

COACH UNILANG - A Standard Language for Coaching a (Robo)Soccer Team

Luis Paulo Reis¹ and Nuno Lau²

¹ LIACC - Artificial Intelligence and Comp. Sci. Lab. - Oporto University, Portugal
lpreis@fe.up.pt, <http://www.ncc.up.pt/liacc/>

² DET - Electronics and Telecommunications Dep. - Aveiro University, Portugal
lau@det.ua.pt, <http://www.det.ua.pt/>

Abstract. This document introduces COACH UNILANG, a standard language for coaching (Robo)Soccer teams. This language was developed with two main objectives: to coach FC Portugal 2001 team and as a proposal to be used in Fukuoka 2002 RoboCup coach competition. This language enables high-level and low-level coaching through coach instructions. High-level coaching includes changing tactics, formations used in each situation and changing player behavior. Low-level coaching includes defining formations, situations, player behavior and positioning with high detail. The language also enables the coach (functioning like an assistant coach) to send opponent modeling information and game statistical information to the players.

1 Introduction

A standard language for coaching must be prepared considering the state of the art in simulated RoboCup [1] and its predictable evolution [7]. Games in RoboCup2001 will be, most likely, similar to the FC Portugal vs Brainstormers games in Melbourne (similar to real soccer games). So, a standard language definition must be sufficiently powerful to enable coaching teams like FC Portugal [2,3], Brainstormers [4], CMUnited [6] or other good teams, which are able to play well without using the coach. To be useful, the information and advice given by the coach to the players must be high-level consisting of things that the players are not able individually (or through communication) to spot by themselves.

This paper introduces COACH UNILANG, a general language for coaching robosoccer teams. In Section 2 some possible coaching architectures are discussed along with usual coaching concepts. Section 3 fully describes the requirements of our language. Section 4 introduces COACH UNILANG and describes its message types. The syntax and semantics of the language are described in Section 5. Finally we give some conclusions and give an outlook to future research.

2 Coaching Architectures, Levels, and Concepts

COACH UNILANG enables coaching at several different levels. We introduce the assistant coach agent that is able to analyze a game (directly or through

a log file) and gather opponent modeling and game statistics information at several levels. This agent may be used to assist the coach providing him with high-level information that may be used more efficiently. The assistant coach may be used as a single agent (Fig. 1a) or integrated into the principal coach (functioning both as a single agent). This architecture (Fig. 1b) will be FC Portugal coach architecture for RoboCup 2001. The principal coach may also be included inside the player agents (Fig. 1c). If the principal coach only needs the opponent modeling and game statistical information provided by the assistant coach then he might be included inside the players. In this architecture, players are responsible for changing the team tactic in use (using the same algorithm). In the last architecture (Fig. 1d) the coach is not an autonomous agent. This was FC Portugal 2000 coaching architecture, in which players used simple game statistics and opponent modeling information to change team tactic.

COACH UNILANG enables coaching at several different abstraction levels:

- **Level 1 - Instructions.** Level 1 coaching is intended to train teams with intelligent players that are used to play together. It enables high-level coaching but it is not very flexible because it uses generic (and fixed) soccer concepts.
- **Level 2 - Statistics + Opponent Modeling.** Level 2 coaching is intended to help a coach to train a team of players. It is the language used by our assistant coach to talk with the coach (although the rules do not allow two coaches, they are implemented as a single agent)! The assistant coach is concerned with calculating game statistics (like ball possession by regions, action result, etc.) and trying to get opponent modeling information.
- **Level 3 - Definitions + Instructions.** This is a very flexible coaching level. It enables the coach to redefine the standard soccer concepts at a much more low-level. It enables both high-level coaching and low-level coaching.

COACH UNILANG is based on several real soccer concepts. These include:

- **Field Regions** - corresponding to areas of the field (mostly rectangles but also points and other type of regions are allowed)
- **Time Periods** - correspond to periods of the game (with a given beginning time and duration);
- **Tactics** - enabling high-level configuration of team behavior by defining offensive or defensive characteristics, game pace, pressure placed on the opponent's team, formations used for different situations, etc.

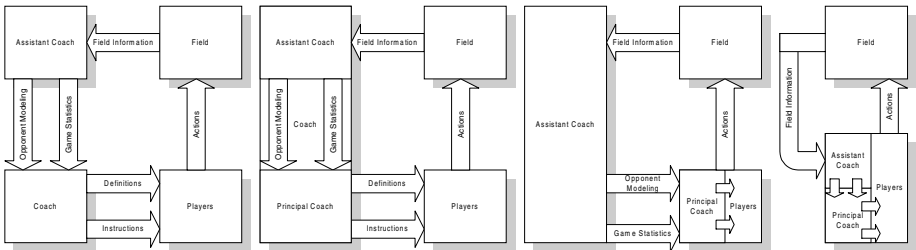


Fig. 1. Coaching architectures possible using COACH UNILANG

- **Formations** - spatial distributions of the players in the field.
- **Situations** - high-level analysis of the game state. Typical situations include offensive and defensive situations, from defense to attack, corners, etc.
- **Player Types** - defining players' behaviors while in possession of the ball and without the ball.

3 Requirements for Our Coaching Language

Our format tries to compromise between generality and simplicity. It is sufficiently general to allow representing the most common variants of coaching strategies but has also a simple syntax and enables coaching at different abstraction levels. The main requirements in the elaboration of this format were:

- It should be independent of implementation details and coaching strategies;
- It should be small but easy to extend including new concepts and definitions;
- Existing languages and coaching strategies should be easily expressed;
- It should be easily readable by the agents but also by the human user;
- Information not directly concerned with coaching should not be included;
- The format should be general enough and multi-resolution to enable the representation of coaching strategies at different abstraction levels;
- Semantics should be completely clear. All concepts must be clear to people with basic soccer and robosoccer knowledge;
- It should enable to represent regions, time periods, tactics, formations, player types and situations;
- Normal soccer regions, tactics, formations, player types and situations should be included as default but new ones must be easily definable;
- It should be robust with simple data validation and override of usual errors.

4 Message Types

COACH UNILANG has four main message types:

- **Definition.** Enables the coach to define a given concept (region, time period, tactic, formation, situation or player type);
- **Statistics.** Enables the coach to broadcast some game statistics;
- **Opponent Modeling.** This message type is used to inform the players about opponent (or own) team characteristics (at several levels).
- **Instruction.** Instruction messages enable the coach to change the different parts of the team tactic (formations used, team pressure, game pace, playing style, player behaviors, etc.)

Each message is composed of a time, identification and one or several parts. Messages may have one of four types: Definition, statistics, opponent modeling or instructions. The language uses several types that include integers, sets, qualitative scales (very high, high, medium, low and very low), etc.

```

<MESSAGE> ::= (<TIME> <ID> <MESSAGE PART> {<MESSAGE PART>})
<MESSAGE PART> ::= <DEFINITION_MESS> | <STATISTICS_MESS> | <OPP_MOD_MESS> | <INSTRUCTION_MESS>
<DEFINITION_MESS> ::= (definition <ID> <DEFINITION >)
<STATISTICS_MESS> ::= (statistics <ID> <STATISTICS>)
<OPP_MOD_MESS> ::= (opp_mod <ID> <OPP_MOD>)
<INSTRUCTION_MESS> ::= (instruction <ID> <INSTRUCTION>)
<ID> ::= [integer] <TIME> ::= [integer] <VALUE> ::= vhi|hi|mid|low|vlow
<TEAM> ::= our|oppon|any <PLAYER> ::= [1..14]|any <POS_NUMB> ::= [1..11]|any

```

5 Coach Language

5.1 Definition Messages

Definition messages allow the coach to define concepts used during the game. These concepts include regions, periods, tactics, formations, situations and player types (active and recovery behavior). Concepts are identified by a name so that they may be referenced in future coach messages.

```
<DEFINITION> ::= (region <REGION_NAME> <REGION>) | (period <PERIOD_NAME> <PERIOD>) |
  (tactic <TACTIC_NAME> <TACTIC>) | (formation <FORMATION_NAME> <FORMATION>) |
  (situation <SITUATION_NAME> <SITUATION>)|(active_type <ACTIVE_TYPE_NAME> <ACTIVE_TYPE>)|
  (recovery_type <RECOVERY_TYPE_NAME> <RECOVERY_TYPE>)
```

5.2 Regions

One of the most common soccer concepts is the region concept. Usually, coaches, players and journalists communicate speaking about known regions like the penalty box, the left wing, etc. COACH UNILANG provides a predefined usual soccer field regions and a simple mechanism to define new regions. Predefined regions include a simple multi-resolution field description.

Regions may be defined as simple regions or as a list of simple regions (interpreted as a union of those simple regions). Simple regions may be the predefined regions included in the language but also user defined regions and regions defined directly. Regions may be defined using points, rectangles, quadrangles, circles or arcs. In the standard region definition, trying to let it remain simple, we included only rectangles:

```
<REGION> ::= <SIMPLE_REGION> | (<SIMPLE_REGION> {<SIMPLE_REGION>})
<SIMPLE_REGION> ::= <PREDEFINED_REGION> | <REGION_NAME> | <REGION_DEFINITION>
<PREDEFINED_REGION> ::= 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 | ...
<REGION_DEFINITION> ::= <POINT> | (rect <POINT> <POINT>) |
  (quad <POINT> <POINT> <POINT> <POINT>) | (circle <POINT> [real]) |
  (arc <POINT> [real] [real] [real] [real])
<POINT> ::= (point [real] [real])
```

5.3 Time Periods

Time is very important in simulated robosoccer and most of the analysis that a coach must perform, are connected with time periods. COACH UNILANG provides a simple mechanism for defining time periods. They may be predefined time periods (defined as general concepts in the language), periods defined by the coach during the game (using the define directive) or periods defined directly when used. Predefined periods include the whole game, first half, second half, extra time, last 1000 cycles, etc. To define other periods the coach may define them in terms of the begin and end times or in terms of length:

```
<PERIOD> ::= <PREDEFINED_PERIOD> | <PERIOD_NAME> | <PERIOD_DEFINITION>
<PREDEFINED_PERIOD> ::= game | first_half | second_half | extra_time | last_1000 |
  last_500 | last_300 | last_100 | last_50 | last_20
<PERIOD_DEFINITION> ::= (<TIME> <TIME>) | <LENGTH>
<TIME> ::= [integer] | now          <LENGTH> ::= [integer]
```

5.4 Tactics

Tactics are probably the most complex soccer concept. In a given tactic, other complex concepts are included, like: formations, game pace, team pressure, game type, playing style, offside trap use, etc. At each moment in a game, a team uses a given tactic. However, the coach according to the game course and opponent team behavior may change this tactic. Our definition of what is a tactic is based on common soccer concepts like team mentality, game pace, team pressure, field use to develop collective moves, offside trap, risk taken, etc. The following tactic characteristics are used:

- **Team mentality** (very offensive, offensive, normal, defensive, very defensive). A team with an offensive mentality is thinking about attacking while it is defending. Some players may stay in the middle field waiting to attack instead of defending. In a team with a defensive mentality players are mainly concerned with defending. Even while the team is attacking players remain in useful defending positions trying to prevent opponent's counter attacks.
- **Game pace** is concerned with the speed imposed to the offensive and defensive game of the team. Teams with high game pace (for example usual English teams) use high speed for all moves. Teams with low game pace will move the ball slowly. Players will first stop the ball and take their time to decide what to do with the ball. Also, players will only move fast in the field if their active participation may be important for the team, otherwise they will prefer to keep their stamina for more important moves.
- **Team pressure** defines the pressure made over the ball when the team is not in possession. When high team pressures are used, the team players tend to be much more concentrated in the region where the ball is located in the field and more players challenge the opponent in possession. More than one player tries to pressure the attacker with the ball and also several players must give support covering his possible passes and dribbles. Defense may get somewhat unbalanced but ball recovery is usually much faster if the opponent team is not a very good one.
- **Field use** concerns the use of the different regions of the field to develop offensive moves. Teams may use a balanced system or use systems that give priority to moves developed using the wings of the field. Also teams may be more enthusiastic to use the middle of the field to develop offensive moves (for example if the opponent covers well both wings).
- **Playing style** is concerned with the type of attack and defense used. Attacking style is mostly concerned with the type of actions used by the players to attack. These actions may be mainly short passes and dribbles (passing_game also known as possession play) or they may be mainly long passes, forwards and runs by the players (long_ball also known as direct play). Defense type is mainly concerned with the type of marking of the players (individual marking or zonal defense).
- **Risk taken** defines the risk players are willing to take while conducting defensive and offensive moves. If a high-risk tactic is used, then players will try risky interceptions and risky passes. When the team is losing a given game, risk is needed to recover quickly the ball and to construct fast offensive moves. If a low degree of risk is used, then players will only try safe moves.
- **Offside tactic** concerns the measures taken by the team to catch opponents in offside and to prevent team members from being caught in offside. The offside-trap pushes defenders up in a line and they will be attempting to push out of defense

quickly after an opposition attack in the hope that they can catch opposition attackers offside.

- **Positioning exchange** use is concerned with the freedom of the players to exchange positions with their teammates. If positioning exchange is high, players will try to exchange positioning to confuse the opponent’s defense.
- **Formations.** One of the more important characteristics of the tactic is concerned with the formations used for different situations. For example, a team may use a narrow 433 formation to defend and a wide 235 formation to attack. Our language lets the coach define the formations that may be used in each situation and their shape (in terms of horizontal and vertical opening).

```
<TACTIC> ::= <PREDEFINED_TACTIC> | <TACTIC_NAME> | <TACTIC_DEFINITION>
<PREDEFINED_TACTIC> ::= normal | scoreLots | counter_attack | keep_possession |
<TACTIC_DEFINITION> ::= <MENTALITY> <GAME_PACE> <PRESSURE> <FIELD_USE> <PLAYING_STYLE>
  <RISK_TAKEN> <OFFSIDE_TACTIC> <POS_EXCHANGE_USE> <FORMATIONS_USED>
<MENTALITY> ::= very_offensive | offensive | normal | defensive | very_def
<GAME_PACE> ::= sleep | ease_up | normal | speed_up | give_evrthng
<PRESSURE> ::= no_pressure | standoff | normal | pressure | high_press
<FIELD_USE> ::= wings | middle | balanced | right_wing | left_wing
<PLAYING_STYLE> ::= (<ATTACKING_STYLE> <DEFENDING_STYLE>)
<ATTACKING_STYLE> ::= short_passing | passing | normal | long_ball | verylong_ball
<DEFENDING_STYLE> ::= zonal_defense | normal | individual_marking
<RISK_TAKEN> ::= very_safe | safe | normal | risky | very_risky
<OFFSIDE_TACTIC> ::= (<OFFSIDE_TRAP_USE> <OFFSIDE_TRAP_CAREFUL>)
<OFFSIDE_TRAP_USE> ::= <VALUE> <OFFSIDE_TRAP_CAREFUL> ::= <VALUE>
<FORMATIONS_USED> ::= { (<SITUATION> <FORMATION> <FORMATION_SHAPE> ) }
<FORMATION_SHAPE> ::= <FORMATION_WIDTH> <FORMATION_HEIGHT>
<POS_EXCHANGE_USE> ::= <VALUE> <FORMATION_WIDTH> ::= <VALUE> <FORMATION_HEIGHT> ::= <VALUE>
```

Although we have several predefined tactics in FC Portugal, the definition of common soccer tactics is not an easy task and there are no globally accepted definitions for standard soccer tactics.

5.5 Formations

Formations describe the spatial distribution of the players in each situation. COACH UNILANG includes predefined common soccer formations (433, 442, 343, etc.) It also includes a high-level formation definition mechanism based on an imaginary 7*5 rectangular board. Figure 2 shows the use of this board to define some common soccer formations.

Formation definition also includes the definition of player types for each of the player in the formation. Usually these player types are something like a striker, a sweeper, a central defender, a left-winger, etc. In COACH UNILANG the definition of player types (section 5.7) is performed at two different levels: behavior with the ball and behavior without the ball. This way we may state that a player is an aggressive central defender or a positional striker, etc:

```
<FORMATION> ::= <PREDEF_FORMATION> | <FORMATION_NAME> | <FORMATION_DEFINITION>
<PREDEF_FORMATION> ::= 433 | 433att | 442 | 442att | 442diamond | 343 | 4123 | 352 | 532 |
  41212 | 4132 | 424 | 451 | 541 | 3223 | 3232
<FORMATION_DEFINITION> ::= { (<PLAYER> <POSITIONING_NUMBER> <PLAYER_POSITIONING> <PLAYER_TYPE> ) }
<PLAYER_POSITIONING> ::= <VERTICAL_POSITIONING> <HORIZ_POSITIONING>
<VERTICAL_POSITIONING> ::= gk | sweeper | defense | defense_middle | middle | middle_forward | forward
<HORIZ_POSITIONING> ::= left | left_center | center | right_center | right
```

5.6 Situations

Situations refer to a high-level analysis of the world state. Basic situations include attack, defense, passing from attack to defense, scoring opportunities, goalie free kicks, goal kicks, kickoffs, corners, kickins and free kicks.

```

<SITUATION> ::= <PREDEF_SITUATION | <SITUATION_NAME> | <SITUATION_DEFINITION>
<PREDEF_SITUATION> ::= attack | defend | defense_to_attack | attack_to_defense |
    our_opportunity | their_opportunity | our_goalie_kick | their_goalie_kick
<SITUATION_DEFINITION> ::= <CONDITION>
<CONDITION> ::= (true) | (false) | (team_in_possession <TEAM><PERIOD>) |
    (ball_pos <REGION>) | (ball_velocity <VALUE>) | (ball_possession <TEAM> <PLAYER>) |
    (player_position <TEAM> <PLAYER> <REGION>) | (not <CONDITION>) |
    (and <CONDITION> <CONDITION>) | (or <CONDITION> <CONDITION>)
    
```

Using our language we may define easily situations like "attack": (situation attack (and (ball_position their_middle_field) (ball_possession our any))).

5.7 Player Types and Actions

Player types refer to player behaviors with and without ball according to what is happening in the field. Active type refers to the player behavior with ball and recovery type refers to the player behavior without ball. We define seven possible actions with ball: shoot, pass, forward (pass not directly to a player but to a point where the player is supposed to collect the ball almost stopped), dribble, run with the ball, hold the ball and clear (through the ball deliberately outside the field). We also define seven possible actions without the ball (recovery actions). These include: interception (intercept a moving ball in the field), tackle (steal the ball that an opponent has controlled), mark a given passing line, mark individually a player, go to the ball position (although not being able to intercept it) and move strategically in the field (respecting the formation in use).

We also defined six different action results (with ball): success (if the action is a shoot then a goal was achieved, etc.), opp_goalie_catch (opponent goalie catch the ball), our_goalie_catch, ball went out of field bounds, interception by the opponent, stolen by the opponent (by a tackle).

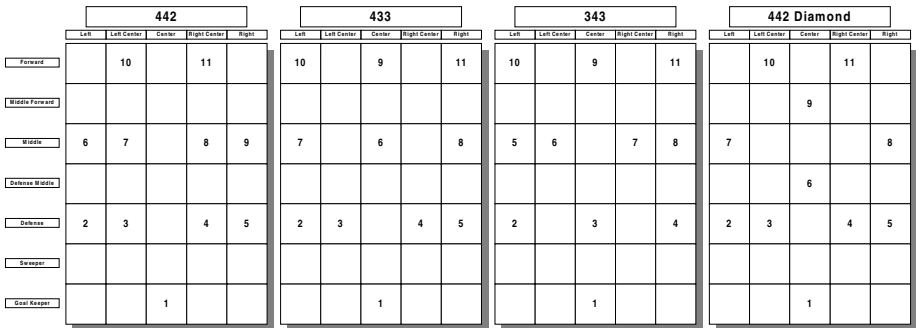


Fig. 2. Definition of some common soccer formations

For recovery actions, besides normal action results, the following action results are also defined: `stolen_other` (other player took the ball from the opponents while this player was conducting the given recovery action) and `intercepted_other` (other player intercepted the ball while the player was conducting the recovery action):

```
<ACTION> ::= shoot|pass|forward|dribble|run|hold|clear|any
<RECOVERY> ::= interception|tackle|mark_pass_line|mark_player|goto_ball|strategical_move|any
<ACTION_RESULT> ::= success|opp_goalie_catch|our_goalie_catch|out|interception|stolen|fail|any
<REC_RESULT> ::= <ACTION_RESULT>|stolen_other|intercepted_other
<ACTION_PARAMETERS> ::= (<DISTANCE> <POWER> <SAFETY>)
<RECOVERY_PARAMETERS> ::= (<DISTANCE> <POWER> <SAFETY>)
```

To define a player type, the coach must define the player behavior with ball (active type) and without ball (recovery type). Some simple predefined behaviors are available for active types and recovery types. Some qualitative parameters are available for the different ball possession actions and recovery actions available. These include the distance, power and safety of the action to perform. For example, a coach may state that in the team's defensive middle field, the passing action shall be safe. He may also state that in the attacking third, safety of actions is not a concern.

```
<PLAYER_TYPE> ::= <ACTIVE_TYPE> <RECOVERY_TYPE>
<ACTIVE_TYPE> ::= <PREDEF_ACTIVETYPE>|<ACTIVE_TYPE_NAME>|<ACTIVETYPE_DEF>
<PREDEF_ACTIVETYPE> ::= basic|normal|shooter|dribbler|passer
<ACTIVETYPE_DEF> ::= ({{<ACTION> <REGION_FROM> <REGION_TO> <ACTION_PARAMETERS> <VALUE>}})
<RECOVERY_TYPE> ::= <PREDEF_RECTYPE>|<RECOVERY_TYPE_NAME>|<RECTYPE_DEF>
<PREDEF_RECTYPE> ::= normal | positional | marker | aggressive
<RECTYPE_DEF> ::= ({{<RECOVERY> <REGION> <RECOVERY_PARAMETERS> <VALUE>}})
```

The definition of ball possession behaviors in simulated robosoccer is rather complex. This way, our predefined active types definition is quite basic relatively to what is needed to construct good soccer players. It is completely impossible to define or configure the behavior of a decent soccer player using only 2 or 3 conditions. Definition of good ball recovery behaviors is simpler than the definition of good ball possession behaviors. The following statement shows a simple definition of a recovery player type: (recovery_type aggressive ((interception field (any any any) vhi) (tackle field (any any any) vhi) (goto_ball field (any any any) hi) (mark_player field (any any any) vlow) (mark_passline field (any any any) vlow) (strategical_move field (any any any) low))).

5.8 Assistant Coach Language

The assistant coach gathers game statistical information and opponent modeling information and communicates it to the coach and/or to the players. Opponent Modeling Information is concerned with opponent characteristics spotted by the coach. We divide this information into four abstraction levels: social level (team information), individual decision level (players and goalie), low-level skills (not concerned with decision mechanisms) and physical characteristics (like speed, recovery, decay, inertia, acceleration, size, kickable margin and kicking accuracy).

Statistical information contrarily to opponent modeling information is not directly concerned with the opponent behavior but is more concerned with the game course at several levels. The definition of the statistical information included in COACH UNILANG is based on the information used by real soccer

coaches and information available in computerized video analysis soccer systems like Match Analysis [5]. Statistical information includes game occurrences, action number and results (by region, team, player and period), ball possession (of each team and player in each region and period), player positions in the game, assistances to shoots on goal, ball losses, ball recoveries and ball circulation information. Due to space limitation we cannot describe here the syntax and semantics of our assistant coach language.

5.9 Coach Instructions

One of the innovations of COACH UNILANG is the fact that it enables very high-level coaching (although not limiting in any way, low level coaching). To change the team strategy, the coach may use the instruction directive.

```
<INSTRUCTION> ::= (change_tactic <TACTIC>)|(change_mentality <MENTALITY>)|
  (change_game_pace <GAME_PACE>)|(change_risk_taken <RISK_TAKEN>)|
  (change_field_use <FIELD_USE>)|(change_playing_style <PLAYING_STYLE>)|
  (change_pressure <PRESSURE>)|(change_offside_tactic<OFFSIDE_TACTIC>)|
  (change_pos_exchange_use <POS_EXCHANGE_USE>)|
  (change_formation <SITUATION_NAME> <FORMATION> <FORMATION_SHAPE>)|
  (change_formation_pos <FORMATION_NAME> <POS_NUMBER> <PLAYER_POS> <PLAYER_TYPE>)|
  (change_player_type <PLAYER_TYPE_NAME> <ACTIVETYPE_DEF> <RECTYPE_DEF>)
```

The coach may change the tactic in use completely (`change_tactic`) or he may change the different parts of that tactic. He may change the team mentality, game pace, team pressure, field use, playing style, offside tactic and, positioning exchange use. Also he may change the formation used (on the tactic in use) for a given situation and that formation's shape.

The coach may also change a single positioning and player behavior of one of the formations. This operation is performed using the `change_formation_pos` directive. Using this directive, for example, the coach may say that (`change_formation_pos 433 10 (left_center middle_forward) (striker aggressive)`). This means that for the formation 433, the positioning number 10 that had the value (left forward) will now be (left_center middle_forward). Also this instructs the player occupying that positioning to behave like an aggressive (ball recovery) striker (ball possession). The final directive enables the change of a given player type, changing some of its parameters.

6 Conclusions and Future Work

This is, to our knowledge, the first attempt to create a language that enables high-level coaching of robo(soccer) teams. Another attempt to create a standard language for coaching robosoccer teams was made by Patrick Riley, Gal Kaminka and Timo Steffens [8]. This later language does not include high-level concepts like tactics, formations, player types, time periods or situations (essential to define the behaviour of a soccer team). It only enables the definition of player's home positions and several directives concerning individual action behaviours (like which player to pass the ball or which player to mark). Also, this language includes some concepts derived from COACH UNILANG (whose

draft proposal was made public on January 2001) like definitions for conditions, actions and directives, marking passing lanes and play mode based conditions. This "standard language" was used in a "coaching competition" in Seattle. As expected, no visible tactical changes were seen in any of the teams that did use this "standard language" and the competition was considered by most of the participants as "somewhat random". Opposite, FC Portugal 2001 did change its tactic and even played, when losing near the end of a game, in a very aggressive 433 formation (without a goalie)!

Some people may argue that the definition of tactics and formations is too high-level. However, we argue that COACH UNILANG enables the definition of tactics and formations at any level. The coach may use the high-level instruction mechanisms or he may define a tactic, defining all used formations and the behavior of each player used in each formation at low level (defining for each region what are the player preferred actions and configuring them). This enables virtually the definition of any soccer tactic. Other critique that can be made to this language is that it is too complex. However, soccer is a complex game. So, there is no interest in developing a language that is able to train only teams that use simple decision mechanisms and is not able to play like real soccer teams (capturing and using the real soccer complexity).

Future work is related with the discussion of this language with soccer experts: professional coaches, players and journalists enabling us to refine the language. Future work will also be related with the adaptation of FC Portugal team enabling it to be fully trained using COACH UNILANG. We also plan to extend COACH UNILANG with information regarding free kick takers, studied offensive moves, goal keeper instruction and player vision advice. Also we plan to develop player-coach and player-player languages. These languages will effectively enable the construction of teams with coach, coach assistant and players developed by different people that are able to play together as a real soccer team.

References

1. Itsuki Noda et al.: Soccer server: A Tool for Research on Multiagent Systems. Applied Artificial Intelligence, Vol. 12, pp.233-250, 1998
2. Luis Paulo Reis and Nuno Lau, FC Portugal Team Description: RoboCup 2000 Simulation League Champion, pp.29-40, in [7]
3. Luis Paulo Reis, Nuno Lau and Eugenio Oliveira. Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team, in M. Hannebauer et al eds, Balancing Reactivity and Social Deliberation in MAS, Springer LNAI 2103, pp.175-197, 2001
4. Martin Riedmiller et al. Karlsruhe Brainstormers 2000 - A Reinforcement Learning Approach to Robotic Soccer. pp.367-372, in [7]
5. Match Analysis, On Line, available as URL: <http://www.matchanalysis.com/>
6. Peter Stone, Patrick Riley and Manuela Veloso, The CMUnited99 Champion Simulator Team, RoboCup99: Robot Soccer World Cup III, LNAI 1856, Springer, 2000
7. Peter Stone, Tucker Balch and Gerhard Kraetzschmar, editors, RoboCup-2000: Robot Soccer World Cup IV, Springer, LNCS 2019, 2001
8. Noda et al, RoboCup Soccer Server Users Manual, RoboCup Federation, June 2001