# NADCO-2D Soccer 2D Simulation Team Description Paper 2012

Mohammad Ali Sadeghi Marasht[1]

[1] Department Of Computer Software Engineering
Islamic Azad University - Northern Tehran Branch
[1] Department of Artificial Intelligence and Multi-Agent Simulation
NAD Educational Robotics Corporation

< sadeghi.mahdi@gmail.com >

Abstract. This paper outlines the improvements of previous high-level skills, emphasizes on the use of the coach for Online Roles Assignments, and highlights a proposed Soccer Development Model designed by our team within last year.

## 1 Introduction

The NADCO-2D team leader started his experiences in the Soccer 2D Simulation league in 2004. He has been away from the competitions for a few years and began continuing to work again with other teams and founded his team under the name of NADCO-2D. He has participated in several worldwide competitions so far including RoboCup2011 competitions and has placed 6$^{th}$. We have chosen Agent2D-2.1.0 base source code [2] to work on according to the great implementation of low-level layer. Our RC2011 source code is released to public due to our software cycle policy to help newbies start their experiences in this field which is accessible [14].

## 2 Previous Skills Improvements

Since last year, we have modified many parts of our team's source code to provide more enhanced skills and obtain better results. Some of these skills' improvements are described here in more details.

### 2.1 Soccer Roles

We have totally changed our previous soccer roles. According to our team's TDP last year, we have a Dynamic Formation [3] which makes changes to our fixed formation in run-time according to a few parameters. In new roles, we are using 4-4-2 diamond instead of default 4-3-3 which changes smoothly from 6-2-2 in deep defense to 4-2-4 on heavy attacks. Our roles are “Goalie” in target, “Defenses” and “Sides” on defense line, “Middles” and “Wings” on middle line and

“Attacks” on offense line. Also a new Online Roles Assignment ability is added to our coach which is discussed under section 4.

### 2.2 Block

In our team's TDP last year, we had described how we implemented our block skill in two sub parts which were to identify the blocker and to make the physical blocking [3]. We had also presented a Hassle skill [3] which was to keep a close contact to opponent offenders to prevent them from receiving through passes. This year we have merged them into one skill to gain better results. In this new Block skill, Sides are responsible for keep hassling opponents until they become the blockers. Defenders are open blockers which cover the danger area.

### 2.3 Dash (AKA Long Dribble)

A few changes are employed in our previous Dash skill [3] to predict more and better points for fast moving. A calculation method similar to our through pass point prediction is designed which is to not limit it to forward moving but to change the dribble angle to the best one.

### 2.4 Play-On Mark

Since we have had implemented our Hassle skill according to MWBM [1,3] described in our team's TDP last year, we recognized no more play-on mark is suitable for our players which cause them a stamina shortage. Therefore, a simple Cross Mark is designed which manages the Defenses and Middles to take position aside the opponents in danger area and prevent them from receiving passes.

### 2.5 Coach Environment Analysis

Last year we emphasized on how our coach is capable of analyzing opponent offenders and remembering their routes of movements to help defenders mark them with predicted points to gain better results [3]. This year we have merged this skill with one of our old Formation Learning [4] methods and our coach is now able to draw a matrix filled by teammates and opponents positions during the past cycles and use it as an input of dribble routes and pass points, beside the Match Statistics Review for human expert notifications.

## 3 Soccer 2D Simulation Development Model

As of Software Engineering [5], a Soccer 2D Simulation source code is a well-proposed software piece of work which needs correct development, enhancement, documentation, iteration approach, flexibility to changes and so on which fits into the Agile software development models. Extreme Programming (XP) [6] is here referred to as almost the best development model that meets the Soccer 2D Simulation development requirements.

Passing aside the introduction, this method is organized on rapidness and early goal achievement, excluding the source code documentation and lack of design specification. This method is appropriate for small projects where staffing and equipment are small. While documentation is exhausting and a time challenging procedure, developers tend to comment and discuss on their codes in

short and benefit forms. This method is one of incremental kinds where it advocates frequent “releases” in a large number of short cycles until it meets the needs. Referring to the number of team members, 2 to 12, it may be applied to Soccer 2D Simulation teams which are encouraged to keep quantity low but to work more effective. Another fitness is in Unit and Acceptance tests which face compiling, debugging, logging and run-time observation. Therefore, Soccer 2D Simulation Development Model (SSDM) inheritances four major activities described below in more details.

Let us assume a team consisting six members attempts to follow SSDM activities. It is recommended to divide them into two subgroups, each filled by two programmers guided by a leader. According to the team's policy, subgroups may choose different levels of research, i.e. Kickable and Non-Kickable skills, Offensive and Defensive skills, Calculating and Learning skills, Skills Coding and Strategies Planning, Skills Coding and Mathematics Modeling, etc.

### 3.1 Listening

Each team member is allowed to propose ideas, discuss algorithms, bring up methodologies and accept or decline other comments. The whole commitments are gathered in Story Cards and discussed in Iteration Plans Meetings each week. The team leader is in charge of representing ideas and/or asking the idea owner to come up with full technical details. By the assumption of team's research level in mind; it may be relevant to discuss each idea in the appropriate subgroup in the presence of the leader. The most suited ideas are checked in a TODO list to be coded later.

### 3.2 Designing

Regarded to team's division in subgroups, the ideas' designing is the major responsibility of each leader and by a refer to team's research level assumption, it is possible to divide ideas into subgroups and discuss them in proper time and place among subgroups members.

### 3.3 Coding

Although Pair Programming [7] is common in agile development methods, it is advised to let programmers work in parallel and fetch resulting codes and insert them to the source code together which holds team's productivity up in short time and reserves more time for debugging level.

SSDM excludes documentations, as its father does, but strongly emphasizes on use of inline comments with full details and test examples if possible.

### 3.4 Testing

Coding and Testing activities are tied up together. Each line of code is tested in Unit Test modules during compilation. Every single coded class and method is tested during log-based run-time Unit Tests. Memory management, block outputs, returned values and other internal calculation results are logged in a human-readable form in text files, with companying run-time console outputs for rapid performance observation. While Unit Tests have to be successfully passed to be able to run the binary, Acceptance Tests are executed after Stand up Meetings each day, helping all team members have an overview on binary performance, coming up with new ideas or implementations and/or bug fixes.

## 4 Online Roles Assignments

Our coach is now able of Online Roles Assignments [8, 15, 16]. Agent2D base source code is a hardened one to handle roles in which each player receives a role based on its uniform number and decides the movement positions according to its fixed formations records on formation files. This year we have implemented a dynamic roles assignment opportunity in which the coach decides which player is responsible to take which role on the field during run-time. For this purpose we had to integrate a Stamina Predictor algorithm on coach to make it able to decide the best role for each agent. After that, the coach sends new roles to the players by say command in a sequential form. Each player decodes the command and retrieves its new role. The advantage of the Online Roles Assignments skill is its flexibility occur during run-time as many times as the coach can send a say command helping team to take position in different defense, middle and offense strategies rapidly. These sub skills are brought here in more details.

Fig. 1. Online Roles Assignments - After Coach's Command

### 4.1 Coach Stamina Predictions

As the main part of this skill, the coach has to know each player agent's current stamina and stamina capacity according to its type. As the coach's noiseless sense of stadium, it can observe each agent's amount of movement each cycle significantly which is the only requirement to calculate its current stamina.
Stamina decreasing for player agent p happens by sending dash command as follows:

$$Length(LastPos(p) - CurrentPos(p)) = Length(LastVel(p) - DashAccel)$$
$$DashAccel = Vector(AccelMag, AccelAng)$$
$$AccelMag = Max\big(Accel, MaxAccel(p)\big)$$
$$Accel = Abs\big(DashPower \times DashDirRate(DashPower) \times DashRate(p)\big)$$
$$AccelAng = BodyDir(p) + Angle\big(LastPos(p) - CurrentPos(p)\big)$$
$$StaminaDecrease(p) = DashPower$$

The amount of stamina recovery is related to player agent's type and happens each cycle by wait until it reaches to stamina capacity limitation and is simply calculable as above.

### 4.2 Coach New Roles Encoding

The less the player agent's current stamina, the less its roles importance is. At the end of the decision making procedure of the coach, a freeform say command is sent to the server in the following form:

`"(freeform "[PlayerNumber1Role]...[PlayerNumber11Role]""`

The new roles assigned to player agents are valid until the next role assignment command. This will gives coach varieties of team's high-level formation regarded to defense and offense strategies according to remaining stamina of the player agents, limiting the opponent pressure team caused by stamina shortage of defenders and sides and/or bring more attacks up through wings and attacks.

### 4.3 Coach Substitute Strategy

With the knowledge of player agents' remaining stamina and referring to their stamina capacity, coach will be able to think better for substitute strategies. WrightEagle2011 introduced a working strategy last year [8], which is now employed in our team.

On this scope, the player agent p is inserted into tired players list if:

$$\frac{Capacity(p) + Stamina(p)}{LeftCycles} < \frac{TotalCapacity - Capacity(p)}{PassedCycles}$$

On multiple candidates, the list is sorted by players' roles importance which is depended on the remaining cycles and the result of the match. According to coach substitute limitation, top players will be substituted to new types.

## 5 Developed Graphical Online Strategy Analyzer

A simple online soccer monitor is developed using Microsoft® Visual Studio C# .NET which is plugged into rcssserver-15.0.1 through TCP/IP connection. This online monitor uses a solid picture of soccer field as the background and moves small circles with pre-defined uniform numbers and team side on it according to the player agents positions received and parsed from soccer server. It is also able to retrieve the result of the match and the number of passed cycles.

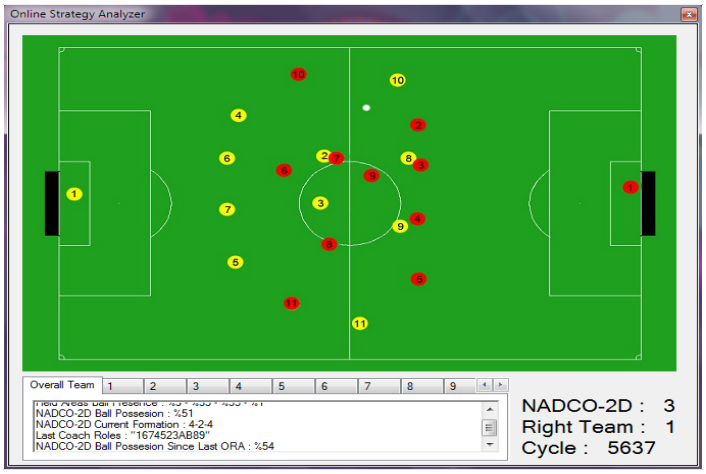


Fig. 2. Graphical Online Strategy Analyzer

The main responsibility of this Online Strategy Analyzer is to monitor the ball possession of the left team (which is meant to be our team) after each ORA command sent from coach and the overall passed cycles. This monitor is able to fetch our coach commands by a simple tweak in rcssserver-15.0.1 codes.

Each left team player agent has a tab to show its statistical progress during match, i.e. current stamina, total amount of movement, average speed, previous roles, yellow card, red card and total kickable cycles.

## 6 Conclusion and Future Works

In this paper, we described our improvements on previous implemented skills, proposed SSDM as a workflow for new and professional teams, discussed our researches and implementations of LFS in this field and outlined the use of coach in ORA in our team.

Nemesis has proposed a framework for Learning Formation Strategy [9] using BPNN through a modular scheme to reduce the number of complex features included in learning process [10]. This year, we are cooperating with this team on completing and integrating this framework in Soccer 2D Simulation field. Although the formation output is not ready yet, we hope we can complete it and use it as our main team formation in RoboCup2012 competitions.

As future plans, several soccer skills improvements using Genetic learning algorithms are under development on the basis of gpc++ [11] and galib [12]. We are also working on "PySoccer", a Python Soccer 2D Simulation framework as the first interpreter base source code in this field [13].

## References


1. Molitor, P.: Algorithmen für das Maximum Weight Matching Problem in bipartiten Graphen. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1989.
2. RoboCup tools Project Top Page (http://sourceforge.jp/projects/rctools) last visited 4th March 2012.
3. M. A. Sadeghi Marasht: "NADCO-2D Soccer 2D Simulation Team Description Paper", Robocup2011, Istanbul, Turkey 2011.
4. M. A. Sadeghi Marasht: "NADCO-2D Soccer 2D Simulation Team Description Paper", IranOpen2011, Tehran, Iran 2011.
5. Software engineering (http://en.wikipedia.org/wiki/Software_engineering) last visited 4th March 2012.
6. Extreme programming (http://en.wikipedia.org/wiki/Extreme_Programming) last visited 4th March 2012.
7. Pair programming (http://en.wikipedia.org/wiki/Pair_programming) last visited 4th March 2012.
8. A. Bai, G. Lu, H. Zhang, X. Chen: "WrightEagle 2D Soccer Simulation Team Description", Robocup2011, Istanbul, Turkey 2011.
9. M. Norouzitallab , A. Javari, A. Noroozi, S.M.A. Salehizadeh, K. Meshgi: "Nemesis2010 Team Description", Robocup2010, Singapore, 2010.
10. M. Norouzitallab: "Dynamic Multi-Agent Formation Strategy in Real-Time Applications",AUTCup2010, Tehran, Iran, 2010.
11. Genetic Programming C++ Class Library (http://www.cs.ucl.ac.uk/staff/ucacbbl/ftp/weinbenner/gp.html) last visited 4th March 2012.
12. GAlib: Matthew's Genetic Algorithms Library (http://lancet.mit.edu/ga/) last visited 4th March 2012.
13. M. A. Sadeghi Marasht: "NADCO-2D Soccer 2D Simulation Team Description Paper", ChinaOpen2011, China, 2011.
14. Official Data Repository (http://www.socsim.robocup.org/files/2D/binary/RoboCup2011/source/NADCO-2D.tar.gz) last visited 4th March 2012
15. L. P. Reis, N. Lau, E. C. Olivera: "Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents", Springer LNAI, Vol 2103, pp. 175-197, Berlin, 2001.
16. P. Stone, M. Veloso,: "Task Decomposition, Dynamic Roles Assignment and Low-Bandwidth Communication for Real-Time Strategic Teamwork, Art, Intelligence", Vol.110, No.2, pp.241-273, 1999.