

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Coordination Methodologies applied to RoboCup: a Graphical Definition of Setplays

Rui Lopes

FINAL VERSION

Report of Dissertation
Master in Informatics and Computer Engineering

Supervisor: Luís Paulo Reis
Co-Supervisor: Nuno Lau

July of 2009

**Coordination Methodologies applied to RoboCup: a
Graphical Definition of Setplays**

Rui Lopes

Report of Dissertation
Master in Informatics and Computer Engineering

Approved in oral examination by the committee:

Chair: Rosaldo José Fernandes Rossetti (Auxiliar Professor)

External Examiner: César Analide Freitas Silva Costa Rodrigues (Auxiliar Professor)

Internal Examiner: Luís Paulo Gonçalves dos Reis (Auxiliar Professor)

24th July of 2009

Abstract

The fall of the chess as the main domain for Artificial Intelligence (AI) research after the victory of Deep Blue against Kasparov arose robotic soccer as a new challenging domain for AI researchers. Robotic Soccer is very appropriate for researching in Distributed Artificial Intelligence problems, as for example coordination methodologies, that can later be applied to other domains. Coordination can be defined as several agents working together as a team to accomplish a common goal. In the context of robot soccer, several aspects may benefit from appropriate coordination methodologies since team play is decisive for deciding the games.

Based on previous work, in this thesis it is presented how the robotic soccer agents coordinate their positioning in the field and how they may execute predefined setplays, i.e. how to make them act in a coordinate way using flexible predefined actions. Both methods are used in real soccer and sports in general, commonly known as *formations* and *studied plays* respectively.

The FC Portugal team with the new positioning system has higher performance than with the old one. Several matches were played against ourselves and against other teams showing that the team with the new positioning system scored more goals and conceded fewer.

A graphical tool to define formations and setplays was fully developed in the context of this work. It offers a quicker and more appealing way to define, both formations and setplays. The tool's interface was inspired on blackboards applications, used in real sports nowadays, used for the same purpose. For evaluation purposes the time of definition of a similar setplay using the tool and writing the configuration file using a text file editor was compared. The results show, as expected, that the file written by the last method took much more time and contained errors.

The tool enables very easy definition of setplays that may then be executed in real-time during the games. Examples of complete setplays and its practical execution are also described in the work, showing the usefulness of the setplay concept and the utility of the tool developed.

Resumo

A queda do xadrez como principal domínio para investigação em inteligência artificial quando o Deepblue venceu Kasparov levou à ascensão do futebol como novo domínio. O futebol robótico é muito apropriado para investigação em problemas de inteligência artificial distribuída, como por exemplo metodologias de coordenação que poderão ser mais tarde aplicadas a outros domínios. Coordenação pode ser definida como vários agentes a trabalhar em equipa para atingir um objectivo comum. No contexto do futebol robótico há vários aspectos que podem beneficiar de metodologias de coordenação apropriadas uma vez que o jogo em equipa é decisivo para decidir o resultado dos jogos.

Baseado em trabalho prévio, nesta tese é apresentado como os agentes em futebol robótico coordenam a sua posição em campo e como poderão executar jogadas predefinidas, i.e. como fazer a equipa agir de modo coordenado através da execução de várias acções flexíveis e definidas previamente. Ambos os métodos são usados em futebol real e em outros desportos, usualmente conhecidos como *formações* e *jogadas estudadas* respectivamente.

A equipa FC Portugal com o novo método de posicionamento tem um desempenho melhor que com o antigo. Vários jogos contra a própria equipa e contra a equipa Yowai foram feitos e a equipa com o novo sistema de posicionamento marcou um maior número de golos e sofreu menos.

Uma ferramenta para a definição gráfica de formações e jogadas foi desenvolvida no contexto deste trabalho. A ferramenta oferece um método mais rápido e mais apelativo para a definição, quer de formações quer de jogadas. A interface desta ferramenta foi baseada em aplicações, usadas actualmente em desportos reais para o mesmo propósito. Como medida de avaliação o tempo de definição de uma setplay idêntica usando a ferramenta e escrevendo o ficheiro usando um editor de texto foi comparado. O resultado mostra, como era esperado, que o ficheiro escrito utilizando o último método demorou mais tempo e continha erros.

A ferramenta possibilita a fácil definição de jogadas que podem ser executadas em tempo real pela equipa. Exemplos de jogadas completas e a sua execução prática são descritos neste trabalho que mostram a potencialidade do conceito de jogada e a utilidade da ferramenta desenvolvida.

Acknowledgements

I would like to thank my advisor and co-advisor, Luís Paulo Reis and Nuno Lau respectively, specially the first, for all the knowledge and experience shared and availability at every moment. I also show my gratitude for João Certo and Luís Mota, I am glad for the opportunity of working with such a great team.

My family is included in the group of people I really thank and love, for all the trust, support and love they always showed me, regardless of all the times my little sister drive me crazy, and they are not few, so for them a *Muito Obrigado*.

For those with who I shared nights, with some only afternoons of work, Pires, Grilo, Ferreira, João, Nuno, Nelson and Cerqueira thank you.

Last, but not the least, a special thank to my brothers Ricardo, Paulo and João for being there in the ups and the downs. I thank my friends, hope I do not forget anyone, Ana Cristina, Diogo, Filipe, João, Jore, Kalinha, Mariana, Rosalba, Rute, Steeve and Zé. Those not mentioned are also part of my life, they are just not part of my thoughts right now.

Rui Lopes

Contents

1	Introduction.....	1
1.1	Coordination in MAS.....	1
1.2	Motivation and Objectives.....	2
1.3	Thesis Outline.....	2
2	RoboCup.....	3
2.1	RoboCupSoccer.....	3
2.1.1	Small Size Robot League.....	4
2.1.2	Middle Size Robot League.....	4
2.1.3	Standard Platform League.....	5
2.1.4	Humanoid League.....	6
2.1.5	Simulation League.....	6
2.2	RoboCupRescue.....	7
2.2.1	Rescue Simulation League.....	7
2.2.2	Rescue Robot League.....	8
2.3	RoboCup@Home.....	8
2.4	RoboCupJunior.....	9
2.5	Conclusion.....	9
3	The 2D Simulation League.....	11
3.1	RCSoccerSim.....	11
3.1.1	RCSoccerServer.....	11
3.1.2	RCSoccerSimulatorMonitor.....	12
3.2	New Simulation Features and Challenges.....	13
3.2.1	New Stamina Model.....	13
3.2.2	Goalkeeper Catch Area.....	14
3.2.3	New Dash Model.....	14
3.3	Other tools.....	15
3.3.1	RCSSLogPlayer.....	15
3.3.2	Classic RCSSMonitor.....	15
3.3.3	Soccer Viewer.....	16
3.3.4	SoccerScope.....	16

3.3.5 Team Assistant.....	17
3.3.6 FCP Debugger.....	18
3.4 Conclusion.....	19
4 State of the Art.....	21
4.1 High-level Positional Coordination.....	21
4.2 Plan-based Coordination.....	22
4.3 Tools.....	26
4.3.1 B-Smart Strategist.....	26
4.3.2 Coach Assistant.....	27
4.4 Conclusion.....	28
5 FC Portugal.....	29
5.1 Strategic Agents.....	29
5.2 Setplays.....	31
5.3 Conclusion.....	33
6 PlayMaker.....	35
6.1 Architecture.....	35
6.2 Setplay Mode.....	37
6.2.1 Condition Panel.....	39
6.2.2 Transition Panel.....	40
6.2.3 Absolute vs Relative Positions.....	40
6.2.4 Setplay file format.....	41
6.2.5 Setplay Example.....	41
6.3 Formation Mode.....	42
6.3.1 Creating/Editing a formation.....	43
6.3.2 Delaunay Triangulation.....	43
6.3.3 Linear Interpolation.....	45
6.3.4 Formation file format.....	46
6.3.5 Formation example.....	47
6.4 Conclusion.....	47
7 Results.....	49
7.1 Formation	49
7.2 Setplays.....	52
7.2.1 Square Passing.....	52
7.2.2 Simple Corner.....	52
7.3 Conclusion.....	53
8 Conclusion.....	55
8.1 Future Work.....	56
References.....	57
Appendix A: Competitions.....	61

A.1	Winners of 2D Simulation League RoboCup World Championship.....	61
Appendix B:	Setplay Grammar.....	63
Appendix C:	Formation	69
Appendix D:	Example Setplays.....	75
D.1	Square Passing.....	75
D.2	Simple Corner.....	77

List of Figures

Figure 2.1: Kick Off of Small Size League Game [SSRL, 2009].....	4
Figure 2.2: Medium Size Robot [ROBOCUP, 2009].....	5
Figure 2.3: NAO Robot [TZI, 2009].....	5
Figure 2.4: Humanoid Robots [http://www.er.ams.eng.osaka-u.ac.jp].....	6
Figure 2.5: Rescue Simulation League Snapshot (Bremen 2006) [RESCUE, 2009].....	8
Figure 3.1: SoccerServer Architechure [Reis, 2003].....	12
Figure 3.2: RCSoccerSimulatorMonitor screenshot.....	13
Figure 3.3: RCSSLogPlayer.....	15
Figure 3.4: Classic Monitor.....	
[http://www.cs.rtu.lv/dssg/images/research/robocup/rcssmonitor_classic1.png].....	15
Figure 3.5: Soccer Viewer [http://www.cs.rtu.lv/dssg/images/research/robocup/smonitor2.png]	16
Figure 3.6: SoccerScope2.....	
[http://www.cs.rtu.lv/dssg/images/research/robocup/soccerscope2.png].....	17
Figure 3.7: Team Assistant.....	
[http://www.cs.rtu.lv/dssg/images/research/robocup/sbcmonitor1.png].....	18
Figure 3.8: FCP Debugger.....	18
Figure 4.1: Team strategy definition.....	22
Figure 4.2: Example of CMDragon's play [Bowling et al., 2003]	23
Figure 4.3: Plan structure [Stoye and Elfers, 2007].....	24
Figure 4.4: Plan example [Riley and Veloso, 2002].....	24
Figure 4.5: Play example [McMillen and Veloso, 2006].....	25
Figure 4.6: Example of value rule [Kok et al., 2003].....	26
Figure 4.7: B-Smart Strategist [Stoye and Elfers, 2007].....	27
Figure 4.8: Coach Assistant 2.1.....	28
Figure 5.1: FC Portugal Strategic Agent.....	30
Figure 5.2: Setplays Structure.....	31
Figure 5.3: Action definition.....	32
Figure 5.4: Region definition.....	33
Figure 6.1: PlayMaker Architecture.....	36
Figure 6.2: PlayMaker Data Structures.....	36

Figure 6.3: Step operation buttons.....	37
Figure 6.4: Conditions on PlayMaker.....	38
Figure 6.5: Transitions on PlayMaker.....	38
Figure 6.6: Roles in PlayMaker.....	39
Figure 6.7: Players positions in setplay.....	41
Figure 6.8: Steps in setplay.....	42
Figure 6.9: Navigation buttons and formation name.....	42
Figure 6.10: Vertex added inside and lying on an edge of a triangle [Kreveld et al. 2000].....	44
Figure 6.11: Edge Flip [Kreveld et al. 2000].....	44
Figure 6.12: Screenshot of Playmaker - formation mode with respective triangulation.....	45
Figure 6.13: Example of Delaunay Triangulation and Voronoy Diagram.....	45
Figure 6.14: Gouraud Shading Algorithm. Adapted from [Akiyama and Noda, 2007].....	45
Figure 6.15: Four positionings of a defined formation.....	47
Figure 7.1: Diference between the two positionings for the old and new formation.....	50
Figure 7.2: Goals scored.....	51
Figure 7.3: Square Passing Setplay	52
Figure 7.4: Simple Corner Setplay.....	53

List of Tables

Table 2.1: Comparison of Chess and Soccer	4
Table 7.1: Matches analysis against ourselves.....	50
Table 7.2: Matches Analysis against Yowai2008.....	51

Abbreviations

AI	Artificial Intelligence
MAS	Multi-Agent System
FIFA	Fédération International de Football Association
RCSS	RoboCup Soccer Server
UDP	User Datagram Protocol
IP	Internet Protocol
XML	Extensible Markup Language
SPAR	Strategic Positioning by Attraction and Repulsion
SBSP	Situation Based Strategic Positioning
IEETA	Instituto de Engenharia Electrónica e Telemática de Aveiro
LIACC	Laboratório de Inteligência Artificial e de Ciência de Computadores
DPRE	Dynamic Positioning and Role Exchange
STN	Simple Temporal Network
ATAC	Adaptive Team-Adversarial Coaching
DT	Delaunay Triangulation

1 Introduction

“People who work together will win, whether it be against complex football defenses, or the problems of modern society”

- **Vince Lombardi**

Chess was for many years the main domain for research in the Artificial Intelligence (AI) methodologies, however after the victory of DeepBlue against Kasparov [DeepBlue, 1997], who was in the year of 1997 the world champion in chess, more domains were necessary. This was when robotic soccer came up as a challenging domain for (distributed) AI researchers. The robotic soccer, specially the simulation league is appropriate to apply coordination methodologies since there are two well-defined teams and their composition is known, including 11 players and a coach in each side.

Soccer is a well-known sport over the world, allied to the fact of the existence of a competition where teams play against other teams and only one will be the winner makes this domain more appealing for research than others with more social component. However coordination methodologies even if specifically developed to the robot soccer can later be applied to problems involving more social issues, like rescue in catastrophes [Lau et al., 2005].

1.1 Coordination in MAS

A Multi-Agent System (MAS) is a system composed by several agents. These agents have an autonomous behaviour and interact with other agents in the same system. These agents must be capable of taking their own actions and at the same time being capable of interacting based on social rules, approaching their behaviour to humans, by negotiation, coordination and cooperation.

Introduction

Coordination can be defined as “The act of working together in an harmonious way to achieve a common agreement or goal” [Reis, 2002]. In the context of this thesis the agents act as a team, have common goals. Agents which have this kind of behaviour are cooperative agents and cooperation methodologies focus on this cooperation between agents, trying to solve the global problem in a distributed way.

1.2 Motivation and Objectives

The author's personal passion for soccer allied to the fact that soccer became the standard problem to Distributed Artificial Intelligence and Intelligent Robotics, so it is the main domain to test coordination methodologies in cooperative agents, was the main motivation. Additionally, the opportunity to participate in an international event/challenge like RoboCup was a factor of great motivation.

The goals of the work were to apply methodologies to achieve positional and plan-based coordination of the players/agents of the FC Portugal 2D simulated soccer team, and also the development of a software application that would allow anyone, without any knowledge and understanding in AI fields, to create formations and cooperative soccer plays that could be executed by the FC Portugal 2D simulation team. The final goal is to have a high-level process to define formations and setplays that may work as a competitive advantage enabling the team to easily exploit other teams' weaknesses.

1.3 Thesis Outline

After this introductory chapter the thesis is organized as follows. The next chapter presents a general and global overview to RoboCup and to the leagues that integrate the initiative. In chapter three a more detailed description to the 2D simulation League, including some new rules, and some related tools. Chapter four contains the state of the art relative to cooperation in MAS, more specifically to high-level positional and plan-based coordination. Chapter five includes an introduction to the FC Portugal's team project, the architecture of the agents and the specification and usage of setplays in the team. At chapter six is described the PlayMaker, the tool used to graphically define setplays and formations to FC Portugal team. In chapter seven RoboCup09's results are presented as well as some results of the usage of the tool. Finally chapter eight presents the conclusions and future work.

2 RoboCup

RoboCup is an international initiative to promote the development of AI and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. The first official RoboCup games took place in the year of 1997. The ultimate goal of RoboCup Initiative is that “By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, comply with the official rule of the FIFA, against the winner of the most recent World Cup” [Kitano, 1997].

Currently the RoboCup Initiative is divided in four major domains: RoboCupSoccer, RoboCupRescue, Robocup@Home and RoboCupJunior [ROBOCUP, 2009].

2.1 RoboCupSoccer

The reason for choosing soccer as its primary domain is that in order to a team of robots perform a soccer game several technologies must be incorporated such as design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics, and sensor-fusion [Kitano et al., 1995]. The choice for soccer as the standard problem is a stimulating method of promote the research in AI and Robotics. It is a quite distinct domain from chess, that was for years the most used problem in AI.

Table 2.1: Comparison of Chess and Soccer

	<i>Chess</i>	<i>Soccer</i>
<i>Environment</i>	Static	Dynamic
<i>State Change</i>	Turn Taking	Real Time
<i>Info. Accessibility</i>	Complete	Incomplete
<i>Sensor Readings</i>	Symbolic	non-symbolic
<i>Control</i>	Central	Distributed

2.1.1 Small Size Robot League

The Small Size Robot League, or F180, focuses on the problem of intelligent multi-agent cooperation and control in a highly dynamic environment with a hybrid centralized/distributed system.



Figure 2.1: Kick Off of Small Size League Game [SSRL, 2009]

A small size robot game is played between two teams of 5 players. Each robot must fit in a 180mm diameter circle and its height depend on the presence of local vision sensors or not. The robots without local vision sensors use an overhead camera connected to an off-field computer , that communicates with the robots wirelessly, typically using commercial FM transceivers. The field is a green carpet of 6.05m x 4.05m and the ball is an orange golf ball.

It is a very interesting and challenging domain as to build a successfully team requires clever design, implementation and integration of many hardware and software sub-components into a robustly functioning whole [SSRL, 2009].

2.1.2 Middle Size Robot League

The middle size robot league follows the FIFA rules, with the needed changes to the playing robots. Each team has 5 robots, including the goal keeper. The projection of the robot on the floor must be inside a square of side 30cm and maximum 50cm. The maximum height

RoboCup

allowed is 80cm and the weight is 40kg. The field is a 12m x 18m rectangle. Teams must be able to play with any orange FIFA standard size 5 football ball.

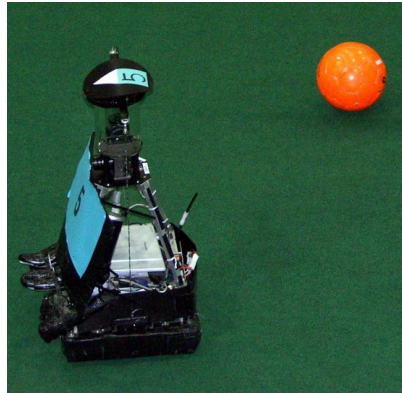


Figure 2.2: Medium Size Robot
[ROBOCUP, 2009]

Robots must recognise objects, locate themselves, choose the actions to perform, controls motors and actuators autonomously. The intervention of humans is not allowed except for placing or removing from the field [MIDDLE, 2009].

2.1.3 Standard Platform League

The RoboCup Standard Platform League is a league in which all teams compete with identical robots. The current standard platform used is the humanoid NAO by Aldebaran Robotics (<http://www.aldebaran-robotics.com>). [TZI, 2009]



Figure 2.3: NAO Robot
[TZI, 2009]

A match in this league is played between two teams of three players each. The field dimension is similar to the small-size league: 6.05m x 4.05m [NAO, 2009]. In addition to the

league, there are three technical skill challenges: “Any Ball”, the Passing and the Localization with Obstacle Avoidance challenges [CHALLENGES, 2009].

2.1.4 Humanoid League

The Humanoid League is one of the most dynamically progressing leagues and the one closest to the 2050 goal. The robots are autonomous and have human-like body plan and senses.

The matches take place in a field of 6m x 4m and each team has no more than three players. The robots are divided into two size classes: KidSize (30-60cm height) and TeenSize (100-160cm height).



Figure 2.4: Humanoid Robots

[<http://www.er.ams.eng.osaka-u.ac.jp>]

Some issues investigated in this league are dynamic walking, running, and kicking the ball while maintaining balance, visual perception of the ball, other players, and the field, self-localization, and team play among others. Also in this league there are some technical challenges. [TZI, 2009] [HUMANOID, 2009].

2.1.5 Simulation League

In this league there is not any real robot, each player is a software agent. Unlike the other leagues, this has a common server that players interact with. The server provides a simulation of the game, giving feedback to the agents of the actions they performed and sensor information.

There are two simulation leagues: 2D and 3D. The 2D league will be detailed in the next chapter as this work is based on it. 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 behaviours of in playing soccer towards the low level control of humanoid robots and the creation of basic behaviours like

walking, kicking, turning and standing up but also provides a test platform for the teams of the Standard Platform League [RC2009, 2009].

2.2 RoboCupRescue

The intention of the RoboCupRescue project is to foster research and development in one of the most serious social issue, disaster rescue at various levels involving multi-agent team work coordination, physical robotic agents for search and rescue, information infrastructures, personal digital assistants, a standard simulator and decision support systems, evaluation benchmarks for rescue strategies and robotic systems that are all integrated into a comprehensive systems in future.

It is considered to be the grand project, as the accomplishment of its long range goal contributes directly to society and may save people's lives. Some characteristics of the problem are similar to the ones encountered on RoboCupSoccer such as a dynamic environment and noisy and incomplete information. There are also some missing features in soccer that are present in RoboCupRescue problem, these are logistics, heterogeneous agents, long-range planning, and emergent collaboration between teams of agents [RSYSTEM, 2009] [ROBOCUP, 2009] [RESCUE, 2009].

There are currently two leagues of RoboCupRescue: Simulation League and the Robot League.

2.2.1 Rescue Simulation League

The simulation league has two purposes. First, it aims to develop simulators that form the infrastructure of the simulation system and emulate realistic phenomena predominant in disasters. Second, it aims to develop intelligent agents and robots that are given the capabilities of the main actors in a disaster response scenario.

It has two subunits:

- Agent simulation: develop agents to undertake the role of Police Forces, Fire Brigades, and Ambulance Teams each with their own capabilities.
- Virtual robots: based on a high fidelity simulator, is the meeting point between the Agent Simulation and the Robot League [RESCUE, 2009].



Figure 2.5: Rescue Simulation League Snapshot
(Bremen 2006) [RESCUE, 2009]

2.2.2 Rescue Robot League

Real robots explore a disaster site to find simulated victims in unstructured environments. It requires robots to demonstrate their capabilities in mobility, sensory perception, planning, mapping, and practical operator interfaces, while searching for simulated victims in unstructured environments [ARENAS, 2009] [RESCUE, 2009].

2.3 RoboCup@Home

The RoboCup@Home league aims to develop service and assistive robot technology with high relevance for future personal domestic applications. A set of benchmark tests and an open challenge is used to evaluate the robots' abilities and performance in a realistic non-standardized home environment. The test should:

- include human machine interaction;
- be socially relevant;
- be application directed/oriented;
- be scientifically challenging;
- be easy to set up and low in costs;
- be simple and have self-explaining rules;
- be interesting to watch;
- take a small amount of time.

Anyone with an autonomous robot can participate [HOME, 2009].

2.4 RoboCupJunior

RoboCupJunior is a project-oriented educational initiative that sponsors local, regional and international robotic events for young students. It is designed to introduce RoboCup to primary and secondary school children.

RoboCupJunior offers three challenges, each emphasizing both cooperative and competitive aspects. Contrasting the one-child-one-computer scenario typically seen today, RoboCupJunior offers participants the opportunity of work together as a team to achieve a common goal.

The challenges developed are [RC2004, 2009]:

- Dance: One or more robots perform to music, in a display that emphasizes creativity of costume and movement.
- Rescue: Robots race to rescue victims from artificial disaster scenarios, varying in complexity from line-following on a flat surface to negotiating paths through obstacles on uneven terrain.
- Soccer: Teams of autonomous mobile robots play games in a rectangular field colour-coded in shades of grey.

2.5 Conclusion

The RoboCup initiative has not only stimulated research in the artificial intelligence field but also has turned media and crowd attention to it. This is clearly observed by checking the growth number of participants and attendants. In RoboCup97 only 38 teams participated, last year the number of participant teams was 373. The choice of soccer as research domain is also responsible for this worldwide acceptance because soccer in “the twentieth century... emerged as the most popular team sport” [Dunning, 1999].

There are not only benefits from the RoboCup, since it is a challenge and the point of compete is to win, it may be a factor to not share the research, this can be supported by the fact that some teams do not release their source code after the competition.

3 The 2D Simulation League

The simulation league is based on the RoboCup Soccer Simulator, a soccer simulation system. It simulates a soccer field, that supports a competition among two teams of eleven autonomous agents each in an uncertain world. As it is a simulated environment allows researchers to focus more on high-level problems as coordination and learning.

There is a virtual referee, and the presence of a human referee to evaluate and decide situations that the artificial referee is not capable of detect like deadlocks. All participating teams play accordingly to a gentleman's agreement.

It is one of the oldest leagues in RoboCup, the first server was completed in the year of 1995, and the year that followed already held a pre-RoboCup competition (See Appendix A for competitions and winners) [Chen et al., 2003].

3.1 RCSoccerSim

The RCSoccerSimulator can be divided in two parts. The RCSoccerServer [Noda et al., 1998] which is the main part of the simulator and it runs the actual simulation. The other part is the monitor that is used to view the simulation.

3.1.1 RCSoccerServer

The server is a software system that allows autonomous agents (regardless the language of implementation) to play a soccer match. Each match is “carried out in a client/server style”. The RCSoccerServer acts as server, providing the simulated field and movements of all players. The clients, are the players and only control their movements and actions. It is also possible to connect a coach, this is a special agent that has a global view of the world state and is able to

communicate with the other team agents when the game stops. The communication is done via UDP/IP sockets, therefore the agents may be implemented in any language that supports UDP/IP.

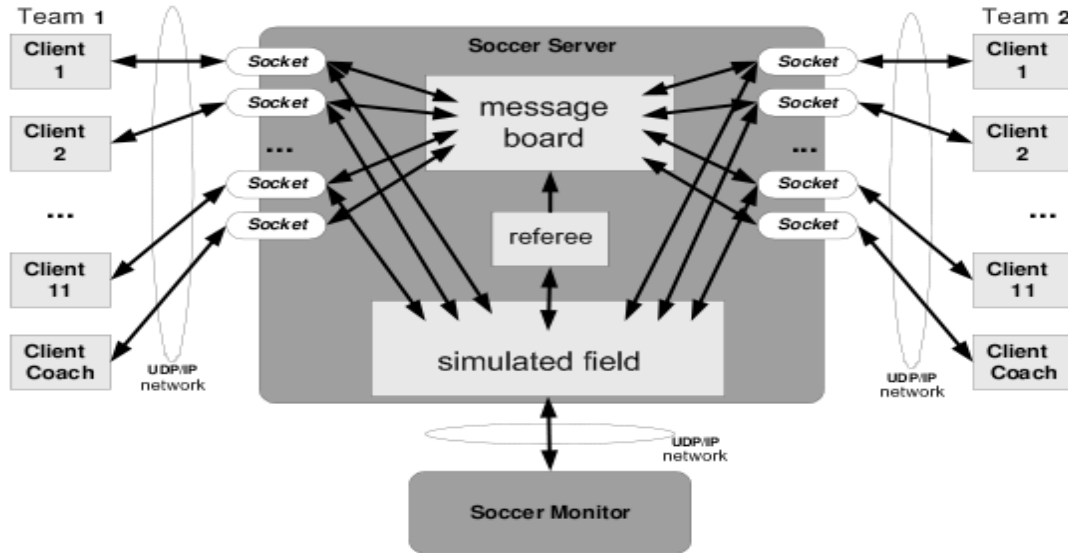


Figure 3.1: SoccerServer Architecture [Reis, 2003]

The client is the thinking unit of the player, and uses the socket to send low-level control-commands, to the server, receiving as feedback sensorial information. Each client is an independent unit, it only controls one player [Chen et al., 2003]. The last available version is 13.2.1 [SSERVER, 2009].

3.1.2 RCSoccerSimulatorMonitor

It is based on the FrameView by Artur Merke. The monitor is connected to the server and is used to see the simulation as it takes place or to view a playback if connected to the log player. A n number of monitors can be connected to the same server. It displays a virtual soccer field based on X window system¹.

It allows some control over the ball, like drop it anywhere and to provide free-kicks to both teams in the standard mode. The view mode lets the user to choose a player, and its vision cone will be displayed. The other mode is the move mode that has the functions of place the players in the mouse position by pressing its number on the keyboard. These three modes, turn the monitor into a good testing tool on run time.

¹ computer software system and network protocol that provides a graphical user interface for networked computers



Figure 3.2: RCSoccerSimulatorMonitor screenshot

It also has some zoom options, as well as scale and level of detail that is desired to see about players information. The essential information about the match that is being showed is also available, the participant teams, the score, the game mode and the time. The latest version released was 13.1.0 [Chen et al., 2003] [SSERVER, 2009].

3.2 New Simulation Features and Challenges

With the attempt to provide a more realistic simulation some major changes were done to the server since the last participation of FC Portugal in official competitions. In this section are presented some of the changes that affected the performance of the team in the qualification to the World Championship.

3.2.1 New Stamina Model

In the previous model, only three variables were important: stamina, recovery and effort. When a player dashes his stamina is consumed and then is replenished by the recovery value each cycle. The effort is related to how effective the dash will be.

In the newer version a new parameter was added, the `stamina_capacity`. This slightly changed the stamina model. The new variable added represents the total amount of stamina that a player is able to recover. If this parameter reaches zero, the player will never recover any

The 2D Simulation League

stamina. This value is set to maximum before kick off and at half time. If after the normal halves the game is not over yet, it is restored to its initial value, but not in the half time of extra halves or before penalty shoot-outs.

In qualification, in a high number of games, FC Portugal's players got tired too soon, causing the team's performance to fell drastically, and some goals were suffered because of the slowness of the tired players. This was due to the fact that the players were reasoning believing that the old stamina model was being used.

3.2.2 Goalkeeper Catch Area

The game parameters keep changing every time a new version of the server is released. The area around the goalkeeper in which he can catch the ball, was reduced from 2.0 to 1.2, almost half of the previous value. As consequence, FC Portugal's goalkeepers in the qualifying matches thought he could catch the ball 2 meters away when he only could when it was 1.2 meters apart.

FC Portugal conceded 64 goals in 30 qualification matches. 30% of them was due to the goalkeeper's behaviour, has he got out of the goal thinking he could catch the ball and in reality he didn't. Not only a high number of goals, but also important goals, as the third goal suffered against DAInamite, when the counter marked a draw on to goals.

3.2.3 New Dash Model

The dash was performed in the direction of the player's body with a power that was given in the command sent. With the new model a new parameter was added to the command, the dash direction. This allows the players to dash in all directions, and not only forwards or backwards. The angle is relative to the players' body angle when the power is higher than 0 and reverse side angle is the power is negative.

The effectiveness of the dash is affected by the desired direction to dash. In future versions of the server, the minimum power allowed will be 0, there will be no negative dash power. Another parameter was added in order to balance with older binaries and to enable flexible settings, the *dashangle step*. It defines the discreteness of the dash angle. If this parameter is setted to 90.0 the player could only dash to four directions, 0.0 forward, 90.0 and 270.0 sideways and 180.0 backwards.

3.3 Other tools

3.3.1 RCSSLogPlayer

The RoboCup Soccer Simulator Log Player is used to replay games that have been logged by the simulator. It is a powerful debugging tool, the information displayed is highly configurable, it allows tracing the ball and the players. Displays all the information about the players, from its number to the number of actions performed. Available at http://sourceforge.net/project/showfiles.php?group_id=24184&package_id=27089.



Figure 3.3: RCSSLogPlayer

3.3.2 Classic RCSSMonitor



Figure 3.4: Classic Monitor

[http://www.cs.rtu.lv/dssg/images/research/robocup/rcssmonitor_classic1.png]

The 2D Simulation League

The classic monitor is a 2D monitor that can be used as monitor to watch running games or as a log player. Like the RCSoccerSimulatorMonitor it shows the the score, team names, and the positions of all the players and the ball. It also shows the players' stamina and runs under X11. There is a Windows version available too which allows to show the player's kickable area. Available at http://sourceforge.net/project/showfiles.php?group_id=24184&package_id=27088.

3.3.3 Soccer Viewer

Another option to see running games or to play logs. Based on the classic monitor with the possibility to show an offside line. It only runs on UNIX systems.



Figure 3.5: Soccer Viewer [<http://www.cs.rtu.lv/dssg/images/research/robocup/smonitor2.png>]

3.3.4 SoccerScope

SoccerScope [Miyazawa et al., 2002] is a monitor/log player written in Java. It is more sophisticated than the described before. It has all the features present in the other monitors, showing stamina, the offside line, players' kickable area and visibility regions and some new features like trajectory of passes, velocities and trajectories of objects, players' messages and dominant regions. Due to all the information it can display and being highly configurable to users' desires it is a good visual debugger tool. It is available at <http://ne.cs.uec.ac.jp/~Ekoji/SoccerScope2/SoccerScope2.zip>.

The 2D Simulation League

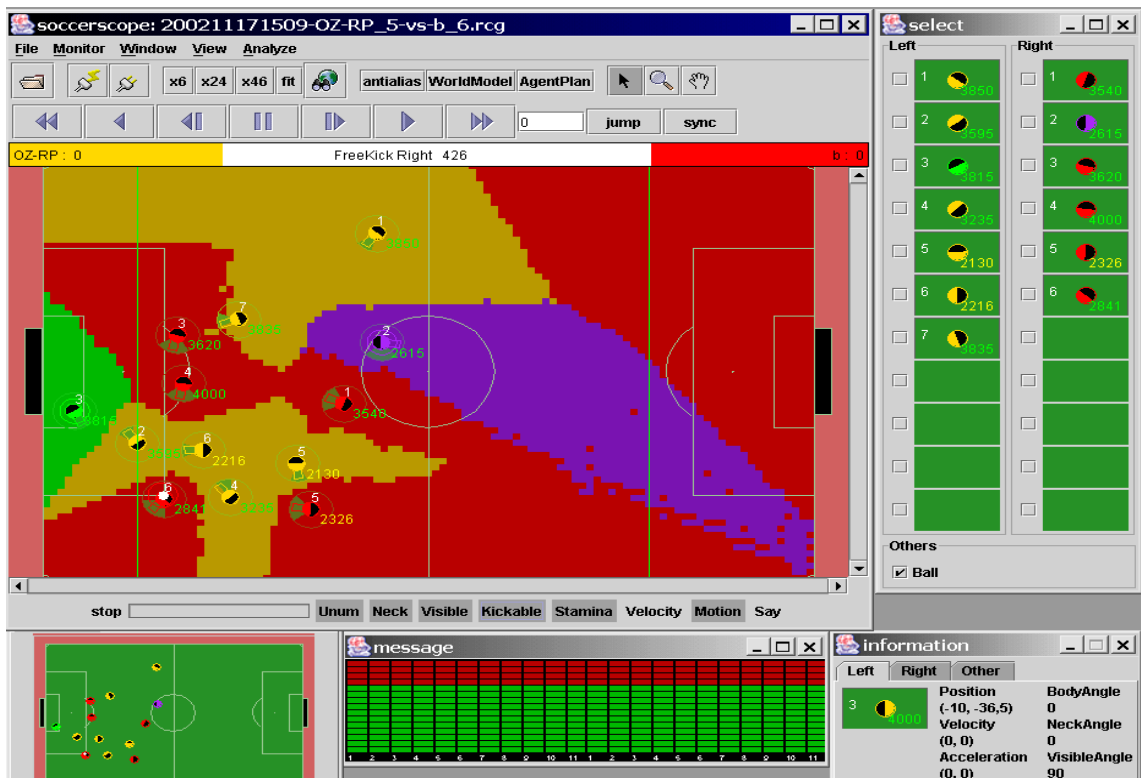


Figure 3.6: SoccerScope2

[<http://www.cs.rtu.lv/dssg/images/research/robocup/soccerscope2.png>]

3.3.5 Team Assistant

Team Assistant 2003 [Zareian et al., 2003] is a multi function tool for the soccer simulation league. As a logplayer it is able to show the players' views/attributes and graphically trace them. It also includes a graphical debugger which allows graphical representation of statements stored in a Player log. The generic grammar which is proposed for the construction of this log consists of statements for declaring: a world model, intended player behaviours, performed skills, etc.

It has an analysis component and it can recognize *ball possession* and actions like *passing*, *dribbling*, *shooting*, etc. and display them graphically on the field. Reports can be generated to evaluate the players individually based on the events listed above.

The 2D Simulation League

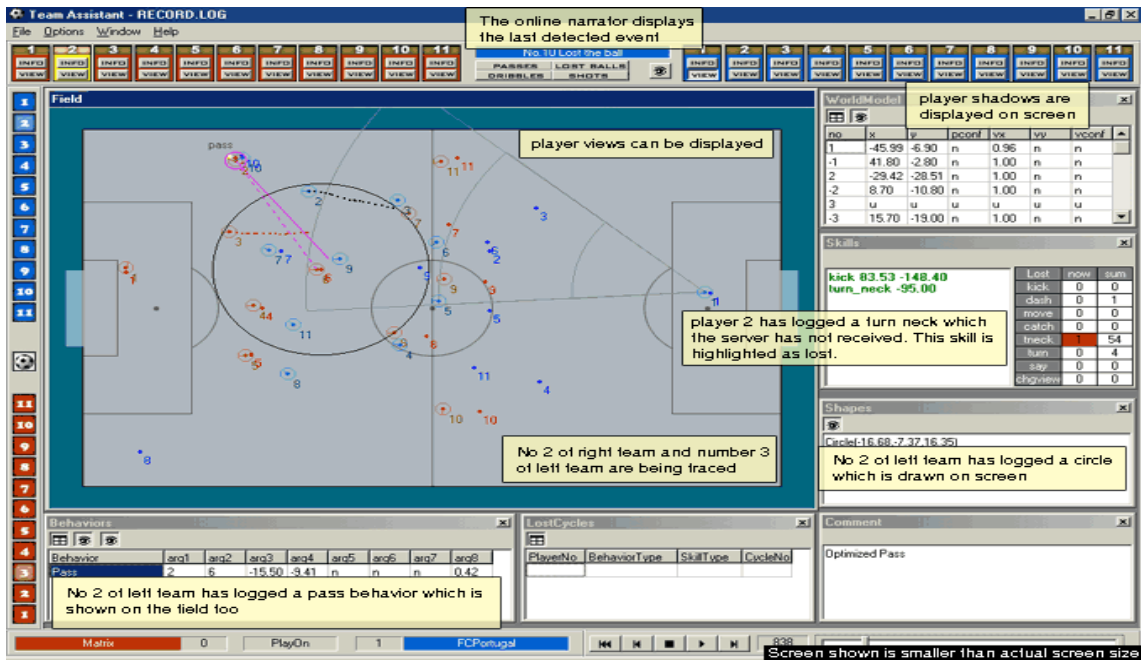


Figure 3.7: Team Assistant

[<http://www.cs.rtu.lv/dssg/images/research/robocup/sbcmonitor1.png>]

3.3.6 FCP Debugger

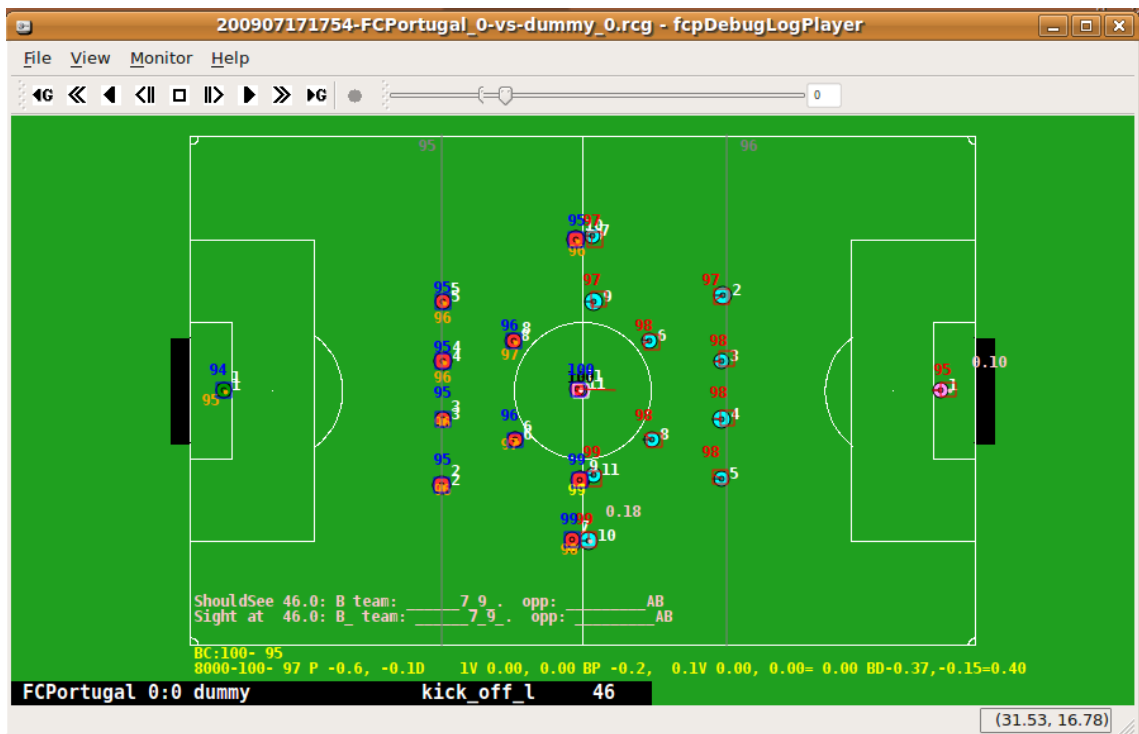


Figure 3.8: FCP Debugger

The actual version is based on the RCSSLogPlayer, this visual debugger has all the features as the original one [Lau et al., 2007a] [Lau et al., 2007b]. Additionally it reads log files

written by the FC Portugal players containing debugging information. Other customized features that make this debugger unique and very important to the FC Portugal team are also available, e.g. the visual display of a player's world state in comparison to the real world state.

3.4 Conclusion

A simulated environment by the organization provides the chance of researchers to focus on more high-level tasks. The permanent evolution of the server provides, each day more, an approach to real soccer.

There are lots of tools that can be used to improve development and debugging of 2D simulation teams. The reason for this is that teams develop their own tool based on a monitor or a log player, like some showed above, and configure them to do exactly what they want. A customised tool is more efficient than a tool released for all teams. A great catalyst for the developing of these tools is the fact that they are open-source under the GNU general public licence so that can be modified to accomplish team's objectives.

The 2D Simulation League

4 State of the Art

Coordination inside a group of robots is challenging, specially in a dynamic and uncertain environment like robot soccer where a team of robots must work together to achieve a common objective, the victory in the current match.

4.1 High-level Positional Coordination

[Stone, 1998] introduced a method to achieve coordination based on repulsions and attractions called *Strategic Positioning by Attraction and Repulsion* (SPAR). When an agent is positioning itself using SPAR, the agent maximizes the distance to other players and minimizes the distance to ball and to goal. This is achieved evaluating several forces: repulsion from opponents and team mates, attraction to active team mate, ball and opponents' goal. It also uses other constraints that have influence in agent's positioning: stay in an area near home position, stay within the field boundaries, avoid being at an offside position and stay in a position where is possible to receive a pass.

Later it was introduced the *Situation Based Strategic Positioning* (SBSP) by [Reis et al., 2001] [Lau and Reis, 2007]. If an agent is not involved, and won't be soon, in an active situation it will try to occupy its strategic position relative to the actual situation of the game. Through the analysis of the tactic, formation, self positioning in the formation and player type, a player is able to define its base strategic positioning. This position is then adjusted accordingly to ball's position and velocity and situation (i.e. attack, defence, etc...). The player type defines the player's strategic characteristics like ball attraction, admissible regions in the field, specific positional characteristics for some regions in the field, tendency to stay behind the ball, alignment in the offside line, and attraction by specific points in the field in some situations. Using a strategic positioning like SBSP the players will be more well distributed over the field

State of the Art

than using a active one like SPAR, this is the reason why other teams adopted SBSP as the standard positioning method.

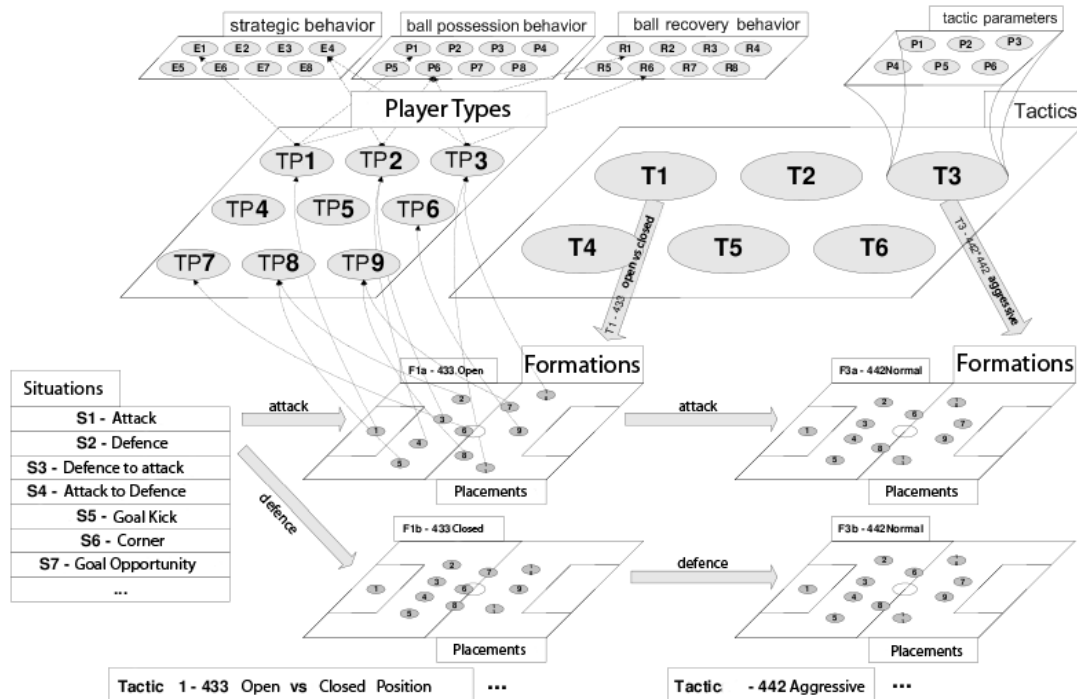


Figure 4.1: Team strategy definition

The SBSP is supported by Dynamic Positioning and Role Exchange (DPRE) [Reis et al., 2001] [Lau and Reis, 2007] which is based on the work of [Stone and Veloso, 1999] suggesting the use of flexible agent roles. Extending that concept, now players exchange their positionings and player types in the actual formation if the utility of that exchange is positive to the team. The utility is calculated using distance from present position to strategic position of both players and the importance of their positionings in the formation.

4.2 Plan-based Coordination

Different authors refer to a plan with different terminology, the reader must interpret plan, play or setplay with the same meaning as a set of coordinated actions performed by the agents when some conditions are satisfied.

In [Bowling et al., 2003] is introduced the concept of play as a multi agent plan, combining reactive and deliberating principles. The CMDragons team, a small size league team, used plays to increase performance. Some sub-goals were defined: “coordinate team behaviour”; “execute temporary extended sequences of actions”; “allow for special behaviour for certain

circumstances”; “allow ease of human design and augmentation”; “enable exploitation for short-lived opportunities” and “allow on-line adaptation to the specific opponent”.

A play is specified using the play language, that is and must be an easy-to-read text format. A play has keywords, that separate different group of information. *Applicable* defines when the play can be executed, *done* when the play must stop. The keyword *role* holds a sequence of active and non-active tactics. Tactics may be interpreted as actions, e.g. shoot and defend_point. Each player has a non-static role, in this approach, and each role has a sequence of actions the player must execute.

```
PLAY Naive Offense

APPLICABLE offense
DONE aborted !offense

ROLE 1
  shoot A
  none
ROLE 2
  defend_point {-1400 250} 0 700
  none
ROLE 3
  defend_lane {B 0 -200} {B 1175 -200}
  none
ROLE 4
  defend_point {-1400 -250} 0 1400
  none
```

Figure 4.2: Example of CMDragon's play [Bowling et al., 2003]

It was also introduced the concept of *playbook*, providing a method for choosing the best plays based on the history of execution and the actual opponent. Basically each play was associated with a weight. The plays weight increased when the play was successfully executed and decreased when failed. A newer approach for play selection, based on the “the experts” or “k-armed bandits” problem was introduced by the same authors in [Bowling et al., 2004].

[Stoye and Elfers, 2007] with the purpose of enhance the B-Smart² software and based on the work related previously, presented their approach that differs from the previous in the definition of the plays and the adaptive selection. To create the plays a tool is used, the *Strategist*, that will be described in the next section.

A *planbase* is the highest level and base structure, it contains multiple plans. A plan is composed by variants and plan steps, and each plan step is defined as a group of conditions and actions. To enter a plan step its conditions must be satisfied. The variant is a list of conditions, it must have at least one element. Variants are the conditions that must be satisfied to start a plan. Each variant must contain exactly one condition containing the ball carrier position. This will be used to filter only the possible plans that can be executed. After all variants of these

² <http://www.b-smart.de/>

State of the Art

applicable plans are checked, and only plans with at least one consistent variant are in the set of applicable plans.

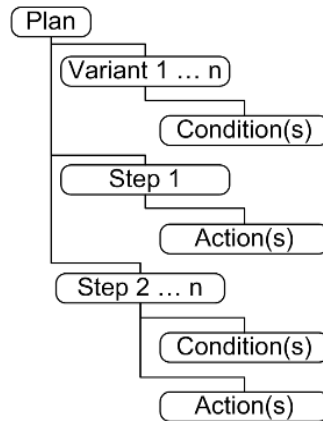


Figure 4.3: Plan structure [Stoye and Elfers, 2007]

[Riley and Veloso, 2002] introduced an *Adaptive Team-Adversarial Coaching* (ATAC). It is an on-line method to provide the team agents with a plan that the coach has generated in response to the specific opponent team. It takes advantage of the opportunity that the soccer simulator offers to let the coach communicate with the players when the game stops for some reason. The coach is equipped with some pre-defined opponent models that are a probabilistic representation of opponents' predicted positions. The coach matches the models to observations of the players' positions and when the game stops it generates a plan through a hill-climbing search that predicts the most likely opponent's model. This model is also used to predict opponent players' behaviours.

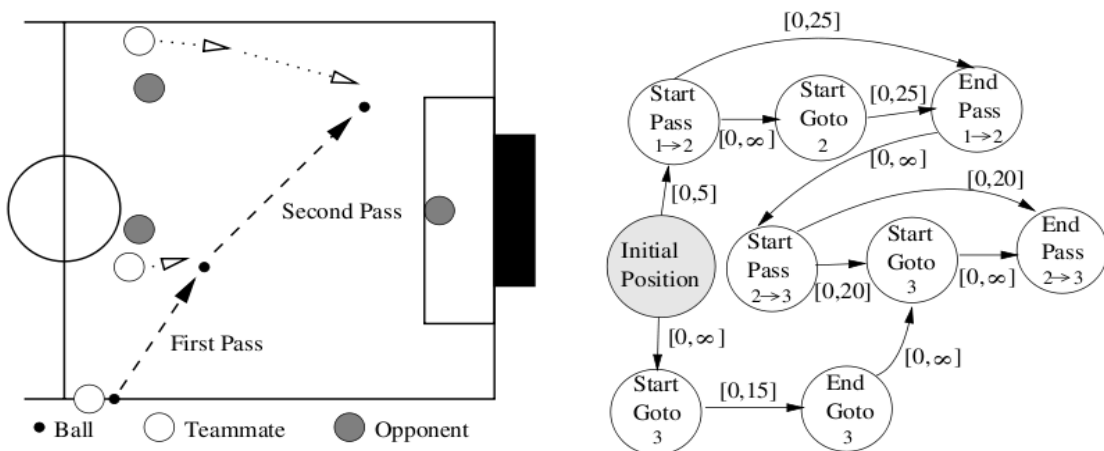


Figure 4.4: Plan example [Riley and Veloso, 2002]

State of the Art

A plan in this approach is represented as a *Simple Temporal Network (STN)* [Dechter et al., 1991]. The STNs are directed graphs where nodes are events and edges temporal constraints between events. Figure 4.4 shows a pictorial representation and the corresponding *STN*.

In this example the number in the nodes are the agents involved in that event, other information about the events is not shown in the Figure, e.g. the position to where the agent must go, when the action is *Goto*. The edges have the time constraints, the two values are the minimum and the maximum time steps the destination node event must occur after the previous. These time values can be negative or infinite too.

As an extension of [Bowling et al., 2003] work, [McMillen and Veloso, 2006] applied a play-based coordination method to a team in the now extinct four-legged league. The objective was to develop a play system that could work in a distributed team. The textual definition of the plan is strongly inspired in the work of Bowling. It is possible to define *applicability conditions*, the *roles* to be used and the play's *weight*. The applicability conditions denote when a play can be executed and the weight is used to decide what play to be used if several are available. The roles define what roles to be used, even if the team loses some of the robots, due to crashes or penalties.

Differing from Bowling's work is the absence of keywords that specify the end of the play. In this approach the play selector continually runs and a play is considered to be complete as soon as another play is chosen. This can happen because of non-satisfaction of the conditions or if this play has a greater weight. This play selector runs in a robot, that is chosen randomly when the match starts and it broadcast the current play and role assignments. If a player misses to receive the information after a predefined amount of time it plays accordingly to a *default play*.

```
PLAY Guard
APPLICABLE winning fewerPlayers
APPLICABLE secondHalf winningBy20rMoreGoals
ROLES 1 Goalkeeper
ROLES 2 Goalkeeper Defender
ROLES 3 Goalkeeper Defender Independent
ROLES 4 Goalkeeper Defender Midfielder Independent
WEIGHT 3
```

Figure 4.5: Play example [McMillen and Veloso, 2006]

In Figure 4.5 is an example of a play to be applied when the team is winning and has fewer players or when is winning by twenty or more goals and the game is in the second half. The keyword *roles* is followed by the number of players available and by the role names, i.e. if there is only one active player in the team *Goalkeeper* role will be assigned to it.

The UvA Trilearn team [Kok and Vlassis, 2006] used a coordination graph framework to achieve coordination between the players [Kok et al., 2003]. The team strategy was defined using value rules. A value rule is a propositional rule over state and action variables. Each of these rules is associated with a local pay-off that is obtained when the action and state variables can be applied.

$$\langle \rho_1; \neg \text{passBlocked}(1,2) \wedge a_1 = \text{passTo}(2) \wedge a_2 = \text{receivePass}(1) : 10 \rangle$$

Figure 4.6: Example of value rule [Kok et al., 2003]

With a set of value rules a coordination graph can be constructed, where nodes are agents and edges indicate that the correspondent agents must coordinate their actions. This approach decomposes the global coordination problem to the sum of smaller local terms, as it maximizes the sum of the pay-offs. The determination of the joint action to execute is done, first by eliminating all the rules that are not applicable in the current state. Then a variable elimination algorithm is applied, it iteratively eliminates an agent by solving a local coordination problem and passing a message to the neighbours in the graph with the results. When only one node is left, its strategy is fixed and is passed in the reverse order. Agents fix their strategy by conditioning on the strategies of the neighbours. This will return the joint of actions that maximize the sum of the pay-offs.

4.3 Tools

4.3.1 B-Smart Strategist

Software developed by [Stoye and Elfers, 2007] for the creation, edition, management of plans. The objective of its development was to provide an easy and intuitive way of complete the tasks it was designed to. The interface was inspired on blackboards applications used in “real” sports.

The plans must be designed by humans users. It uses a qualitative world model [Schiffer et al., 2006] with grid rastering, this means that the field is divided into equal rectangles. In this software were used 8x10 regions and every region was numbered consecutively to be more explicit.

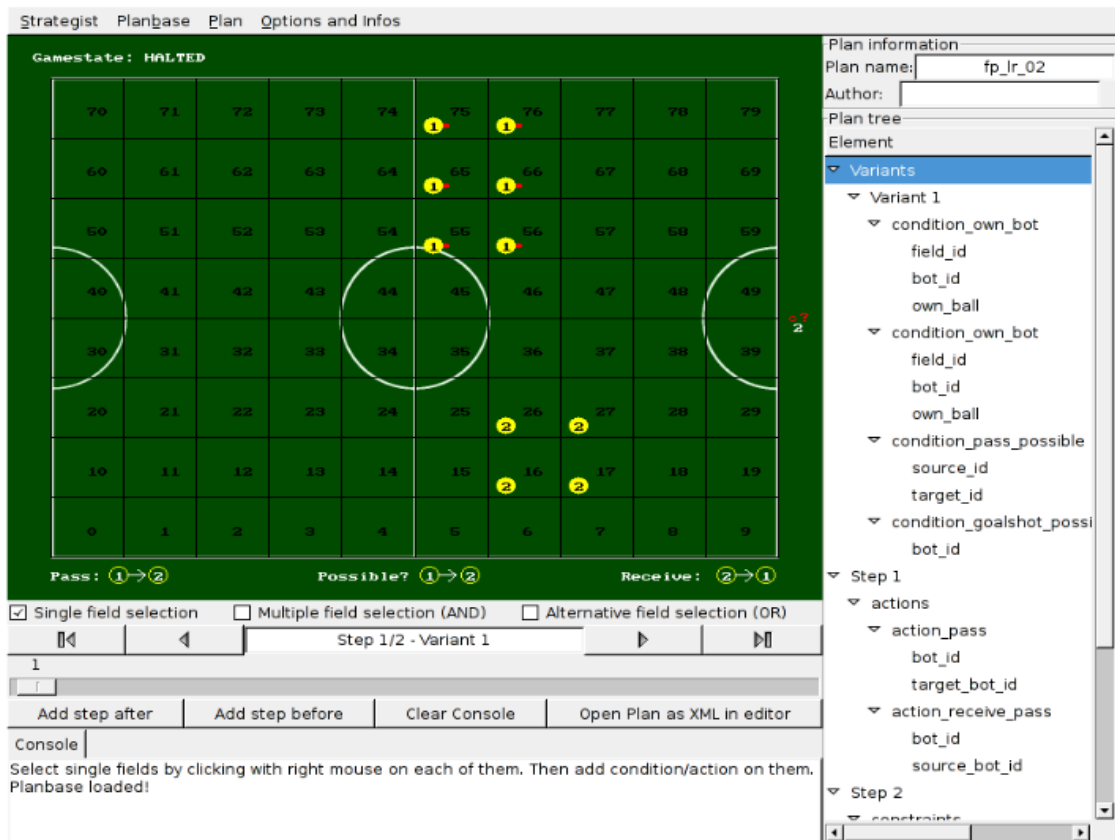


Figure 4.7: B-Smart Strategist [Stoye and Elfers, 2007]

4.3.2 Coach Assistant

Coach assistant [IRAVANIAN, 2009] is a tool for creating positioning patterns and visualizing them for a team of coachable players. It can also serve as a tool for visual analysis of the players and their relative positions. The user defines the ball region and players regions, and the tool creates an one-to-one map from the points inside the ball region and the player region. This map is created by performing some arithmetic on the ball's coordination, based on attraction values. It generates Coach Language (Clang) [Chen et al., 2003] statements for the strategy assigned to the team. The generated CLang may need little modification to form a complete team's strategy.

Earlier versions of Coach Assistant were Windows based applications written in C++ using Borland C++Builder. The current version is written in Java, and hence is platform independent. It is available in <http://sina.iravanian.googlepages.com/robocup>. The source code is also published under GNU General Public License.

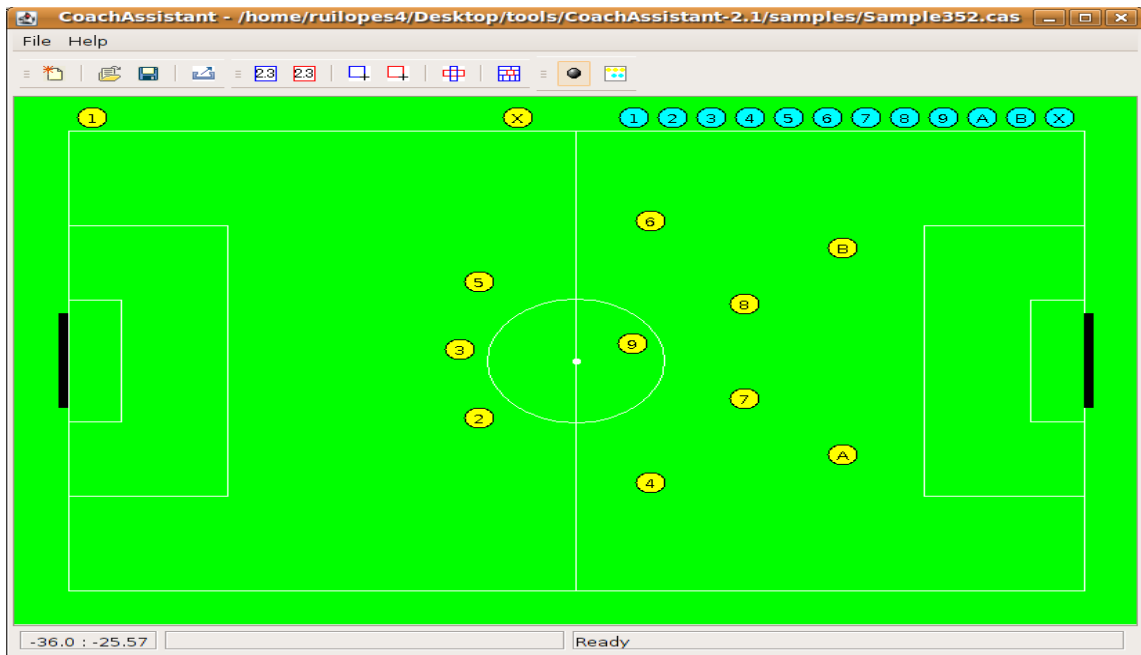


Figure 4.8: Coach Assistant 2.1

4.4 Conclusion

The idea of using setplays to achieve coordination is not recent, in a real sports perspective and in the context of this thesis, it is probably as older as the soccer itself. It is easy to imagine a two players planning a very simple play, even in the origins of soccer, before the game starts whispering “If you have the ball and I am in that position pass it to me”. This is a basic definition of a setplay, when some conditions are satisfied (ball possession and position of the receiver), an action (pass) is performed. Of course this is a very simple example, but was just given to make a point.

In the simulated soccer was introduced in the RoboCup99, by the winning team CMUnited99 [Stone et al., 2000] however it was not used in the official competition. This first approach consisted in the coach informing the players of ball's real position, when the game stopped, and based on this the players executed one of several possible setplays.

The lack of software tools to define setplays and formations is obvious from the analysis made on this chapter. Each team has their own definition of plan, different formats, different implementation for this reason the tools to define plans must be customized to individually and teams probably focus their efforts in other aspects of the team than this.

5 FC Portugal

FC Portugal is a RoboCup Simulated League team developed at IEETA / University of Aveiro and LIACC / University of Porto. This project started in February 2000.

The FC Portugal team won the European and World Championship in 2000, the first time it participated [Reis e Lau, 2001a]. A year later, won the German Open and placed 3rd in the World Championship. In 2002 World championship placed in 5th. In 2006, the 3D team won the Dutch Open and the World Championship [IEETA, 2009].

5.1 Strategic Agents

This subsection is based on the description of the agent presented in [Reis, 2003].

The characteristics of the environment, objectives of the agents and the cooperative task that they need to perform forces the development of an agent which can balance the reactivity with the individual capability to act deliberately. Since it is a specific domain the agent needs specific modules that can deal with advanced coordination methodologies and specific knowledge.

The FC Portugal *Strategic Agent's* architecture divides the necessary knowledge in two independent structures: coordination knowledge and domain knowledge. In the domain knowledge is included the dynamics of the world that is known by the agent and its capability of perception and action. In the coordination knowledge is included the strategy to follow, the available tactics, situations, formations and roles that will allow to use the strategic coordination. It is also included opponent modelling, intelligent communication and perception strategies.

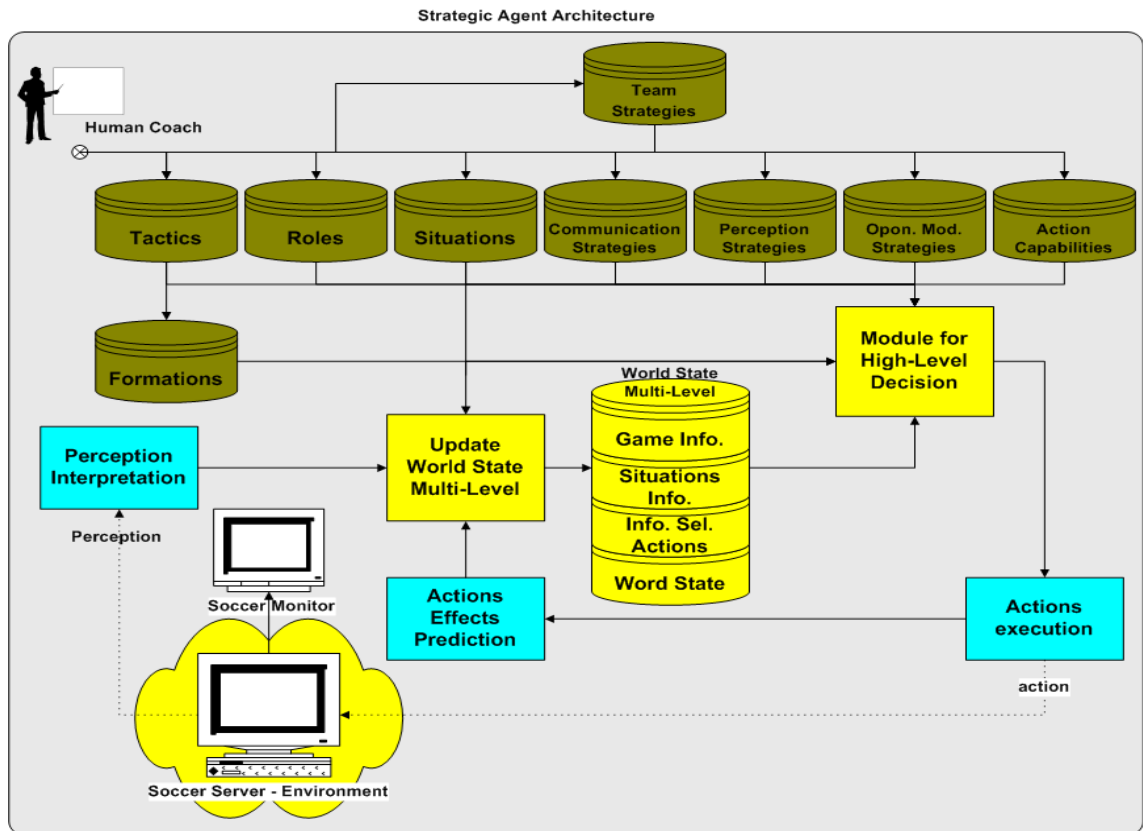


Figure 5.1: FC Portugal Strategic Agent

A agent's world state is defined using a multi-level structure that contains low-level information like positions, velocities and orientation of the objects in the field, and also strategy-level (high-level) information. The world state is actualized based on agent's perception, received information through communication, world dynamics and the prediction of self-actions and other agents actions.

The strategic decision module is responsible for selecting at every moment the tactic to be used as well as define the current situation, and based on this information choose the team formation, role and strategic positioning of the agent. In this module are also made decisions about communication and perception. The team-mates' and opponents' actions are predicted in this module accordingly to the available information in the world state and retrieved from the mutual modeling. When an agent is in an active situation, is the ball carrier or is going to intercept or receive the ball, the decisions from the strategic module is complemented with individual decision, adding specific domain knowledge and information of the low-level world state. It is responsible for specific domain decisions, e.g. where to shot the ball, how to intercept, among others. These individual decisions are transformed in commands that will be sent to the simulator.

5.2 Setplays

The setplay framework [Mota and Reis, 2007] used by FC Portugal was designed to be flexible, general, parametrizable and applicable to any soccer league. An overview of the global structure can be seen in Figure 5.2.

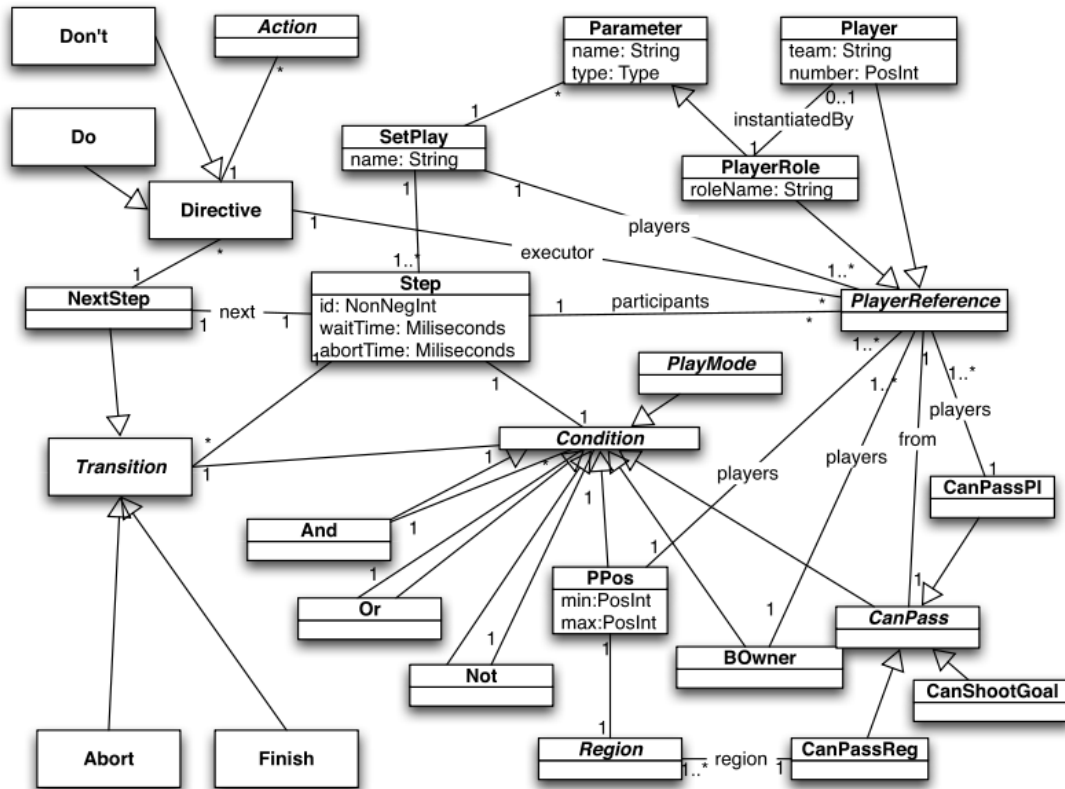


Figure 5.2: Setplays Structure

A *setplay* is identified by a name and has parameters. It also have *player references*, identifying the players taking part in the setplay, pointing to specific players or *player roles* identified by name.

A *setplay* has a list of *steps*, which are it's main blocks. They can be seen as states of the setplay execution. Each *step* is identified by an *id*. The first *step* of all *setplays* is by convention identified by the number 0. *WaitTime* and *abortTime* were introduced to control the execution of the *step*, the first is the time the players must wait after entering a *step* before starting the *transition* to another *step*. The *abortTime* is the maximum time the players will wait before aborting the *setplay* if it is not possible to move another *step*. A *step* has a *condition* that must be satisfied to enter that *step*. The players taking part of the step are called *participants*.

There are several ways to progress between *steps* or finish the *setplay* defined as *transitions*. These *transitions* may have a *condition* that needs to be satisfied to follow that *transition*. There are three kinds of *transitions*: *abort*, *finish* and *nextStep*. An *abort transition*

The last concept in the core of the setplay framework is a *region*, also based in Clang, and depicted in Figure 5.4. A *region* is a spatial entity, a *point*, a *triangle*, an *arc* or a *rectangle*. It was introduced the concept of *dynamic point*, referring to the location of the ball or a player. The concept *named region* models an intuitive region, e.g. our midfield or our penalty box as defined in [Reis and Lau, 2001b].

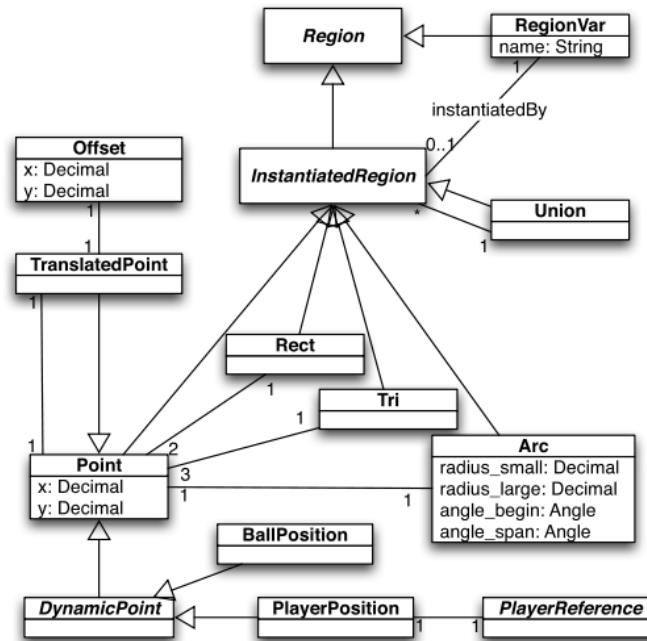


Figure 5.4: Region definition

A complex setplay must follow several steps and the players participating must be synchronized. The communication and synchronization policy in the framework is as follows. Each step is led by one player, the ball owner, called lead player because it is the one that can take the most important decisions. The lead player is responsible for instruct the other players of setplay begin, step entries and transition choices. Entering into a new step usually implies changing the lead player.

5.3 Conclusion

The low-level of the team is based on the code released by CMUnited although it has suffered modifications. The description above reveals that the members that started and continued developing the team value the high-level. The setplay framework presented is highly complex when compared to the related work in the previous chapter. The development of a tool to reduce the time defining high-level aspects, in this case formations and setplays, is perfectly acceptable and justified.

FC Portugal

6 PlayMaker

PlayMaker is the name chosen to the tool developed, as its main function is to provide a chance to define setplays in a graphical way, without the mess of write long text files, avoiding syntax errors and saving time.

Currently it is possible to choose two run modes, the setplay definition and the formation definition. A formation is nothing more than a set of positionings. A positioning is the players' position in the field relative to where the ball is. A formation can be seen as a “special” defensive setplay. These “special” kind of setplays wouldn't have transitions and the only condition would be to the ball be at a point of the field. As only the ball's position and players' position are important to this kind of setplays, and the tool's output is also different, it was decided to divide the tool in two modes. The user can choose the mode wanted after starting the tool.

6.1 Architecture

PlayMaker uses the same interface with two distinct modes. The data structures used in both modes are the same, however the formation mode does not uses the complete structures. The GUI is used to fill the data structures and when the setplay is saved it is generated a configuration file, containing the setplay or the formation definition.

PlayMaker

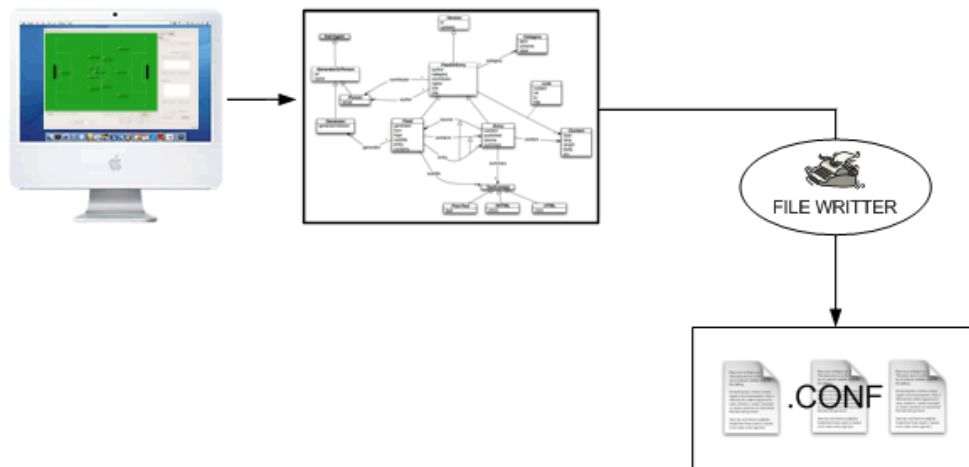


Figure 6.1: PlayMaker Architecture

The data structures used are a simple version of the ones used in the implementation in the FC Portugal team.

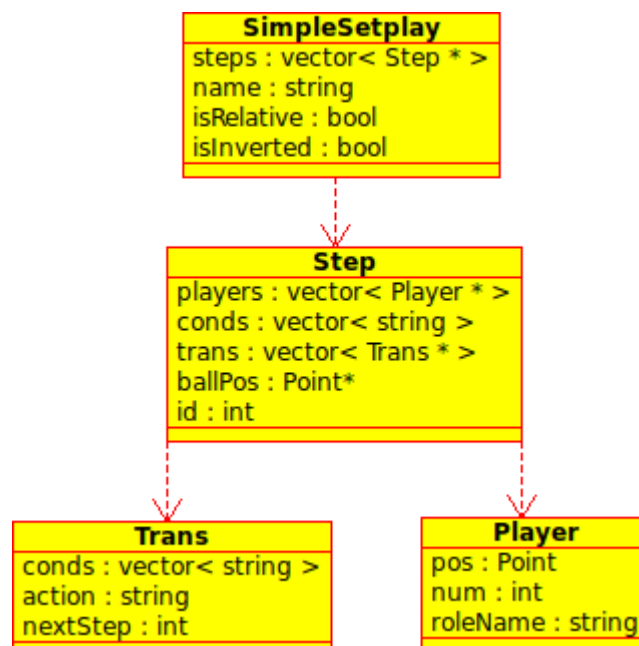


Figure 6.2: PlayMaker Data Structures

A SimpleSetplay object represents a setplay or a formation depending on the mode that the tool is running. In the string name is stored the setplay name. The boolean variables represents whether the setplay is defined in relative coordinates and if it can be inverted. As example, suppose the right defender passes the ball to the right wing, if a setplay can be inverted it means

that there is no need to define a setplay where the left defender passes the ball to the left wing because that is already defined in the value of the boolean `isInvertible`.

The steps vector, contains the pointers to the steps that are part of that setplay. Each Step has an id, an integer value that represents the step number. The ball position is stored in each step as the players positions. A player is identified by a number or role name and has a position in the field. All steps also have a group of conditions that must be satisfied to enter that step. If a setplay is represented in a directed graph the edges are the actions executed, formerly represented as transitions. A Step can have multiple transitions.

A Transition is an action performed to move to another step, so it must has an identifier of the step it evolves the setplay to. To an action be performed it may have some conditions to be satisfied so a transition has a group of condition too.

6.2 Setplay Mode

The creation of setplays was the main object of the PlayMaker. The interface was drawn to this specific mode, further adapted to the formation mode. The purpose was to create a graphical way of defining setplays. The tool developed saves an enormous amount of time because a setplay file can be extremely big and when “hand-written” could take some time and probably have errors.

After initialize the tool in setplay mode, the next step is to create a new setplay, by accessing menu File and then clicking New. By this time the setplay has one step, the step 0. when a step is created, it is created empty of transitions and conditions, only the players are kept in the same positions. The reason to do so is to save time by avoid placing players in the same position as the previous step if it is desired to it to stand still, to avoid human errors a player may be forgotten and to simulate a continuous evolving of the play. When elaborating a play, the user's thought probably will be “at step 0 player X is here, and in the next step he will be here”, this is easier to identify if the previous position is kept when adding a step, it simulates with the user interaction the player's movement. The current step that is currently being edited is identified by its number, and the user can change step using the navigation buttons showed in Figure 6.3.

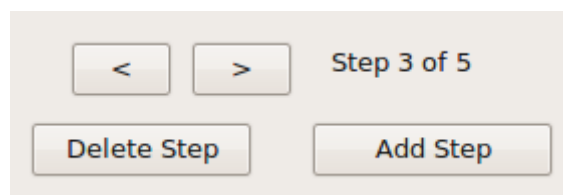


Figure 6.3: Step operation buttons

PlayMaker

Before adding any step, condition or transition to the setplay, it is possible to define whether the player's positions are going to be global or relative, by changing the state of the **Relative** check box. The difference on player placement will be described in a further section.

The **Invertible** check box when ticked identifies the current setplay as an inverted play. This means that if a play is defined in the left of the field, the same play can be executed on the right side.

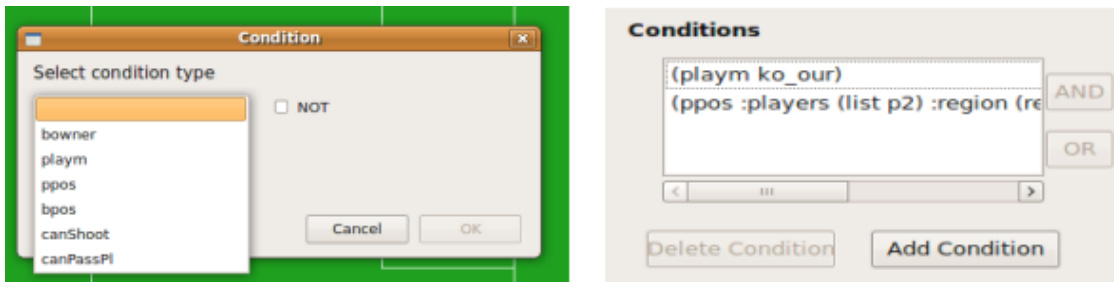


Figure 6.4: Conditions on PlayMaker

The conditions of the step are shown in a list in the condition section. To add a condition just press the **Add Condition** button and a panel will appear. The characteristics of this panel will be seen in detail in a further section. It is possible to delete a condition by selecting it from the box and pressing the **Delete Condition** button. The **AND** and **OR** buttons are used to join conditions by the selected operand.

A similar process can be done to add a transition to the step. The step panel will be detailed in a further section. The transition added is displayed in text in the transition section. The conditions of the transition are exactly like step's conditions, and the method to add and delete them is the same. To select to which step the play will evolve, it is used the combo box, it holds the values of all created steps and two special steps, **Finish** and **Abort**. The finish step indicates that the play was realized successfully and the abort step that for some reason, e.g. ball lost, the play needs to be aborted. It is also possible to select the transition to a new step, that will be created automatically. Each step can have several transitions, and the navigation through these transitions is also possible.

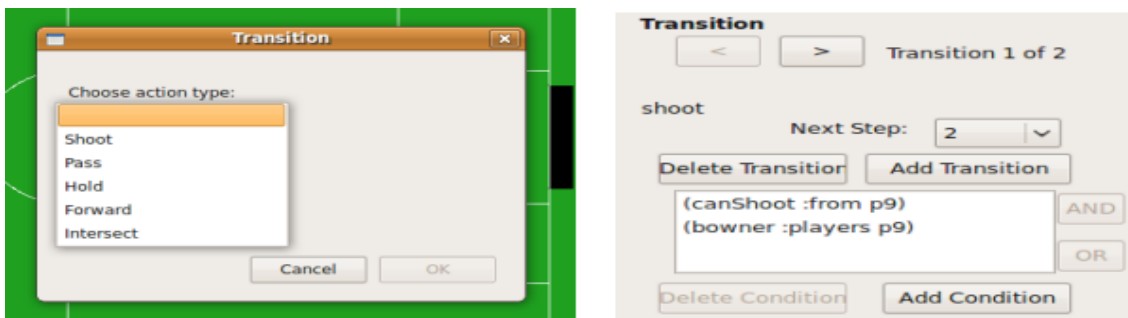


Figure 6.5: Transitions on PlayMaker

PlayMaker

The available **Add Role** button, opens a panel that allows the user to add a role. These roles, are identified by a name. After a role is added, it can be used in all setplay operations that involve players (Figure 6.6).

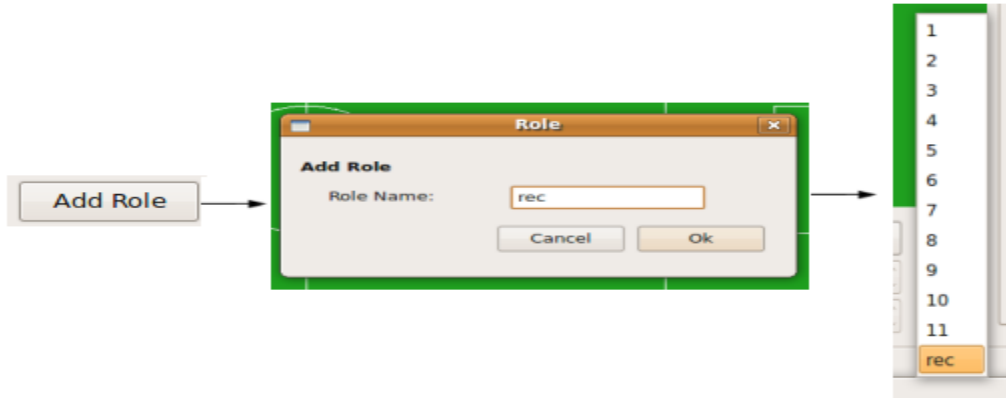


Figure 6.6: Roles in PlayMaker

There is also available a second tab named **Graph**, that presents the graph of the setplay providing the user with a global view of the setplay. The graph nodes are steps and the edges transitions.

6.2.1 Condition Panel

This is the panel shown when one of the **Add Condition** buttons is pressed (Figure 6.4 left). This panel is quite simple. The user just need to choose the condition type from the combo box filled with the keywords.

Bowner refers to the ball owner, the player who has the ball possession. The *playm* keyword identifies the play mode. The left keywords are paired up in terms of meaning, *ppos* is the player position and *bpos* is the ball position. The last two conditions available are the possibility to perform a pass to a player (*canPassPl*) or a shoot to goal (*canShoot*).

All the conditions need more information than the one provided by just choosing the type, i.e. it is needed to select the players that own the ball, can perform passes or shoots, the player to be in a region and that same region, and the desired play mode. As the condition type is changed combo boxes appear/disappear to be inserted the information necessary to the condition to be complete.

6.2.2 Transition Panel

The panel shown when the user wants to add a transition is displayed in Figure 6.5 left. There are actually four types of available action to add. Pass to a player (number or role), hold the ball, forward to a point or shoot to goal. Actions hold and shoot, does not need any further input from the user, the team will hold the ball in the best position avoiding adversary interception and shoot to the best point accordingly to the other team's goalkeeper.

The action pass, implies the definition of the player who will receive the ball, this is done by choosing his number or role name in the combo box available in the specification area when this action is chosen.

Other action that needs extra information is the forward action, it needs the point to which the ball will be forwarded. The input to this action is done by drawing a line in the field, finishing where is desired the ball to be forwarded.

6.2.3 Absolute vs Relative Positions

When the player's positions are absolute, meaning that do not depend on ball position, to add a player to a specific step of a setplay, after choosing that step, what the user needs to do is to select the player number or role from the combo box near the **Player** button. After the desired player is chose just press the **Player** button and click on the field to place the player in that specific position. This is a toggle button, so after placing the player it needs to be clicked again to disable the player placement mode. To edit a player position, the same actions needs to be performed.

On the other hand if the setplay is marked as relative, the ball is placed the same way the players are in the absolute coordinates setplays but the players' positions are defined by the value on the spin boxes below the **Player** button. The player must be selected the same way as if the positions were absolute, also the **Player** button must be pressed to enable the spin boxes. After defining the values that will separate the player from the ball, on exiting the player placement the player is drawn in the field. It is quite important to not forget to include the ball when the positions are relative.

It is easy do notice that, for a setplay defined with relative positions, to be successfully the ball must be in that exact position. To avoid this situation can be added an offset to the ball, this is interpreted as the ball area, quite useful if the user wants to define a setplay that can be executed no just in one occasion, e.g. a throw in, to be executed when the ball leaves in the opponent midfield. This can be done within an only setplay, the ball is placed in the middle of the opponent midfield and is defined the offset on the x axis of $\text{field_width}/4$.

6.2.4 Setplay file format

The setplay files have the suffix *.conf* and are written in *S-Expression*. In a global point of view, a setplay defined by its name, have a comment, initial parameters, the list of all players that are part of the setplay and an abort condition. Most important than the characteristics referred above is the sequence of steps. Each step has an id, a wait and an abort time, followed by a list of the players, the conditions that must be satisfied to enter that step and the lead player. Each step also has a list of transitions, characterized by the next step to go, the actions that must be taken (directives) and the conditions necessary to take that action. The full grammar for the formation of setplay files can be found in Appendix B.

6.2.5 Setplay Example

In Figure 6.7 and Figure 6.8 is presented a partial scenario of a simple setplay defined in PlayMaker. There are four participants in this setplay, one of them is the goalie. The player's positions in step 0 is depicted in Figure 6.7.

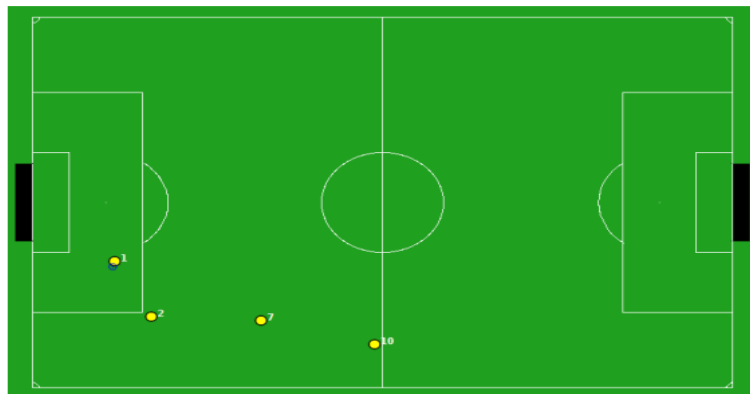


Figure 6.7: Players positions in setplay

The setplay is started with the condition (*playm gc_our*), i.e. when the goalie catches the ball. Then the goalie will perform a pass to player 2 if he thinks he can success in passing, as this transition has the condition (*canPassPl :from p1 :to p2*) associated. If the pass is performed, and player 2 receives the ball, the setplay is on step 1.

The conditions to enter this step are the play mode being play on and player 2 have the ball (*owner :players 7*). There are two transitions possible in this step, passing the ball to player 7 or passing the ball to a region near player 10. If the second transition is chosen the setplay finishes, on the other hand if the player decides to pass to player 7 and it receives the ball, the condition to enter step 2 is satisfied.

This last step only has one transition, a pass to a region near player 10, with the condition that the pass can be successfully performed.

PlayMaker

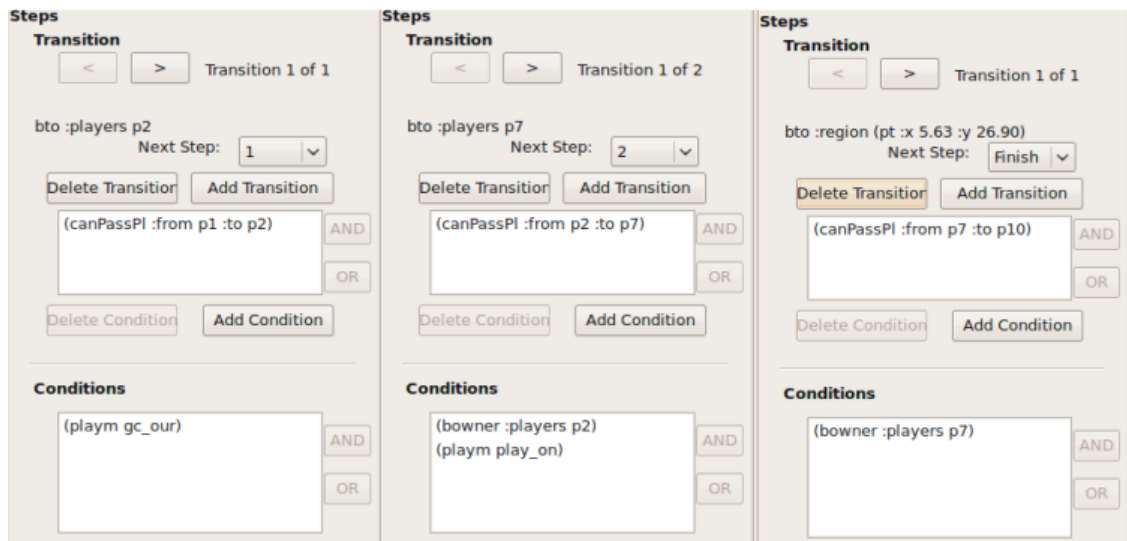


Figure 6.8: Steps in setplay

6.3 Formation Mode

The main objective of this mode is to provide a quick and easy way of defining formations so that they can be interpreted by FC Portugal's players when they are playing. One of the main advantages, regardless time saving, is the on-define visualization of the players position.

In this mode, the user can edit or create formations. Actually, when a formation is opened or created, the user is manipulating a “formation's book”, which can contain more than one formation. The tool provides full navigation through formations, and positionings inside a formation. The formation name is displayed and can be edited in the the text box between the formation's navigation buttons.

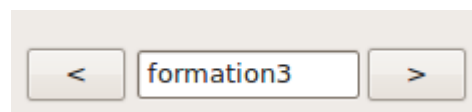


Figure 6.9: Navigation buttons and formation name

Near each player, is not only its number, but also its position since it is a important factor that the position is the desired and not approximately.

6.3.1 Creating/Editing a formation

After creating a formation, are automatically created five positionings. The first where the ball is placed in the centre of the field, and the players are displayed in FC Portugal's 4-3-3. The other four, the ball is placed in the four corners of the field, and the players position is calculated by players attraction to ball from the original positioning. In these five positionings, it is not possible to re-collocate the ball, only player placement is available, has they are five essential and need positionings to calculate where the players are to all other ball positions inside the field.

The **TestForm** button, lets the user place the ball an shows the players' position accordingly to place where the ball was placed. However this is just a test, it does not saves the current positioning to the formation. To do so, the user must select the desired formation and click the **Add Step** button. This will create an empty positioning. Then, use the **Ball** button to place the ball. This will add the positioning to the formation. After placing the ball it is possible to edit the players' position by clicking the **Player** button. This will enter the player edition mode and by clicking in the field the closest player to that point will be placed there.

The player's default positions when the ball is placed, exception to the first five positionings in each formation is calculated using a *Delaunay triangulation* and *linear interpolation* as proposed by [Akiyama and Noda, 2007], using librcsc-3.0.1 available at <http://rctools.sourceforge.jp/>. The Delaunay triangulation is used to find the triangle to which the new ball position is inserted. After is used the linear interpolation algorithm to calculate the player position. The triangles are displayed in the field so the user can choose more strategic points to the place the ball in the new formations and as a way of testing if the triangulation is being calculated correctly.

6.3.2 Delaunay Triangulation

The Delaunay triangulation is one way of triangulate a plan. Being \mathbf{P} a set of points, belonging to the same plan, $\mathbf{DT}(\mathbf{P})$ is the *Delaunay triangulation* of \mathbf{P} , if none of the points in \mathbf{P} is inside the circumcircle of the triangles in $\mathbf{DT}(\mathbf{P})$. A circumcircle of a triangle is a circle that intersects all the corners. This algorithm maximizes the minimum angle of all angles of the triangles. With a set of three or more point it is possible to find a unique *Delaunay triangulation*, with obvious exceptions to when there are four points in the same circumcircle, like the case of a rectangle where there are two possible triangulations. It was used an incremental algorithm, one of the fastest to calculate the *Delaunay triangulation*.

It starts by creating a large triangle, that involves all the points in set, with two new vertices and a vertex from the set. The algorithm adds the points one by one randomly, in this case by the order defined in the formation. Each time a point is added, it finds the triangle in the

triangulation that surrounds it, and edges from this point to the vertices of its surrounded triangle are added. If it falls on an edge, edges for the opposite edges of the two triangles are added, in Figure 6.10 are visible the two case scenarios when a vertex \mathbf{Pr} is added. Assuming that all edges are legal, now the only thin to do to have a *Delaunay triangulation* for the initial set is to remove the points added in the first step and their edges.

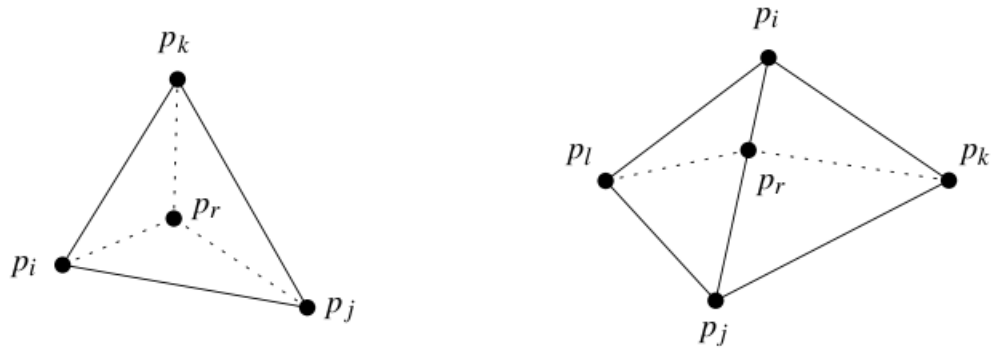


Figure 6.10: Vertex added inside and lying on an edge of a triangle [Kreveld et al. 2000]

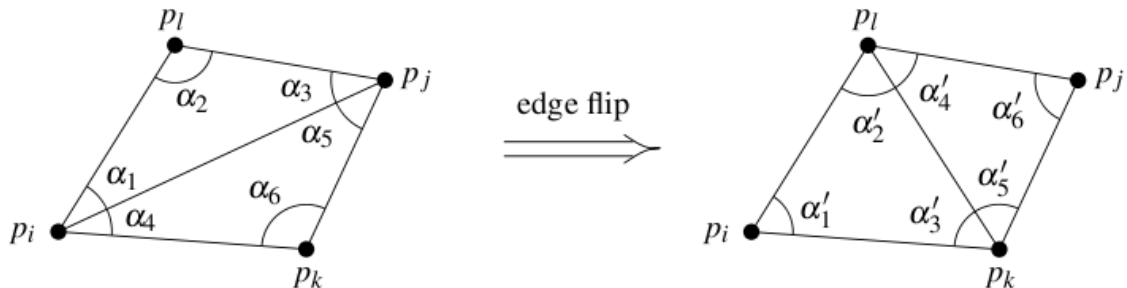


Figure 6.11: Edge Flip [Kreveld et al. 2000]

The addition of the new does not implies that we continue to have a *Delaunay triangulation*. It may be possible to increase the minimum angle of all triangles. This is called an illegal edge. When an illegal edge is added it needs to be flipped. Flipping an edge is changing the internal edge of a convex quadrilateral.

When all edges of the triangles are legal, it is not possible to increase the minimum angle, the triangulation is angle-optimal therefore a *Delaunay triangulation* [Kreveld et al. 2000].

There is a duality between the *Delaunay triangulation* and the *Voronoy diagram*. If the centre of the circumcircles of the triangulation are connected, it results in the Voronoy diagram for that set of points. In the context of formations, each point of the set used to calculate the triangulation is a ball position.

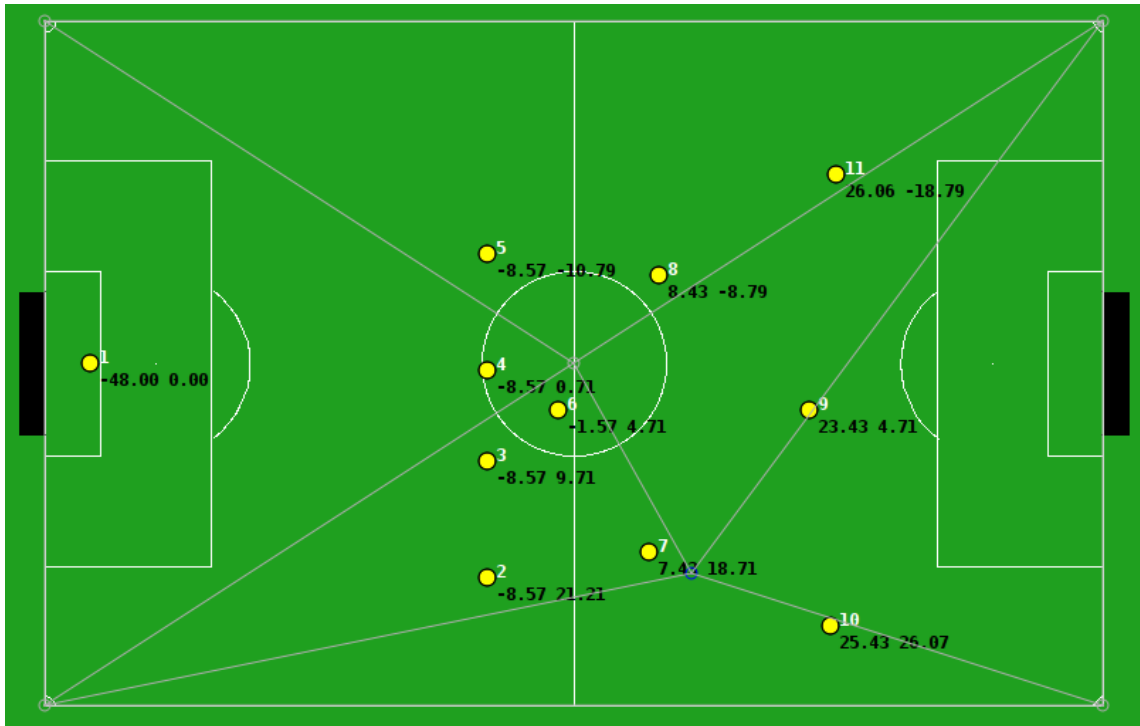


Figure 6.12: Screenshot of Playmaker - formation mode with respective triangulation

6.3.3 Linear Interpolation

The linear interpolation algorithm used is the same as the *Gouraud Shading* algorithm [GOURAUD, 2008] used in computer graphics domain to simulate the differing light and colour effects along a surface. Figure 6.14 presents a linear interpolation by *Gouraud Shading* algorithm.

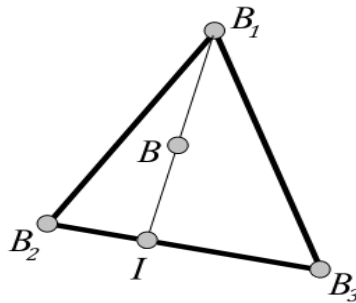


Figure 6.14: Gouraud Shading Algorithm.
Adapted from [Akiyama and Noda, 2007]

Within the context, the vertices of the triangle, **B1**, **B2**, **B3** are ball positions. **P(B1)**, **P(B2)** and **P(B3)** are respectively the position of a player when the ball is in the point. Point **B** is the new ball position, so algorithm will calculate the **P(B)**, this is the position of a player when the ball is in point **B**. The algorithm is as follows:

1. Calculate **I**, the intersection point between the line **B1B** and the segment **B2B3**.
2. Calculate interpolation of points **B1** and intersection point from 1, by

$$P(I) = P(B2) + (P(B3) - P(B2)) * \left(\frac{|\overrightarrow{B2I}|}{|\overrightarrow{B2I}| + |\overrightarrow{B3I}|} \right)$$

3. In addition, **P(B)** is calculated by,

$$P(B) = P(B1) + (P(I) - P(B1)) * \left(\frac{|\overrightarrow{B1B}|}{|\overrightarrow{B1B}| + |\overrightarrow{B1I}|} \right)$$

P(B) is the position of the player when the ball is at **B**.

6.3.4 Formation file format

When all the formations are as the user wants, they must be saved to a configuration file, so it can be read after by the team. The file must have the suffix “.conf”. It stores each formation's name, and all the positionings of the formation. The file format is as it follows:

```
#
# FORMATIONS FILE
#
$tf # Number of formations
$f1 # $f2
$tp # Number of points
# (ballx bally) (pl1x pl1y) (pl2x pl2y) (pl3x pl3y) (pl4x pl4y) (pl5x pl5y) (pl6x pl6y)
  (pl7x pl7y) (pl8x pl8y ) (pl9x pl9y) (pl10x pl10y) (pl11x pl11y)
$p ($b1 $b2) ($p1x $p1y) ($p2x $p2y) ($p3x $p3y) ($p4x $p4y) ($p5x $p5y) ($p6x
  $p6y) ($p7x $p7y) ($p8x $p8y) ($p9x $p9y) ($p10x $p10y) ($p11x $p11y)
```

Where \$tf is the number of formations in the file, \$f1 and \$f2 are the number and name of the formation which will be described below. \$tp is the number of positionings of the current formation and \$p is the number of the positioning. The other values are ball's position, \$ballx \$bally, and the player's number i position \$pix and \$piy.

6.3.5 Formation example

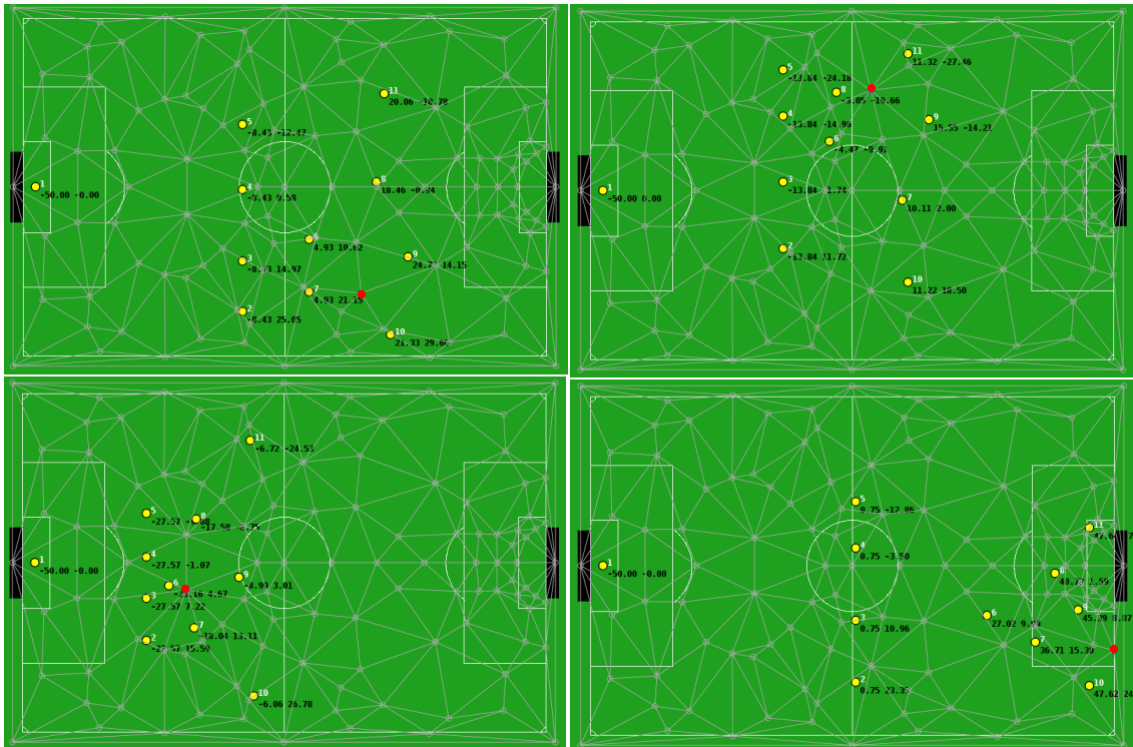


Figure 6.15: Four positionings of a defined formation

As it can be seen in Figure 6.15 PlayMaker enables to define very complex formations using a huge number of points and triangles to completely define the team's formation. Figure 6.15 displays the PlayMaker's field with the player's positions for four different ball positions. The ball was marked as a red dot to be easier identified. The triangulation is the grey lines drawn in the field and the corners of all triangles are all ball positions for this formation.

6.4 Conclusion

It was presented the tool and algorithms developed for creating advanced formations and setplays to be used in the context of FC Portugal or other robotic soccer teams in RoboCup competitions. The tool is in constant development and improvement and may be easily extended to accommodate new types of formations, setplays or other similar concepts. The graphical definition of setplays and formations is a great help, by looking at the formation and setplay files it is perfectly understandable that it is quite difficult and very plausible to have errors when defining them “*by hand*”.

PlayMaker

7 Results

In this year edition of the RoboCup, the FC Portugal team did not go further than the ninth place. Due to some bad luck in the seeding round and draw it had to play and was defeated by the WrightEagle team, the actual world champion, in a play-off match to access the quarter finals.

7.1 Formation

In order to evaluate the new positioning system compared to the old one, several matches were realized between two FC Portugal teams, one with each positioning systems. To avoid misunderstandings the team with the new positioning system will be called *FCP09* and the other *FCP*. It is visible in Figure 7.1 the differences for two ball positions between the old (above) and the new (below) formations.

It is important to focus the fact that on the matches played the goalie did not act as usual. It was deactivated, it only scanned the field, i.e. span around himself. This decision was taken so that it could not interfere with the ball and alienate the results, as the goal of these matches was to evaluate the positioning of the team. If the ball crossed the defensive line in the direction of the goal, probably there would be a goal, so this is a very effective manner of test the defensive positioning. The formation used in *FCP09* team was the one used in most of the matches in this year edition of RoboCup, a 4-3-3 based on the formation used by the Helios08 team. This formation has 115 defined points and can be found in Appendix C. The *FCP* team used the normal *SBSP*.

Results

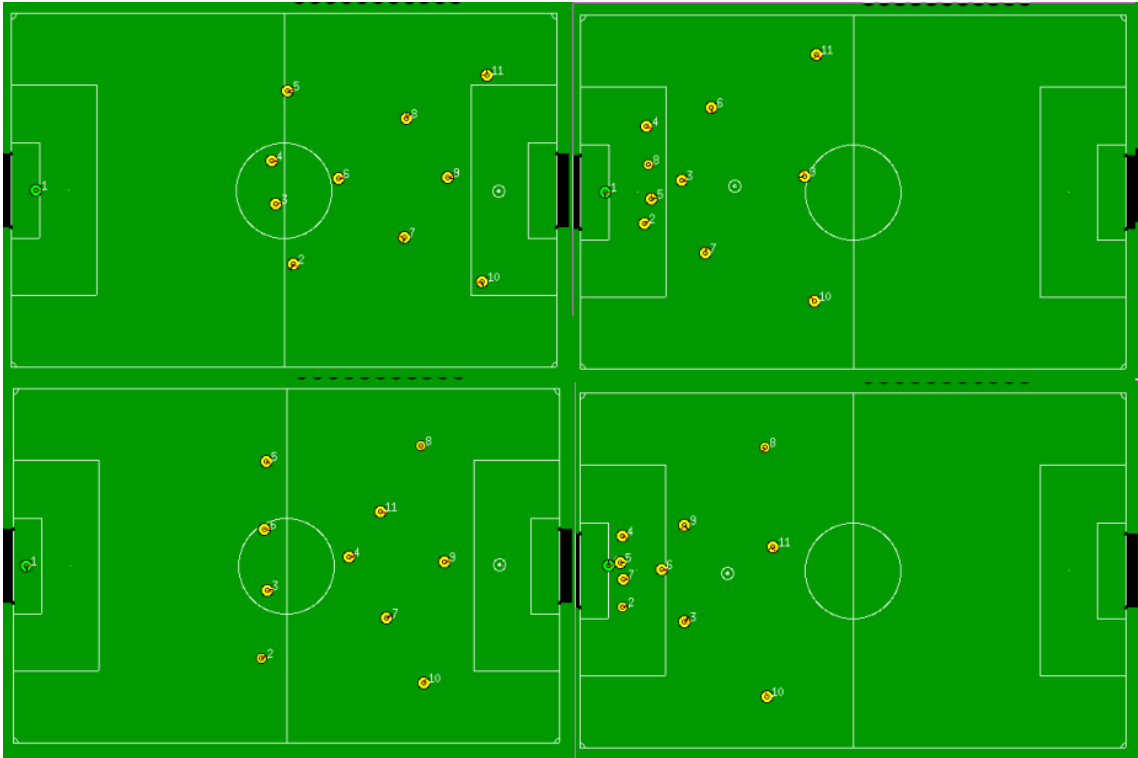


Figure 7.1: Difference between the two positionings for the old and new formation

A total of 117 games were played between these two teams. The result was 116 winnings and 1 defeat to the *FCP09* team, an impressive 99.15% of winnings.

Table 7.1: Matches analysis against ourselves

	<i>FCP09</i>	<i>FCP</i>
<i>Winnings</i>	116	1
<i>Draws</i>	0	0
<i>Defeats</i>	1	116
<i>Goals Scored</i>	602	99

The fact that the goalie played a passive role in these matches and may lead to the wrong thought that both teams would score in all matches. In Figure 7.2 it is clearly seen that in more than 50 matches *FCP* team did not score any goal. On the other hand *FCP09* scored in all matches. Also the number of goals scored by *FCP09* was approximately 6 times more than by the *FCP*. *FCP09* team scored on average 5.15 goals per match against the 0.85 of *FCP*.

Results

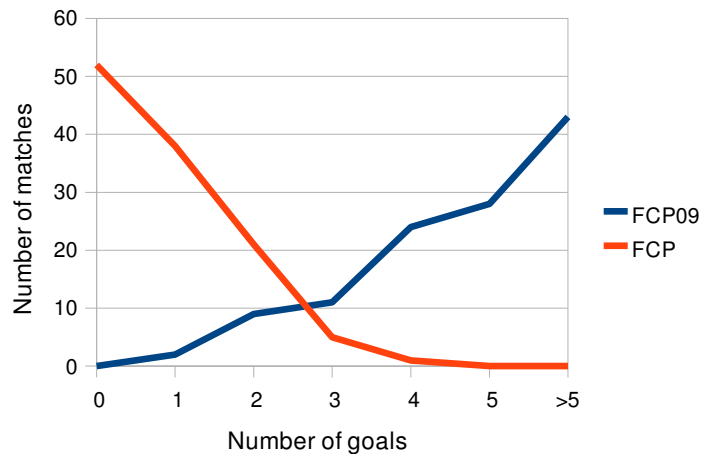


Figure 7.2: Goals scored

As the matches were between the two teams that are being compared, the goals scored by one team are the goals conceded by the other team, so all statistics presented above about scored goals, can be seen as conceded goals of the other team.

Another test conducted was a set of ten matches against the team *Yowai2008*. Both FC Portugal teams, as expected, won all the matches but the number of goals scored and conceded are different and can be used for comparison.

Table 7.2: Matches Analysis against *Yowai2008*

	<i>FCP09</i>	<i>FCP</i>
<i>Winnings</i>	10	10
<i>Draws</i>	0	0
<i>Defeats</i>	0	0
<i>Goals Scored</i>	79	61
<i>Goals Conceded</i>	0	5

In these 10 matches *FCP09* did not conceded any goal opposing the 5 of *FCP*. The number of scored goals is slightly higher for the *FCP09* also.

The huge discrepancy of values between the two teams in the final results of the matches and the number of goals scored/conceded, allyng to the difference in the performance against *Yowai2008* team seem enough to justify the usage of the new positioning system.

7.2 Setplays

To test if the team could perform the setplays defined by the PlayMaker two setplays were defined. With some limitations the team is actually able to follow the steps of the setplays defined. The setplays files can be found in Appendix D. The setplays were tested when playing against no one, as the objective is just to test the executions of setplays and not the performance and efficiency of the defined setplays.

7.2.1 Square Passing

This setplay has four participants placed in the corners of a square in the team midfield. Each player has two passing options, passing the ball to the player on one of the edges. This setplay has simple conditions like ball owning and possibility of passing. It does not have any transition to a finish state, it is supposed to keep running.

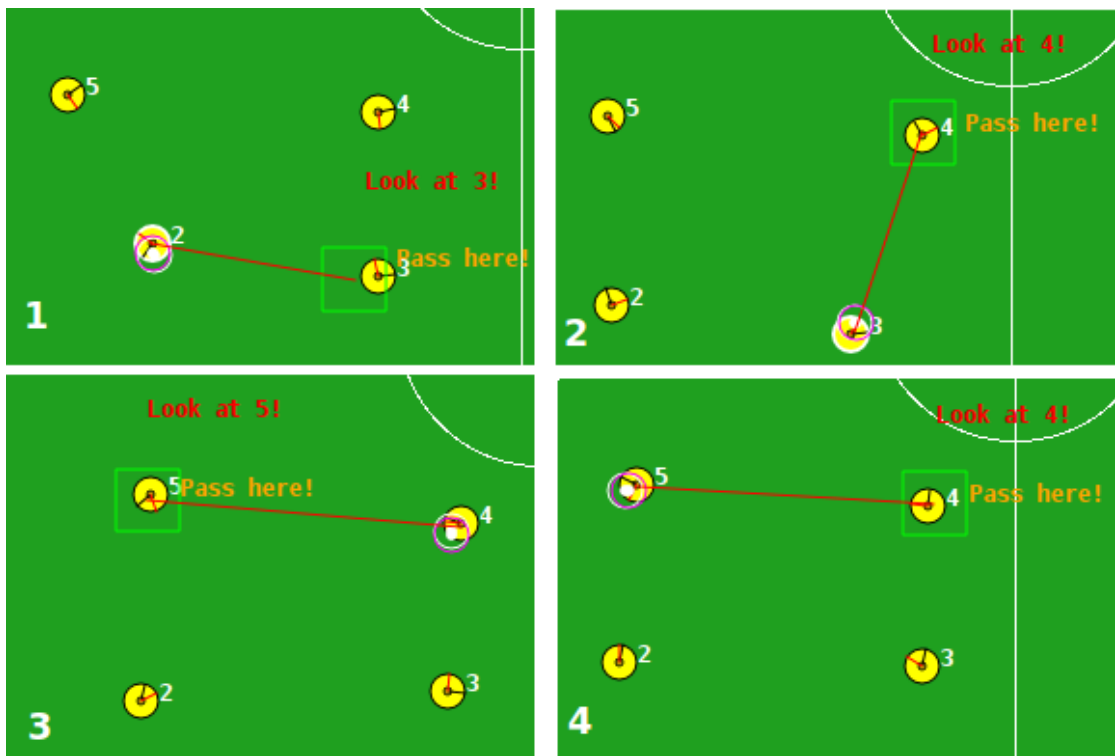


Figure 7.3: Square Passing Setplay

7.2.2 Simple Corner

The other setplay used as example is a corner. The corner taker passes the ball to a player near him (1). After intercept the ball, the new ball owner and leader of the setplay passes to a player near the penalty marker (2). The last participant just shoots at goal (3).

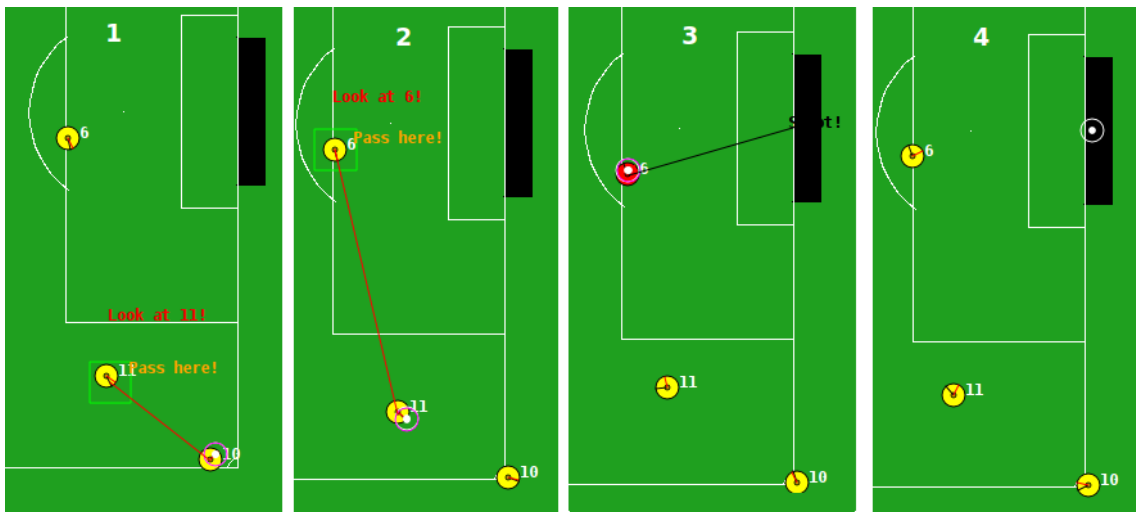


Figure 7.4: Simple Corner Setplay

7.3 Conclusion

This chapter presented a simple evaluation of the tool developed. From the evaluation performed it is simple to conclude that PlayMaker enables to define advanced formations that may hugely benefit the team performance in competitions. The tool also enables to define complex setplays that, if correctly executed by the players may improve the team's offensive play.

Results

8 Conclusion

This thesis presented a software tool that allowed the definition of setplays and formations in a graphical way. From the practical analysis performed it was concluded that to define a good formation to a competitive team like FC Portugal, around a hundred points are needed. Thus, to define by hand a formation that has half of these points it is completely unthinkable by obvious reasons: time and errors. Graphical definition of formations is the answer to rapidly define robust and effective formations.

The definition of setplays is even a more obvious case of the need of a graphical tool due to the complexity of the grammar that supports this theory. Defining a setplay with more than 4 steps can be pretty hard and the probability of the presence of errors is high. The tool developed solves these problems and it is easier to define as the user can visually observe the evolution of the definition. It is much more intuitive and perceptible than to look at a text file.

A single experiment was performed, when defining the setplay examples. Two persons, one using the PlayMaker tool and the other, who was familiar with the setplay's defining grammar and S-expression syntax, using a text editor. As expected it was a lot faster to define both example setplays in PlayMaker than just writing the file.

Another great advantage of a tool like this is that motivates the interactions of AI experts with sports, specifically soccer, experts. If a team could have, experts in soccer area defining their formations and setplays, it would be a great improvement in the quality of the game. This is possible because its user friendly interface and since it is based on real sports software it should not be strange to them.

The new positioning system seems better than the previous. By analysing the results of the matches against ourselves and against the Yowai team it is perfectly visible that the team with the new positioning system has a better performance, by comparing the number of winnings and goals.

8.1 Future Work

Although the PlayMaker interface is user-friendly, there are some aspects that could be improved to become more intuitive, like use drag-and-drop to place and move the players and the ball around the field. A way of reduce the time of creation of setplays would be to user the graph view to edit the setplay, also by dragging edges and nodes, this would be quite useful to create setplays with high number of transitions, has by looking at the graph the user has a more global view of all the setplay.

PlayMaker only writes files in S-Expression, it could be expanded to other languages that could support data in this kind of hierarchy, an example is XML. The tool was designed having has domain of application the 2D simulation league, in the future it should be able to create setplays and formations to any other RoboCup soccer league.

References

[Akiyama and Noda, 2007] Akiyama, H. and Noda, I., Multi-Agent Positioning Mechanism in a Dynamic Environment, In RoboCup 2007: Robot Soccer World Cup XI, pp. 377-384, Springer Berlin, 2007

[ARENAS, 2009] <http://robotarenas.nist.gov/>, accessed on June 2009.

[Bowling et al., 2003] Bowling M., Browning B., Chang A. and Veloso M.: Plays as Team Plans for Coordination and Adaptation. RoboCup 2003: Robot Soccer World Cup VII. Springer. Volume 3020. 2004. (p. 686-693)

[Bowling et al., 2004] Bowling M., Browning B., and Veloso M.: Plays as effective multiagent plans enabling opponent-adaptive play selection. Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'04). 2004.

[CHALLENGES, 2009] Technical Challenges for the RoboCup 2009 Standard Platform League Competition. 2009.

[Chen et al., 2003] Chen, M., Dorer, K., Foroughi, E., Heintz, F., Huang, Z., Kapetanakis, S., Kostiadis, K., Kummeneje, J., Murray, J., Noda, I., Obst, O., Riley, P., Steffens, T., Wang, Y. and Yin, X., RoboCup Soccer Server 2003 for Soccer Server Version 7.07 and later. 2003.

[Dechter et al., 1991] Dechter, R.; Meiri, I.; and Pearl, J. Temporal constraint networks. Artificial Intelligence 49:61–95. 1991.

[DeepBlue, 1997] <http://www.research.ibm.com/deepblue/>, accessed on June 2009.

[Dunning, 1999] Dunning, E., Sport Matters: Sociological Studies of Sport, Violence and Civilisation, ISBN: 0415093783, pp. 103-103. Routledge-London. 1999.

References

- [GOURAUD, 2008] Gouraud, H. Continuous shading of curved surfaces. In Rosalee Wolfe, editor, *Seminal Graphics: Pioneering efforts that shaped the field*. ACM Press, 1998.
- [HOME, 2009] <http://www.ai.rug.nl/robocupathome/>, accessed on June 2009.
- [HUMANOID, 2009] RoboCup Soccer Humanoid League Rules and Setup for the 2009 competition in Graz, Austria . 2009.
- [IEETA, 2009] <http://www.ieeta.pt/robocup/>, accessed on June 2009.
- [IRAVANIAN, 2009] <http://sina.iravanian.googlepages.com/catut.html>, accessed on June 2009.
- [Kitano et al., 1995] Kitano, H.; Asada, M. ; Kuniyoshi, Y.; Noda, I.; Osawa E., *Robocup: The Robot World Cup Initiative*. Proceedings of IJCAI' 95 Workshop on Entertainment and AI/Alife, pp. 19-24, 1995
- [Kitano, 1997] Kitano, H., *RoboCup: The Robot World Cup Initiative*, Proceedings of the 1st International Conference on Autonomous Agent (Agents' 97), Marina del Ray, The ACM Press, 1997.
- [Kok and Vlassis, 2006] Kok, J. R. and Vlassis, N., *UvA Trilearn Team Description*, in *RoboCup 2005: Robot Soccer World Cup IX*, 2006.
- [Kok et al., 2003] Kok, J., Spaan, M. and Vlassis, N., *Multi-robot decision making using coordination graphs* in Proceedings of the 11th International Conference on Advanced Robotics, Coimbra, Portugal, 2003.
- [Kreveld et al. 2000] Kreveld, M., Berg, M., Schwarzkopf, O. and Overmars. M., *Computational Geometry: Algorithms and Applications*. Springer Verlag, second edition, chapter 9. 2000.
- [Lau and Reis, 2007] Lau, N. and Reis, L. *FC Portugal - High-level Coordination Methodologies in Soccer Robotics*, *Robotic Soccer*, Book edited by Pedro Lima, Itech Education and Publishing, Vienna, Austria, pp. 167-192, December 2007, ISBN 978-3-902613-21-9.
- [Lau et al., 2005] Lau, N., Reis, L. and Reinaldo, F.: *FC Portugal 2005 Rescue Team Description: Adapting Simulated Soccer Coordination Methodologies to the Search and Rescue Domain*, CD Proceedings of the RoboCup International Symposium, 2005.
- [Lau et al., 2007a] Lau, N., Reis, L., and Certo, J., *Multi-Level, Functional, Spatial and Temporal Agent's Reasoning Debugging*, In Neves J, Santos M.F., Machado J.M. (eds.), *New Trends in Artificial Intelligence*, Proceedings of the 13th Portuguese Conference on Artificial Intelligence, EPIA 2007, Guimarães, Portugal, December 3-6 2007, pp. 716-726, APPIA, ISBN 978-9899561809.
- [Lau et al., 2007b] Lau, N., Reis, L., and Certo, J., "Understanding Dynamic Agent's Reasoning", In Neves J, Santos M.F., Machado J.M. (eds.), *Progress in Artificial Intelligence*, 13th Portuguese Conference on Artificial Intelligence, EPIA 2007, Guimarães, Portugal,

References

- December 3-6, 2007, Springer LCNS, Vol. 4874, pp. 542-551, 2007, ISBN: 978-3-540-77000-8.
- [McMillen and Veloso, 2006] McMillen, C. and Veloso, M., Distributed, Play-Based Role Assignment for Robot Teams in Dynamic Environments. Distributed Autonomous Robotic Systems 7, pp. 145-154. Springer Japan. 2006.
- [MIDDLE, 2009] Middle Size Robot League Rules and Regulations for 2009 Version - 13.1 20081212. 2009
- [Miyazawa et al. 2002] Miyazawa, S., Nakayama, K. and Takeuchi, I. SoccerScope3, In Team Description Paper of RoboCup 2002: Robot Soccer World Cup VI. 2002.
- [Mota and Reis, 2007] Mota, L. and Reis, L. P., Setplays: Achieving Coordination by the appropriate Use of arbitrary Pre-defined Flexible Plans and inter-robot Communication, RoboComm 2007 - First International Conference on Robot Communication and Coordination, Athens, Greece, October 15-17, 2007.
- [NAO, 2009] RoboCup Standard Platform League (Nao) Rule Book. 2009.
- [Noda et al., 1998] Noda, I., M., H., Hiraki, K. and Frank, I., Soccer Server: A Tool for Research on Multiagent Systems. Applied Artificial Intelligence, Vol. 12, pp.233-250, 1998
- [RC2004, 2009] <http://www.robocup2004.pt/robocupJunior>, accessed on June 2009.
- [RC2009, 2009] <http://www.robocup2009.org/212-0-general>, accessed on June 2009.
- [Reis and Lau, 2001a] Reis, L. P. and Lau, N., FC Portugal Team Description: RoboCup 2000 Simulation League Champion, in Peter Stone, Tucker Balch and Gerhard Kraetzschmar, editors, RoboCup-2000: Robot Soccer World Cup IV, Springer Verlag Lecture Notes in Artificial Intelligence, Vol. 2019, pp.29-40, Berlin, 2001.
- [Reis and Lau, 2001b] Reis, L. P. and Lau, N., COACH UNILANG – A Standard Language for Coaching a (Robo) Soccer Team, in Andreas Birk, Silvia Coradeschi and Satoshi Tadokoro, editors, RoboCup-2001: Robot Soccer World Cup V, Springer Verlag LNAI, Vol. 2377, pp. 183-192, Berlin, 2002, ISBN 3-540-43912-9.
- [Reis et al., 2001] Reis, L., Lau, N. and Oliveira, E., Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents in Markus Hannebauer, Jan Wendler and Enrico Pagello Editors, Balancing Reactivity and Social Deliberation in Multi-Agent System – From RoboCup to Real-World Applications, Springer LNAI, Vol. 2103, pp. 175-197, Berlin, 2001, ISBN 3-540-42327-3.
- [Reis, 2002] Reis, L. P., Coordenação de Agentes Cooperativos e Trabalho de Equipa, LIACC (NIAD&R) – Research Report, Maio de 2002 .
- [Reis, 2003] Reis, L. P., Coordination in Multi-Agent Systems: Applications in University Management and Robotic Soccer, University of Porto, Porto, PhD Thesis, 2003.

References

- [RESCUE, 2009] <http://www.robocuprescue.org/>, accessed on June 2009.
- [Riley and Veloso, 2002] Riley, P., and Veloso, M. Planning for distributed execution through use of probabilistic opponent models. In ICAPS-02. 2002.
- [ROBOCUP, 2009] www.robocup.org, accessed on June 2009.
- [RSYSTEM, 2009] <http://www.rescuesystem.org/>, accessed on June 2009.
- [Schiffer et al., 2006] Schiffer S., Ferrein A. and Lakemeyer G.: Qualitative World Models for Soccer Robots. Conference Paper - KI 2006 WS on Qualitative Constraint Calculi. Knowledge-based Systems Group, RWTH Aachen. 2006.
- [SSERVER, 2009] <http://sserver.wiki.sourceforge.net/>, accessed on June 2009.
- [SSRL, 2009] <http://small-size.informatik.uni-bremen.de/>, accessed June 2009.
- [Stone and Veloso, 1999] Stone, P. and Veloso, M., Task Decomposition, Dynamic Role Assignment and Low-Bandwidth Communication for Real-Time Strategic Teamwork. Artificial Intelligence, Vol. 110 (2), pp. 241-273, June of 1999.
- [Stone et al., 2000] P. Stone, P. Riley, and M. Veloso. The CMUnited-99 champion simulator team. In Veloso, Pagello, and Kitano, editors, RoboCup-99: Robot Soccer World Cup III. Springer, 2000.
- [Stone, 1998] Stone, P., Layered Learning in Multi-Agent System, PhD thesis, School of Computer Science, Carnegie Mellon University, December of 1998.
- [Stoye and Elfers, 2007] Stoye, K. and Elfers, C., Intuitive Plan Construction and Adaptive Plan Selection, In RoboCup 2007: Robot Soccer World Cup XI, pp. 278-285, Springer Berlin. 2007.
- [TZI, 2009] <http://www.tzi.de/>, accessed on June 2009.
- [Zareian et al., 2003] Zareian, A., Samimi, R., Shiva, F., Hedayati, A., Kazemi, V., Nazemi, E. and Adibi, J., Team Assistant 2003, In Team Description Paper of RoboCup 2003: Robot Soccer World Cup VII. 2003.

Appendix A: Competitions

A.1 Winners of 2D Simulation League RoboCup World Championship

- 1996 – Osaka, Japan – Ogalets
- 1997 – Nagoya, Japan – AT Humboldt
- 1998 – Paris, France – CMUnited98
- 1999 – Stockholm, Sweden – CMUnited99
- 2000 – Melbourne, Australia – FC Portugal
- 2001 – Seattle, USA – Tsinghuaeolus
- 2002 – Fukuoka/Busan, Japan - Tsinghuaeolus
- 2003 – Padova, Italy – Uva Trilearn03
- 2004 – Lisbon, Portugal - STEP
- 2005 – Osaka, Japan – Brainstormers2D
- 2006 – Bremen, Germany – WrightEagle
- 2007 – Atlanta, USA - Brainstormers
- 2008 – Suzhou, China – Brainstormers
- 2009 – Graz, Austria - WrightEagle

Appendix A: Competitions

Appendix B: Setplay Grammar

Differences from CLang# - All references to players are done by PLAYER_REFERENCE

- A new region type was added: NAMED_REGION

- New conditions: CLEAR_SHOT_AT_GOAL, CLEAR_PASS_TO

- All operators (+, -, *, /, ==, !=, <, <=, >, >=) are prefix

- Arguments have added labels (like :label)

- All lists are built through the s-expression (list <args>)

Syntax: (from the definition of regular expression, in Wikipedia)

?: 0 or 1

*: any amount

+: 1 or more

<SETPLAY>: (setplay <PARAMETER_DEFINITION_LIST>?

<PLAYER_REFERENCE_DEFINITION_LIST> <STEP_LIST>)

<PARAMETER_DEFINITION_LIST>: :parameters (list <PARAMETER_DEFINITION>+)

<PARAMETER_DEFINITION>: (parameter :name <VAR> :type <TYPE>)

<TYPE>: integer | decimal | region | point

Appendix B: Setplay Grammar

<PLAYER_REFERENCE_DEFINITION_LIST>:
:players (list <PLAYER_REFERENCE_DEFINITION>+)

<PLAYER_REFERENCE_DEFINITION>: (playerRole :name <VAR>)
| (player :team <TEAM> :number <PLAYER_NUM>)

<TEAM>: our | opp

<PLAYER_NUM>: [0-9] | 10 | 11

<STEP>: (step :order <NON_NEG_INTEGER>
:waitTime <NON_NEG_INTEGER>
:abortTime <NON_NEG_INTEGER>
:participants <PLAYER_REFERENCE_LIST>
:condition <CONDITION>
:nextSteps <NEXT_STEP_LIST>)

<STEP_LIST>: (list <STEP>+)

<TRANSITION_LIST>: (list <TRANSITION>+)
| <TRANSITION>

<TRANSITION>: <NEXT_STEP> | <FINISH> | <ABORT>

<NEXT_STEP>: (nextStep :nextStepNumber <NON_NEG_INTEGER>
:condition <CONDITION>
:directives <DIRECTIVE_LIST>)

< FINISH >: (finish :condition <CONDITION>
:directives <DIRECTIVE_LIST>)

< ABORT >: (abort :condition <CONDITION>
:directives <DIRECTIVE_LIST>)

Appendix B: Setplay Grammar

<ACTION> : (pos :region <REGION>)
| (bto :region <REGION>)
| (bto :players <PLAYER_REFERENCE_LIST>)
| (mark :players <PLAYER_REFERENCE_LIST>)
| (markl :players <PLAYER_REFERENCE_LIST>)
| (markl :region <REGION>)
| (markGoal)
| (oline :region <REGION>)
| (clear :region <REGION>)
| (hold)
| (dribble :region <REGION>)
| (shoot)
| (tackle :players <PLAYER_REFERENCE_LIST>)
| (intercept)
| (stop)
| (attentionTo :region <REGION>)
| (attentionTo :object <OBJECT_LIST>)
| (htype <INTEGER>)
| (seq <ACTION>+)

<ACTION_LIST> : (list <ACTION>+) | <ACTION>

<OBJECT_LIST>: (list <OBJECT>+) | <OBJECT>

<OBJECT>: <STATIC_OBJECT> | <MOBILE_OBJECT>

<STATIC_OBJECT>: <POST> | <FLAG> | ...

<MOBILE_OBJECT>: <BALL> | <PLAYER_REFERENCE>

<CONDITION> : (true)
| (false)
| (ppos :players <PLAYER_REFERENCE_LIST> :min <INTEGER>

Appendix B: Setplay Grammar

```
:max <INTEGER> :region <REGION>)
| (bpos :region <REGION>)
| (bowner :players <PLAYER_REFERENCE_LIST>)
| (playm <PLAY_MODE>)
| (canShoot :from <PLAYER_REFERENCE_LIST>)
| (canPassPl :from <PLAYER_REFERENCE_LIST>
      :to <PLAYER_REFERENCE_LIST>)
| (canPassReg :from <PLAYER_REFERENCE_LIST>
      :to <REGION>)
| (and <CONDITION_LIST>)
| (or <CONDITION_LIST>)
| (not <CONDITION>)
| (<COND_COMP>)

<COND_COMP> : <TIME_COMP>
      | <OPP_GOAL_COMP>
      | <OUR_GOAL_COMP>
      | <GOAL_DIFF_COMP>

<TIME_COMP> : (time <COMP> <INTEGER>)

<OPP_GOAL_COMP> : (opp_goals <COMP> <INTEGER>)

<OUR_GOAL_COMP> : (our_goals <COMP> <INTEGER>)
      | <COMP> <INTEGER> our_goals

<GOAL_DIFF_COMP> : (goal_diff <COMP> <INTEGER>)

<COMP> : < | <= | == | != | >= | >

<PLAY_MODE> : bko | time_over | play_on | ko_our | ko_opp
      | ki_our | ki_opp | fk_our | fk_opp
      | ck_our | ck_opp | gk_opp | gk_our
      | gc_our | gc_opp | ag_opp | ag_our
```

Appendix B: Setplay Grammar

<CONDITION_LIST> : (list <CONDITION>+)

<DIRECTIVE> :

(do | dont :players <PLAYER_REFERENCE_LIST>

:actions <ACTION_LIST>)

<DIRECTIVE_LIST> : (list <DIRECTIVE>+)

<REGION> : <VAR>

| (regVar :name <CLANG_STR>)

| (regVar :name <CLANG_STR> :value <REGION>)

| <POINT>

| (regNamed :name <REGION_NAME>)

| (arc :center <POINT> :radius_small <REAL> :radius_large <REAL>

:angle_begin <REAL> :angle_span <REAL>)

| (triang :points (list <POINT> <POINT> <POINT>))

| (rec :points (list <POINT> <POINT>))

<REGION_NAME>: field | outside | our_middle_field | their_middle_field | our_leftwing |

our_middle | our_rightwing | our_penalty_box | our_leftwing_back |

our_leftwing_middle | our_leftwing_front | our_rightwing_back |

our_rightwing_middle | our_rightwing_front | our_middle_back | our_middle_front |

our_middle_back_right | our_middle_back_left | our_middle_front_right |

our_middle_front_left | our_left_box | our_center_box | our_goalie_area |

our_right_box | our_goal_left | our_goal_center | our_goal_right | defensive_third |

middle_third | attacking_third | their_leftwing | their_middle | their_rightwing |

their_penalty_box | their_leftwing_back | their_leftwing_middle |

their_leftwing_front | their_rightwing_back | their_rightwing_middle |

their_rightwing_front | their_middle_back | their_middle_front |

their_middle_back_right | their_middle_back_left | their_middle_front_right |

their_middle_front_left | their_left_box | their_center_box | their_goalie_area |

their_right_box | their_goal_left | their_goal_center | their_goal_right

Appendix B: Setplay Grammar

<POINT> : <VAR>

l (ptVar :name <CLANG_STR>)
l (ptVar :name <CLANG_STR> :value <POINT>)
l (pt :x <REAL> :y <REAL>)
l (pt ball)
l (pt :player <PLAYER_REFERENCE>)
l (ptRel :x <REAL> :y <REAL> :pt <POINT>)
l (<POINT_ARITH>)

<POINT_ARITH> : (<OP> <POINT> <POINT>)

<OP> : + | - | * | /

<REGION_LIST> : (list <REGION>+)

<PLAYER_REFERENCE_LIST> : (list <PLAYER_REFERENCE>+)

<PLAYER_REFERENCE> : <VAR>

l (player :team <TEAM> :number <PLAYER_NUM>)

<CLANG_STR> : "[0-9A-Za-z\(\)\.\+\-*\V?\<\>_]+"

<VAR> : [A-Z][a-zA-Z0-9_]*

Appendix C: Formation

```
#
# FORMATIONS FILE
#
1 # Number of formations
1 # heli433
115 # Number of points
# (ballx bally) (pl1x pl1y) (pl2x pl2y) (pl3x pl3y) (pl4x pl4y) (pl5x pl5y) (pl6x pl6y)
(pl7x pl7y) (pl8x pl8y) (pl9x pl9y) (pl10x pl10y) (pl11x pl11y)
1 ( 54.50 -36.00) (-50.00 0.00) ( 10.00 8.00) (-0.84 1.08) (-0.72 -12.00) ( 4.90 -27.30) ( 27.43
-16.50) ( 38.22 -3.50) ( 33.12 -27.00) ( 46.28 -14.00) ( 46.00 6.80) ( 44.22 -30.85)
2 ( 54.50 36.00) (-50.00 -0.00) ( 4.90 27.30) (-0.72 12.00) (-0.84 -1.08) ( 10.00 -8.00) ( 27.43
16.50) ( 33.12 27.00) ( 38.22 3.50) ( 46.28 14.00) ( 44.22 30.85) ( 46.00 -6.80)
3 ( 0.00 0.00) (-50.00 0.00) (-11.56 15.78) (-15.53 5.42) (-15.53 -5.42) (-11.56 -15.78) (-6.73
-1.87) ( 2.83 10.81) ( 2.83 -10.81) ( 9.41 -3.12) ( 9.30 23.78) ( 9.30 -23.78)
4 ( 54.50 0.00) (-50.00 0.00) ( 7.21 18.58) ( 2.74 6.07) ( 2.74 -6.07) ( 7.21 -18.58) ( 26.86 -3.20)
( 40.73 3.77) ( 40.73 -3.77) ( 45.60 -1.65) ( 48.97 9.82) ( 48.97 -9.82)
5 ( 36.57 -12.09) (-50.00 0.00) ( 7.36 15.67) ( 0.52 3.20) (-1.25 -9.96) ( 5.09 -23.21) ( 18.98
-7.30) ( 32.40 1.86) ( 27.73 -14.57) ( 38.10 -8.45) ( 42.41 10.78) ( 39.17 -19.85)
6 ( 36.57 12.09) (-50.00 -0.00) ( 5.09 23.21) (-1.25 9.96) ( 0.52 -3.20) ( 7.36 -15.67) ( 18.98
7.30) ( 27.73 14.57) ( 32.40 -1.86) ( 38.10 8.45) ( 39.17 19.85) ( 42.41 -10.78)
7 ( 48.51 -15.92) (-50.00 0.00) ( 12.83 13.63) ( 3.07 3.38) ( 0.51 -10.77) ( 2.27 -23.46) ( 25.20
-9.33) ( 39.34 -1.01) ( 34.83 -14.95) ( 43.86 -8.86) ( 46.88 6.76) ( 45.03 -20.81)
8 ( 48.51 15.92) (-50.00 -0.00) ( 2.27 23.46) ( 0.51 10.77) ( 3.07 -3.38) ( 12.83 -13.63) ( 25.20
9.33) ( 34.83 14.95) ( 39.34 1.01) ( 43.86 8.86) ( 45.03 20.81) ( 46.88 -6.76)
9 ( 42.76 0.00) (-50.00 0.00) ( 6.85 19.24) ( 0.98 5.97) ( 0.98 -5.97) ( 6.85 -19.24) ( 21.77 -2.76)
( 35.57 5.84) ( 35.57 -5.84) ( 40.72 -2.36) ( 44.37 9.81) ( 44.37 -9.81)
10 ( 48.66 -5.01) (-50.00 0.00) ( 8.38 17.49) ( 2.33 5.02) ( 1.54 -7.25) ( 6.06 -20.44) ( 24.70
-4.89) ( 39.20 2.37) ( 37.70 -7.40) ( 43.57 -4.31) ( 46.80 0.42) ( 46.09 -15.60)
11 ( 48.66 5.01) (-50.00 -0.00) ( 6.06 20.44) ( 1.54 7.25) ( 2.33 -5.02) ( 8.38 -17.49) ( 24.70
4.89) ( 37.70 7.40) ( 39.20 -2.37) ( 43.57 4.31) ( 46.09 15.60) ( 46.80 -0.42)
```

Appendix C: Formation

12 (50.57 -6.78) (-50.00 0.00) (9.23 16.82) (2.75 4.77) (1.66 -7.71) (5.36 -20.80) (25.62 -5.67) (40.06 1.43) (38.20 -8.13) (44.40 -4.93) (47.39 -0.36) (47.00 -17.67)
13 (50.57 6.78) (-50.00 -0.00) (5.36 20.80) (1.66 7.71) (2.75 -4.77) (9.23 -16.82) (25.62 5.67) (38.20 8.13) (40.06 -1.43) (44.40 4.93) (47.00 17.67) (47.39 0.36)
14 (52.49 -17.10) (-50.00 0.00) (14.73 12.86) (3.74 3.50) (0.95 -10.96) (0.75 -23.35) (27.02 -9.99) (40.72 -1.59) (36.71 -15.30) (45.39 -8.87) (47.64 7.59) (47.62 -24.13)
15 (52.49 17.10) (-50.00 -0.00) (0.75 23.35) (0.95 10.96) (3.74 -3.50) (14.73 -12.86) (27.02 9.99) (36.71 15.30) (40.72 1.59) (45.39 8.87) (47.62 24.13) (47.64 -7.59)
16 (52.49 -7.96) (-50.00 0.00) (10.02 16.31) (3.10 4.66) (1.82 -7.99) (4.65 -20.95) (26.49 -6.22) (40.75 0.80) (38.78 -8.58) (45.16 -5.26) (47.99 0.19) (48.07 -19.82)
17 (52.49 7.96) (-50.00 -0.00) (4.65 20.95) (1.82 7.99) (3.10 -4.66) (10.02 -16.31) (26.49 6.22) (38.78 8.58) (40.75 -0.80) (45.16 5.26) (48.07 19.82) (47.99 -0.19)
18 (49.25 -9.29) (-50.00 0.00) (9.92 16.07) (2.73 4.33) (1.25 -8.48) (4.73 -21.62) (25.18 -6.62) (39.69 0.74) (37.01 -10.02) (43.96 -6.11) (47.11 1.01) (46.18 -17.93)
19 (49.25 9.29) (-50.00 -0.00) (4.73 21.62) (1.25 8.48) (2.73 -4.33) (9.92 -16.07) (25.18 6.62) (37.01 10.02) (39.69 -0.74) (43.96 6.11) (46.18 17.93) (47.11 -1.01)
20 (46.74 0.00) (-50.00 0.00) (7.10 19.05) (1.62 6.08) (1.62 -6.08) (7.10 -19.05) (23.75 -0.90) (37.73 4.99) (37.73 -4.99) (42.56 -2.13) (45.70 9.75) (45.70 -9.75)
21 (42.61 -5.60) (-50.00 0.00) (7.57 17.56) (1.36 4.72) (0.50 -7.52) (6.18 -20.97) (21.91 -4.90) (36.27 3.21) (34.21 -9.06) (40.91 -5.04) (45.41 2.80) (43.47 -15.10)
22 (42.61 5.60) (-50.00 -0.00) (6.18 20.97) (0.50 7.52) (1.36 -4.72) (7.57 -17.56) (21.91 4.90) (34.21 9.06) (36.27 -3.21) (40.91 5.04) (43.47 15.10) (45.41 -2.80)
23 (45.86 -3.54) (-50.00 0.00) (7.68 18.07) (1.78 5.23) (1.23 -6.89) (6.49 -20.18) (23.37 -4.21) (37.80 3.47) (36.61 -7.09) (42.32 -3.85) (45.96 2.43) (45.04 -13.82)
24 (45.86 3.54) (-50.00 -0.00) (6.49 20.18) (1.23 6.89) (1.78 -5.23) (7.68 -18.07) (23.37 4.21) (36.61 7.09) (37.80 -3.47) (42.32 3.85) (45.04 13.82) (45.96 -2.43)
25 (46.89 -6.49) (-50.00 0.00) (8.52 17.12) (2.17 4.70) (1.14 -7.70) (5.83 -20.99) (23.98 -5.41) (38.57 2.07) (36.48 -8.62) (42.89 -5.11) (46.51 0.08) (45.21 -15.64)
26 (46.89 6.49) (-50.00 -0.00) (5.83 20.99) (1.14 7.70) (2.17 -4.70) (8.52 -17.12) (23.98 5.41) (36.48 8.62) (38.57 -2.07) (42.89 5.11) (45.21 15.64) (46.51 -0.08)
27 (38.63 0.00) (-50.00 0.00) (6.41 19.34) (0.18 5.93) (0.18 -5.93) (6.41 -19.34) (19.71 -2.62) (32.73 6.84) (32.73 -6.84) (38.61 -2.57) (42.20 11.81) (42.20 -11.81)
28 (39.22 -5.75) (-50.00 0.00) (6.93 17.60) (0.71 4.57) (-0.15 -7.62) (6.10 -21.14) (20.20 -4.84) (34.05 3.87) (31.80 -9.98) (39.22 -5.35) (44.01 6.35) (41.58 -16.28)
29 (39.22 5.75) (-50.00 -0.00) (6.10 21.14) (-0.15 7.62) (0.71 -4.57) (6.93 -17.60) (20.20 4.84) (31.80 9.98) (34.05 -3.87) (39.22 5.35) (41.58 16.28) (44.01 -6.35)
30 (30.37 -15.92) (-50.00 0.00) (6.00 14.49) (-0.92 2.05) (-3.06 -11.84) (3.83 -24.40) (15.44 -8.70) (27.44 1.55) (21.07 -17.96) (34.70 -10.66) (34.71 14.57) (34.11 -24.30)
31 (30.37 15.92) (-50.00 -0.00) (3.83 24.40) (-3.06 11.84) (-0.92 -2.05) (6.00 -14.49) (15.44 8.70) (21.07 17.96) (27.44 -1.55) (34.70 10.66) (34.11 24.30) (34.71 -14.57)
32 (0.00 -36.00) (-50.00 0.00) (-7.16 6.63) (-16.68 -6.05) (-17.18 -19.96) (-13.42 -30.30) (-8.38 -16.63) (-4.81 -1.58) (-7.37 -25.18) (14.62 -20.60) (10.92 18.27) (13.45 -32.14)
33 (0.00 36.00) (-50.00 -0.00) (-13.42 30.30) (-17.18 19.96) (-16.68 6.05) (-7.16 -6.63) (-8.38 16.63) (-7.37 25.18) (-4.81 1.58) (14.62 20.60) (13.45 32.14) (10.92 -18.27)
34 (44.53 -22.41) (-50.00 0.00) (14.30 11.33) (2.77 2.31) (-0.79 -13.56) (0.88 -25.30) (23.64 -11.92) (36.76 -1.94) (30.42 -20.33) (42.36 -11.75) (45.59 7.46) (41.44 -22.71)
35 (44.53 22.41) (-50.00 -0.00) (0.88 25.30) (-0.79 13.56) (2.77 -2.31) (14.30 -11.33) (23.64 11.92) (30.42 20.33) (36.76 1.94) (42.36 11.75) (41.44 22.71) (45.59 -7.46)

Appendix C: Formation

36 (44.09 -29.78) (-50.00 0.00) (15.32 8.96) (3.08 1.45) (-1.70 -16.43) (2.15 -27.12) (23.78 -15.03) (35.10 -2.94) (28.49 -25.40) (42.31 -14.34) (44.65 7.22) (39.72 -25.13)

37 (44.09 29.78) (-50.00 -0.00) (2.15 27.12) (-1.70 16.43) (3.08 -1.45) (15.32 -8.96) (23.78 15.03) (28.49 25.40) (35.10 2.94) (42.31 14.34) (39.72 25.13) (44.65 -7.22)

38 (29.19 -34.36) (-50.00 0.00) (9.40 8.12) (-0.74 -0.94) (-5.36 -18.80) (2.30 -27.83) (15.16 -16.50) (26.04 -3.11) (19.03 -27.25) (34.81 -17.65) (29.88 13.71) (36.79 -29.36)

39 (29.19 34.36) (-50.00 -0.00) (2.30 27.83) (-5.36 18.80) (-0.74 0.94) (9.40 -8.12) (15.16 16.50) (19.03 27.25) (26.04 3.11) (34.81 17.65) (36.79 29.36) (29.88 -13.71)

40 (33.03 -31.26) (-50.00 0.00) (11.01 8.83) (0.28 -0.08) (-4.18 -17.73) (2.01 -27.43) (17.29 -15.35) (28.24 -2.72) (20.92 -26.07) (36.65 -16.30) (34.68 12.10) (38.57 -27.83)

41 (33.03 31.26) (-50.00 -0.00) (2.01 27.43) (-4.18 17.73) (0.28 0.08) (11.01 -8.83) (17.29 15.35) (20.92 26.07) (28.24 2.72) (36.65 16.30) (38.57 27.83) (34.68 -12.10)

42 (23.00 -5.16) (-50.00 0.00) (2.22 17.45) (-3.76 4.10) (-4.35 -7.68) (2.99 -20.69) (10.52 -4.14) (18.29 8.05) (15.33 -13.53) (29.04 -6.00) (24.74 20.06) (23.07 -25.95)

43 (23.00 5.16) (-50.00 -0.00) (2.99 20.69) (-4.35 7.68) (-3.76 -4.10) (2.22 -17.45) (10.52 4.14) (15.33 13.53) (18.29 -8.05) (29.04 6.00) (23.07 25.95) (24.74 -20.06)

44 (28.16 0.00) (-50.00 0.00) (4.32 19.24) (-2.39 5.82) (-2.39 -5.82) (4.32 -19.24) (13.80 -2.33) (22.61 9.60) (22.61 -9.60) (32.25 -1.00) (29.71 21.47) (29.71 -21.47)

45 (34.65 -5.75) (-50.00 0.00) (5.89 17.64) (-0.31 4.40) (-1.13 -7.70) (5.74 -21.22) (17.72 -4.69) (30.34 4.95) (27.93 -11.15) (36.70 -5.66) (40.20 11.53) (37.52 -18.78)

46 (34.65 5.75) (-50.00 -0.00) (5.74 21.22) (-1.13 7.70) (-0.31 -4.40) (5.89 -17.64) (17.72 4.69) (27.93 11.15) (30.34 -4.95) (36.70 5.66) (37.52 18.78) (40.20 -11.53)

47 (19.91 -28.60) (-50.00 0.00) (4.35 9.94) (-4.43 -1.60) (-7.44 -17.45) (-1.88 -26.53) (8.44 -13.83) (21.83 -1.79) (11.60 -24.14) (28.57 -16.62) (24.91 17.54) (28.02 -30.94)

48 (19.91 28.60) (-50.00 -0.00) (-1.88 26.53) (-7.44 17.45) (-4.43 1.60) (4.35 -9.94) (8.44 13.83) (11.60 24.14) (21.83 1.79) (28.57 16.62) (28.02 30.94) (24.91 -17.54)

49 (14.30 -11.06) (-50.00 0.00) (-2.42 15.19) (-7.15 1.87) (-8.06 -10.45) (-1.04 -21.99) (4.24 -6.29) (13.68 6.13) (5.05 -17.13) (22.95 -9.41) (18.09 19.60) (17.23 -26.90)

50 (14.30 11.06) (-50.00 -0.00) (-1.04 21.99) (-8.06 10.45) (-7.15 -1.87) (-2.42 -15.19) (4.24 6.29) (5.05 17.13) (13.68 -6.13) (22.95 9.41) (17.23 26.90) (18.09 -19.60)

51 (11.35 -25.07) (-50.00 0.00) (-2.46 11.03) (-8.68 -1.89) (-10.43 -16.57) (-5.48 -25.58) (1.71 -12.12) (16.29 -0.28) (3.25 -22.20) (22.04 -15.92) (17.57 18.46) (18.90 -29.64)

52 (11.35 25.07) (-50.00 -0.00) (-5.48 25.58) (-10.43 16.57) (-8.68 1.89) (-2.46 -11.03) (1.71 12.12) (3.25 22.20) (16.29 0.28) (22.04 15.92) (18.90 29.64) (17.57 -18.46)

53 (9.58 0.00) (-50.00 0.00) (-4.36 17.44) (-9.77 5.58) (-9.77 -5.58) (-4.36 -17.44) (0.91 -0.62) (5.26 12.10) (5.26 -12.10) (17.81 -1.03) (14.64 25.04) (14.64 -25.04)

54 (18.58 0.00) (-50.00 0.00) (0.77 18.60) (-5.66 5.71) (-5.66 -5.71) (0.77 -18.60) (7.52 -2.11) (11.95 11.59) (11.95 -11.59) (25.23 -0.34) (20.57 25.76) (20.57 -25.76)

55 (3.83 -20.20) (-50.00 0.00) (-9.54 11.72) (-13.20 -1.74) (-13.84 -14.96) (-9.32 -24.18) (-4.47 -9.91) (10.11 2.00) (-3.05 -19.66) (15.55 -14.21) (11.22 18.50) (11.32 -27.46)

56 (3.83 20.20) (-50.00 -0.00) (-9.32 24.18) (-13.84 14.96) (-13.20 1.74) (-9.54 -11.72) (-4.47 9.91) (-3.05 19.66) (10.11 -2.00) (15.55 14.21) (11.32 27.46) (11.22 -18.50)

57 (6.19 -10.32) (-50.00 0.00) (-7.77 14.20) (-11.54 1.49) (-11.99 -10.37) (-6.34 -20.75) (-2.15 -5.86) (9.32 6.62) (-0.16 -16.34) (16.23 -9.25) (12.63 18.96) (11.66 -29.11)

58 (6.19 10.32) (-50.00 -0.00) (-6.34 20.75) (-11.99 10.37) (-11.54 -1.49) (-7.77 -14.20) (-2.15 5.86) (-0.16 16.34) (9.32 -6.62) (16.23 9.25) (11.66 29.11) (12.63 -18.96)

59 (10.47 -29.78) (-50.00 0.00) (-1.47 9.34) (-9.28 -3.00) (-11.25 -18.18) (-6.48 -26.52) (0.92 -14.14) (14.34 -1.54) (3.65 -23.76) (21.84 -17.79) (18.15 18.08) (19.43 -31.04)

Appendix C: Formation

60 (10.47 29.78) (-50.00 -0.00) (-6.48 26.52) (-11.25 18.18) (-9.28 3.00) (-1.47 -9.34) (0.92 14.14) (3.65 23.76) (14.34 1.54) (21.84 17.79) (19.43 31.04) (18.15 -18.08)
61 (13.27 -33.18) (-50.00 0.00) (1.37 8.33) (-7.76 -3.20) (-10.40 -19.06) (-4.63 -27.14) (3.20 -15.64) (15.53 -2.38) (7.31 -25.26) (24.21 -18.79) (21.76 17.76) (22.52 -31.88)
62 (13.27 33.18) (-50.00 -0.00) (-4.63 27.14) (-10.40 19.06) (-7.76 3.20) (1.37 -8.33) (3.20 15.64) (7.31 25.26) (15.53 2.38) (24.21 18.79) (22.52 31.88) (21.76 -17.76)
63 (-16.96 -30.52) (-50.00 0.00) (-22.14 9.35) (-23.18 -4.34) (-23.33 -18.45) (-23.94 -27.93) (-19.24 -13.98) (-14.10 -0.07) (-16.33 -22.81) (0.06 -9.81) (-4.28 17.04) (0.44 -30.41)
64 (-16.96 30.52) (-50.00 -0.00) (-23.94 27.93) (-23.33 18.45) (-23.18 4.34) (-22.14 -9.35) (-19.24 13.98) (-16.33 22.81) (-14.10 0.07) (0.06 9.81) (0.44 30.41) (-4.28 -17.04)
65 (-4.28 -16.81) (-50.00 0.00) (-12.63 11.13) (-14.88 -3.68) (-12.71 -15.79) (-13.76 -23.07) (-8.80 -11.13) (-3.98 5.90) (-3.79 -19.50) (5.59 -8.48) (2.60 22.30) (6.03 -29.81)
66 (-4.28 16.81) (-50.00 -0.00) (-13.76 23.07) (-12.71 15.79) (-14.88 3.68) (-12.63 -11.13) (-8.80 11.13) (-3.79 19.50) (-3.98 -5.90) (5.59 8.48) (6.03 29.81) (2.60 -22.30)
67 (-7.08 -27.57) (-50.00 0.00) (-13.48 10.45) (-16.57 -4.96) (-15.15 -19.34) (-15.35 -27.21) (-10.91 -14.26) (-6.22 1.56) (-6.54 -22.94) (6.91 -10.52) (1.32 18.51) (6.84 -30.97)
68 (-7.08 27.57) (-50.00 -0.00) (-15.35 27.21) (-15.15 19.34) (-16.57 4.96) (-13.48 -10.45) (-10.91 14.26) (-6.54 22.94) (-6.22 -1.56) (6.91 10.52) (6.84 30.97) (1.32 -18.51)
69 (-7.96 -31.41) (-50.00 0.00) (-13.66 11.00) (-17.14 -5.03) (-15.89 -20.09) (-15.83 -28.57) (-11.44 -14.98) (-6.86 0.18) (-7.45 -24.07) (8.36 -10.67) (1.09 17.04) (7.39 -31.30)
70 (-7.96 31.41) (-50.00 -0.00) (-15.83 28.57) (-15.89 20.09) (-17.14 5.03) (-13.66 -11.00) (-11.44 14.98) (-7.45 24.07) (-6.86 -0.18) (8.36 10.67) (7.39 31.30) (1.09 -17.04)
71 (-23.89 -34.21) (-50.00 0.00) (-28.55 8.60) (-29.37 -4.33) (-31.05 -17.31) (-29.93 -27.83) (-24.69 -13.70) (-19.22 -1.39) (-22.94 -23.29) (-3.41 -8.93) (-7.83 15.33) (-3.39 -30.07)
72 (-23.89 34.21) (-50.00 -0.00) (-29.93 27.83) (-31.05 17.31) (-29.37 4.33) (-28.55 -8.60) (-24.69 13.70) (-22.94 23.29) (-19.22 1.39) (-3.41 8.93) (-3.39 30.07) (-7.83 -15.33)
73 (-54.50 0.00) (-50.00 0.00) (-49.99 6.88) (-47.53 3.17) (-47.53 -3.17) (-49.99 -6.88) (-44.06 1.02) (-41.64 8.61) (-41.64 -8.61) (-30.03 4.57) (-23.83 22.20) (-23.83 -22.20)
74 (-19.61 -5.46) (-50.00 0.00) (-27.52 9.88) (-27.37 1.07) (-26.74 -7.22) (-27.57 -15.59) (-23.16 -4.67) (-17.58 8.75) (-18.04 -13.11) (-8.99 -3.01) (-6.72 24.53) (-6.06 -26.78)
75 (-19.61 5.46) (-50.00 -0.00) (-27.57 15.59) (-26.74 7.22) (-27.37 -1.07) (-27.52 -9.88) (-23.16 4.67) (-18.04 13.11) (-17.58 -8.75) (-8.99 3.01) (-6.06 26.78) (-6.72 -24.53)
76 (-7.96 -7.37) (-50.00 0.00) (-16.52 12.59) (-17.18 0.07) (-15.64 -9.85) (-16.91 -19.09) (-12.21 -6.37) (-7.25 9.27) (-7.34 -15.32) (0.76 -4.50) (0.38 24.81) (1.69 -28.01)
77 (-7.96 7.37) (-50.00 -0.00) (-16.91 19.09) (-15.64 9.85) (-17.18 -0.07) (-16.52 -12.59) (-12.21 6.37) (-7.34 15.32) (-7.25 -9.27) (0.76 4.50) (1.69 28.01) (0.38 -24.81)
78 (-5.31 0.00) (-50.00 0.00) (-14.59 16.26) (-14.82 4.50) (-14.82 -4.50) (-14.59 -16.26) (-9.26 -1.26) (-4.86 12.66) (-4.86 -12.66) (2.20 -0.09) (2.61 26.72) (2.61 -26.72)
79 (-2.06 -11.35) (-50.00 0.00) (-11.39 13.02) (-13.56 -2.16) (-11.37 -12.91) (-12.20 -21.01) (-6.73 -8.68) (-2.03 8.39) (-1.72 -17.72) (6.06 -6.47) (4.06 24.04) (6.48 -29.12)
80 (-2.06 11.35) (-50.00 -0.00) (-12.20 21.01) (-11.37 12.91) (-13.56 2.16) (-11.39 -13.02) (-6.73 8.68) (-1.72 17.72) (-2.03 -8.39) (6.06 6.47) (6.48 29.12) (4.06 -24.04)
81 (-3.39 -5.90) (-50.00 0.00) (-12.79 14.39) (-14.14 0.58) (-12.70 -8.99) (-13.20 -18.92) (-7.76 -5.48) (-3.16 10.48) (-3.04 -15.36) (4.70 -3.75) (3.43 25.44) (4.62 -28.05)
82 (-3.39 5.90) (-50.00 -0.00) (-13.20 18.92) (-12.70 8.99) (-14.14 -0.58) (-12.79 -14.39) (-7.76 5.48) (-3.04 15.36) (-3.16 -10.48) (4.70 3.75) (4.62 28.05) (3.43 -25.44)
83 (-9.44 -24.77) (-50.00 0.00) (-15.92 9.67) (-17.92 -4.50) (-16.56 -18.28) (-17.58 -26.25) (-13.12 -13.42) (-8.23 2.36) (-8.86 -21.71) (3.59 -10.02) (-0.29 19.43) (4.27 -30.45)

Appendix C: Formation

84 (-9.44 24.77) (-50.00 -0.00) (-17.58 26.25) (-16.56 18.28) (-17.92 4.50) (-15.92 -9.67) (-13.12 13.42) (-8.86 21.71) (-8.23 -2.36) (3.59 10.02) (4.27 30.45) (-0.29 -19.43)
85 (-12.39 -12.39) (-50.00 0.00) (-20.15 9.72) (-20.37 -1.72) (-18.61 -12.64) (-20.71 -20.92) (-16.37 -8.83) (-11.06 6.74) (-11.58 -16.74) (-2.10 -6.61) (-2.47 23.22) (-0.53 -28.52)
86 (-12.39 12.39) (-50.00 -0.00) (-20.71 20.92) (-18.61 12.64) (-20.37 1.72) (-20.15 -9.72) (-16.37 8.83) (-11.58 16.74) (-11.06 -6.74) (-2.10 6.61) (-0.53 28.52) (-2.47 -23.22)
87 (-16.37 -15.78) (-50.00 0.00) (-23.62 8.03) (-23.54 -2.52) (-22.13 -13.78) (-24.22 -21.95) (-19.87 -9.96) (-14.35 5.07) (-15.41 -17.57) (-4.48 -7.55) (-4.82 22.01) (-2.60 -28.70)
88 (-16.37 15.78) (-50.00 -0.00) (-24.22 21.95) (-22.13 13.78) (-23.54 2.52) (-23.62 -8.03) (-19.87 9.96) (-15.41 17.57) (-14.35 -5.07) (-4.48 7.55) (-2.60 28.70) (-4.82 -22.01)
89 (-19.91 -18.28) (-50.00 0.00) (-26.83 6.95) (-26.70 -3.02) (-25.82 -14.13) (-27.37 -22.39) (-22.83 -10.50) (-17.16 3.88) (-18.77 -18.12) (-6.47 -7.96) (-6.79 21.05) (-4.47 -28.73)
90 (-19.91 18.28) (-50.00 -0.00) (-27.37 22.39) (-25.82 14.13) (-26.70 3.02) (-26.83 -6.95) (-22.83 10.50) (-18.77 18.12) (-17.16 -3.88) (-6.47 7.96) (-4.47 28.73) (-6.79 -21.05)
91 (-32.73 -29.19) (-50.00 0.00) (-37.47 4.45) (-38.23 -4.58) (-39.69 -13.44) (-37.42 -22.97) (-32.04 -11.42) (-26.10 -0.09) (-30.22 -20.74) (-12.00 -7.92) (-13.04 16.63) (-10.38 -28.63)
92 (-32.73 29.19) (-50.00 -0.00) (-37.42 22.97) (-39.69 13.44) (-38.23 4.58) (-37.47 -4.45) (-32.04 11.42) (-30.22 20.74) (-26.10 0.09) (-12.00 7.92) (-10.38 28.63) (-13.04 -16.63)
93 (-24.03 -17.55) (-50.00 0.00) (-30.99 5.87) (-31.07 -2.95) (-30.55 -12.75) (-31.18 -20.78) (-26.38 -9.76) (-20.54 3.87) (-22.45 -17.38) (-9.46 -7.39) (-9.15 21.01) (-7.22 -28.24)
94 (-24.03 17.55) (-50.00 -0.00) (-31.18 20.78) (-30.55 12.75) (-31.07 2.95) (-30.99 -5.87) (-26.38 9.76) (-22.45 17.38) (-20.54 -3.87) (-9.46 7.39) (-7.22 28.24) (-9.15 -21.01)
95 (-31.26 0.00) (-50.00 0.00) (-38.28 8.03) (-39.65 2.16) (-39.65 -2.16) (-38.28 -8.03) (-32.56 -0.43) (-27.38 9.85) (-27.38 -9.85) (-18.33 1.17) (-13.07 24.74) (-13.07 -24.74)
96 (-29.34 -15.33) (-50.00 0.00) (-36.13 4.57) (-36.79 -2.72) (-36.56 -10.26) (-35.95 -17.42) (-30.72 -8.28) (-24.81 4.35) (-26.82 -15.97) (-13.30 -6.21) (-12.11 21.27) (-10.73 -27.40)
97 (-29.34 15.33) (-50.00 -0.00) (-35.95 17.42) (-36.56 10.26) (-36.79 2.72) (-36.13 -4.57) (-30.72 8.28) (-26.82 15.97) (-24.81 -4.35) (-13.30 6.21) (-10.73 27.40) (-12.11 -21.27)
98 (-37.01 -33.03) (-50.00 0.00) (-40.14 3.64) (-40.82 -4.62) (-42.77 -12.81) (-40.07 -23.04) (-34.61 -11.31) (-28.64 -1.11) (-33.61 -21.66) (-13.43 -7.26) (-14.85 14.96) (-12.13 -28.47)
99 (-37.01 33.03) (-50.00 -0.00) (-40.07 23.04) (-42.77 12.81) (-40.82 4.62) (-40.14 -3.64) (-34.61 11.31) (-33.61 21.66) (-28.64 1.11) (-13.43 7.26) (-12.13 28.47) (-14.85 -14.96)
100 (-54.50 -36.00) (-50.00 0.00) (-48.06 2.31) (-44.48 -1.69) (-46.24 -10.29) (-48.45 -21.85) (-43.60 -8.28) (-38.36 -1.28) (-43.89 -21.18) (-22.80 -4.37) (-22.49 12.44) (-20.87 -26.29)
101 (-54.50 36.00) (-50.00 -0.00) (-48.45 21.85) (-46.24 10.29) (-44.48 1.69) (-48.06 -2.31) (-43.60 8.28) (-43.89 21.18) (-38.36 1.28) (-22.80 4.37) (-20.87 26.29) (-22.49 -12.44)
102 (-48.66 -22.71) (-50.00 0.00) (-47.31 2.84) (-45.54 -2.25) (-46.06 -8.98) (-47.21 -15.63) (-41.76 -7.28) (-36.69 1.74) (-40.32 -17.03) (-22.29 -5.05) (-21.01 17.44) (-19.94 -26.01)
103 (-48.66 22.71) (-50.00 -0.00) (-47.21 15.63) (-46.06 8.98) (-45.54 2.25) (-47.31 -2.84) (-41.76 7.28) (-40.32 17.03) (-36.69 -1.74) (-22.29 5.05) (-19.94 26.01) (-21.01 -17.44)
104 (-39.52 -28.16) (-50.00 0.00) (-42.42 2.74) (-42.60 -4.34) (-43.82 -11.32) (-42.05 -20.13) (-36.63 -10.01) (-30.74 0.19) (-35.11 -19.75) (-16.44 -6.95) (-16.49 16.49) (-14.51 -27.72)
105 (-39.52 28.16) (-50.00 -0.00) (-42.05 20.13) (-43.82 11.32) (-42.60 4.34) (-42.42 -2.74) (-36.63 10.01) (-35.11 19.75) (-30.74 -0.19) (-16.44 6.95) (-14.51 27.72) (-16.49 -16.49)
106 (-39.22 -22.12) (-50.00 0.00) (-42.88 2.69) (-43.11 -3.81) (-43.64 -10.02) (-42.46 -17.18) (-36.90 -8.77) (-31.14 1.89) (-34.57 -17.60) (-17.55 -6.43) (-16.77 18.47) (-15.29 -27.15)
107 (-39.22 22.12) (-50.00 -0.00) (-42.46 17.18) (-43.64 10.02) (-43.11 3.81) (-42.88 -2.69) (-36.90 8.77) (-34.57 17.60) (-31.14 -1.89) (-17.55 6.43) (-15.29 27.15) (-16.77 -18.47)

Appendix C: Formation

108 (-41.58 -7.22) (-50.00 0.00) (-45.08 3.99) (-45.51 -0.60) (-45.59 -4.05) (-44.96 -8.55) (-38.84 -3.44) (-33.99 6.61) (-35.13 -11.83) (-21.98 -1.47) (-18.33 22.22) (-17.94 -24.94)
109 (-41.58 7.22) (-50.00 -0.00) (-44.96 8.55) (-45.59 4.05) (-45.51 0.60) (-45.08 -3.99) (-38.84 3.44) (-35.13 11.83) (-33.99 -6.61) (-21.98 1.47) (-17.94 24.94) (-18.33 -22.22)
110 (-34.06 -7.37) (-50.00 0.00) (-40.73 4.73) (-41.60 -0.80) (-41.55 -5.15) (-40.20 -10.73) (-34.41 -4.35) (-28.91 6.92) (-29.99 -12.46) (-17.91 -2.58) (-14.67 22.91) (-14.14 -25.77)
111 (-34.06 7.37) (-50.00 -0.00) (-40.20 10.73) (-41.55 5.15) (-41.60 0.80) (-40.73 -4.73) (-34.41 4.35) (-29.99 12.46) (-28.91 -6.92) (-17.91 2.58) (-14.14 25.77) (-14.67 -22.91)
112 (-48.22 -9.88) (-50.00 0.00) (-47.87 4.14) (-46.74 -0.12) (-46.90 -4.98) (-47.87 -9.24) (-41.91 -3.75) (-37.64 5.54) (-39.23 -12.38) (-24.49 -1.57) (-21.27 20.92) (-20.84 -24.54)
113 (-48.22 9.88) (-50.00 -0.00) (-47.87 9.24) (-46.90 4.98) (-46.74 0.12) (-47.87 -4.14) (-41.91 3.75) (-39.23 12.38) (-37.64 -5.54) (-24.49 1.57) (-20.84 24.54) (-21.27 -20.92)
114 (15.33 -21.38) (-50.00 0.00) (-0.76 12.47) (-6.57 -0.58) (-8.43 -14.97) (-2.37 -25.05) (4.93 -10.62) (18.46 0.94) (4.93 -21.15) (24.72 -14.15) (20.06 18.78) (21.33 -29.66)
115 (15.33 21.38) (-50.00 -0.00) (-2.37 25.05) (-8.43 14.97) (-6.57 0.58) (-0.76 -12.47) (4.93 10.62) (4.93 21.15) (18.46 -0.94) (24.72 14.15) (21.33 29.66) (20.06 -18.78)

Appendix D: Example Setplays

D.1 Square Passing

```
(setplay :name passQuadSetplay
  :players(list (playerRole :roleName first)
               (playerRole :roleName second)
               (playerRole :roleName third)
               (playerRole :roleName fourth) )
  :abortCond (bpos :region (regNamed :name their_middle_field))
  :steps (seq
    (step :id 0 :waitTime 5 :abortTime 50
      :participants (list (at first (pt :x -25.100000 :y 23.400000)) (at second (pt :x -5.900000 :y
23.800000)) (at third (pt :x -5.800000 :y 12.700000)) (at fourth (pt :x -25.200000 :y
11.100000)) )
      :condition (playm play_on)
      :leadPlayer first
      :transitions (list
        (nextStep :id 1 :condition (canPassPl :from first :to second)
          :directives (list
            (do :players first :actions (bto :players second) )
            (do :players second :actions (intercept) ) ) )
          (nextStep :id 2 :condition (canPassPl :from first :to fourth)
            :directives (list
              (do :players first :actions (bto :players fourth) )
              (do :players fourth :actions (intercept) ) ) ) ) )
      (step :id 1 :waitTime 5 :abortTime 50
```

Appendix D: Example Setplays

```
:participants (list (at first (pt :x -25.100000 :y 23.400000)) (at second (pt :x -5.900000 :y
23.800000)) (at third (pt :x -5.800000 :y 12.700000)) (at fourth (pt :x -25.200000 :y
11.100000)) )
:condition (owner :players second)
:leadPlayer second
:transitions (list
  (nextStep :id 0 :condition (canPassPl :from second :to first)
  :directives (list
    (do :players second :actions (bto :players first) )
    (do :players first :actions (intercept) ) ) )
  (nextStep :id 3 :condition (canPassPl :from second :to third)
  :directives (list
    (do :players second :actions (bto :players third) )
    (do :players third :actions (intercept) ) ) ) ) )
(step :id 2 :waitTime 5 :abortTime 50
:participants (list (at first (pt :x -25.100000 :y 23.400000)) (at second (pt :x -5.900000 :y
23.800000)) (at third (pt :x -5.800000 :y 12.700000)) (at fourth (pt :x -25.200000 :y
11.100000)) )
:condition (owner :players fourth)
:leadPlayer fourth
:transitions (list
  (nextStep :id 0 :condition (canPassPl :from fourth :to first)
  :directives (list
    (do :players fourth :actions (bto :players first) )
    (do :players first :actions (intercept) ) ) )
  (nextStep :id 3 :condition (canPassPl :from fourth :to third)
  :directives (list
    (do :players fourth :actions (bto :players third) )
    (do :players third :actions (intercept) ) ) ) ) )
(step :id 3 :waitTime 5 :abortTime 50
:participants (list (at first (pt :x -25.100000 :y 23.400000)) (at second (pt :x -5.900000 :y
23.800000)) (at third (pt :x -5.800000 :y 12.700000)) (at fourth (pt :x -25.200000 :y
11.100000)) )
:condition (owner :players third)
:leadPlayer third
:transitions (list
  (nextStep :id 2 :condition (canPassPl :from third :to fourth)
  :directives (list
    (do :players third :actions (bto :players fourth) )
    (do :players fourth :actions (intercept) ) ) )
  (nextStep :id 1 :condition (canPassPl :from third :to second)
  :directives (list
    (do :players third :actions (bto :players second) )
    (do :players second :actions (intercept) ) ) ) ) ) )
```


D.2 Simple Corner

```

(setplay :name simpleCorner
:players (list (playerRole :roleName CornerP)
              (playerRole :roleName receiver)
              (playerRole :roleName shooter) )
:steps (seq
  (step :id 0 :waitTime 15 :abortTime 70
:participants (list (at CornerP (pt :x 52.500000 :y 34.100000)) (at receiver (pt :x 40.300000 :y 25.400000))
                  (at shooter (pt :x 36.200000 :y 2.700000)))
:condition (playm fk_our)
:leadPlayer CornerP
:transitions (list
              (nextStep :id 1 :condition (canPassPl :from CornerP :to receiver)
:directives (list
              (do :players CornerP :actions (bto :players receiver) )
              (do :players receiver :actions (intercept) ) ) ) )
  (step :id 1 :waitTime 5 :abortTime 70
:participants (list (at CornerP (pt :x 52.500000 :y 34.100000)) (at receiver (pt :x 40.300000 :y 25.400000))
                  (at shooter (pt :x 36.200000 :y 2.700000)) )
:condition (playm play_on)
:leadPlayer receiver
:transitions (list
              (nextStep :id 2 :condition (canPassPl :from receiver :to shooter)
:directives (list
              (do :players receiver :actions (bto :players shooter) )
              (do :players shooter :actions (intercept) ) ) )
              (abort :condition (canShoot :players receiver)
:directives (list
              (do :players receiver :actions (shoot) ) ) ) )
  (step :id 2 :abortTime 70
:participants (list (at CornerP (pt :x 52.500000 :y 34.100000)) (at receiver (pt :x 40.300000 :y 25.400000))
                  (at shooter (pt :x 36.200000 :y 2.700000)) )
:condition (and (bowner :players shooter) (playm play_on) )
:leadPlayer shooter
:transitions (list
              (nextStep :id 3 :condition (canShoot :players shooter)
:directives (list
              (do :players shooter :actions (shoot) ) ) ) )
  (step :id 3 :waitTime 5 :abortTime 70
:participants (list (at CornerP (pt :x 52.500000 :y 34.100000)) (at receiver (pt :x 40.300000 :y 25.400000))
                  (at shooter (pt :x 36.200000 :y 2.700000)) )
:condition (playm play_on)
:leadPlayer shooter

```

Appendix D: Example Setplays

```
:transitions (list  
              (finish ) ) ) )
```