

# A Soccer Simulation for Comparison of Game Strategies

Martin Bruckner<sup>1,2 \*</sup>, Nikolas Popper<sup>2</sup>

<sup>1</sup> Inst. For Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

<sup>2</sup> dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria; \* [martin.bruckner@dwh.at](mailto:martin.bruckner@dwh.at)

**Abstract.** Our goal was to develop a simple simulator to show capabilities of simulation methods and to introduce a simple game for understanding how various strategies within a rather complex system of individuals can be tested. A simple soccer game was chosen, which was motivated by the 2008 European Football Championship in Austria. With aspects from Cellular Automata and Agent Based Modelling it was possible to show, that changes in the model are very simple to implement by changing the transitional function, in our case by adding or omitting individual rules. So different strategies but also other changes in the system can be easily tested and compared.

## Introduction

‘The Goal’ of this work is to define a Model of a soccer game. There are - not surprisingly - 22 players, including two goalkeepers, one ball and two goals. Foul play or offside will be ignored. Out balls are avoided by introduction of a ‘board’ where the ball is rebounded. Rules could easily be added, but were ignored to test the reactions of the system in a first stage. The model was introduced to test the modeling of different strategies of team line up and was influenced by hosting the 2008 UEFA European Football Championship in Austria and Switzerland. The model is realized as a JAVA Applet.

## 1 Cellular Automata Theory

As a first approach a cellular automaton was planned to implement the simulation. A cellular automaton (plural: cellular automata) is a discrete mathematical model to simulate problems for example in theoretical biology and microstructure modeling. The automaton consists of a discrete lattice of identical cells; each is in one of a finite number of states. The dimension of the grid is finite. Time is also discrete, and the state of a cell at time  $t$  is depending on the states of a finite number of cells (the neighborhood of the cell) at time  $t-1$ .

The neighbourhood is classed as either Moore or Von Neumann neighborhood. The Moore neighborhood comprises the eight cells surrounding the central cell on a two-dimensional area, during Von Neumann comprises the four cells orthogonally surrounding the cell. In the transition function it is precisely defined how the transition from one state to the next state takes place. The transitions of the states are made for all cells with the same transition function simultaneously.

## 2 Advantages/Disadvantages of the System – Introducing Agents

An advantage of the system is, that changes in the model are very simple to implement by changing the transitional function. In our case this can be done by adding or omitting individual rules. For this reason different strategies can be tested and compared. But individual rules mean, that for the various players different behaviour should be possible to implement. This is not basically possible with a cellular automaton. The virtual manager could also be able to compare the advantages or disadvantages of 4-4-2 vs. 3-4-3 with his – limited – pool of players. By doing so it can be shown, that some strategies only work with players like Ronaldo and not with Prödl and so the strategies can be adjusted to the pool of players, as different players have different possibilities (see Section 3).

Regarding to this it appears that not a cellular automaton but some kind of mixture between cellular automaton and an agent based system is appropriate. A great advantage of this simulation method is that this relatively simple system, often with only a few states, has the capability to simulate very complex systems and nevertheless it is possible to introduce different behaviour for different ‘players’. As a common problem in using such systems the ‘simultaneous application of the cells’ is scarcely feasible.

The state of the cells is also depending on the processing sequence of the cells.

### 3 The User Interface

Figure 1 shows the interface and the visualisation of the simulation, which is implemented as a Java applet. The screenshot depicts the model during the situation of a pass between German forwarders. Informations about the players are shown on the left side and can be achieved by clicking on the player. Informations like name, age, size or number help to identify the various players. More important is the possibility to change two variables for each player. The first possible change can be done on 'playing ability', the second change can be done on 'pass force'. This change can be done on every different player separately.

The user can change the overall system in the succession of computing of the different cells, shown in the lower left part of the picture.

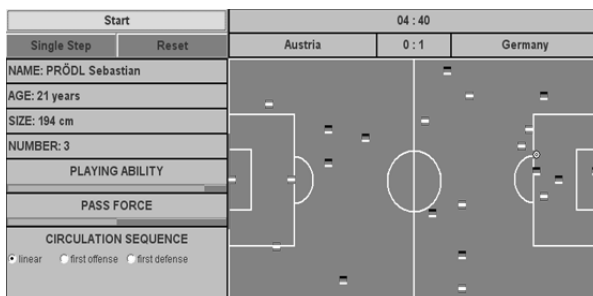


Figure 1. Screenshot – Situation of a pass.

### 4 Implementation

In order to realize the simulation a combination of a cellular automaton and an agent-based model was used. In contrast to other types of modeling in an agent-based model many small units (agents) have decisions or options for action. The system behavior results from the behavior of individual agents and will not be provided at the system level. Both modelling methods are 'bottom up' methods, so changes on system behaviour can be implemented easily by changing rules of the cells or agents in the system.

The cellular automaton is constructed by definition of the size of the automaton, the used neighbourhood definition, the different kind of states a cell can be in and the applied rules.

**Cell space.** the size of the playing field is defined by 50x29 cells, which is approximately the relation for an international playing field.

**Neighborhood.** Moore neighborhood (8-neighboring Cells) as we have tried to improve the influence of other players, which are near to the player holding the ball.

**Set of states.** empty, AUT player with ball, AUT players without ball, BRD player with ball, BRD player without ball, ball

**Transitional function:** Is defined in the form of rules

In order to limit the movement of the player the target field has been divided into three parts and each player is allocated by one part. The only adjustment in the APPLET is the processing sequence from the individual cells.

**Linear:** The cells pass through, as usual, from top left to bottom right.

**First Offense.** The playing field will be passed through twice in a linear process. In the first pass the cells of the attacking team will be processed, in the second step the ones of the defending team.

**First defense.** Analog to the first offense; the only difference is the reverse order.

The described characteristics of each player (name, age, size, number) as well as the playing ability and strength of pass will be stored in additional text files.

### 5 The Model

As mentioned, the basic structure of the model is 50x29 cells. At any pass, each non-empty cell is classified (team related, ball possession, attacking or defending team, goalie or field players) and, depending on the type of the cell, the matching routine, with the rule for this cell, is called. These rules evaluate the adjacent fields and allocate each field a positive natural number which indicates the unease of the player to go on this field. After evaluation of all the adjacent fields the cell with the lowest value will selected as a new site for the player. In case two or more cells have identical numbers a random process decides how the next step looks like.

In order to compensate the problems with the parallel processing of cells, the result will be stored in a second field, also 50x29 cells.

Thus, the move by more than one field of a player is avoided. This may occur when a player has moved from a line in the underlying line and will process again. Going forward, the whole game will be simulated step by step until the end. This, while also possible, shooting and kicking to a friendly player is a combination of a random process and the distance to the gate or the border area of the player. (Remember, each player is in one of the three areas.)

The direction of the shot or pass is calculated by an algorithm, in which the positions of the team mates and the opponents are taken into account. After the completion of a cycle, after all cells are processed, it is checked whether a defendable player is next to the ball carrier. If so, there will be a 'duel' in a random process, depending on the strength of the players which decides who is in possession of the ball.

In Figure 2 the situation of the German team is in possession of the ball. The player holding the ball is indicated by a small blue square. The red stop button indicates that the model is running. The buttons 'Single Step' and 'Reset' are unavailable in the running modus and therefore grayed out. The simulation can be stopped at any time to run the simulation in the single step mode or to reset the simulation run.

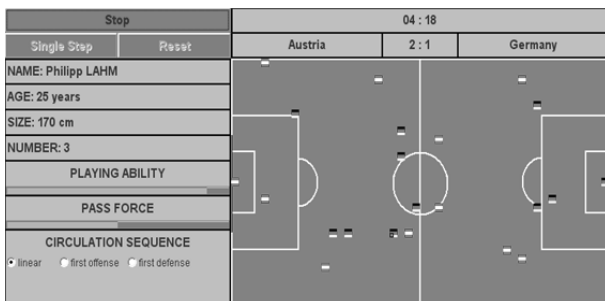


Figure 2. Screenshot – German Player be in possession of the ball.

## 6 Example of a Rule

The most important part of the model is the possibility to define and to integrate various rules. As mentioned at the beginning the rules can be adopted or changed more or less easily, which is one of the main advantages of the modelling technique.

In the following some examples of the mentioned 'rules' for the different types of cells (Offence with ball - Offence without ball - Offence Goal-Keeper – Defence - Defence Goal-Keeper - Player (only when Ball is free)) will be considered in more detail.

This is only one example of the whole set of rules, which have been implemented in the system. These examples for the integrated rules present how rules are described and how they influence the behaviour of the whole system.

### Rule for Offence with ball

The higher the value is the less suited the cell is for the next move:

(STEP 1) If the cell is already occupied, it makes no sense to consider the cell for the next step.

```

If (cell is not empty)
{
    Value = max.
} Else {
    Value = 0
}
    
```

(STEP 2) To constrain the range of each player on his third each cell will be controlled.

```

If (cell still in my third)
{
    Value = Value + 0
} Else {
    Value = Value + max.
}
    
```

(STEP 3) To reach a balanced distribution of players in their third, it is undesirable to approach to other teammates.

```

For (all teammates with distance < 7)
{
    Switch (approach to other teammates)
    {
        Case lower distance:    temp = temp
        + 6
        Case same distance:
        temp = temp + 3
        Case greater distance:  temp = temp
        + 0
    }
}
    
```

```

Value = Value + (temp/ teammates with distance < 7)
    
```

(STEP 4) This rule compares the strength of the attacker with his opponent's strength. If the attacker is very strong he tends to get a more, perhaps weaker, opponent to approach.

```

For (all opposing players)
{
  Switch (distance Enemy player)
  {
    Case 1 cell:
      Value = Value + (5 x force Opponents x (1
        - force Attackers))

    Case 2 cell:
      Value = Value + (3 x force Opponents x (1
        - force Attackers))

    Case 3 cell:
      Value = Value + (2 x force Opponents x (1
        - force Attackers))

    Case 4 cell:
      Value = Value + (1 x force Opponents x (1
        - force Attackers))

    Default:
      Value = 0
  }
}

```

(STEP 5) One of the most important rules is that the Offence comes closer to the goal.

```

Switch (come closer to goal)
{
  Case lower distance:
    Value = Value + 0
  Case same distance:
    Value = Value + 2
  Case greater distance:
    Value = Value + 4
}

```

The whole set of rules depending on additive system description (there are three different areas) and the individual abilities (playing ability, passing force) defines the behaviour of the system.

## 7 Experiment

In the experiment a number of successive games with different players'-strengths were realized and are shown in Table 1. The first 20 games were performed with the same skill and pass level (average 0.7 and 13.6) for both teams. The results of these games are shown in the table in the second column.

Afterwards 20 games with a reduced average strength (average 0.5) of the German team were performed. The strength of passport remained unchanged. These results are shown in the third column.

Game Nr.	Final outcome Equal force	Final outcome Advantage AUT
1	4:3	5:4
2	4:4	6:4
3	10:6	6:4
4	8:5	6:1
5	6:6	5:4
6	5:6	5:2
7	4:4	10:2
8	2:7	4:5
9	4:6	4:5
10	3:4	4:3
11	3:4	6:2
12	5:5	2:5
13	4:4	9:4
14	6:5	3:8
15	5:4	5:6
16	3:6	7:3
17	2:10	6:2
18	5:8	6:8
19	7:6	7:2
20	8:4	10:3
Won AUT	7	14
Won GER	8	6
Draw	5	0

Table 1. Result Table of Games.

As we can easily see the behaviour of the system (the result of the game) is influenced directly by the ability of the players. This is only one example of changing the parameters and sets of the simulation. In other experiments it can be analyzed f.e. how integrating one superstar (in teams with the same skills for the other players) would affect the results. Another point which could be analyzed is that the shown results in Table 1 are higher than we would expect in international games. This could be changed on one hand by variation of skills between defeners and other players, but also by redefining the areas of the place.

## References

- [1] S. Wolfram. *Cellular Automata*. 1983, <http://www.stephenwolfram.com/publications/articles/ca/83-cellular/1/text.html>
- [2] W. Golder. *Beziehungskiste aka Party-Planer*. <http://www.golder.de/psion/bkiste/main.html>
- [3] Wikipedia. *Zellulärer Automat*. [http://de.wikipedia.org/wiki/Zellulärer\\_Automat](http://de.wikipedia.org/wiki/Zellulärer_Automat)
- [4] Wikipedia. *Agentenbasierte Modelle*. [http://de.wikipedia.org/wiki/Agentenbasierte\\_Modelle](http://de.wikipedia.org/wiki/Agentenbasierte_Modelle)

Submitted: August, 2010

Revised: February, 2011

Accepted: March3, 2011