies, produced by Sotiropoulos et al. [SMT05] and Dufour [Duf93], indicated that approximately 80% of goals were scored from inside the penalty area and 16% from the goal area.

7.6 Conclusions

In this chapter two statistical tools capable of automatically calculating a huge set of game statistics were presented. These statistics were previously defined by a soccer experts board and after that, the log files related to the RoboCup 2D simulation competitions games were used to produce the results. In both cases the final results were quite interesting. The main goal of the developed of the first statistical tool was to prove that is possible to construct a reliable tool only using players and ball cartesians coordinates, while the main goal of the second tool was to calculate a huge set of statistics capable to characterized the robotic teams in the RoboCup 2D simulation league competition. Finally it is important to note that this tool will occupy an important role in subsequent phases of this thesis at the characterization of how the teams play.

Also in this chapter and, having the SSSET tool as a base, a comparison study between the human and the 2d simulation reality was executed. The final results concluded that the Robotic teams' behaviors still present some statistical differences with the Human Soccer, especially in what concerns the frequency of outside situation, frequency of shots and also in the frequency of successful and missed passes.

Chapter 8

Opponent Modelling

In recent years, the growing need and interest in performance analysis have led to new forms of match analysis techniques. Modern-day techniques include video-based statistical analysis systems, video-based tracking and electronic tracking systems [CWR07][AVM+09]. Nowadays, as previously mentioned, there are many types of analysis software, capable of calculating a large amount of game statistics. However, they still present difficulties in transferring such data to the training session and, not rarely, different software systems analyzing the same data produce different results, as shown by Randers et al. [RMH+10], which means that depending on the analysis software being used, the user can obtain contradictory information about his next opponent.

Sun Tzu in his book The Art of War [Gri63] claims that "If you know/study your enemy in a hundred battles, you will win all of them". Despite the competitiveness gap being greater among robotic soccer teams, the importance of preparing a robotic team for the next opponent occupies a similar role in achieving a game victory. Normally, in the preparation of a match, a soccer coach divides it in two distinct phases: the first consists of, using past games, detecting an opponent play pattern in order to prepare his team to select the best strategy to neutralize it (offline phase). The second phase consists of, during the game, analyzing the behavior of his opponent and adapting his team strategy in order to win the game (online mode).

In this chapter, respecting the specific phases used for the soccer coach's, it will be presented an approach with the main goal of improving the performance of a specific robotic team (FCPortugal). Primarily a literature review description will be held as well as a brief description of the FCPortugal team (used as a case study in this project). From this, a complete explanation of the two distinct phases that compose the project will be held and finally the results will be exposed.

8.1 Literature Review

Research communities currently show a large interest in automatic modeling of a human players or even of other agents. A strong area in this domain is the human imitation area. Following Aler et al. [AVCL09], this topic is present in several fields, such as cognitive science [BB78], user modeling [WPB01] or robotics [KII94].

With regards to the cognitive area, Brown and Buton [BB78] presented a work that tried to model the internal cognition of students. However, over the years, researchers have exploited different perspectives, including the Agent Modeling in terms of relationships between Inputs and Outputs (IOAM).Treating the cognition problem as a black box, the IOAM approach can include feature-based modeling [WK96], C4.5 - IOAM [WCK97] among others. It is important to note that the IOAM approach only deals with discrete and static domains (the opposite in comparison to a robotic environment).

In the user modeling field there were two major areas: user profile creation and behavior cloning. The first area is related to the World Wide Web. With the fast growth in research in information retrieval on the web, companies assume the need to create user profiles in order to group them into communities with common interests [PKPS99]. Consequently, users can be classified into stereotypes and their interests can then be predicted.

The behavioral cloning is related to the ability to reproduce exactly a set of actions previously executed by an agent or a human. One example of a research project in this area was produced by Sammut et al. [SHKM92], which basically consists of learning to pilot a simulator through cloning a human pilot simulation experience. The authors used many piloting styles and because of this the learning process did not present good results. Furthermore, this kind of problem (learning from humans) presents three main issues (when compared to machines or agents): humans are less systematic, their behavior varies more and they make more errors. These three tasks constituted important issues in this application domain. However, in 2004 these limitations were overcome with the Bauckhage et al. [BTS03] research. These authors provided good imitation play skills on different levels such as reactive, strategy and motion modeling. Using Quake II (a very popular first person shooter game) [1] as the project base, these researchers allowed a human to play the game and record his pairs of state vectors and actions. Following this, and in order to reduce the vectors dimension, they used a self-organizing map and a multilayer neural network to map these vectors in actions. At the end, the results appeared promising.

Through the years, modeling soccer teams is a research problem in professional soccer that many researchers have tried to solve [Gar97b]. In 1982, Maher [Mah82] tried to model goal scoring using univariate and bivariate Poisson distributions in order to simulate attacking and defensive capabilities used by soccer teams. Although this work was capable of estimating the goal scoring after the match (using final game statistics), it was not able to predict the final result or goal scoring before the match was played. Other studies have tried to quantify the effect of home advantage on match outcomes in the variation of goal scoring [CN95][DR98]. Today, in real soccer, many automatic tools exist that are capable of calculating a huge amount of game statistics. One of the most famous ones is the ProZone[2] software (previously analyzed in chapter 5), which has been used in many research studies, such as the ones presented in [BSB$^+$09] and

[1] More information about the Quake Game available at `http://planetquake.gamespy.com/quake2/`

[2] More informations available at `http://www.prozonesports.com/product-prozone3.html`

[GDAS10] . Although being very powerful in what concerns the number of calculated statistics, this type of software still presents two major issues: the first one is concerned to the inexistence of a simulation module, which means that this software requires a soccer expert to interpret the data it produces; the second issue concerns to the standardization of the calculation process, which leads to different opponent statistics being generated, depending on the analysis software being used [RMH+10]. Also, it is important to note that the Pro Zone system uses twelve cameras spread over the soccer field in order to calculate statistics. This increased economical investment can be a deterrent factor for poorest/small teams, and in some Leagues the adoption of this system will not be possible due to the dimensions and structural conditions of some soccer fields.

Over the years in the robotic field two different areas have appeared: human imitation and opponent classifier. For the first area, and similar to the approach presented in the previous paragraph, Aler et. al. [AVCL09] presented an IOAM approach to model a human playing RoboSoccer. This approach consists of allowing a human to play soccer using the soccer server (the platform that supports the RoboCup 2D simulation league) and a set of low level commands that includes dash, turn and kick. Using the Part machine learning algorithm, these authors were able to construct the human player model and to implement it in a computer agent. From the results it was proved that the modulated agent was capable of scoring more goals than the other ones presented in the same environment. It is important to note that this approach constitutes the first imitation approach in the RoboCup environment.

A different area is opponent behavior classification. The main goal of this topic is not to imitate the opponents but to model them in order to choose the best team strategy to defeat them. Normally these models are not learned but predefined and they are used to classify an opponent (usually the team as a whole, and not an individual agent).

Over the years, in the RoboCup soccer environment, a lot of research has been related to opponent modeling, it mostly focused on a coach agent (how a coach agent with limited and restricted communication with its players can improve the performance of his/her team). Stone et al. [SRV00] presented a low-level positioning and interaction agent approach based on an ideal world (where the performance of the opposing team is the maximum). However, in this approach, the process of positioning adaptation does not change throughout the game and it is independent of the opposing team (it is a generic approach). This constitutes severe approach limitations. An extension of this work is proposed by Ledezma et al. [LASB05] with the main goal of improving the low level skills of the modeled agent. Druecker et al. [DHN+00] used a neural network to identify the opposing team formation (however the information obtained seems to be very limited in it ability to improve the performance of a team). Similar to Druecker, Riley presented a learning formation approach based on players' positions [RVK02]. However, the limitations presented by this study are similar to the previous one.

In conclusion, it is clear that many studies have tried to solve the problem of modeling opponent behavior team, detecting single variables that can improve the team performance, like, for instance, detecting the team formation, or trying to analyze the relationship between the "home or away" effect. However, this problem incorporates many complex and correlated variables, and to improve team performance a combination of them will be needed. Also, there is not a framework that is capable of automatically detecting the pattern behavior of an opponent and after that, to indicate which is the best tactic/strategy to adopt in order to defeat that opponent (optimizing the team performance).

8.2 FCPortugal Team Description

FCPortugal is a robotic team developed by Porto and Aveiro Universities' researchers that through the years had achieved good performances in the RoboCup Competition leagues. Doing a brief summary about the participation of this team in this competition, it won the 2D simulation league in the RoboCup 2000 in Melbourne and became two times European Champion (Amsterdam 2000, Paderborn 2001). This team also won awards in other RoboCup competitions such as the coach simulation league in 2002 (Fukuoka) and achieved two second places in the same competition (Padova 2003 and Lisbon 2004). Regarding the Simulation Rescue league it became European Champion in 2006 and in the 3D Simulation league became World Champion (2006) and two times European Champion (2006 and 2007).

In order to perform complex cooperative tasks in complex dynamic environments, each FCPortugal player includes three levels of knowledge: individual action execution, individual decision-making and cooperation [RLML10]. The first level is concerning to the specific commands capable of performing low-level actions, the second level is concerned with the way agents choose the action to execute (regarding the available set of actions) and finally the third level is concerned with tactics, situations, dynamic formations, roles, dynamic plans and communication protocols [RLO01][LR07].

Through the years, the FCPortugal architecture followed an idea of a common framework for Cooperative Robotics [MRB06] [MR08] enabling its use in different robotic leagues. This framework included low level skills which are defined for each type of robot (in each league) and high level skills which are chosen through the same decision-making component. In this thesis only the high level will be subject to a deeper analysis.

8.2.1 High-level decision and Cooperation

In the next sections four mechanisms used by FC Portugal – Strategy Coordination, Situation Based Strategic Positioning (SBSP), Dynamic Positioning and Role Exchange (DPRE) and Flexible SetPlays– will be exposed.

8.2.1.1 Strategical Coordination

FC Portugal team presented a strategical model that uses a multi-level hierarchical approach (Figure 8.1). The lower level uses the concept of roles to reflex the agent's usual activities. The middle level introduces a sub-tactic that aggregates agents with various roles to solve partial objectives

[CLR07a][CLR07b]. On top of the sub-tactics, the concept of formations (this concept will be explained in offline mode section) is used to distribute agents through the sub-tactics. Over that level a tactical level employs a hybrid method to switch formations. This method is based on events situations and precedences. On top of all those levels a strategical level is defined which allows communication between tactics according to scenario conditions.

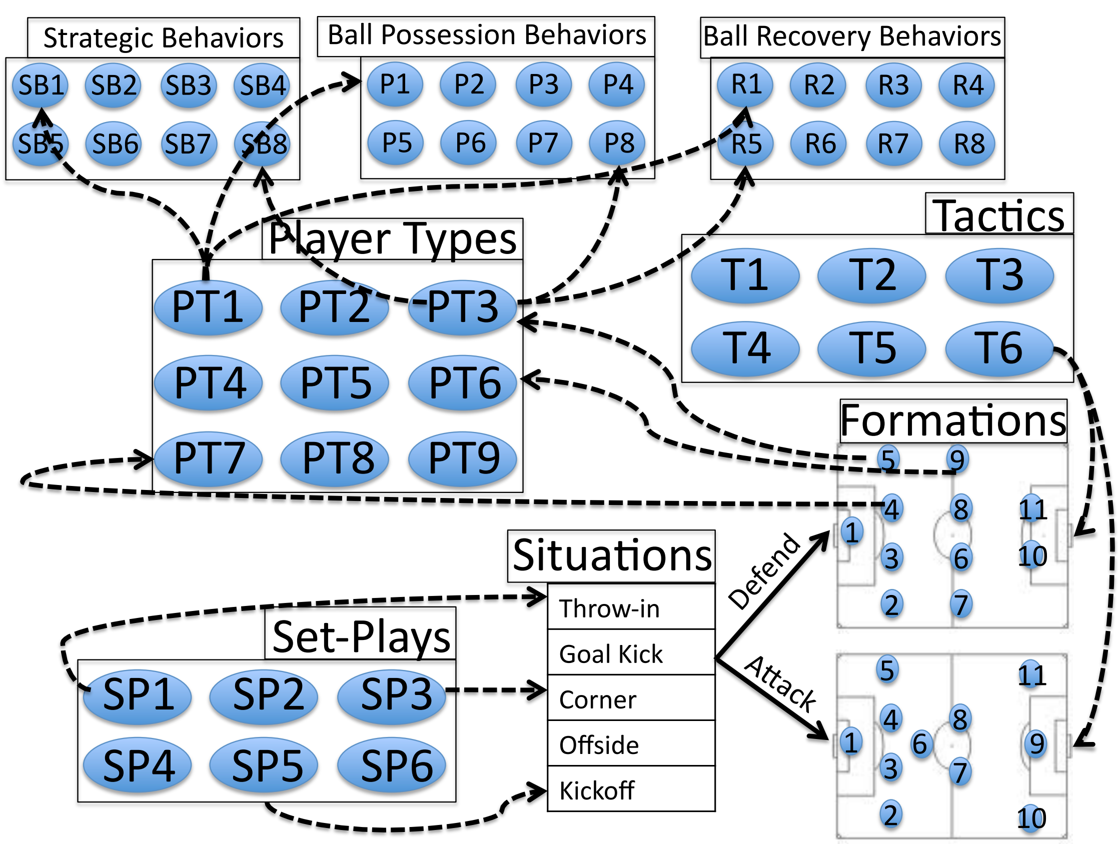


Figure 8.1: FC Portugal's Strategical Model (adapted from [RL01])

As we can see in the next sections, in order to increase the flexibility of our team strategy, two distinct high-level variables were used: Team Formation, related to the players' position in the field; and Set-Plays, a set of player movements combination. In order to define this information, the Playmaker tool may be used [RLML10]. Basically, this tool is comprised of two distinct modes: formation definition and set play definition.

8.2.1.2 Situation Based Strategic Positioning

Situation Based Strategic Positioning (SBSP) is a mechanism (used for strategic situations) [RL01] [RLO01] that allows the change of the spatial position of a team using different formations for different situations. Normally, for each game situation, the agent adjusts its position according to the ball position and velocity. Whereas, at each time, only a few players are in active behavior (conducting or trying to recover the ball) the other players are nearly to their strategic positioning. The players' positions are calculated through a Delaunay Triangulation [AN08] and a linear interpolation algorithm. For this task we used the same algorithm that was used in the Gouraud Shading algorithm [Gou98], which simulates the differing light and color effects along a surface.

The formation definition is based on the ball position. All ball positions included in the formation definition are used as vertices to create the triangulation. After determining the triangle (B1, B2, B3) that encompasses the current ball position (B), player positions are calculated according to the interpolation algorithm. Figure 8.2 illustrates an example of the interpolation process. This

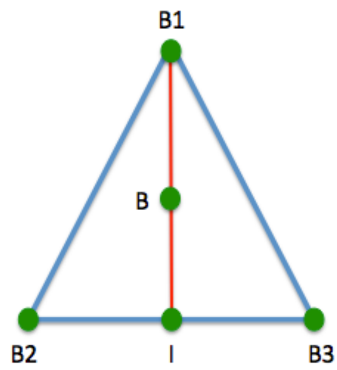


Figure 8.2: Gouraud Shading Interpolation [RLML10]

process is divided into three parts [RLML10]:

1. Calculate the I value (the intersection point between the line B1 and segment B2B3);

2. Calculate the interpolated target position regarding I as the current ball position and B2 and B3 as reference points (8.1):

$$P(I) = P(B2) + (P(B3) - P(B2)) * \frac{|\overrightarrow{B2I}|}{|\overrightarrow{B2I}| + |\overrightarrow{B3I}|} \tag{8.1}$$

3. Calculate the new player position in relation to the ball position B (8.2):

$$P(B) = P(B1) + (P(I) - P(B1)) * \frac{|\overrightarrow{B1B}|}{|\overrightarrow{B1B}| + |\overrightarrow{BI}|} \tag{8.2}$$

In this research, two distinct formations were defined (1-4-3-3 and 1-4-4-2) and 115 points were used for the definition of each formation. An example of a 1-4-3-3 formation definition using the Playmaker tool is illustrated in Figure 8.3.

8.2.1.3 Dynamic Positioning and Role Exchange

The Dynamic Positioning and Role Exchange (DPRE), and Dynamic Covering mechanism [RL01] are extended from the work done by Stone [Sto00]. This enables the agents to switch their relative positions (for a given formation) and roles (that define agent behavior at several levels), at runtime, on the field, if the utility of that exchange is positive for the team.

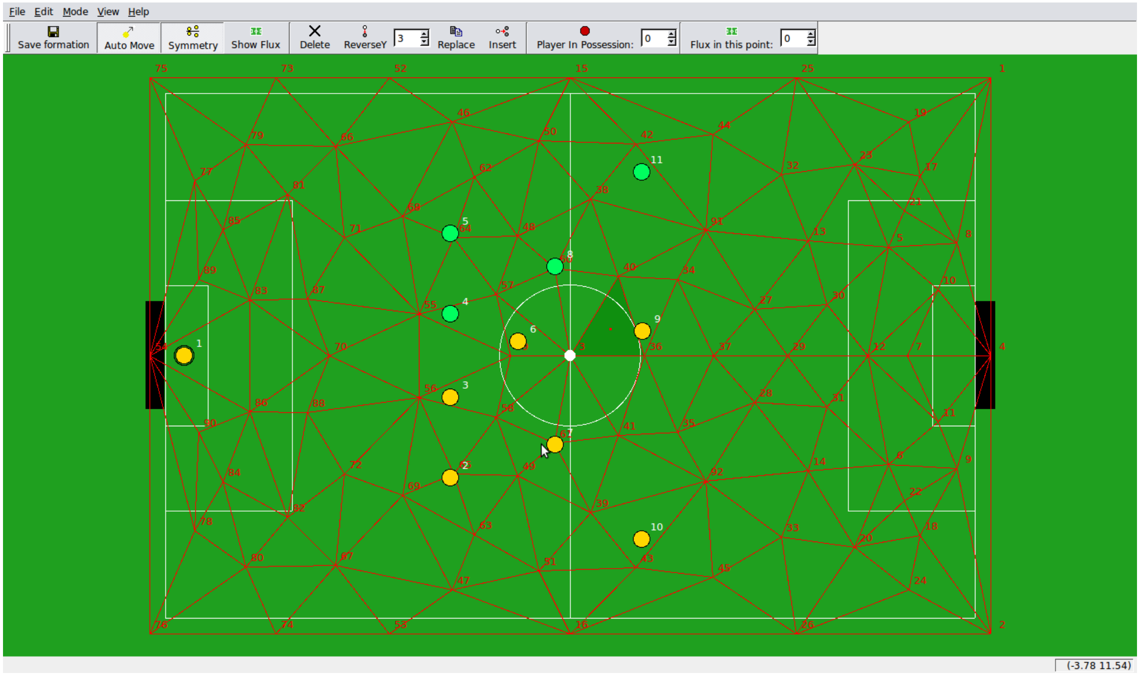


Figure 8.3: Example of 1-4-3-3 formation definition

8.2.1.4 Flexible Set-Plays

A Set-Play can be seen as a flexible team plan for a specific game situation that can involve: a specific game period, the number of scored goals, a game situation like a set piece or even the opponent players position in the field.

Normally, a Set-Play is identified by a name, a set of parameters (conditions), and players. . Furthermore, a set play can be seen as a list of states. The possible transaction between these steps can be divided into three groups: (1) abort if not all the conditions to continue with the set play are met; (2) the next step transaction when all the conditions to continue are met; and finally (3) the finish transaction which indicates that a set play execution is completed [MLR10][MRL11].

In this research 8 different set plays were used. They can be divided into four groups relating to a specific game situation. Only one SetPlay of each group was used in each simulation game. For each representative SetPlay figure the black and white line symbolizes the movement of the player and the ball respectively. It is important to note that if, for some reason, the conditions for the realization of a Set-Play are not met it will be aborted:

1. **Kick-off** is a situation that characterizes the beginning or recommencement of a soccer match (e.g. after a goal is scored). In this work two set plays relating to this game situation were used:

   (a) **Kick-off to winger composed by 4 steps** (Figure 8.4)

      i. The kick-off taker passes the ball to the midfielder, while the defender positions himself to the left of the midfielder.

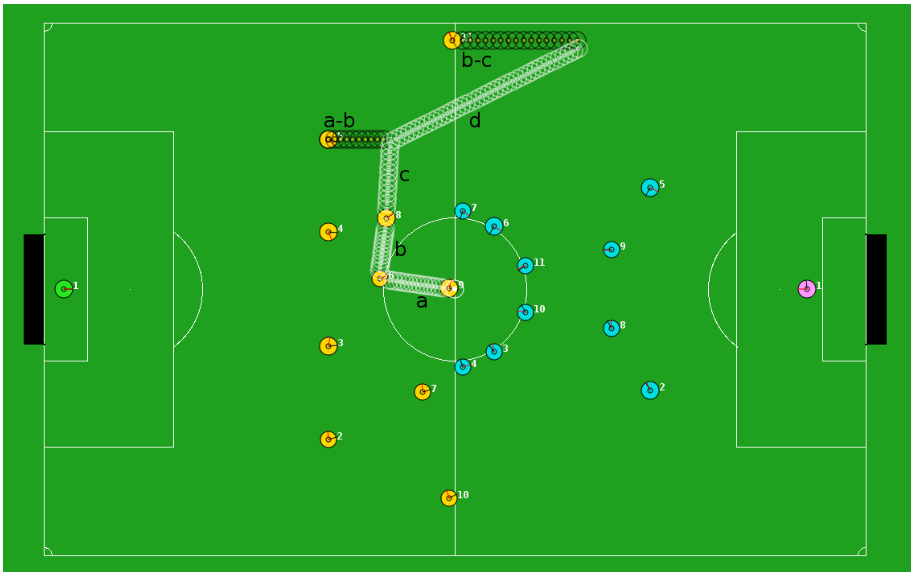


Figure 8.4: Example of a Kick-off Set Play involving 4 steps

ii. The midfielder passes the ball to the striker, while the winger moves nearer to the offside line to the left. The defender keeps himself to the left of the midfielder.

iii. The striker passes the ball to the defender, while the winger keeps heading towards the offside line to the left.

iv. The defender passes the ball to the winger.

(b) **Kick-off to winger composed by 2 steps** (Figure 8.5)

i. The attacker makes a direct pass to the midfielder. The winger positions himself near the offside line to receive a pass from the midfielder.

ii. The midfielder passes the ball to the winger.

2. **Free Kick** is a situation that occurs when a player violates the game play rules. For this situation, we have used two distinct set plays taking into account that it takes place in the offensive region, from our team's perspective.

(a) **Free Kick direct to the goal** (Figure 8.6)

i. The taker passes the ball to a receiver near the goal.

ii. The receiver shoots at the goal.

(b) **Free Kick where is privileged the ball possession**(Figure 8.7)

i. The taker passes the ball to the striker.

ii. The striker passes the ball to the left-winger or to the right-winger; if this is not, he passes the ball back to the taker.

3. **Goal Kick** occurs when the ball has completely crossed the goal-line without a goal having been scored and having last been touched by an attacking team player. Two different Goal Kick plays were used:

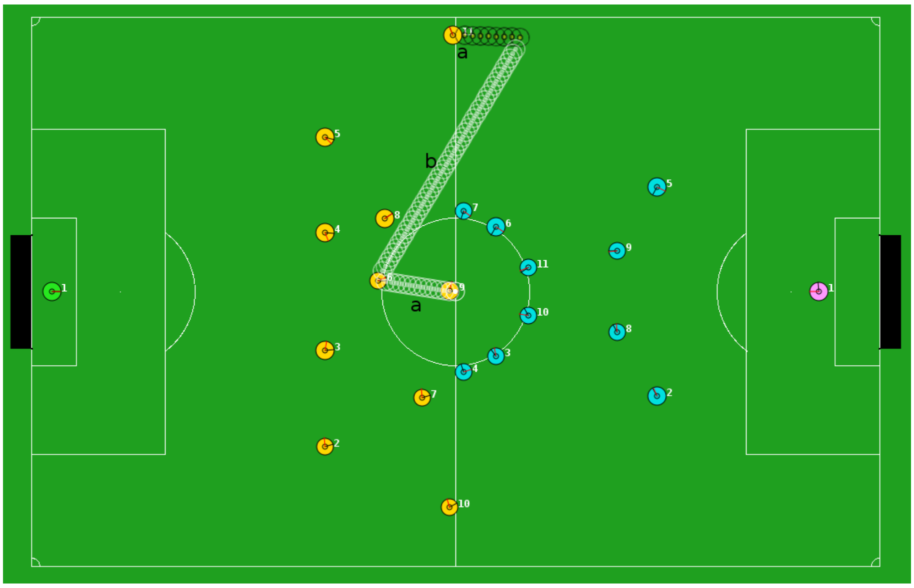


Figure 8.5: Example of a Kick-off Set Play involving 2 steps

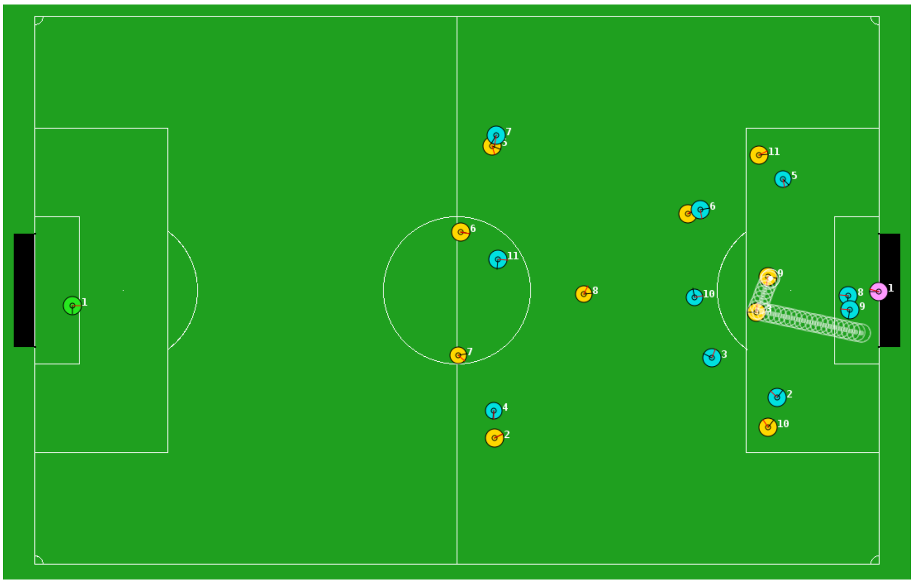


Figure 8.6: Example of a Free Kick Set Play direct to the goal

(a) **Goal Kick composed by 7 steps** (Figure 8.8)

i. The goalkeeper and the left defender move to the left region of the field (the left defender more to the left and in front of the keeper);

ii. The goalkeeper kicks the ball to the region where the left defender is running. If the left defender intercepts the ball, the left midfielder moves to the left and front of the left defender;

iii. The left defender kicks the ball to the place where the left midfielder is running. If the left midfielder intercepts the ball, the left forward moves to the front and left of the midfielder;

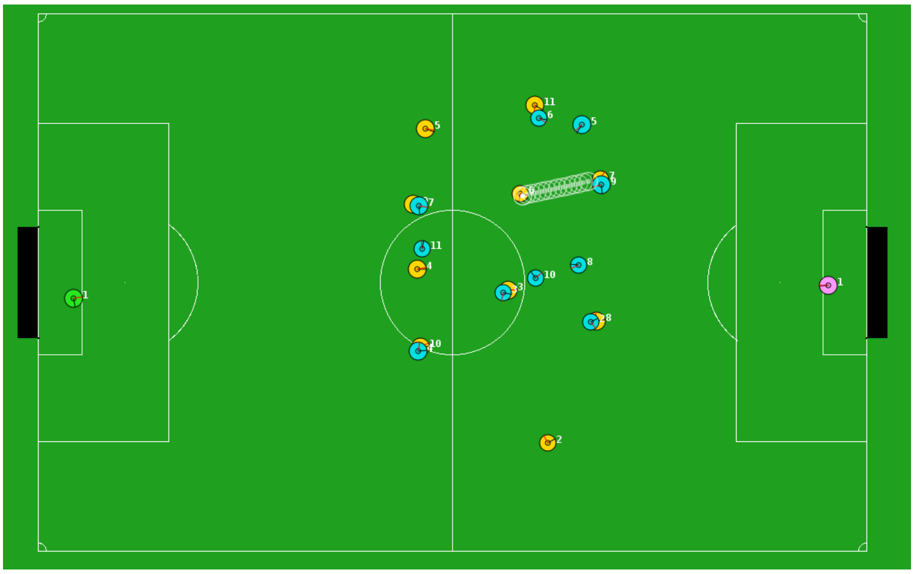


Figure 8.7: Example of a Free Kick Set Play where the ball possession is first priority

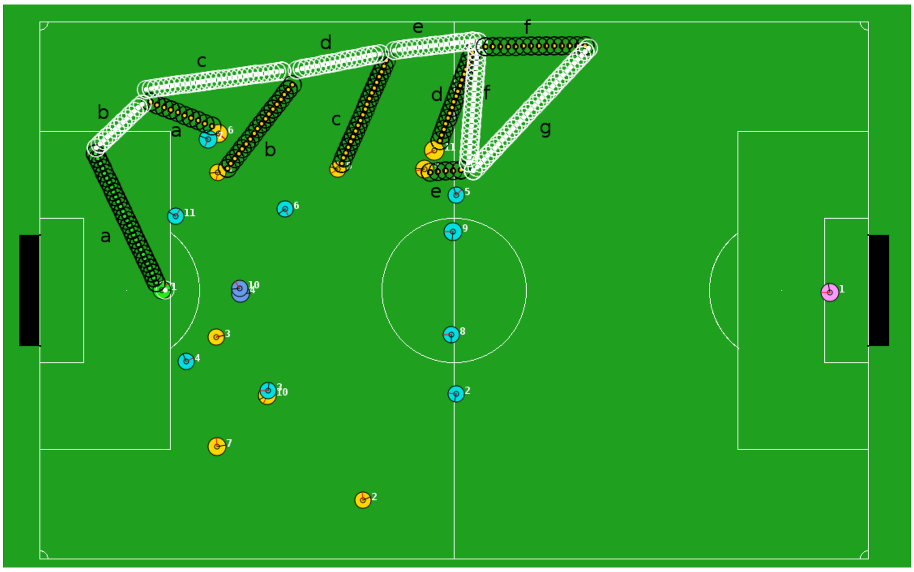


Figure 8.8: Example of a Goal Kick Set Play involving 7 steps

iv. The left midfielder kicks the ball to be intercepted by the left forward. A runner moves to the front of the left forward;

v. The left forward kicks the ball forward to be intercepted by the runner;

vi. The runner kicks the ball to his right and it is received by a kicker. The runner keeps running forward;

vii. The kicker kicks the ball in a left and forward direction to be intercepted by the runner.

(b) **Goal Kick composed by 3 steps** (Figure 8.9)

i. The players position themselves in two rows perpendicular to the goal line and

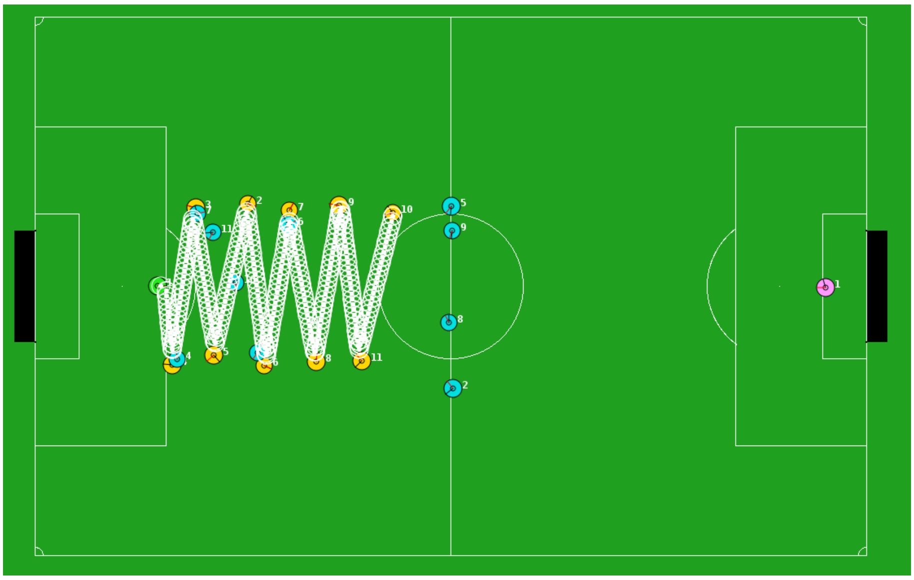


Figure 8.9: Example of a Goal Kick Set Play involving 3 steps

in front of the penalty area (except the goalkeeper who stays inside the penalty area). The goalkeeper kicks the ball to the player closest to him on his right;

ii. The player that has the ball, passes it to the player in front of him (on the other row) and slightly ahead of him (it is a zig zag slalom);

iii. The next step is repeated until the last player receives the ball.

4. **Corner Kick** is awarded when the whole of the ball passes over the goal line without a goal being scored, and it was last touched by a player from the defending team. Two corner kick situations were used:

   (a) **Corner Kick to the left striker**.

      i. The taker passes the ball to the nearest receiver.

      ii. The receiver passes the ball to the left striker who is in front of the opponent's goal;

      iii. If the left striker has space to shoot, he will shoot at the goal, if not he passes the ball to the center striker;

      iv. The center striker receives the ball and shoots;

   (b) **Corner Kick**

      i. The taker passes the ball to the nearest receiver;

      ii. The receiver dribbles the ball back to the middle of the penalty box;

      iii. The receiver passes the ball to either the left, right or center striker;

      iv. Whoever receives the ball shoots at the goal.

8.3 Methodology

As previously mentioned this approach has two distinct phases: offline and online (Figure 8.10). Basically the offline phase has 5 steps and the online phase is composed by 2 steps. Finally it is important to note that all the data processing in this project was performed using the R software version 2.11.0 [3].

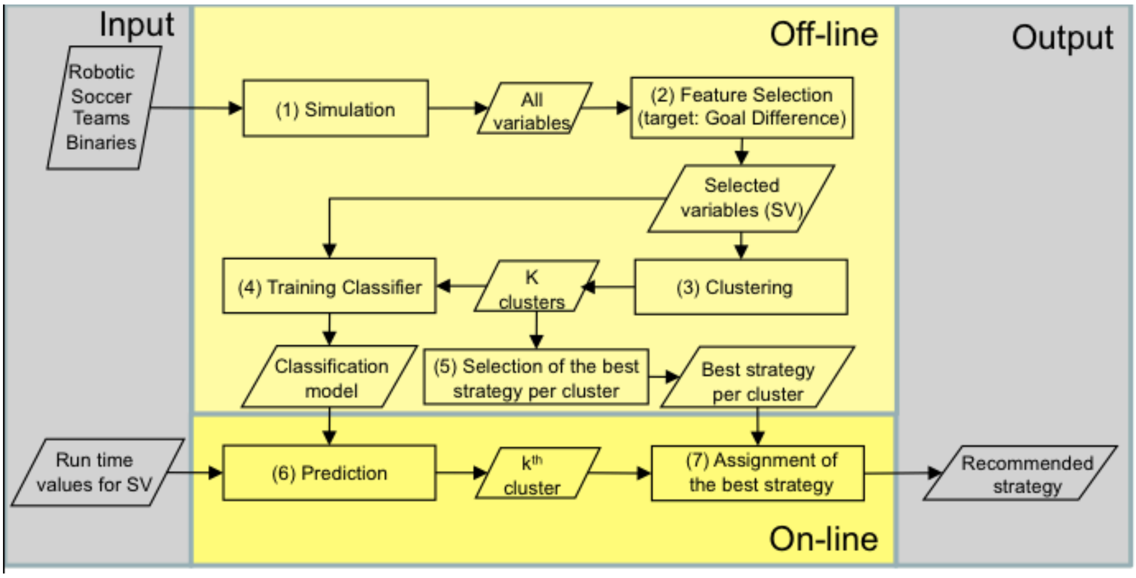


Figure 8.10: Project Architecture

8.3.1 Simulation

In this phase more than 280 Robocup 2D competition log files (between 2006-2009) were chosen selecting, at least 3 games per each team per year. After that and, using the SSSET tool (previously explained in the chapter 7) 56 statistics were calculated according to a soccer experts board. This board defined 5 groups of statistics. The first group is designated as the *Passes Group* which is basically composed by statistics related to the pass event (Table 8.1). In this group the successful and missed passes were filtered according to the half of the game (first or second half) and also according to the soccer field regions where they were executed (defensive or offensive).

The total number of passes chain and the wing chain (concepts previously explained in the chapter 7) were also included in this group as well as the year of the competition. Another group defined by the soccer board is the *Shots Group* which basically defined three types of shots (shot, intercepted shot and shot on target) filtered by the half of the game (first and second half) (Table 8.2).

Table 8.3 exposes the *Set Pieces Group*. In this group some game situations were defined (Throw-in, Goal Kick, Corner, Offside and Intercepted Offside) according once again, to the half

[3] More informations available at `http://www.r-project.org/`

Statistics Group	Acronym	Definition
Passes	Year	Competition Year
	1-GoodPassTot	Total number of sucessfully executed passes in the first half of the game
	1-GoodDef	Total number of sucessfull passes in the first half of the game executed in the defensive zone
	1-GoodOff	Total numberof sucessfull passes in the first half of the game executed in the offensive zone
	2-GoodPassTot	Total number of sucessfully executed passes in the second half of the game
	2-GoodDef	Total number of sucessfull passes in the second half of the game executed in the defensive zone
	2-GoodOff	Total number of sucessfull passes in the second half of the game executed in the offensive zone
	1-BadPassTot	Total number of missed passes in the first half of the game
	1-BadDefDefensive	Total number of missed passes in the first half of the game executed in the defensive zone
	1-BadOff	Total number of missed passes in the first half of the game executed in the offensive zone
	2-BadPassTot	Total number of missed passes in the second half of the game
	2-BadDef	Total number of missed passes in the second half of the game executed in the defensive zone
	2-BadOff	Total number of missed passes in the second half of the game executed in the offensive zone
	PassChain	Total number of consecutive passes executed by a team
	WingChain	Total number of sucessfull passes between two defined regions

Table 8.1: The Passes Group

Statistics Group	Acronym	Definition
Shots	1-Shot	Total number of executed shots in the first half of the game
	1-IntShot	Total number of executed intercepted shots in the first half of the game
	1-ShotTarget	Total number of executed shots on target in the first half of the game
	2-Shot	Total number of executed shots in the second half of the game
	2-IntShot	Total number of executed intercepted shots in the second half of the game
	2-ShotTarget	Total number of shots on target executed in the second half of the game

Table 8.2: The Shots Group

of the game. Finally two more groups were defined. The *Goal Group* (Table 8.4) which is composed by the total number of scored goals in each half of the game as well as the soccer field region where the goals are executed. The Goal Opportunity concept is also defined in this group (previously explained in chapter 7).

Statistics Group	Acronym	Definition
Set Pieces	OutTot	Total number of outside situations in a soccer game
	1-GoalKick	Total number of goal kicks in the first half of the game
	1-Corner	Total number of corners in the first half of the game
	1-ThrowIn	Total number of Throw-ins in the first half of the game
	2-GoalKick	Total number of goal kicks in the second half of the game
	2-Corner	Total number of corners in the second half of the game
	2-ThrowIn	Total number of Throw-ins in the second half of the game
	1-Offside	Total number of offsides in the first half of the game
	2-Offside	Total number of offsides in the second half of the game
	1-OffInt	Total number of intercepted offsides in the first half of the game
	2-OffInt	Total number of intercepted offsides in the second half of the game

Table 8.3: The Set Pieces Group

Statistics Group	Acronym	Definition
Goals	GoalsTot	Total number of scored goals
	1-Goals	Total number of scored goals in the first half of the game
	2-Goals	Total number of scored goals in the second half of the game
	PenBoxBack	Total number of scored goals executed inside the Penalty Box Area
	PenArea	Total number of scored goals executed inside the Penalty Area
	OutPenArea	Total number of scored goals executed outside the Penalty Area
	GoalsOpp	Total number of goal opportunities

Table 8.4: The Goal Group

The last group is concerned with the Ball Possession concepts (Table 8.5). Inside of it, the soccer experts board included the different types of attack (broken, fast, medium and slow) as well as the ball possession throughout 12 field regions (6 defensive and 6 offensive regions).

8.3.2 Feature Selection

In this phase the MARS algorithm (see section 2.10.4.1) is used to select the statistics that most influence the final game results (using the difference between the goals scored as a target) from an initial set of 60 final game statistics. The use of this algorithm is supported in experimental results that are presented in the next section.

Statistics Group	Acronym	Definition
Ball Possession	BroAtt	Total number of broken attacks per game
	FasAtt	Total number of fast attacks per game
	MedAtt	Total number of medium attacks per game
	SloAtt	Total number of slow attacks per game
	AttTot	Total number of attacks per game
	1-LeftBposs-Def	Ball possession throughout a left defensive field zone
	2-LeftBposs-Def	Ball possession throughout a left defensive field zone
	3-LeftBposs-Attack	Ball possession throughout a left offensive field zone
	4-LeftBposs-Attack	Ball possession throughout a left offensive field zone
	1-MiddBposs-Def	Ball possession throughout a middle defensive field zone
	2-MiddBposs-Def	Ball possession throughout a middle defensive field zone
	3-MiddBposs-Attack	Ball possession throughout a middle offensive field zone
	4-MiddBposs-Attack	Ball possession throughout a middle offensive field zone
	1-RightBposs-Def	Ball possession throughout a right defensive field zone
	2-RightBposs-Def	Ball possession throughout a right defensive field zone
	3-RightBposs-Attack	Ball possession throughout a right offensive field zone
	4-RightBposs-Attack	Ball possession throughout a left offensive field zone

Table 8.5: The Ball Possession Group

8.3.3 Clustering

After the feature selection phase, the data calculated are grouped into K clusters using the K-means algorithm (see section 2.10.4.3). Since clustering algorithms are sensitive to the existence of irrelevant features, the previous phase (step 2) is crucial in order to enhance clustering results.

8.3.4 Training Classifier

In this phase a classifier is trained to predict the group that best characterizes a given input data. For that three different classification algorithms were tested: Bagging, Random Forest (RF) and Support Vector Machines (SVM) – see Sections 2.10.4.4, 2.10.4.5 and 2.10.4.6. These algorithms are used according to the results exposed in a benchmark study [MLH03] that compares 17 state-of-the-art classifiers on 21 datasets.

8.3.5 Selection of the Best Strategy per Cluster

In this step, the best strategy expected for a group of opponents with similar behaviors is defined. The best strategy was chosen according to the maximum number of goals scored.

8.3.6 Prediction

In this step, for which the given data is expected to be more similar with is predicted. The model used for prediction was trained in step 4;

8.3.7 Assignment of the Best Strategy

Finally, in this phase the best strategy expected for a particular way of playing of the opponent is assigned. For this, we have used the predicted group (obtained in the previous step) and the best strategy per cluster (step 5) in order to obtain the strategy that optimizes the difference of goals scored for a given runtime input data. To choose the best strategy per cluster, the following criteria were used in a sequential manner until only one strategy was selected:

1. Pick up the one with the maximum difference of goals scored from the FC Portugal team's perspective;
2. Pick up the one that appears more in each group;
3. Pick up the one with the maximum value for the evaluation MARS function.

8.4 Experimental Results

With regards to Figure 8.10, the first step of our approach was to develop a framework capable of automatically calculating the final game statistics through RoboCup 2009 2D Simulation League log files. An exhaustive description about this step is available in chapter 7. Before starting step (2) of our project, which included the identification of a subset of the calculated statistics (56) that most influence the game result through a feature selection algorithm. A data standardization was executed, according to Equation 8.3 (where μ_x is the sample average and σ_x is the sample standard deviation). As mentioned before an analysis between the input variables (56) and the target variable (goal scored difference) was executed. It is important to note that this set of statistics can be divided into five groups: Passes, Shots, Goals, Set Pieces and Ball Possession analyzed for each part of the game (for further information, please consult the chapter 7). Figure 8.11 illustrates those relationships. From the analysis of this figure one can conclude that the relationship between those variables are non-linear (only the goal variables presented an almost linear relationship with the target variable – difference of goals scored). Due to this fact it is not possible to use linear regression and the use of other parametric algorithms does not seem appropriate. The use of non-parametric techniques capable of constructing an evaluation function of the problem with good prediction accuracy was the selection criteria.

$$x = \frac{x - \mu_x}{\sigma_x} \tag{8.3}$$

8.4.1 Feature Selection Algorithm

In order to select a feature subset to avoid irrelevant features that can decrease the performance of the clustering algorithm (step 3) and obtain an evaluation function that is used in step 7 a

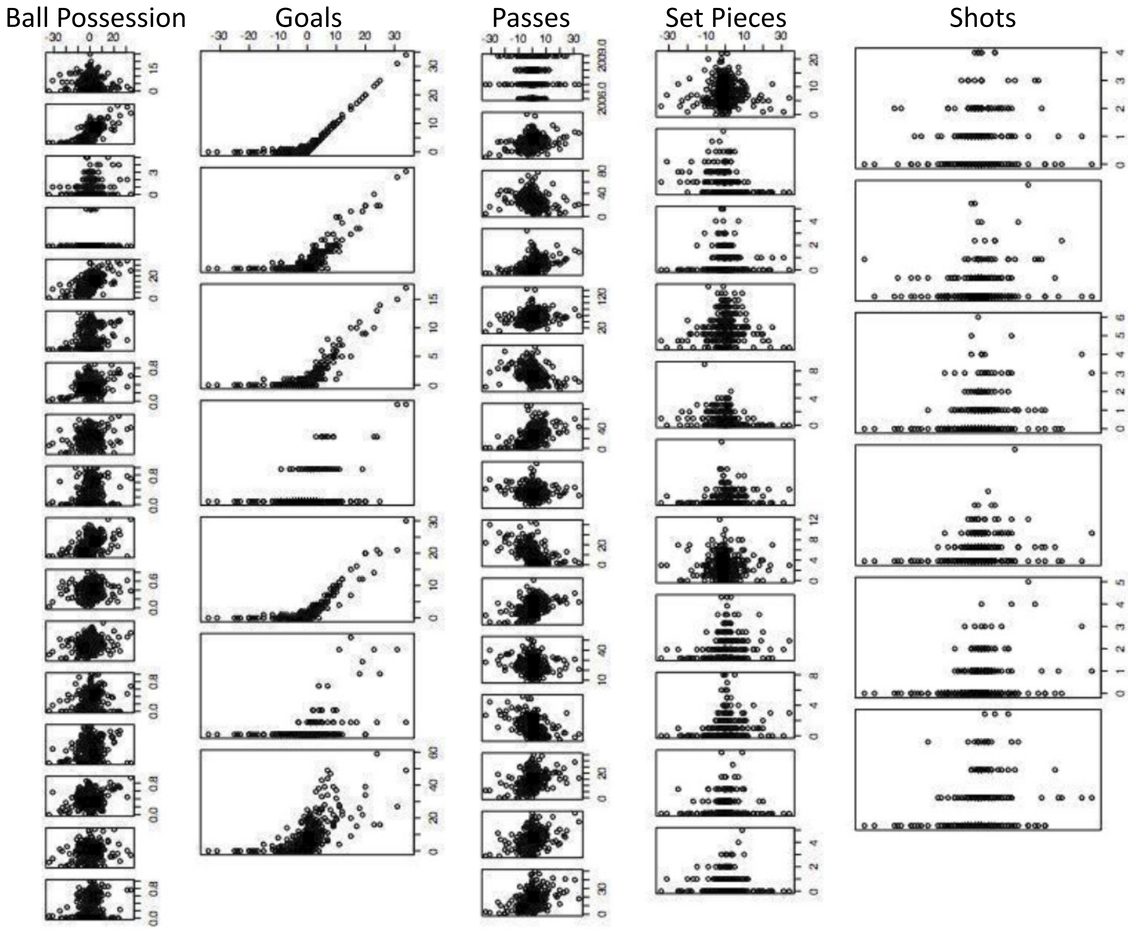


Figure 8.11: Relation between Input Variables and Target Variable divided by groups (Ball Possession, Goals, Passes, Set Pieces, Shots respectively)

comparison study between two feature selection algorithms (RreliefF and MARS) was performed. For that, 235 log files of RoboCup 2D Simulation League were collected (including at least three games of each team per competition) and a set of final game statistics were calculated (previously mentioned). Using the MARS algorithm an evaluating function was obtained (Table 8.6). It is important to note that to evaluate MARS and RreliefF the following measures are used: the RSq, coefficient of determination, GRSq, a measure of how well the next value can be predicted using the structural part of the model and the past values of the residuals. The GRSq and RSq values vary between 0 and 1(the higher the better). In order to increase the interpretability of the evaluation formula, the correlation matrix was analyzed and the variables with highest correlation values were identified(Table 8.7). After some tests, and as not to undermine the RSq value, the best evaluation formula, obtained through the elimination of the 1-BadDefDefensive, is represented in Table 8.8. To produce another evaluation function in this research work, the RreliefF algorithm was also used. Typically this algorithm is used for feature selection problems; however, and following the work described in [RSK03], it is also possible to use the solution produced by this algorithm in a form of an evaluation function. The process consists in 2 steps: run the RreliefF algorithm and

Weight	Statistic	Interval
1.3236	Year	MAX[0, 2007-value]
0.1758	1-GoodPassTot	MAX[0, value-28]
-0.4432	1-GoodPassTot	MAX[0, 28-value]
0.1942	1-GoodDef	MAX[0, 45-value]
0.5634	1-GoodOff	MAX[0, 13-value]
-0.1778	1-GoodOff	MAX[0, value-19]
-0.0729	2-GoodDef	MAX[0, value-24]
0.485	1-BadPassTot	MAX[0, 38-value]
-0.3821	1-BadDefDefensive	MAX[0, 19-value]
0.3927	1-BadOff	MAX[0, value-6]
-0.652	1-BadOff	MAX[0, 6-value]
-0.5733	2-IntShot	MAX[0, 2-value]
1.08	GoalsTot	MAX[0, value-2]
-1.4054	GoalsTot	MAX[0, 2-value]
-0.2652	1-Throwin	MAX[0, 6-value]
-1.733	FasAtt	MAX[2-value]
-0.5532	AttTot	MAX[0, value-6]
-0.776	AttTot	MAX[0, 6-value]
0.5831	AttTot	MAX[0, value-9]
-9.4419	1-LeftBPoss-Def	MAX[0, 0.1489-value]
-30.8963	2-LeftBPoss-Def	MAX[0, 0.1439-value]
-5.9118	1-MiddBPoss-Def	MAX[0, 0.2911-value]
-26.7951	2-MiddBPoss-Def	MAX[0, 0.1794-value]
22.9079	3-MiddBPoss-Attack	MAX[0, value-0.5229]
-11.517	3-MiddBPoss-Attack	MAX[0, value-0.4092]
-6.5698	4-MiddBPoss-Attack	MAX[0, 0.2230-value]
GRSq	0.8000512	
RSq	0.8486588	

Table 8.6: Statistics that Composed the Mars Function

obtain the evaluation function (Table 8.9).

After that, and in order to increase the interpretability of the function (as suggested by [Hal00]), the variables that present values below 0.01 are eliminated (considering that they are irrelevant in order to explain the target variable). Next, and before running the RreliefF again (to obtain a new evaluation function with less variables), a prediction algorithm was run (in this case MARS) as to evaluate the RSq (coefficient of determination) – a measure of how well the next value can be predicted using the structural part of the model and the past values of the residuals – Values of GRSq and RSq above 0.80 are considered good [SK92].

This process was repeated for seven times and the final evaluation function is represented in Table 8.10. The threshold used to stop the process was 0.8 for the RSq. The initial value of RSq was higher than 0.83. The use of an evaluation function with fewer variables has priority, mainly due to restrictions related to the real-time mode (explained below).

Doing a comparison between the two evaluation functions, it is easy to note that both of them present an RSq value greater than 0.80, which composes excellent perspectives, since these functions attempt to model the 2D competition between 2006 and 2009. It is also important to note that the MARS evaluation function uses only 19 variables, 11 of them also present in the RreliefF evaluation function (this relation can increase to 18-11 if we don't considered the variable "Year" which after 2007 disappear in the MARS function). This fact could be explained by the nature

Variables	Correlation Value
FastAtt & AttTot	0,768
GoalsTot & FastAtt	0,658
1-GoodDef & 2-GoodDef	0,648
1-BadPassTot & 1-BadDefDefensive	0,605

Table 8.7: The Highest Correlation Coefficient Table for Soccer Variables

Weight	Statistic	Interval
1.6443	Year	MAX[0, 2007-value]
-0.3069	1-GoodPassTot	MAX[0, value-28]
0.5093	1-GoodDef	MAX[0, value-45]
-0.2746	1-GoodDef	MAX[0, 45-value]
0.4722	1-GoodOff	MAX[0, value-13]
-0.1705	1-GoodOff	MAX[0 value-21]
-0.0594	2-GoodDef	MAX[0, value-22]
0.1417	1-BadPassTot	MAX[0, 38-value]
0.571	2-BadPassTot	MAX[0, value-38]
-0.5607	2-BadDef	MAX[0, value-23]
-0.721	2-IntShot	MAX[0, 1-value]
1.0879	GoalsTot	MAX[0, value-2]
-1.5165	GoalsTot	MAX[0, 2-value]
-0.2664	1-Throwin	MAX[0, 6-value]
-1.1549	1-OffInt	MAX[0, 1-value]
-1.6546	FasAtt	MAX[0, 2-value]
-0.6579	AttTot	MAX[0, 6-value]
-11.6797	1-LeftBPoss-Def	MAX[0, 0.01489-value]
-41.8	2-LeftBPoss-Def	MAX[0, 0.1439-value]
-29.2901	2-MiddBPoss-Def	MAX[0, 0.1794-value]
3.4187	2-RightBPoss-Def	MAX[0, 0.5647-value]
GRSq	0.7967068	
RSq	0.837157	

Table 8.8: Statistics that Composed the Mars Function without 1-BadDefDefensive

of these two algorithms. The MARS algorithm uses a greedy approach, which means that once a variable with a substantial weight is found, all correlated variables are more likely to be discarded. On the other hand, RreliefF employs a different process – when the algorithm identifies two correlated variables with a substantial weight, the weight is (approximately) divided in two, maintaining both variables in the evaluation functions. This fact is the main reason that explains the high number of variables in the RreliefF (compared to the other approach) and the similarities between the variables presented in both algorithms. Doing a more deep analysis in the 11 common variables in the two evaluation functions it is important to note that 4/11 are variables related to passes (1-GoodPassTot, 1-GoodDef, 2-GoodDef and 2-BadDef), 4/11 related to Ball Possession (AttTot, 1-LeftBPoss-Def, 2-LeftBPoss-Def, 2-MiddBPoss-Def, 2-RightBPoss-Def), 1 related to set piece (throwin), 1 related to shots (2-intshot) and finally 1 related to the number of scored goals (goalsTot). Doing a parallelism between these results and professional soccer, if we analyzed the top soccer teams it is easy to conclude that they present some common characteristics: a high percentage of ball possession and a good percentage in successful passes in any field region (independent of the half of the game). Also, throughout the years, these teams have spent many

Weight	Statistic	Weight	Statistic
0.0026	Year	0.1057	2-Corner
0.0815	1-GoodPassTot	0.0202	2-Throwin
0.0382	1-GoodDef	0.0002	1-Offside
0.1523	1-GoodOff	0.0342	2-Offside
0.0517	2-GoodPassTot	0.1281	1-Offint
0.0143	2-GoodDef	0.0577	2-Offint
0.1201	2-GoodOff	0.0151	BroAtt
0.0171	1-BadPassTot	0.0018	FasAtt
0.0432	1-BadDefDefensive	0.0888	MedAtt
0.0059	1-BadOff	0.0087	SloAtt
0.1198	2-BadPassTot	0.0154	AttTot
0.1028	2-BadDef	0.0939	1-LeftBPoss-Def
0.0051	2-BadOff	0.0703	2-LeftBPoss-Def
0.0377	1-Shot	0.0563	3-LeftBPoss-Attack
0.0745	1-IntShot	0.0234	4-LeftBPoss-Attack
0.1272	2-ShotTarget	0.046	1-MiddBPoss-Def
0.1518	GoalsTot	0.1016	2-MiddBPoss-Def
0.0781	1-Goals	0.0118	3-MiddBPoss-Def
0.0969	2-Goals	0.0602	4-MiddBPoss-Attack
0.0444	PenBoxBack	0.1259	1-RightBPoss-Def
0.1269	PenArea	0.0988	2-RightBPoss-Def
0.0183	OutPenArea	0.0153	3-RightBPoss-Attack
0.0968	OutTot	0.0846	4-RightBPoss-Attack
0.1271	1-GoalKick	0.0465	GoalsOpp
0.1396	1-Corner	0.0127	PassChain
0.0427	1-Throwin	0.0724	WingChain
0.0287	2-GoalKick		
GRSq	0.814361		
RSq	0.834508		

Table 8.9: Statistics that Composed the RreliefF Function

Weight	Statistic	Weight	Statistic
0.1085	1-GoodPassTot	0.2173	PenArea
0.0333	1-GoodDef	0.0976	OutPenArea
0.0796	2-GoodPassTot	0.1052	1-Corner
0.0128	2-GoodDef	0.0734	1-Throwin
0.0083	2-GoodOff	0.0593	2-Corner
0.0274	1-BadDefDefensive	0.0309	2-OffInt
0.0708	2-BadPassTot	0.0064	BroAtt
0.0448	2-BadDef	0.0173	MedAtt
0.0577	1-Shot	0.0845	AttTot
0.0481	1-IntShot	0.009	2-LeftBPoss-Def
0.0018	1-ShotTarget	0.0761	3-LeftBPoss-Attack
0.0098	2-Shot	0.0366	2-MiddBPoss-Def
0.0099	2-IntShot	0.0458	4-MiddBPoss-Attack
0.0828	2-ShotTarget	0.0759	1-RightBPoss-Def
0.2604	GoalsTot	0.0009	2-RightBPoss-Def
0.1799	2-Goals	0.05999	GoalsOpp
0.0273	PenBoxBack		
GRSq	0.7733726		
RSq	0.8143865		

Table 8.10: Statistics that Composed the RreliefF Function after seven iterations

training hours worked set pieces situations. Throw-ins is a particular set piece that normally takes a team performed long throws into the penalty area creating goal opportunities situations. Finally and, having the main goal of the game in mind, the number of scored goals would be always an important factor that a professional coach must be aware.

In order to validate this approach, the values of the evaluation function were calculated for 52 games of the best, average and worst teams – see Table 8.11. The main reason to pick these games was to analyze the variation of the evaluation functions values, according to different kinds of strategy used by these distinct teams (with different competition goals and final results). For this analysis, the best three games of each team throughout the tournaments (according to the difference of scored goals) were selected, and for the teams from the middle of the classification table, all games were analyzed (as these teams have a greater variation in results compared to others, the authors considered important to conduct a more detailed analysis). To perform a comparison between the function values, the mean squared error (MSE) function (8.4) was also calculated. Nowadays, there are several ways to quantify the difference between the estimator (in this particular case the value calculated by the evaluation function) and the true value of the quantity being estimated.

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{f}(Xi)-f(Xi))^2 \tag{8.4}$$

The MSE is a possible evaluation metric that measures the average of the squared error. The error is the amount by which the estimator differs from the real value (in this case the goal scored difference). The values obtained were 14, 68 for MARS and 384,4 for RreliefF respectively (from a pairwire ***t*-test** for a significance level of 5%. The **p-value** for this test was 2.2e-16. This value shows that the MARS algorithm is a better approach for this particular scenario. In consequence

	2006	2007	2008	2009
Better Team	WE2006	Brainstormers	Brainstorm08	WE
Average Team	DAINAMITE	NCL	Hfut_Engine	NemesisRC09
Worst Team	Mersad	Brasil2D	ATHumbold	Bahia2D

Table 8.11: Performance Comparison Table between the Best and the Worse Teams

of that, in step (2) we used the MARS algorithm to discover which were the final game statistics that most influence the game results (using as target the difference of scored goals and 2009 competition logs). The obtained expression is shown in Table (8.12). This expression includes variables related to total number of bad passes (BadPassTot), Pass Chains (PassChainTot), outside situations (OutTot), number of goals (GoalsTot), number of attacks (AttTot) and, finally statistics related to ball possession per zones (1-LeftBposs-Def, 2-MiddBposs-Def, 3-RightBposs-Attack etc.).

8.4.2 Clustering

Having the previous obtained knowledge as a base, the input of the system consists of three robotic team's binaries (Wright Eagle, Nemesis and Bahia2D) that occupied distinct positions in the final classification table (RoboCup 2009 2D simulation league) in order to have a high spectrum of final game simulation results (these teams will also be used in the online mode). Following this,

Weight	Statistic	Interval
1.5131	BadPassTot	MAX[0, value-75]
0.0944	BadPassTot	MAX[0, 75-value]
1.5339	BadPassTot	MAX[0, value-70]
0.3496	BadPassTot	MAX[0, value-62]
1.048	GoalsTot	MAX[0, value-2]
-1.6261	GoalsTot	MAX[0, 2-value]
-0.1608	OutTot	MAX[0, 10-value]
2.2987	GoalKick	MAX[0, value-3]
-1.6205	GoalKick	MAX[0, value-2]
0.9517	OffInt	MAX[0, value-2]
-2.021	FasAtt	MAX[0, 1-value]
-0.7101	AttTot	MAX[0, value-6]
-1.2009	AttTot	MAX[0, 6-value]
0.7653	AttTot	MAX[0, value-9]
-10.278	1-LeftBPoss-Def	MAX[0, 0.1337-value]
-29.2997	2-LeftBPoss-Def	MAX[0, 0.1439-value]
-2.4553	1-MiddBPoss-Def	MAX[0, 0.55-value]
32.5304	2-MiddBPoss-Def	MAX[0, 0.1794-value]
-6.0374	4-MiddBPoss-Attack	MAX[0, 0.2307-value]
-2.8867	3-RightBPoss-Attack	MAX[0, value-0.2977]
-8.8114	3-RightBPoss-Attack	MAX[0, 0.2977-value]
0.3035	PassChain	MAX[0, value-13]
-0.249	PassChain	MAX[0, value-7]
GRSq	0.7827846	
RSq	0.8317884	

Table 8.12: Statistics that Composed the Mars Function

16 distinct team strategies (8 set plays and 2 formations) were defined and, in order to avoid outliers, each team used the same tactic in 10 simulations. As such, we simulated 480 games – 3 teams x 16 strategies x 10 simulations. Using the obtained subset of statistics, the data instances are then grouped according to their similarity (step (3)). We have used the K-means algorithm. In order to determine which is the optimal number of k clusters (the optimal number of cluster is a compromise between the minimum sum square errors and the minimum number of clusters), the GAP algorithm was used. According to [TWH01] the number that maximized the GAP should be used as the number of clusters (in our case this value was 9).

8.4.3 Training Classifer

In the next step (4) and according to the guidelines presented in [MLH03] three data mining methods (SVM, Random Forest and Bagging) were trained for the prediction of the group that would best characterizes the data input (using as factor the k produced in the previous step). In the training process a Cross Validation *10-fold* was used and more than 1920 log files, including matches of the RoboCup 2D simulation league between 2006 and 2009 were also used.

With reference to the percentage rate in the train process, SVM is the one that presents the highest result (96,4%), followed by RandomForest (93,59%) and Bagging (80,15%). The statistical validation of these results was done using the Friedman rank test with the statistic derived by Iman and Davenport as described in [Dem06]. The null hypothesis of equivalence between the three predictors is rejected with a ***p-value*** of 0.038. The 480 tested games were divided into 12 groups (4 groups per team and 40 games per group). For each group, the algorithms were ranked according to their percentage rate (the results are shown in Table 8.13). Comparing the three predictors for a 5% significance level with the Nemenyi test[Dem06], we have obtained CD = 0.96 . CD is the critical value for the difference of mean ranks between the three predictors and with regards to their average value, it is easy to conclude that these predictors present significant statistical differences. Consequently, it is easy to note that, the only difference between the analyzed predictors that is smaller than the critical value (CD) is related to the SVM and Random Forest algorithms presenting a difference of 0.8334 of difference. The differences with regards to the other two pairs Bagging/SVM and Bagging/RandomForest are 1.91667 and 1.08333 respectively.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Mean**
Random Forest	2	2	2	2	2	2	2	2	2	1	2	2	1.91667
SVM	1	1	1	1	1	1	1	1	1	2	1	1	1.08333
Bagging	3	3	3	3	3	3	3	3	3	3	3	3	3

Table 8.13: Ranks of the Friedman test

8.4.4 Assignment of the Best Strategy

After the results obtained, in this step (5), the best strategy expected (combination of Set Play and Formation) was defined according to the similarity of opposing group's behavior (this step ends the offline phase).

For step (6), the first one from the online phase, the SVM algorithm was used (based on the results produced in step (4)) to predict the group where each game (input data) is expected to be more similar to. Finally, in step (7), the expected best strategy was assigned according to the opponent's behavior.

To produce experimental results for the online components, 840 games between FC Portugal and Wright Eagle, Nemesis and Bahia 2D team were simulated and the SVM algorithm was used with three different frequencies SVM500 (500 in 500 cycles), SVM1000 (1000 in 1000 cycles) and SVM2000 (2000 in 2000 cycles) – remember that a simulated robotic soccer has 6000 cycles. These three predictors were compared with a Baseline algorithm, which is basically an approach that chooses the strategy that a FC Portugal will use before the game and, during the game it does not make any changes. The comparison is done using the Friedman rank test. We have compared the best predictor of each algorithm (four different predictors). The 840 simulations were divided into 12 groups according to their opponent (for each opponent 280 simulation games were performed spread throughout 4 distinct groups).

The results were evaluated according to the number of wins, draws and defeats. If, for some reason, there was a draw between the results produced by at least two algorithms, a draw scale was used. This scale consisted of assigning 4 points for a victory, 2 points for a draw and 1 point for a loss. These values increased 0.1 per goal scored (increasing with the difference of goals scored in case of victory and depreciating in case of defeat). For example, if FC Portugal wins two games by the difference of 2 or 12 goals, these games will be ranked with 4.2 and 5.2 points respectively. On the other hand, if this team loses two games by the difference of 2 or 12 goals, these games will be ranked with 0.8 and -0.2 points respectively. The ranks obtained are shown in Figure 8.14. The null hypothesis of equivalence between the four predictors is rejected with a ***p-value*** of 0.00000303. Comparing the three predictors against the baseline for a 5% significance level with the Bonferroni-Dunn test [Dem06], we have obtained CD = 1.2617 . CD is the critical value for the difference of mean ranks between the baseline and any other of the three predictors. It is proved that SVM 500 and SVM2000 are better than the baseline.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Mean**
Baseline	4	4	4	4	4	3	3	4	4	3	4	2	3.583333333
SVM500	2	2	2	2	2	1	1	3	1	1	2	1	1.666666667
SVM1000	1	3	1	3	1	4	4	2	3	4	1	3	2.5
SVM2000	3	1	3	1	3	2	2	1	2	2	3	4	2.25

Table 8.14: Ranks of the Friedman test

8.5 Conclusions

In this chapter, an approach capable of improving a soccer team performance was presented. From a soccer team coach perspective it is important to obtain all possible knowledge regarding to how the opponent plays. This means preparation can be made for future training sessions and a strategy can be chosen for early in the game (offline phase). Furthermore, it also helps to execute the best team strategy changes during the game (online phase), improving at the end the team's performance. Focusing on these two phases, the main goal of this research is to prove that if a coach agent during the game periodically changes his team strategy using the knowledge previously generated in the offline phase, it is possible to improve his team's performance. As outlined in the previous section the SVM proved to be the best algorithm in the prediction step (step (4)) and also the SVM 500 and SVM 2000 proved to be better than a baseline algorithm with reference to the increase of the FC Portugal performance task (step (7)).

Chapter 9

Conclusions and Future Work

This chapter gives an overview of the work reported in this thesis, referring to the original contributions. Also some future work directions will be exposed.

Evaluating the Initial Objectives

In the next sections, the thesis objectives, separated by priorities, will be analyzed.

Main Objectives

The main objective of this thesis is to prove that using previous soccer knowledge it is possible to improve a soccer team's performance by modifying its strategy during a soccer game. To achieve that goal, a statistical tool and an online model were created:

1. **Construct a Statistical Generator Tool - SSSET** (chapter 7) : A statistical generator tool was developed, able to automatically calculate final game statistics. Using RoboCup log files, this tool is able to automatically calculate more than 60 statistics. It is important to note that this statistics set was defined by a soccer expert board, constituted by academic and sport researchers.

2. **Construct a FC Portugal Online Model** (chapter 8): Using the final game statistics previously calculated through the generator tool, a FC Portugal online model was constructed and four distinct predictors were used to choose, in real time, the best strategy to use, improving our team performance. Doing a comparison between the distinct predictors used, it was proved that this kind of approach actually improves the team performance.

Secondary Objectives

Throughout this thesis, several secondary objectives were defined regarding the different phases:

1. **Use a Tracking System in a Soccer Field Environment** (chapter 5): At the beginning of this thesis, we used a Wi-Fi tracking system to generate data from a soccer match. However,

this experience proves that the tracking systems haven't yet achieved sufficient maturity to be used in outdoor environments. The software and hardware price, together with technical issues still constituted two major drawbacks in the use of these systems.

2. **Define a Soccer Ontology** (chapter 6): In spite of the fact that the ontology concept emerged over the last two decades, there is not an ontology capable of representing high level soccer concepts. Because of that, and to better understand the relationships between soccer concepts, an ontology able to represent a large set of concepts was developed and a survey was used to validate its concepts and relationships. The majority of the concepts have been validated by the experts and those results composed a good base for the beginning of the work presented in this thesis.

3. **Construct a Cartesian Statistical Tool** (chapter 7): The event detection in a soccer game is one of the most worked research topics over the last years. However, the answer to solve this problem is still far from being achieved. One of the realities that proves this situation is the manual processing of data that tracking systems still use today. In this thesis, an approach using only cartesian coordinates was presented. With this development, we created a tool that through ball and players cartesian coordinates (from a robotic data base or even a tracking system) is able to automatically calculate a large set of high level game statistics. To validate this approach, RoboCup 2D simulation league log files were used; however, this framework can be used in other projects involving for instance human soccer.

4. **Conduct a comparison Study between Human and Robotic Realities** (chapter 7): Having the RoboCup main goal in mind, a comparison study between human and robotic teams was performed using final game statistics. The results showed that these two realities still present some statistical differences and, because of that, to achieve the main goal by 2050, the developers team must pay attention to these factors.

Future Directions

Due to the nature of the developed work in this dissertation, the future work possibilities are diverse. In this section, we summarize the most important ones and some others that seem promising filtering by application domain.

Future Directions within Robotic Soccer

Regarding 2D simulated league which is the environment used in this thesis, there is a large number of possible future works:

1. **Strategics Features Implementation** : In this work two types of strategics features were used (formations and set pieces). An interesting development in the future consists in enlarging these features adding for instance a game flux (related to the prevalence of use of

some field regions relatively to others), game pace (the type of pass more often used by the players that can have a risk association concerning to the field region where it is executed) among others, increasing the range of options that a online coach can execute in his team during the game, and on the other hand turning the team strategy features more similar to the human reality.

2. **Expand this approach to other RoboCup Leagues**: The soccer challenge is almost transversal to all of the RoboCup leagues. Because of that it will be very interesting to expand the approach exposed in the chapter 9 to other leagues such as 3D simulation league, humanoid league among others, and trying to conclude that improve team performance using final game statistics it is not directly influenced by the application soccer domain but it is inherent to the nature of the game.

3. **Comparison Study**: Doing a more exhaustive study comparing the robotic and human soccer realities. In chapter 8 of this thesis we presented a study between the 2D simulation environment and human soccer including some competition finals. In the future a new study can be made including more games (specially human ones) and trying to understand if the improvements of the robotic soccer through the years are similar to the ones that were analyzed in the human soccer not only in the 2D league but also including other leagues. This will constitute a good benchmark for the robotic developers.

Future Directions within Human Soccer

In the human soccer domain many are the future work perspectives:.

1. **Construct a Tracking System**: One of the major difficulties in the development of this thesis is related to obtaining human soccer data. Because of that one research direction in the future can be developed a hybrid tracking system that use a RFID or a Wi-Fi technology to track the players and cameras to track the ball. This system can be used to a soccer coach to improve his team performance not only after a game but also in a training session. For the first approach we will need to construct a hybrid system . After that, we obtain the cartesian coordinates of the different stakeholders and we can calculate final game statistics through the soccer scientia framework presented in chapter 7. It is important to note that the faults events need to be manually processed. To complete this process the steps presented in chapter 9 should be performed.

2. **Expand this Approach to Human Soccer**: If for some reason the previous exposed task did not be possible to achieve, we will try to contact again the official soccer entities. If we have lucky this time, the procedure used in this thesis will be used with human data and we will be capable to conclude if this type of approach is valid in other environment.

3. **Expand this Approach to Other CSG**: We proved that obtained previous opponent information we are able to construct a model capable to improve our simulated robotic team

performance. Expanding this study to other sports like Hockey, HandBall or Basketball will constitute an interesting challenge in the future.

Concluding Remarks

In short, this dissertation contributes for creating a methodology for constructing an online model capable to improve the performance of a simulated robotic team. For that, a framework capable to automatically calculating a set of high level statistics was implemented; this set of statistics was the base of the online model (previously mentioned).

Appendix A

Soccer Concepts Survey

In this appendix, the survey used for validate the soccer concepts presented in the ontology will be exposed.

1. Regarding the next statements, which is more correct to define a "short pass"?
   (a) It is a move from "foot to foot";
   (b) It is a pass that is made for distances of less than 20 meters;
   (c) It is a pass that is made for distances of less than 10 meters;
   (d) It is a pass where the ball always remains close to the ground;
   (e) Another answer.
2. Regarding the next statements, which is more correct to define a "long pass"?
   (a) It is a pass that is made for distances over 10 meters;
   (b) It is a pass that is made for distances over 20 meters;
   (c) It is a pass that is made for distances over 30 meters;
   (d) It is a pass where the ball does not stay close to the ground;
   (e) Another answer.
3. Regarding the next statements, which is more correct to define a "long shot"?
   (a) It is a shot executed from outside the opponent´s penalty area;
   (b) It is a shot executed from a distance of 20 meters from the opponent goal perspective;
   (c) It is a shot executed from a distance of 30 meters from the opponent goal perspective;
   (d) Another answer.
4. In relation to Figure A.1, imagine that a player has the ball and attacking toward the goal represented. In your point of view, select the statements are true:

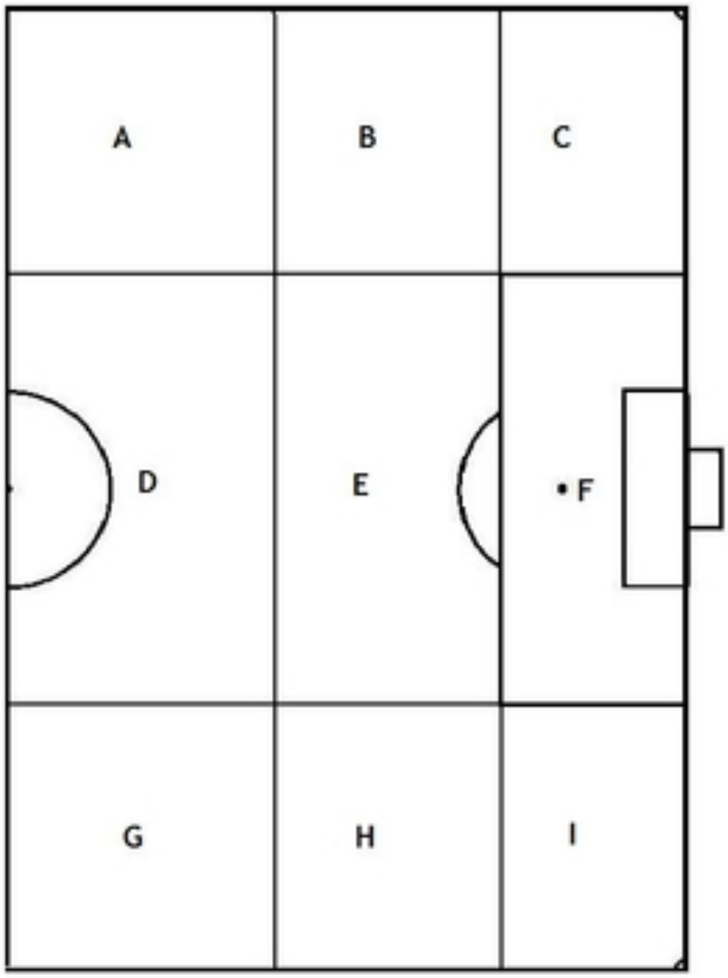


Figure A.1: A Crossing Game Situation Analysis.

(a) If the player is in region C and passes the ball to a teammate in the region E, one can consider that this player executes a crossing;

(b) If the player is in region C and passes the ball to a teammate in the region M, we can consider that this player executes a crossing;

(c) If the player is in region C and passes the ball to a teammate in the region H, we can consider that this player executes a crossing;

(d) If the player is in region C and passes the ball to a teammate in region I, one can consider that this player executes a crossing;

(e) If the player is in region B and passes the ball to a teammate in the region E, one can consider that this player executes a crossing;

(f) If the player is in region B and passes the ball to a teammate in the region F, one can consider that this player executes a crossing;

(g) If the player is in region B and passes the ball to a teammate in the region H, we can consider that this player executes a crossing;

(h) If the player is in region B and passes the ball to a teammate in region I, one can consider that this player executes a crossing;

(i) If the player is in region A and passes the ball to a teammate in the region E, one can consider that this player executes a crossing;

(j) If the player is in region A and passes the ball to a teammate in the region F, one can consider that this player executes a crossing;

(k) If the player is in region A and passing the ball to a teammate in the region H, we can consider that this player is executing a crossing;

(l) If the player is in region A and passes the ball to a teammate in region I, one can consider that a crossroad occurs

5. The Figure A.2 represents a game situation where only a few players are represented. The red team is in attacking process while the blue team is in defensive process. The numbers of the blue team players are between 1 to 6 and the red team are between 9-11 including 7. Player 10 (red team) has the ball. For each statement indicate your answer following a 4 value scale:

   (a) Player 3 is covering the pass line for player 7
      i. Totally Disagree;
      ii. Partially Disagree;
      iii. Partially Agree;
      iv. Totally Agree.

   (b) Player 3 is marking the player 7
      i. Totally Disagree;
      ii. Partially Disagree;
      iii. Partially Agree;
      iv. Totally Agree.

   (c) Player 3 is pressing player 7
      i. Totally Disagree;
      ii. Partially Disagree;
      iii. Partially Agree;
      iv. Totally Agree.

   (d) Player 6 is pressing player 10
      i. Totally Disagree;
      ii. Partially Disagree;
      iii. Partially Agree;
      iv. Totally Agree.

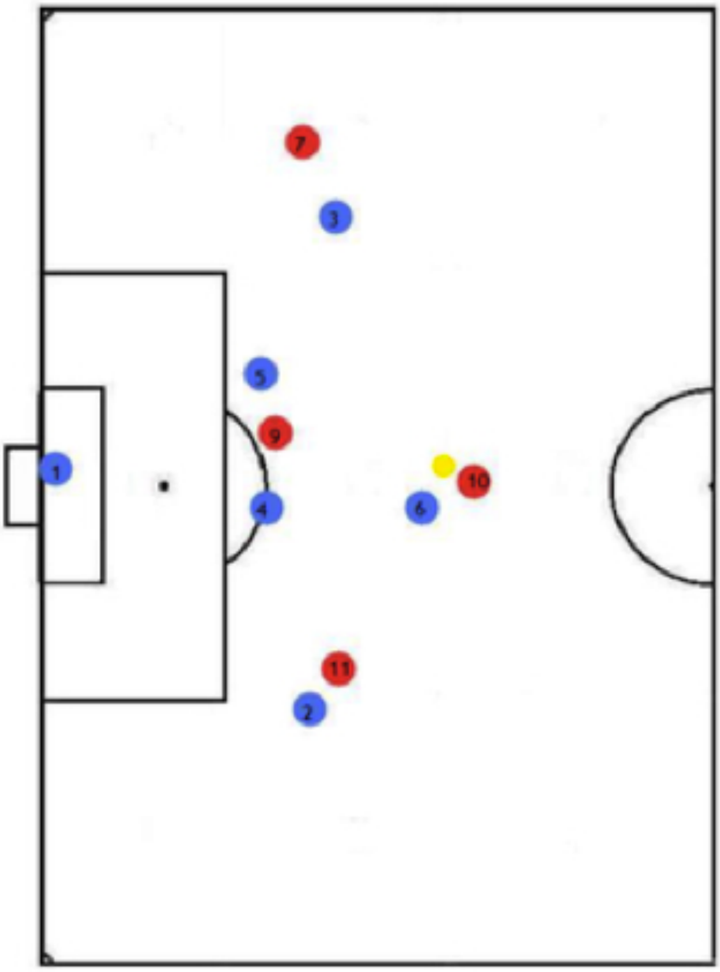


Figure A.2: Defensive Game Situation for a Specific Game Period.

(e) Player 6 is marking player 10

i. Totally Disagree;

ii. Partially Disagree;

iii. Partially Agree;

iv. Totally Agree.

(f) Player 6 is covering the pass line for player 11

i. Totally Disagree;

ii. Partially Disagree;

iii. Partially Agree;

iv. Totally Agree.

(g) Player 2 is covering the pass line for player 11

i. Totally Disagree;

ii. Partially Disagree;

    iii. Partially Agree;
    iv. Totally Agree.

(h) Player 2 is marking player 11

    i. Totally Disagree;
    ii. Partially Disagree;
    iii. Partially Agree;
    iv. Totally Agree.

(i) Player 2 is pressing player 11

    i. Totally Disagree;
    ii. Partially Disagree;
    iii. Partially Agree;
    iv. Totally Agree.

(j) Player 5 is covering the pass line for player 9

    i. Totally Disagree;
    ii. Partially Disagree;
    iii. Partially Agree;
    iv. Totally Agree.

(k) Player 5 is marking player 9

    i. Totally Disagree;
    ii. Partially Disagree;
    iii. Partially Agree;
    iv. Totally Agree.

(l) Player 5 is pressing player 9

    i. Totally Disagree;
    ii. Partially Disagree;
    iii. Partially Agree;
    iv. Totally Agree.

(m) Player 4 is marking player 9

    i. Totally Disagree;
    ii. Partially Disagree;
    iii. Partially Agree;
    iv. Totally Agree.

(n) Player 4 is pressing player 9

    i. Totally Disagree;
    ii. Partially Disagree;

iii. Partially Agree;

iv. Totally Agree.

6. Figure A.3 represents a goal kick situation. The main reason for picking this situation is because before the goal keeper kicks the ball the players are usually placed in their team formation.

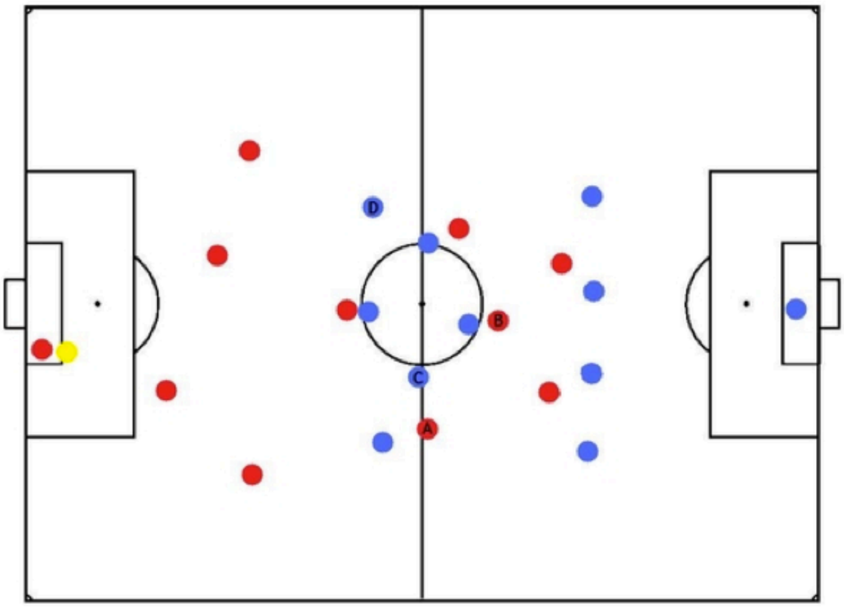


Figure A.3: Team and Players Position - First Game Situation Analysis.

(a) What is the formation that is being used by the red team

i. 4-1-3-2;

ii. 4-1-2-1-2;

iii. 4-4-2;

iv. 4-4-3.

(b) What is the formation that is being used by the blue team

i. 4-1-2-3;

ii. 4-1-4-1;

iii. 4-5-1;

iv. 4-3-3.

(c) How would you rate in this case player A´s position

i. Right Winger;

ii. Right Back Midfielder;

iii. Midfielder Centre;

iv. Right Midfielder;

   v. Right Wing Forward;
   vi. Right Attacking Midfielder.

(d) How would you rate in this case player B´s position
   i. Striker;
   ii. Centre Forward;
   iii. Attacking Midfielder Centre;
   iv. Second Forward;
   v. Second Striker.

(e) How would you rate in this case player C´s position
   i. Defensive Midfielder Centre;
   ii. Right Back Midfielder;
   iii. Left Back Midfielder;
   iv. Attacking Midfielder Centre;
   v. Midfielder Centre;
   vi. Left Midfielder.

(f) How would you rate in this case player D´s position
   i. Right Winger;
   ii. Right Attacking Midfielder;
   iii. Attacking Midfielder Centre;
   iv. Second Forward;
   v. Centre Forward.

7. Figure A.4 illustrates another goal kick situation involving two other teams (orange and green).

   (a) What is the formation that is being used by the orange team
      i. 4-4-2;
      ii. 4-2-4;
      iii. 4-2-2-2;
      iv. Other.

   (b) What is the formation that is being used by the green team
      i. 5-3-2;
      ii. 5-2-3;
      iii. 3-5-2;
      iv. 3-4-3;
      v. 5-2-1-2;
      vi. 3-4-1-2.

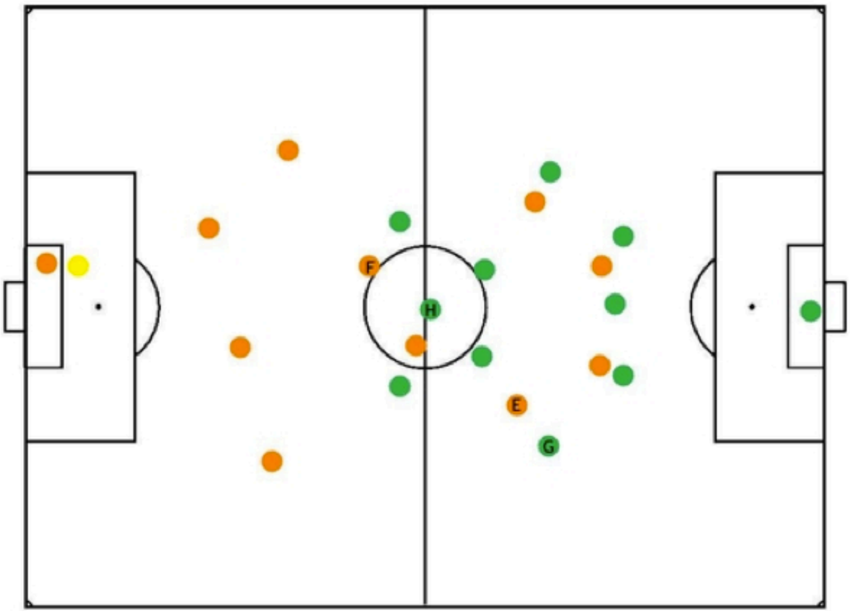


Figure A.4: Team and Players Position - Second Game Situation Analysis.

(c) How would you rate in this case player E´s position
    i. Right Attacking Midfielder;
    ii. Right Winger;
    iii. Right Attacking Midfielder;
    iv. Midfielder Centre;
    v. Attacking Midfielder Centre;
(d) How would you rate in this case player F´s position
    i. Defensive Midfielder Centre;
    ii. Attacking Midfielder Centre;
    iii. Midfielder Centre;
    iv. Left Back Midfielder.
(e) How would you rate in this case player G´s position
    i. Left Back;
    ii. Left Wing Back;
    iii. Left Back Midfielder;
    iv. Left Midfielder.
(f) How would you rate in this case player H´s position
    i. Centre Forward;
    ii. Striker;
    iii. Midfielder Centre;
    iv. Attacking Midfielder Centre;

v. Second Forward;
vi. Second Striker;

8. Now consider a player who shoots frequently. From what values do you consider that the player has a tendency to execute shots?

   (a) when more than 25% of his offensive actions are shots;
   (b) when more than 50% of his offensive actions are shots;
   (c) when more than 75% of his offensive actions are shots;

9. Now consider a player who crosses frequently. From what values do you consider that the player has a tendency to execute crossings?

   (a) when more than 25% of his offensive actions are crossings;
   (b) when more than 50% of his offensive actions are crossings;
   (c) when more than 75% of his offensive actions are crossings;

10. Now consider a player who dribbles frequently. From what values do you consider that the player has a tendency to execute dribbles?

   (a) when more than 25% of his offensive actions are dribbles;
   (b) when more than 50% of his offensive actions are dribbles;
   (c) when more than 75% of his offensive actions are dribbles;

11. The next assertions are related to the attributes or characteristics of the infielders. Please rate each of the following:

   (a) A player with good dribbling, usually has good ball control.
      i. Totally Disagree;
      ii. Partially Disagree;
      iii. Partially Agree;
      iv. Totally Agree.
   (b) A player who shoots well inside the area, and also shoots well far away.
      i. Totally Disagree;
      ii. Partially Disagree;
      iii. Partially Agree;
      iv. Totally Agree.
   (c) A player who is good in executing direct free kicks, he is usually good to shoot from far.
      i. Totally Disagree;

   ii. Partially Disagree;
   iii. Partially Agree;
   iv. Totally Agree.

(d) A good player making tackles, usually is a good player in the marking.

   i. Totally Disagree;
   ii. Partially Disagree;
   iii. Partially Agree;
   iv. Totally Agree.

(e) A player who makes good short passes, is also a good player to make long passes.

   i. Totally Disagree;
   ii. Partially Disagree;
   iii. Partially Agree;
   iv. Totally Agree.

(f) A good player to mark corners usually is a good player in crosses execution.

   i. Totally Disagree;
   ii. Partially Disagree;
   iii. Partially Agree;
   iv. Totally Agree.

(g) A good player to make long passes is also good to make crosses.

   i. Totally Disagree;
   ii. Partially Disagree;
   iii. Partially Agree;
   iv. Totally Agree.

12. Figure A.5 illustrates another goal kick situation involving two other teams (red and blue).

(a) In your opinion which of the following game situations is best to discern whether the defensive line of your team plays retreated or advanced

   i. When the goal keeper executes a goal kick;
   ii. When the attaching team is in its offensive process and the ball is in the last half of the opponent midfield;
   iii. When your team is attacking and the ball is in the last half of the opponent midfield;
   iv. When the opponent team is attacking and the ball is in the opponent penalty area;
   v. When your team is attacking and the ball is in the opponent penalty area;
   vi. Another answer.

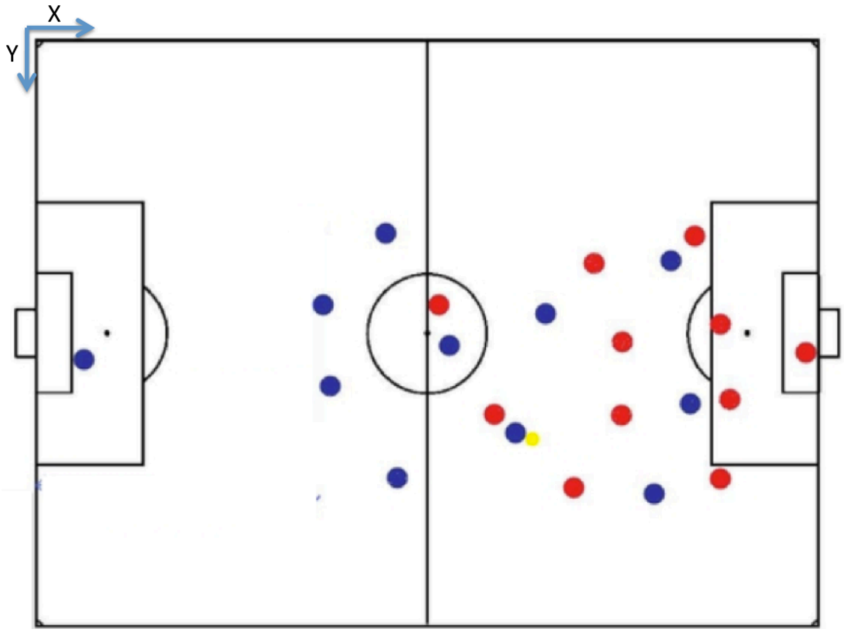


Figure A.5: A Static Game Situation.

(b) According to the game situation that you choose in the previous answer, from that distance value is that X (Figure A.5) may be regarded as the defensive line plays retreated

i. From 20 meters;
ii. From 30 meters;
iii. From 40 meters;
iv. From 50 meters;
v. From 60 meters;
vi. From 70 meters.

(c) From which distance value is that X (Figure A.5) may be regarded, as the defensive line plays normal

i. Between 10 and 20 meters;
ii. Between 10 and 30 meters;
iii. Between 20 and 30 meters;
iv. Between 20 and 40 meters;
v. Between 20 and 50 meters;
vi. Between 30 and 40 meters;
vii. Between 30 and 50 meters;
viii. Between 30 and 60 meters;
ix. Between 40 and 50 meters;
x. Between 40 and 60 meters;
xi. Between 40 and 70 meters;

   xii. Between 50 and 60 meters;
   xiii. Between 50 and 70 meters.

(d) Consider now the concept of Team-width. Which of the following situations is the most ideal for analyzing the width of the team?

   i. When the goal keeper of your team is executing a goal kick;
   ii. When your team is attacking and the ball is in your midfield;
   iii. When your team is attacking and the ball is in the first half of the opponent midfield (situation represented in Figure A.5);
   iv. When your team is attacking and the ball is in the last half of the opponent midfield;
   v. When your team is attacking and the ball is inside the opponent penalty area;
   vi. Another Answer.

(e) According to the situation indicated in the previous question, from that distance value Y (having Figure A.5 as a base), we can say that a team "plays the entire width of the field"

   i. From 30 meters;
   ii. From 35 meters;
   iii. From 40 meters;
   iv. From 45 meters;
   v. From 50 meters;
   vi. From 55 meters;
   vii. From 60 meters;
   viii. From 65 meters.

(f) What should be the distance value Y (having Figure A.5 as a base) for the opposite situation?

   i. Values below to 55 meters;
   ii. Values below to 50 meters;
   iii. Values below to 45 meters;
   iv. Values below to 40 meters;
   v. Values below to 35 meters;
   vi. Values below to 30 meters;
   vii. Values below to 25 meters;
   viii. Values below to 20 meters;

13. A soccer team can implement certain types of passes more often than others. During a match if we calculate the total number of passes, from which values could we say that the team preferentially uses one type of pass?

(a) Short Pass

i. more than 25% of the team passes are of that type;

ii. more than 50% of the team passes are of that type;

iii. more than 75% of the team passes are of that type;

(b) Long Pass

i. more than 25% of the team passes are of that type;

ii. more than 50% of the team passes are of that type;

iii. more than 75% of the team passes are of that type;

14. Consider now the concept of game rhythm. Which of the following statements is the one that better describes this concept?

(a) The game rhythm is associated with the passes´ speed;

(b) The game rhythm is associated with the time that the player takes to execute offensive actions;

(c) The game rhythm is associated with the player´s movements;

(d) The game rhythm is associated with the time that players take to make a pass;

(e) Another Answer.

15. In your opinion what is the passes performed value to suggest that a team preferably attacks to the right flank?

(a) when more than 25% of these passes are executed on the right flank;

(b) when more than 50% of these passes are executed on the right flank;

(c) when more than 75% of these passes are executed on the right flank.

16. In your opinion what is the passes performed value to suggest that a team preferably attacks to the centre of the field?

(a) when more than 25% of these passes are executed on the centre of the field;

(b) when more than 50% of these passes are executed on the centre of the field;

(c) when more than 75% of these passes are executed on the centre of the field.

17. Imagine that we want to measure the risk degree that a team takes over a game. For each range value, select the number of players of a team that should be behind the ball line.

(a) Very Safe

i. 5;

ii. 6;

iii. 7;

iv. 8;
v. 9;
vi. 10;
vii. 11.

(b) Safe
  i. 5;
  ii. 6;
  iii. 7;
  iv. 8;
  v. 9;
  vi. 10;
  vii. 11.

(c) Normal
  i. 5;
  ii. 6;
  iii. 7;
  iv. 8;
  v. 9;
  vi. 10;
  vii. 11.

(d) Risky
  i. 5;
  ii. 6;
  iii. 7;
  iv. 8;
  v. 9;
  vi. 10;
  vii. 11.

(e) Very Risky
  i. 5;
  ii. 6;
  iii. 7;
  iv. 8;
  v. 9;
  vi. 10;
  vii. 11.

References

[ABC+03] J. Assfalg, M. Bertini, C. Colombo, A. Bimbo, and W. Nunziati. Semantic annotation of soccer videos: Automatic highlights identification. *Computer Vision, Graphics and Image understanding Journal*, pages 287–310, 2003.

[ABTi+00] Elisabeth André, Kim Binsted, Kumiko Tanaka-ishii, Sean Luke, Gerd Herzog, and Thomas Rist. Three robocup simulation league commentator systems. *Artificial Inteligence Magazine*, 21(1):57–66, 2000.

[ACCa+10a] P. Abreu, I. Costa, D. Castelão, J. Moreira, L. P. Reis, and J. Garganta. Human vs. virtual robotic soccer: A technical analysis about two realities, 2010. European Journal of Sport Science – Article in Press.

[ACCa+10b] P. Abreu, I. Costa, D. Castelão, L. P. Reis, and J. Garganta. Human vs. robotic soccer: How far are they?a statistical comparison study. In *RoboCup 2010: Robot Soccer World Cup XIV, article in press*, 2010.

[ADM02] G. Abt, G. Dickson, and W. Mummery. Goal scoring patterns over the course of a match: An analysis of the australian national soccer league. In W. Spinks, T. Reilly, and A. Murphy, editors, *Science and Football IV*, pages 107–111. Routledge London, 2002.

[AEGO96] G. Abowd, J. Engelsma, L. Guadagno, and Okon. O. Architectural analysis of object request brokers. *Object Magazine*, pages 44–51, March 1996.

[AFA00] L. Almeida, P. Fonseca, and J. Azevedo. The micro-rato contest: a popular approach to improve self-study in electronics and computer science. In *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, pages 701–705, 2000.

[AFRG10] P. Abreu, M. Faria, L. P. Reis, and J. Garganta. Representação de conhecimento no domínio do futebol: Desenvolvimento de uma ontologia. In *Conferência Ibérica de Sistemas e Tecnologias de Informação*, pages 414–419, 2010.

[AHR88] Elisabeth Andre, Gerd Herzog, and Thomas Rist. On the simultaneous interpretation of real world image sequences and their natural language description: The system soccer. In *Proceedings of the European Conference on Artificial Intelligence*, pages 449–454, 1988.

[AMS+10] P. Abreu, J. Moura, D. C. Silva, L. P. Reis, and J. Garganta. Soccer scientia- an automated tool for professional soccer coaches. In *Proceedings of the IEEE International Conference on Cybernetics and Intelligent Systems Robotics, Automation and Mechatronics*, pages 126–131, 2010.

[AMTY98] S. Araki, T. Matsuoka, H. Takemura, and N. Yokoya. Real-time tracking of multiple moving objects in moving camera image sequences using robust statistics. In *Proceedings of the International Conference on Pattern Recognition*, pages 1583–1591, 1998.

[AN08] H. Akiyama and I. Noda. Multiagent positioning mechanism in a dynamic environment. In U. Visser, F. Ribeiro, T. Ohashi, and F. Dellaert, editors, *ROBOCUP 2007: ROBOT SOCCER WORLD CUP XI*, volume 5001, pages 377–384. Springer, 2008.

[AVCL09] R. Aler, J. Valls, D. Camacho, and A. Lopez. Programming robosoccer agents by modeling human behavior. *Expert Systems with Applications Journal*, 36(2):1850–1859, March 2009.

[AVM+09] P. Abreu, V. Vinhas, P. Mendes, L.P. Reis, and J. Garganta. Tool based analytical support for practical professional soccer training. In *International Conference on FUture BUsiness TEChnology*, pages 68–76, 2009.

[AVM+10] P. Abreu, V. Vinhas, P. Mendes, L. P. Reis, and J. Garganta. *Real-Time Wireless Location and Tracking System with Motion Pattern Detection*, volume 1 of *Robot Localization and Map Building edited by Hanafiah Yussof*, pages 467–492. Intech, first edition, March 2010.

[AYA+08] M. F. Acar, B. Yapicioglu, N. Arikan, S. Yalcin, N. Ates, and M. Ergun. Analysis of goals scored in the 2006 world cup. In *Proceedings of the World Congress on Science and Football*, pages 235–241, 2008.

[AYAS05] V. Armatas, A. Yiannakos, D. Ampatis, and P. Sileloglou. Analysis of the successful counter-attacks in high-standard soccer game. *Sport and Physical Education Journal*, 3(2):187–195, 2005.

[Bal02] P. Balsom. Interview in solna, sweden 13th of july, 2002.

[BAOR03] H. Ben Amor, Murray, O. Obst, and C. Ringelstein. Robolog koblenz 2003 – team description. Academic Press, 2003.

[Bat98] B. Bate. Football change:tactics and strategy. In T. Reilly et al., editor, *Science and Football*, pages 293–301. E % FN Spon London, 1998.

[BB78] J. Brown and R. Burton. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science Journal*, 2(2):155–192, April 1978.

[BER02] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. *IEEE Workshop on Motion and Video Computing*, pages 169–174, 2002.

[BF81] A. Barr and A. Feigenbaum. *The Handbook of Artificial Intelligence Vol. 1*. Kaufmann, William Inc, First edition, 1981.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[BG88] A. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Pub, 1988.

[BG04] D. Brickley and R. Guha. Rdf vocabulary description language 1.0: Rdf schema. Technical report, W3C, February 2004.

[BGV92] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[BJ03] M. Baillie and J. Jose. Audio-based event detection for sports video. In *Proceedings of the international conference on Image and video retrieval*, pages 300–310. Springer, 2003.

[BJM02] A. Borrie, G. Jonsson, and Magnusson. M. Temporal pattern analysis and its applicability in sport: an explanation and exemplar data. *Sport Science Journal*, pages 845–852, October 2002.

[BL00] T. Berners-Lee. Xml semantic web talk, 2000. Available at http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html#, consulted in 10/06/2010.

[BLB98] Kim Blinsted, Sean Luke, and A. V. Williams Building. Character design for soccer commentary. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, volume 1604, pages 23–35. Springer, 1998.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American Journal*, pages 29–37, May 2001.

[BLN10] J. Baltes, M. Lagoudakis, and T. Naruse. *RoboCup 2009: Robot Soccer World Cup XIII*, volume 5949. Springer Verlag, 2010.

[BLS+05] E. Bekris, Z. Louvaris, S. Souglis, K. Hountis, and E. Siokou. Statistical analysis of the ability of shot in high standard matches. In *First International Scientific Congress in Soccer*, 2005.

[BMM+07] R. Barros, M. Misuta, R. Menezes, P. Figueroa, F. A, S. Cunha, R. Anido, and N. Leite. Research article analysis of the distances covered by first division brazilian soccer players obtained with an automatic tracking method. *Sport Science Journal*, pages 233–242, 2007.

[Bor97] W. Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* Enschede, The Netherlands, 1997.

[BPOR08] R. Braga, M. Petry, E. Oliveira, and L. P. Reis. Multi-level control of an intelligent wheelchair in a hospital environment using a cyber-mouse simulation system. In *Proceedings of the International Conference on Informatics in Control, Automation and Robotics*, pages 179–182, 2008.

[BPSM+04] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (XML) 1.0 (third edition), W3C recommendation. Technical report, 2004.

[Bre96] L. Breiman. Bagging predictors. In *Machine Learning*, pages 123–140, 1996.

[Bre98] L. Breiman. Arcing classifiers. *The Annals of Statistics Journal*, 26(3):801–849, 1998.

[Bre01] L. Breiman. Random forests. *Machine Learning Journal*, 45:5–32, 2001.

[Bro91] R. Brooks. Intelligence without reason. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 569–595. Morgan Kaufmann, 1991.

[BS84] G. Bruce and E. Shortliffe. *Rule-Based Expert Systems- The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, 1984.

[BS97] P. Bretier and M. Sadek. A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 189–203, London, UK, 1997.

[BSB+09] P. Bradley, W. Sheldon, W. Blake, P. Olsen, P. Boanas, and P. Krustrup. High-intensity running in english fa premier league soccer matches. *Sport Science Journal*, 27:159–168, 2009.

[BTS03] C. Bauckhage, C. Thurau, and G. Sagerer. Learning humanlike opponent behavior for interactive computer games. In *Pattern Recognition*, pages 148–155. Springer, 2003.

[Bud05] B. Budianto. An overview and survey on multi agent system, 2005.

[CA96] Q. Cai and J. Aggarwal. Tracking human motion using multiple cameras. In *Proceedings of the International Conference on Pattern Recognition*, pages 68–72, 1996.

[Car93] C. Cardie. Using decision trees to improve case-based learning. In *Proceedings of the International Conference on Machine Learning*, pages 25–32. Morgan Kaufmann, 1993.

[Cas96] J. Castelo. *Futebol e Organização do Jogo*. Edição de Autor, 1996.

[Cas03] J. Castelo. *Guia Prático de Exercícios de Treino*. Lisboa: Visão e Contextos, 2003.

[Cas09] A. Castro. *Disruption Management in Airline Operations Control: Designing a Multi-Agent System with a GAIA based Methodology*. VDM Verlag, 2009.

[CCR+06] J. Certo, N. Cordeiro, F. Reinaldo, L.P. Reis, and N. Lau. Fcpx: A tool for evaluating teams' performance in robocup rescue simulation league. *Advances in Artificial Intelligence Journal*, 26:137–148, 2006.

[CCYL02] Y. Cai, J. Chen, J. Yao, and S. Li. Global planning from local eyeshot: An implementation of observation-based plan coordination in robocup simulation games. In *RoboCup 2001: Robot Soccer World Cup V*, pages 12–21, London, UK, 2002.

[CFF+98] V. Chaudhri, A. Farquhar, R. Fikes, P. Karp, and J. Rice. Open knowledge base connectivity 2.0.3,protocol proposed by artificial intelligence center of sri international and the knowledge systems laboratory of stanford university, 1998.

[CFH+03] M. Chen, E. Foroughi, S. Heintz, S. Kapetanakis, K. Kostiadis, J. Kummeneje, I. Noda, O. Obst, P. Riley, T. Steffens, Y. Wang, and X. Yin. Soccer server version 7.07 or latest, 2003. Available at http://sourceforge.net/projects/server/, consulted in 10/07/2010.

[CFLGP07] O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez. Chapter 3: Ontological engineering:what are ontologies and how can we build them. In J. Cardoso, editor, *Semantic Web Services: Theory, Tools and Applications*, pages 44–70. IGI Global, March 2007.

[CFlGpV02] O. Corcho, M. Fernandez-lopez, A. Gomez-perez, and O. Vicente. Webode: An integrated workbench for ontology representation, reasoning, and exchange. In *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management*, pages 138–153. Springer, 2002.

[Cha07] C. and Chao. Determining technology trends and forecasts of rfid by a historical review and bibliometric analysis from 1991 to 2005. *Technovation27 Journal*, 27(5):268–279, 2007.

[CLR07a] J. Certo, N. Lau, and L. Reis. A generic multi-robot coordination strategic layer. In *Proceedings of the international conference on Robot communication and coordination*, volume 318, pages 273–282, 2007.

[CLR07b] J. Certo, N. Lau, and L. Reis. *A Generic Strategic Layer for Collaborative Networks*, volume 243 of *Establishing the Foundation of Collaborative Networks*, pages 273–282. Springer Boston, September 2007.

[CMBT02] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, pages 168–175, 2002.

[CN95] S. Clarke and J. Norman. Home ground advantage of individual clubs in english soccer. *Statistician Journal*, 44:509–521, 1995.

[CO08] H. Cardoso and E. Oliveira. Electronic institutions for b2b: dynamic normative environments. *Artificial Intelligence and Law*, 16(1):107–128, April 2008.

[Con02] Onto Web Consortium. A survey on ontology tools, ist project ist-2000-29243, 2002.

[Cor05] O. Corcho. A layered declarative approach to ontology translation with knowledge preservation, January 2005. Frontiers in Artificial Intelligence and its Applications-PhD Thesis, Available at http://oa.upm.es/197/1/10200405.pdf, consulted in 10/7/2010.

[Cru02] J. Cruyff. *Me gusta el fútbol*. RBA Libros S.A., 2002.

[CRW09] C. Carling, T. Reilly, and A. Williams. *Performance Assessment for Field Sports*. Routledge, First edition, 2009.

[CSSdO09] P. Carvalho, L. Sarmento, M. Silva, and E. de Oliveira. Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In *Proceeding of the international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM, 2009.

[CVR98] M. Crampes, J. Veuillez, and S. Ranwez. Adaptive narrative abstraction. In *Proceedings of the ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, pages 97–105, 1998.

[CWR07] C. Carling, A. Williams, and T. Reilly. *Handbook of Soccer Match Analysis - A Systematic Approach to Improving Performance*. Routledge, Second edition, 2007.

[Dem06] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Machine Learning Research Journal*, 7:1–30, January 2006.

[DF98] R. Dias and J. Freitas. *A evolução das Tácticas*. Rui Santos edições, 1998.

[DHN+00] C. Druecker, S. Huebner, H. Neumann, E. Schmidt, U. Visser, and H. Weland. Virtualweder: Using the online-coach to change team formations. Technical report, University of Bremen, 2000.

[Die97] T. Dietterich. Machine-learning research – four current directions. *Artificial Intelligence Magazine*, 18:97–136, 1997.

[Die98] T. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation Journal*, 10(7):1895–1923, 1998.

[DM90] Y. Demazeau and J. Muller. Decentralized artificial intelligence. 3:3–13, 1990.

[DMG99] J. Domingue, E. Motta, and O. Garcia. Knowledge modelling in webonto and ocml – a user guide – version 2.4, 1999.

[Dor00] K. Dorer. The magmafreiburg soccer team. In *RoboCup-99: Robot Soccer World Cup III*, volume 1856, pages 600–603. Springer, 2000.

[DR98] M. Dixon and M. Robinson. A birth process model for association football matches. *Statistician Journal*, 47:523–538, 1998.

[DST89] H. Dobler, G. Schnabel, and G. Thieb. *Grundbegriffe der Sportspiele*. Sportverlag, 1989.

[DSW+00] A. Duineveld, R. Stoter, M. Weiden, B. Kenepa, and V. Benjamins. Wondertools?: a comparative study of ontological engineering tools. *Human-Computer Studies Journal*, 52(6):1111–1133, June 2000.

[Dua97] O. Duarte. *Futebol, histórias e regras*. Makron Books, 1997.

[Duf93] W. Dufour. Computer- assisted scouting in soccer. In T. Reily., J. Clarys, and A. Stibbe, editors, *Science and Football II*, pages 160–166. E&FN Spon London, 1993.

[EC07] L. Escorcio and J. Cardoso. Chapter 4: Editing tools for ontology construction. In J. Cardoso, editor, *Semantic Web Services: Theory, Tools and Applications*, pages 71–95. IGI Global, March 2007.

[ET94] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 1 edition, May 1994.

[ETM03] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing Journal*, 12(7):796–807, July 2003.

[Fed01] RoboCup Federation. The goals of robocup, 2001. Available at http://www.robocup.org/overview/22, consulted in 17/07/2010.

[Fen01] D. Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, First edition, 2001.

[FFR96] A. Farquhar, R. Fikes, and J. Rice. The ontolingua server: a tool for collaborative ontology construction. In *Human-Computer Studies Journal*, volume 46, pages 707–727, 1996.

[FG97] S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–35. Springer-Verlag, 1997.

[FGM83] I. Franks, D. Goodman, and G. Miller. Human factors in sport systems. In *Proceedings of the Human Factors Society Annual Meeting*, pages 383–386, 1983.

[FGPJ97] M. Fernandez, A. Gomez-Perez, and N. Juristo. Methontology: from ontological art towards ontological engineering. In *Proceedings of the Association for the Advancement of Artificial Intelligence Spring Symposium*, pages 33–40, March 1997.

[FHH+00] D. Fensel, I. Horrocks, F. Harmelen, S. Decker, M. Erdmann, and M. Klein. Oil in a nutshell. In *Proceedings of the European Workshop of Knowledge Acquisition, Modeling, and Management*, pages 1–16. Springer, 2000.

[FM91] I. M. Frank and G. Miller. Training coaches to observe and remember. *Sports Sciences Journal*, 9(3):285–297, 1991.

[FPN90] I. Franks, D. Partridge, and P. Nagelkerke. World cup 90: A computer assisted technical analysis of team performance. Technical report, Canadian Soccer Association, 1990.

[Fra96] I. Franks. Use of feedback by coaches and players. In *Proceedings of the World Congress on Science and Football*, pages 267–278. Taylor and Francis Group, 1996.

[FRBR09] M. Figueiredo, R. Rossetti, R. Braga, and L. P. Reis. An approach to simulate autonomous vehicles in urban traffic scenarios. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 322–327, 2009.

[Fri91] J. Friedman. Multivariate adaptive regression splines. *Annals of Statistics Journal*, 19:1–141, 1991.

[FWW+94] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, S. Shapiro, J. McGuire, R. Pelavin, and C. Beck. Specification of the kqml agent-communication language, 1994.

[Gar97a] J. Garganta. Modelação da dimensão táctica do jogo de futebol. In J. Oliveira and F. Tavares, editors, *Estratégia e Táctica nos Jogos Desportivos*, pages 63–82. Porto: Centro de Estudos dos Jogos Desportivos, 1997.

[Gar97b] J. Garganta. Modelação táctica do jogo de futebol. estudo da organização da fase ofensiva em equipas de alto rendimento, *PhD thesis*, sport faculty of porto university, 1997.

[Gar01a] E. Garcia. Cognição organizacional e ciência da complexidade. *Empresas, caos e complexidade: Gerindo à beira de um ataque de nervos*, pages 35–56, 2001.

[Gar01b] J. Garganta. Tactical modeling in soccer: a critical view. In M. Hughes and F. Tavares, editors, *World Congress of Notational Analysis of Sport*, pages 33–40. Faculdade de Desporto da Universidade do Porto, 2001.

[GBD97] J. Grehaigne, D. Bouthier, and B. David. Dynamic-system analysis of opponent relationships in collective actions in soccer. *Sports Sciences Journal*, pages 137–149, 1997.

[GDAS10] W. Gregson, B. Drust, G. Atkinson, and V. Salvo. Match-to-match variability of high-speed activities in premier league soccer. *Sports Medicine Journal*, 31(4):237–242, 2010.

[GF92] M. Genesereth and R. Fikes. Knowledge interchange format, version 3.0 reference manual, technical report. Technical report, Computer Science Department, Stanford University, 1992.

[GF95] M. Grüninger and M. Fox. Methodology for the design and evaluation of ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.

[GG95] N. Guarino and P. Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing Journal*, pages 25–32, 1995.

[GHN+97] S. Green, L. Hurst, B. Nangle, P. Cunningham, R. Evans, and F. Somers. Software agents: A review, 1997.

[GL02] M. Grüninger and J. Lee. Ontology applications and design. *Communications of ACM*, 45(2):39–41, 2002.

[GMB97] J. Garganta, J. Maia, and F. Basto. Analysis of goal-scoring patterns in european top level soccer teams. In T. Reilly, J. Bangsbo, and M. Hughes, editors, *Science and Footaball II*, pages 246–250. E&FN Spon London, 1997.

[Gou98] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers journal*, pages 87–93, 1998.

[Gre89] J. Grehaigne. Football de mouvement-vers une aproche systémique du jeu, *PhD Thesis*, univ. bourgogne, 1989.

[Gre92] J. Grehaigne. *L´Organization du jeu en football*. Paris:Éditions Actio, 1992.

[Gre99] J. Grehaigne. Systemic approach and soccer. In M. Hughes, editor, *Notation of Sport III*, pages 1–8. Cardiff: Centre of Performance Analysis, UWIC, 1999.

[Gri63] S. Griffith. *Translaction of the book The Art of War by Sun Tzu*. Oxford University Press, 1963.

[Gru93] T. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *Human-Computer Studies Journal*, pages 907–928. Kluwer Academic Publishers, 1993.

[GSC+95] Y. Gong, L. Sin, C. Chuan, H. Zhang, and M. Sakauchi. Automatic parsing of tv soccer programs. In *IEEE International Conference on Multimedia Computing and Systems*, pages 167–174, 1995.

[Gua98] N. Guarino. Formal ontology and information systems. In *Formal Ontology and Information Systems*, pages 3–15. IOS Press, 1998.

[Hal00] M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of International Conference on Machine Learning*, pages 359–366. Morgan Kaufmann, 2000.

[HB02] M. Hughes and R. Bartlett. The use of performance indicators in performance analysis. *Sports Science Journal*, 20(10):739–54, 2002.

[HF01] M. Hughes and I. Franks. *Notational Analysis of Sport: Systems for Better Coaching and Performance in Sport*. Routledge, Second edition, 2001.

[HF05] M. Hughes and I. Franks. Analysis of passing sequences, shots and goals in soccer. *Sports Science Journal*, 23(5):509–514, 2005.

[HLSE08] P. Haase, H. Lewen, R. Studer, and M. Erdmann. The neon ontology engineering toolkit, 2008.

[Hop06] A. Hopcraft. *The Football Man:People and Passions in Soccer*. Aurum Press Ltd, 2006.

[HPSB+04] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. Swrl: A semantic web rule language combining owl and ruleml. Technical report, World Wide Web Consortium, May 2004.

[HPSH02] I. Horrocks, P. Patel-Schneider, and F. Harmelen. Reviewing the design of daml+oil: An ontology language for the semantic web. In *Proceedings of the Artificial Intelligence Conference*, pages 792–797. AAAI Press, 2002.

[HRN88] M. Hughes, K. Robertson, and A. Nicholson. An analysis of the 1984 world cup of association football. In T. Reilly, A. Lees, K. Davids, and W. Murphy, editors, *Science and Footaball*, pages 363–367. E&FN Spon London, 1988.

[HS97] M Huhns and M. (editors) Singh. Readings in agents, 1997. Morgan Kaufmann Publishers.

[HS99] M. Huhns and L. Stephens. Multiagent systems and societies of agents. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 79–120. The MIT Press, 1999.

[IMWZ09] L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou. *RoboCup 2008: Robot Soccer World Cup XII: 5399 (Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence)*. Springer Verlag, 2009.

[IS04] S. Iwase and H. Saito. Parallel tracking of all soccer players by integrating detected positions in multiple view images. In *IEEE International Conference on Pattern Recognition*, pages 751–754. IEEE Computer Society, 2004.

[Jai10] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters Journal*, 31(8):651–666, June 2010.

[JD88] A. Jain and R. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[Jen00] N. Jennings. On agent-base software engineering. *Artificial Intelligence Journal*, 117:277–296, 2000.

[JJ07] P. Jappinen and P. J. Preference-aware ubiquitous advertisement screen. In *Proceedings of IADIS-e-Commerce*, pages 99–105, 2007.

[JJM04] N. James, P. Jones, and S. Mellalieu. Possession as a performance indicator in soccer. *Performance Analysis in Sport Journal*, 4(1):98–102, 2004.

[JOH00] B. Jung, M. Oesker, and H. Hecht. Virtual robocup: Real-time 3d visualization of 2d soccer games. In M. Veloso, E. Pagello, and H. Kitano, editors, *ROBOCUP-99: ROBOT SOCCER WORLD CUP III*, volume 1856, pages 115–128. Springer, 2000.

[Jon98] G. Jonsson. Detecting patterns in complex behavioural processes with the observer and theme. In *Proceedings of the International Conference on Methods and Techniques in Behavioural Research*, pages 176–184, 1998.

[JRSS03] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *IEEE International Conference on Computer Vision*, pages 952–957, 2003.

[JXYM93] X. Jinshan, C. Xiakone, K. Yamanaka, and M. Matsumoto. Analysis of the goals in the 14th world cup. In T. Reilly, J. Clarys, and A. Stibbe, editors, *Proceedings of Science and Football II*, pages 203–205, 1993.

[Kab07] V. Kabilan. Ontology for information systems (o4is) design methodology:conceptualizing, designing and representing domain ontologies, *PhD Thesis*, royal institute of technology, Junho 2007.

[KAK+95] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robotic world cup initiative. In *Proceedings in Workshop on Entertainment and AI/Alife*, pages 19–24, 1995.

[KAK+97] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In *Proceedings of the International conference on Autonomous agents*, pages 340–347. ACM, 1997.

[KCT99] P. Karp, V. Chaudhri, and J. Thomere. Xol: An xml-based ontology exchange language. Technical report, SRI International, 1999.

[KII94] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation Journal*, 10:799–822, 1994.

[KJ97] M Knapik and J Johnson. *Developing Intelligent Agents for Distributed Systems: Exploring Architectures, Techniques, and Applications*. Osborne/McGraw-Hill, 1997.

[KJRS01] S. Khan, O. Javed, Z. Rasheed, and M. Shah. Human tracking in multiple cameras. In *Proceedings of the International Conference on Computer Vision*, pages 331–336, 2001.

[KLW95] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *ACM Journal*, 42:741–843, 1995.

[KMI+02] O. Koichi, K. Miura, I. Ide, S. Sakai, and H. Tanaka. An object detection method for describing soccer games from video. In *Proceedings of the IEEE International Conference on Multimedia and Expo.,Vol.1*, pages 45–48, 2002.

[Kon00] R. Koning. Balance in competition in dutch soccer. *The Statistician Journal*, 49:419–431, 2000.

[KPS+05] A. Kalyanpur, B. Parsia, E. Sirin, B. Grau, and J. Hendler. Swoop: A web ontology editing browser. *Web Semantics Journal*, 4:144–153, 2005.

[KR92] K. Kira and L. Rendell. A practical approach to feature selection. In *Proceedings of the international workshop on Machine learning*, pages 249–256, San Francisco, CA, USA, 1992.

[KTN+99] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research. In *Proceedings of IEEE Conference on Man, Systems, and Cybernetics*, pages 739–743, 1999.

[KTS+97] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada. The robocup synthetic agent challenge 97. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 24–29. Morgan Kaufmann, 1997.

[KV00] J. Kim and P. Vadakkepat. Multi-agent systems: A survey from the robot-soccer perspective. *Intelligent Automation and Soft Computing Journal*, 6(1):3–17, 2000.

[KZ99] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. volume 2, pages 117–123, 1999.

[Lan07] M. Lang. Knoodl.com semantic wiki-creating and using owl vocabularies in a wiki, revelytix sicop presentation drm 3.0 with wordnet senses in a semantic wiki, February 2007.

[LASB05] A. Ledezma, R. Aler, A. Sanchis, and D. Borrajo. Predicting opponent actions by observation. In D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, editors, *ROBOCUP 2004: ROBOT SOCCER WORLD CUP VIII*, volume 3276, pages 286–296. Springer, 2005.

[Les99] V. Lesser. Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering Journal*, 11(1):133–142, 1999.

[LF97] Y. Labrou and T. Finin. Semantics and conversations for an agent communication language. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 235–242. Morgan Kaufmann, 1997.

[LG90] D. Lenat and R. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, Second edition, 1990.

[LH00] S. Luke and J. Heflin. Shoe 1.01 proposed specification. Technical report, Parallel Understanding Systems Group, Department of Computer Science, University of Maryland, 2000.

[LH01] M. Lames and G. Hansen. Designing observational systems to support top-level teams in game sports. *Performance Analysis Journal*, 1(1):85–91, 2001.

[LHSC03] Q. Li, Z. Huang, P. Sun, and X. Chen. Wrighteagle 2003 presentation description: 3d monitor with automatic commentary and graphical debugging system, 2003. Team description for RoboCup 2003.

[LLCF09] N. Lau, L. Lopes, G. Corrente, and N. Filipe. Multi-robot team coordination through roles, positionings and coordinated procedures. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 5841–5847, 2009.

[LLFC09] N. Lau, L. Lopes, N. Filipe, and G. Corrente. Roles, positionings and set plays to coordinate a robocup msl team. In *Proceedings of the Portuguese Conference on Artificial Intelligence*, pages 323–337, 2009.

[LLR00] L. Lopes, N. Lau, and L. P. Reis. Intelligent control and decision-making demonstrated on a simple compass-guided robot. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 2419–2424, 2000.

[LM01] O. Lassila and D. Mcguinness. The role of frame-based representation on the semantic web. Technical report, Knowledge System Laboratory, Report KSL-01-02, 2001.

[Lob07] L. Lobo. *Planeta do Futebol: Em busca da alma, dos magos e das táticas que fizeram história (As origens e o presente do futebol moderno)*. Prime Books, 2007.

[Lou04] S. Louro. Muti agent intelligent camera control, 2004. MsC thesis submitted to the Faculty of Science, University of Porto.

[LPTW96] B. Lemon, D. Pynadath, G. Taylor, and B. Wray. Cognitive architectures, 1996. Available at http://krusty.eecs.umich.edu/cogarch4/toc.html.

[LR07] N. Lau and L. Reis. *FC Portugal - High-Level Coordination Methodologies in Soccer Robotics*, pages 167–192. Robotic Soccer. Itech Education and Publishing, December 2007.

[LS98] O. Lassila and R. Swick. Resource description framework (rdf) model and syntax specification, 1998.

[LSST07] G. Lakemeyer, E. Sklar, D. Sorrenti, and T. Takahashi. *RoboCup 2006: Robot Soccer World Cup X: No. 10 (Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence)*. Springer Verlag, 2007.

[Mac91] R. MacGregor. Inside the loom description classifier. *ACM SIGART Bulletin*, pages 88–92, 1991.

[Mac93] A. MackWorth. On seeing robotics, computer vision: System, theory, and applications. In *Computer Vision: Systems, Theory, and Applications*, pages 1–13. World Scientific Press, Singapore, 1993.

[Mah82] M. Maher. Modeling association football scores. *Statistica Neerlandica Journal*, 36:109–118, 1982.

[Mar79] A. et.al. Marcio. *Os Misters do Universo*. A Bola-Edição Especial, 1979.

[MC91] T. Malone and K. Crowston. Towards an interdisciplinary theory of coordination. Technical report, MIT-Center for Coordination Science, 1991.

[MC94] T. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys Journal*, 26:87–119, 1994.

[McG03] D. McGuinness. *Ontologies Come of Age*. MIT Press, 2003.

[Men79] L. Mendes. *As Tácticas do Futebol*. Ediouro, 1979.

[MFIY06] T. Morita, N. Fukuta, N. Izumi, and T. Yamaguchi. Doddle-owl: A domain ontology construction tool with owl. In *The Semantic Web-A SWC 2006*, pages 537–551. Springer Berlin/Heidelberg, September 2006.

[MGH+09] B. Motik, B. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. Owl 2 web ontology language profiles, w3c recommendation, June 2009.

[Min06] A. Mingkhwan. Wi-fi tracker: an organization wi-fi tracking system. In *Proceedings of the Conference of Electrical and Computer Engineering*, pages 231–234, 2006.

[MLH03] D. Meyer, F. Leisch, and K. Hornik. The support vector machine under test. *Neuro Computing Journal*, 55(1-2):169–186, 2003.

[MLR10] L. Mota, N. Lau, and L. P. Reis. Co-ordination in robocup's 2d simulation league: Setplays as flexible, multi-robot plans. In *Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics*, pages 362–367, 2010.

[MMPP04] C. Michailidis, I. Michailidis, G. Papaiakovou, and I. Papaiakovou. Analysis and evaluation of way and place that goals were achieved during the european champions league of football 2002-2003. *Sports Organization Journal*, 2(1):48–54, 2004.

[Mor86] C. Morelli. *Futebol de campo:histórico e fundamentos*. Campinas edition, 1986.

[MR08] L. Mota and L. P. Reis. A common framework for cooperative robotics: an open, fault tolerant architecture for multi-league robocup teams. In *Proceedings of the International Conference Simulation, Modeling, and Programming for Autonomous Robots*, volume 5325, pages 171–182. Springer Berlin/Heidelberg, November 2008.

[MRB06] L. Mota, L. P. Reis, and H. Burkhard. Communication challenges raised by open co-operative teams in robocup. In *Scientific Meeting of the Portuguese Robotics Open*, April 2006.

[MRL11] L. Mota, L. P. Reis, and N. Lau. Multi-robot coordination using setplays in the middle-size and simulation leagues, 2011. Mechatronics Journal – Article in Press.

[MvH04] D. McGuinness and F. van Harmelen. Owl web ontology language review, February 2004. W3C Recommendation.

[Nee03] C. Needham. Tracking and modelling of team game interactions, October 2003. PhD Thesis on Philosophy submitted in the University of Leeds School of Computing.

[NF03] C. Necib and J. Freytag. Ontology based query processing in database management systems. In *Proceeding on the International conference on Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems*, pages 839–857, 2003.

[NFF+91] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout. Enabling technology for knowledge sharing. *Artificial Intelligence Magazine*, 12(3):36–56, 1991.

[NKFH98] M. Naphade, T. Kristjansson, B. Frey, and T. Huang. Probabilistic multimedia objects (multiobjects): A novel approach to video indexing and retrieval in multimedia systems. In *Proceedings of the International Conference of Image Processing*, pages 536–540, 1998.

[NKS05] B. Nejikovsky, K. Kesler, and J. Stevens. Real time monitoring of vehicle/track interaction. In *Proceedings of the Rail Transportation Conference*, pages 25–31, 2005.

[NM01] N. Noy and D. Mcguinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[NMN05] A. Nicola, M. Missikoff, and R. Navigli. A proposal for a unified process for ontology building: Upon. In *Proceedings of the International Conference on Database and Expert Systems Applications*, pages 655–664, 2005.

[NNMH97] I. Noda, C. Noda, H. Matsubara, and I. Hiraki, K. Frank. Soccer server: a tool for research on multi-agent systems. *Applied Artificial Intelligence Journal*, 12:233–250, 1997.

[NO97] M. Neves and E. Oliveira. A control architecture for an autonomous mobile robot. In *Proceedings of the International Conference on Autonomous agents*, pages 193–200. ACM, 1997.

[Nwa96] H. Nwana. Software agents: An overview. *Knowledge Engineering Review Journal*, 11(3):205–244, 1996.

[Ols98] E. Olsen. An analysis of goal scoring strategies in the world championship in mexico 1986. In T. Reilly, A. Lees, K. Davids, and W. Murphy, editors, *Science and Football*, pages 373–376. London: E. & FN. Spon, 1998.

[OMN+02] J. Ohashi, O. Miyagi, H. Nagahama, T. Ogushi, and K. Ohashi. Application of an analysis system evaluating intermittent activity during a soccer match. In Spinks RaM, editor, *Proceedings of the Science and Football IV Conference*, pages 329–333, 2002.

[OR02] O. Obst and C. Ringelstein. Robolog koblenz 2002 – visualization description. In Gal A. Kaminka, Pedro U. Lima, and Raul Rojas, editors, *RoboCup 2002: Robot Soccer World Cup VI*, 2002.

[Pap02] A. Pappas. Effectiveness of offensive tactic of dead-ball situations in the world cup 2002, *MSc Thesis* – department of physical education and sport science, auth, 2002.

[PARG10] J. Portela, P. Abreu, L. P. Reis, and J. Garganta. An intelligent framework for automatic event detection in robotic soccer games: An auxiliar tool to help coaches improving their teams performance. In *Proceedings of the International Conference on Enterprise Information System*, pages 244–249, 2010.

[PC02] F. Pinheiro and J. Coelho. *A Paixão do Povo: História do Futebol em Portugal*. Simon and Schuster; 1st edition, edições afrontamento edition, 2002.

[PF89a] D. Partridge and I. Franks. A detailed analysis of crossing opportunities from the 1986 world cup (part i). *Soccer Journal*, pages 48–51, 1989.

[PF89b] D. Partridge and I. Franks. A detailed analysis of crossing opportunities from the 1986 world cup (part i). *Soccer Journal*, pages 45–47, 1989.

[Pie83] A. Piecniczk. Preparation of football teams for world competition in 1986, 1983. Communication to 9th UEFA course for National Coaches and Directors of coaching of the Member Associations.

[PKK$^+$03] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, and O. Goranov. Semantic annotation platform. In *Proceedings of the International Semantic Web Conference*, pages 834–849. Springer, 2003.

[PKPS99] G. Paliouras, V. Karkaletsis, C. Papatheodorou, and C. Spyropoulos. Exploiting learning techniques for the acquisition of user stereotypes and communities. In *Proceedings of the International Conference on User Modeling*, volume 407, pages 169–178. Springer-Verlag, 1999.

[Por98] M. Porter. *Competitive Strategy: Techniques for Analyzing Industries and Competitors*. Simon and Schuster Ltd, 1998.

[PPaNC04] C. Penedo, J. Pavão, P. Nunes, and L. Custódio. Robocup advanced 3d monitor. In D. Polani, B. Browning, A. Bonarini, and K. Yoshida, editors, *ROBOCUP 2003: ROBOT SOCCER WORLD CUP VII*, volume 3020, pages 637–644. Springer, 2004.

[Pun07] J. Punuru. Knowledge-based methods for automatic extraction of domain specific ontologies, 2007.

[QLD04] F. Qi, Y. Luo, and H. Du. Visual tracking of players through occlusions in low resolution. In *Proceedings of the International Conference of Science and Technology for Development*, pages 375–380, 2004.

[Rai03] V. Raizer. Validation of two-dimensional microwave signatures. In *Geoscience and Remote Sensing Symposium*, pages 2694–2696, 2003.

[RB68] C. Reep and B. Benjamin. Skill and chance in ball game. *Royal Statistical Society Journal*, pages 581–585, 1968.

[RBGS00] L. Renambot, H. Bal, D. Germans, and H. Spoelder. Cavestudy: an infrastructure for computational steering in virtual reality environments, 2000.

[RDM98] C. Rias, R. Dias, and M. Morim. *Espanha e Itália. O Futebol Espectáculo*. Rui Santos Editions, 1998.

[RE92] B. Read and P. Edwards. *Teaching children to play games*. Leeds: White Line, 1992.

[Rei96] T. Reilly. *Science and Soccer*. London: E and FN Spon, 1996.

[Rei02] L. P. Reis. Agentes autónomos. Technical report, LIACC (NIADR) Research Report, March 2002.

[Rei03] L. P. Reis. Coordination in multi-agent systems: Applications in university management and robotic soccer, *PhD thesis*, engineering faculty of porto university, July 2003.

[RG91] A. Rao and M. Georgeff. Modeling rational agents within a BDI-architecture. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991.

[RJ05] R. Ren and J. Jose. Football video segmentation based on video production strategy. In *Proceedings of the European Conference on IR Research*, pages 433–446, 2005.

[RL01] L. P. Reis and N. Lau. Fc portugal team description: Robocup 2000 simulation league champion. In P. Stone, T. Balch, and G. Kraetzschmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, volume 2103, pages 29–40. Springer, 2001.

[RL02] L. P. Reis and N. Lau. Coach unilang - a standard language for coaching a (robo)soccer team. In *RoboCup 2001: Robot Soccer World Cup V*, volume 2377, pages 251–265. Springer, January 2002.

[RLML10] L.P. Reis, R. Lopes, L. Mota, and N. Lau. Playmaker: Graphical definition of formations and setplays. In *Proceedings of the Iberian Conference on Information Systems and Technologies*, pages 1–6, 2010.

[RLO01] L. P. Reis, N. Lau, and E. Oliveira. Situation based strategic positioning for coordinating a team of homogeneous agents. In *Balancing Reactivity and Social Deliberation in Multi-Agent Systems, LNAI 2103*, pages 175–197. Springer Verlag, 2001.

[RMH$^+$10] M. Randers, I. Mujika, A. Hewitt, J. Santisteban, R. Bischoff, R. Solano, A. Zubillaga, E. Peltola, P. Krustrup, and M. Mohr. Application of four different football match analysis systems: A comparative study. *Sport Science Journal*, 28(2):171–182, 2010.

[RN09] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (3rd Edition)*. Prentice Hall, Third edition, 2009.

[RO01] A. Rocha and E. Oliveira. Electronic institutions as a framework for agents' negotiation and mutual commitment. In *Progress in Artificial Intelligence: Knowledge Extraction, Multi-agent Systems, Logic Programming, and Constraint Solving, LNAI 2258*, pages 232–245. Springer, 2001.

[Ros97] R. Rossetti. An agent-based approach to assess drivers' interaction with pre-trip information systems. *Intelligent Transportation Systems: Technology, Planning, and Operations Journal*, 9(1):1–10, March 1997.

[RSK03] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning Journal*, 53(1-2):23–69, 2003.

[Rud95] I. Rudowsky. Intelligent agents, 1995.

[RVK02] P. Riley, M. Veloso, and G. Kaminka. Towards any-team coaching in adversarial domains. In *Proceedings of the International Joint Conference on Autonomous agents and multiagent systems*, pages 1145–1146, 2002.

[Sal97] S. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery Journal*, 1:317–327, 1997.

[SBF98] R. Studer, V. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *IEEE Transactions on Data and Knowledge Engineering journal*, 25:161–297, 1998.

[SBK01] P. Stone, T. Balch, and G. Kraetzschmar. Overview of robocup-2000, 2001. In RoboCup-2000: Robot Soccer World Cup IV, Springer Verlag, Berlin, 2001.

[SD95] J. Sichman and Y. Demazeau. Exploiting social reasoning to deal with agency level inconsistency. In *Proceedings of the International Conference of Multi-Agent Systems*, pages 352–359. MIT Press, 1995.

[Seb96] J. Sebastián. *Futebol. Entrenamiento fisico baseado en la tactica y la estrategia.1. Los sistemas de juego y su evolucion*. Gymnos, 1996.

[Set03] D. Setterwall. Computerised video analysis of football-techincal and commercial possibilities for football coaching, *MSc Thesis* in computer science submitted in the university of stockholm, 2003.

[SGM+03] M. Sedaghat, N. Gholami, B. Mirbagheri, A. Afkanpour, S. Vaisipour, H. Saffar, V. Montaghami, M. Shahbazi, O. Tourzan, M. Jeddi, A. Valinejad, A. Motamedi, and M. Kangavari. Caspian 2003 presentation description, 2003.

[SHKM92] C. Sammut, S. Hurst, D. Kedzierand, and D. Michie. Learning to fly. In *Proceedings of the International Conference on Machine Learning*, pages 385–393, 1992.

[Sin97] M. Singh. Considerations on agent communication, presented at fipa workshop, 1997.

[Sin03] M. Singh. Agent communication languages: Rethinking the principles. In *Communications in Multiagent Systems*, pages 37–50, 2003.

[SK92] S. Sekulic and B. Kowalski. Mars: a tutorial. *Chemometrics Journal*, 6:199–216, 1992.

[SL92] P Saltas and S. Ladis. Soccer and study in shots, 1992. Thessaloniki, Greece.

[SL06] Y. Sun and J. Li. Iterative relief for feature weighting. In *Proceedings of the International Conference of Machine Learning*, pages 913–920. ACM Press, 2006.

[SMT05] A. Sotiropoulos, M. Mitrotasios, and A. Traulos. Comparison in goal scoring patterns between greek professional and amateur teams. In *First International Scientific Congress in Soccer*, 2005.

[soc04] Soccer simulation online journal from japan (ssojj), 2004. Available at http://www.fs.se.uec.ac.jp/~nishino/robocup/ssojj/SSOJJ-No6.pdf.

[Sou97] M. Sousa. *História do Futebol - Origens, Nomes, Números e Factos*. SporPress, 1997.

[SP01] P. Stone and D. Polani. Robocup-2000: The fourth robotic soccer world championships, 2001.

[SRG+99a] H. Spoelder, L. Renambot, D. Germans, H. Bal, and F. Groen. Interactive and collaborative visualization of robot soccer. In *In Proceedings of the Immersive Projection Technology/Eurographics Virtual Environments Workshop*, 1999.

[SRG+99b] H. Spoelder, L. Renambot, D. Germans, H. Bal, and F. Groen. Man multi-agent interaction in vr: a case study with robocup. In *In Proceedings of the IEEE Virtual Reality conference*, 1999.

[SRG+00] H. Spoelder, L. Renambot, D. Germans, H. Bal, and F. Groen. Real-time interaction in vr with a distributed multi-agent system, 2000.

[SRV00] P. Stone, P. Riley, and M. Veloso. Defining and using ideal teammate and opponent agent models. In *Proceedings of the Innovative Applications of Artificial Intelligence Conference*, pages 1040–1045, 2000.

[SSRO09] D. C. Silva, R. Silva, L. P. Reis, and E. Oliveira. Agent-based aircraft control strategies in a simulated environment. In *AGS*, pages 190–205, 2009.

[Ste56] H. Steinhaus. Sur la division des corp materiels en parties. *Bulletin of the Polish Academy of Sciences journal*, 1:801–804, 1956.

[Ste98] G. Stein. Tracking from multiple view points: Self-calibration of space and time. In *Proceedings of Darpa Workshop*, pages 521–527, 1998.

[Sto98] P. Stone. Layered learning in multi-agent system, *PhD thesis*, school of computer science, carnegie mellon university, December 1998.

[Sto00] P. Stone. Layered learning in multiagent systems: A winning approach to robotic soccer, mit press – isbn:0262194384, 2000.

[SV97] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots Journal*, 8:345–383, 1997.

[SW99] S. Sen and G. Weiss. Learning in multiagent systems, in weiss 1999, 1999.

[SW04] C. Stanton and M. Williams. Grounding robot sensory and symbolic information using the semantic web. In D. Polani, B. Browning, A. Bonarini, and K. Yoshida, editors, *RoboCup 2003: Robot Soccer World Cup VII*, volume 3020, pages 757–764. Springer, 2004.

[SW08] Y. Sun and D. Wu. A relief based feature extraction algorithm. In *Proceeding of the International Conference on Data Mining*, pages 188–195, 2008.

[T.95] Mowbray. T. Essentials of object-oriented architecture. *Object Magazine*, pages 28–32, September 1995.

[Tad92a] A. Tadeia. Em busca dos golos perdidos, expresso, 26-28, 1992.

[Tad92b] A. Tadeia. O jogo invisível, expresso, 29-32, 1992.

[Teo75] L. Teodorescu. Probleme de teorie si metódica in jocurile sportive. *Ed. Sport-Turism*, pages 14–15, 1975.

[THF96] T. Taki, J. Hasegawa, and T. Fukumura. Development of motion analysis system for quantitative evaluation of teamwork in soccer games. In *Proceedings of the IEEE International Conference Image Processing*, pages 815–818, 1996.

[TINF$^+$98] K. Tanaka-Ishii, I. Noda, I. Frank, H. Nakashima, K. Hasida, and H. Matsubara. Mike: An automatic commentary system for soccer. In *Proceedings of the International Conference on Multi Agent Systems*, pages 285–292, Washington, DC, USA, 1998.

[TQ01] V. Tovinkere and R. Qian. Detecting semantic events in soccer games: Towards a complete solution. *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1040–1043, 2001.

[Tsa86] R. Tsai, editor. *An efficient and accurate camera calibration technique for 3-D Machine vision*, 1986.

[TTF09] H. Trevor, R. Tibshirani, and J. Friedman. *The elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, 2009.

[TWH01] R. Tibshirani, G. Walther, and T Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Royal Statisticcal Society B journal*, 63(2):411–423, 2001.

[UG96] M. Uschold and M. Grüninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review Journal*, 11:93–136, 1996.

[Val01] J. Valdano. *Apuentes del balón*. Esfera de Los Libros, 2001.

[VAM09] V. Vinhas, P. Abreu, and P. Mendes. Business intelligence based on a wi-fi real time positioning engine - a practical application in a major retail company. In *Proceedings of the International Conference on Enterprise Information Systems*, pages 11–16, 2009.

[Vap99] V. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, 2nd edition, November 1999.

[Vel00] *RoboCup-99: Robot Soccer World Cup III*, volume 1856. Springer, 2000.

[VGaR06] R. Voser, M. Guimarães, and E. Ribeiro. *Futebol:História, Técnica e Treino de Goleiro*. EDIPUCRS, phorte edition, 2006.

[VHSW97] G. Van Heijst, A. Schreiber, and B. Wielinga. Using explicit ontologies in kbs development. *Human Computer Studies Journal*, 46(2-3):183–292, 1997.

[VROD08] U. Visser, F. Ribeiro, T. Ohashi, and F. Dellaert. *RoboCup 2007: Robot Soccer World Cup XI*, volume 5001. Springer Verlag, 2008.

[Wal92] M. Waldrop. *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster; 1st edition, london:penguin books edition, 1992.

[Wal00] J. Walvin. *The people's game : the history of football revisited*. Mainstream Publishing, 2000.

[WCK97] G. Webb, B. Chiu, and M. Kuzmycz. Comparative evaluation of alternative induction engines for feature based modelling. *Artificial Intelligence in Education journal*, 8:97–115, 1997.

[Wei97] J. Weineck. *Coaching soccer - Conditioning*. Thessaloniki, 1997.

[Wei99] G. Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.

[WG96] G. Weiss and W. Gerhard. Adaptation and learning in multi-agent systems: Some remarks and a bibliography. In *Proceedings of the Conference on Adaptation and Learning in Multi-Agent Systems*, pages 1–21. Springer-Verlag, 1996.

[WGSW02] M. Wellman, A. Greenwald, P. Stone, and P. Wurman. The 2001 trading agent competition. In *Proceedings of the conference on Innovative applications of artificial intelligence*, pages 935–941. AAAI Press / The MIT Press, 2002.

[WJ95] M. Wooldridge and N. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review journal*, 10(2):115–152, 1995.

[WK96] G. Webb and M. Kuzmycz. Feature based modelling: A methodology for producing coherent, dynamically changing models of agents competencies. *User modeling and user-adapted interaction journal*, 5(2):117–150, 1996.

[Woo02] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley and Sons, Second edition, 2002.

[WPB01] G. Webb, M. Pazzani, and D. Billsus. Machine learning for user modeling. *User Modeling and User-Adapted Interaction journal*, 11(1-2):19–29, 2001.

[XLL+02] X. Xing, Q. Li, C. Liu, B. Yang, and X. Chen. An intuitive 3d monitor system with automatic commentary for robocup 2002, 2002. Team description for RoboCup 2002.

[XXC01] P. Xu, L. Xie, and S. Chang. Algorithms and system for segmentation and structure analysis in soccer video. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 928–931, 2001.

[XXC+04] L. Xie, P. Xu, S. Chang, A. Divakaran, and H. Sun. Structure analysis of soccer video with domain knowledge and hidden markov models. *Pattern Recognition Letters*, 25(7):767–775, 2004.

[YA04] A. Yiannakos and V. Armatas. Evaluation of the goal scoring patterns in european championship in portugal 2004. *Performance Analysis in Sport journal*, 6(1):178–188, 2004.

[YHL93] K. Yamaka, M. Hughes, and M. Lott. An analysis of playing patterns in the 1990 world cup for association football. In T. Reilly, J. Clarys, and A. Stibbe, editors, *Proceedings of the Science and Footaball II*, pages 206–214. E&FN Spon London, 1993.

[Yu05] Z. Yu. Gps train location and error analysis which based on the track fitting of the railway´s geometric locus. *Proceedings of the International Conference on Electronic Measurement*, 2005.

[YYYL95] D. Yow, B. Yeo, M. Yeung, and B. Liu. Analysis and presentation of soccer highlights from digital video. In *Proceedings of the Asian Conference on Computer Vision*, pages 499–503, 1995.

[ZSS+03] A. Zareian, R. Samimi, F. Shiva, A. Hedayati, V. Kazemi, E. Nazemi, and J. Adibi. Team assistant 2003, 2003. Team Description Paper of RoboCup 2003: Robot Soccer World Cup VII.