METHODOLOGIES AND APPLICATION

An automatic approach to extract goal plans from soccer simulated matches

**Fernando Almeida · Pedro Henriques Abreu ·
Nuno Lau · Luís Paulo Reis**

Abstract Soccer is a competitive and collective sport in which teammates try to combine the execution of basic actions (cooperative behavior) to lead their team to more advantageous situations. The ability to recognize, extract and reproduce such behaviors can prove useful to improve the performance of a team in future matches. This work describes a methodology for achieving just that makes use of a plan definition language to abstract the representation of relevant behaviors in order to promote their reuse.

F. Almeida
DI, Superior School of Technology of Viseu, Campus
Politécnico de Repeses, 3504-510 Viseu, Portugal
e-mail: falmeida@di.estv.ipv.pt

F. Almeida
IEETA, Campus de Santiago, 3810-193 Aveiro, Portugal

F. Almeida · L. P. Reis
LIACC, Artificial Intelligence and Computer Science
Laboratory, University of Porto, Rua Dr. Roberto Frias s/n,
4200-465 Porto, Portugal
e-mail: lpreis@dsi.uminho.pt

P. H. Abreu (✉)
DEI, CISUC, Faculty of Sciences and Technology,
Rua Sílvio Lima, University of Coimbra, Pólo II,
3030-790 Coimbra, Portugal
e-mail: pha@dei.uc.pt

N. Lau
DETI, IEETA, University of Aveiro, Campus de Santiago,
3810-193 Aveiro, Portugal
e-mail: nunolau@ua.pt

L. P. Reis
EEUM, School of Engineering, University of Minho, DSI,
Portugal Campus de Azurm, 4800-058 Guimarães, Portugal

Experiments were conducted based on a set of game log files generated by the Soccer Server simulator which supports the RoboCup 2D simulated robotic soccer league. The effectiveness of the proposed approach was verified by focusing primarily on the analysis of behaviors which started from set-pieces and led to the scoring of goals while the ball possession was kept. One of the results obtained showed that a significant part of the total goals scored was based on this type of behaviors, demonstrating the potential of conducting this analysis. Other results allowed us to assess the complexity of these behaviors and infer meaningful guidelines to consider when defining plans from scratch. Some possible extensions to this work include assessing which plans have the ability to maximize the creation of goal opportunities by countering the opponent's team strategy and how the effectiveness of plans can be improved using optimization techniques.

Keywords Game analysis · Set-play · Cooperative behavior · Plan extraction

1 Introduction

In recent years, the growing need and interest in performance analysis have led to new forms of match analysis techniques. Modern-day techniques include video-based statistical analysis systems, video-based tracking and electronic tracking systems (Carling et al. 2007; Abreu et al. 2010c).

Despite the performance gap being greater among robotic soccer teams (in comparison to human soccer teams (Abreu et al. 2012), the importance of preparing a robotic team for the next opponent occupies a similar role in achieving a game victory. Normally, in the preparation of a match, a robotic

soccer coach divides it in two distinct phases: the first consists of, using past games, detecting an opponent play pattern in order to select the best strategy to neutralize it (offline phase). The second phase consists of, during the game, analyzing the behavior of his opponent and adapting his team strategy in order to win the game (online phase).

Englobed in a large project related with the automatic improvement of soccer team performance using high level information, the first step of this research work consists of identifying useful complex cooperative behaviors that occur in a soccer match (offline phase). In particular, this work focuses on extracting such complex cooperative behaviors that lead to the scoring of a goal. These behaviors are referred as goal plans from here on. For that a framework that allows the extraction of complex behaviors was developed and to promote the reuse of such behaviors a set-play (Mota et al. 2010) framework was used to characterize these behaviors. Also, a set of RoboCup log files of the 2010 2D competition was used to characterize the behavior of robotic soccer teams.

The achieved results were very promising and proved unequivocally that goal plans that start from set-pieces[1] play an important role in what concerns to team performance (25 % of the total of scored goals in this manner). In the future, this research work will be included in a large soccer analysis framework capable of automatically improving a soccer team performance.

The remainder of this paper is organized as follows: Sect. 2 describes the related work in this research area. Sections 3 and 4 expose the set play framework as well as all the steps related to the set plays extraction (including the event extraction). Section 5 presents the methodology used to validate the proposed approach. Section 6 exposes the achieved results and finally in the last section final conclusions are presented.

2 Related work

Over the years, several researches were developed using the RoboCup soccer environment (Noda et al. 1998), many of which were related to opponent modeling (Riley and Veloso 2001, 2006; Riley et al. 2002; Steffens 2004a, b, 2005; de Dios Bulos et al. 2005; Fathzadeh et al. 2006, 2008; Iglesias et al. 2009a; Ledezma et al. 2009). Most of these opponent modeling works focused on a coach agent, who has a restricted communication with his players, and

on how to improve the performance of a team using a standard language called CLang.[2]

Enclosed in the RoboCup Soccer competition, the latest RoboCup Coach League had as its most relevant goal the development of an advisor agent (called coach), capable of detecting behavioral patterns of simulated soccer teams. To achieve that goal, this agent receives global and noiseless information from the platform that supports the competition (Soccer Server), including ball and players' position and speed. The architecture of this competition depicted in Fig. 1 is built upon two main concepts: Play Pattern and Base Strategy.

Play Pattern describes a predictable behavior executed by a team that can be exploited by a coach while the Base Strategy is more concerned with the general strategy followed by a team. This strategy can be composed of a set of distinct patterns. At the start of the competition, the organization members create a set of strategies that can be used in the base strategies of the patterns and some matches are played to generate No-Pattern Log Files (NPLF). After that patterns are added to basic strategies and some sample matches are played again to generate Pattern Log Files (PLF), creating many NPLF and PLF pairs. When the competition starts (from an agent's perspective), each coach receives a set of PLF containing only one specific pattern activated and its corresponding NPLF (containing the same base strategy but without any activated pattern). After receiving those logs, the coach agent should recognize the pattern and report it (executing a matching between the PLFs and NPLFs). This phase is designated as the offline phase. In the next step (online phase) the coach should recognize the set of patterns used by a test team in a live game. The sooner the coach sends its pattern recognition reports, the higher the score it gets if they are accurate.

Despite the end of this competition in 2006, it sparked a growing interest in the area of agent/team modeling, leading to the pursuit of many research works in several domains such as knowledge representation (Gonzalez et al. 2008; Abreu et al. 2010a), game analysis (Abreu et al. 2010b, 2011), human imitation (Bauckhage et al. 2003; Aler et al. 2009), game event detection (Kuhlmann et al. 2005; Portela et al. 2010) and opponent classification. In the last domain, some works need to be highlighted. Stone et al. (2000) presented a low-level positioning and interaction agent approach based on an ideal world (where the performance of the opposing team is always at its maximum). However, in this approach, the process of positioning adaptation does not change throughout the game and is independent of the opposing team (it is a generic approach) which constitutes severe limitations. An extension of this work is proposed by

---

[1] Game situation in which the ball is returned to open play after a stoppage due to the ball going out-of-bounds or a call made by the referee. In a soccer match this includes kick-offs, goal-kicks, throw-ins, corner-kicks and free-kicks. In this work, the latter is also used to refer to indirect free-kicks.

[2] More information available online at http://sourceforge.net/projects/sserver/.

Fig. 1 Coach competition architecture (adapted from Iglesias et al. 2009b)
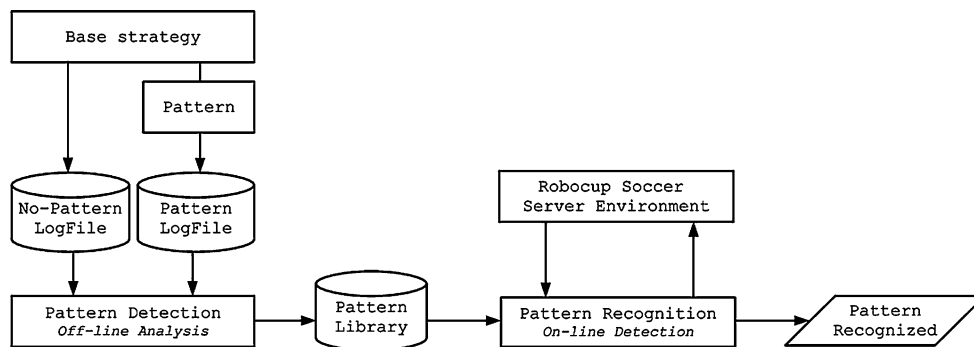
Ledezma et al. (2005), with the main goal of improving the low level skills of the modeled agent. Druecker et al. (2000) used a neural network to identify the opposing team formation (however, the information obtained seems to be very limited in its ability to improve the performance of a team). Similar to Druecker, Riley presented a learning formation approach based on players positions (Riley et al. 2002). However, the limitations presented by this study are similar to the previous one.

In conclusion, it is clear that many studies have tried to solve the problem of modeling the behavior of opponent teams, detecting multi-valued variables that can improve the team performance, such as team formation. However, there is no approach that using high level information (such as set-plays Mota et al. 2010) is capable of automatically detecting the pattern behavior of an opponent and reuse it to advise the best tactic/strategy to adopt in order to defeat an opponent (optimizing the team performance). This research work constitutes a first step towards achieving that goal.

3 Set-play framework

The set-play framework (Mota et al. 2010) provides a language specification for defining plans (set-plays) for the soccer domain, a built-in parser and an execution engine that allows them to be interpreted and executed at run-time.

These set-plays can be particularly useful in some situations (e.g. there are empty spaces in the opponent's penalty area) to help the team achieve a competitive edge. Moreover, the continuous improvement of soccer agents tactics and skills will require the development of new strategies to counter them and thus the definition of set-plays can prove useful for this purpose. Set-plays can also be a means to create mixed teams (formed by heterogeneous players) because players only have to follow the steps in the set-play to be able to cooperate. A set of defined set-plays can be reused in different soccer games and integrated with other existent team strategy mechanisms such as tactics and formations to better cope with opponents.

A **Set-play** is characterized by a name and can have **Parameters** (Integers, Decimals, Points or Regions) and **Player References** which identify the participants.

A **Player Reference** can point to a specific player, characterized by a team and number, or **Player Roles** (abstract representations of a player). **Parameters** and **Player Roles** will be instantiated at run-time.

A **Set-play** is composed of several **Steps** (with at least one entry **Step**) which represent a state of its execution. Each **Step** can have a **Condition** that must be satisfied before entering the **Step** and is lead by a **Leader** (**Player Reference**) (e.g. normally the ball owner since his decisions are the most important). This Lead Player is not necessarily the same throughout the **Set-play**. A list of **Player References** identifies the players participating in the step, including the Leader. The exit of a **Step** is done by following one of possibly several **Transitions**. A **Transition** can have a **Condition** which must be satisfied before its list of **Directives** can be applied. The following three types of **Transitions** are defined:

– **Next Step** (main transition): used to establish links between different **Steps**;
– **Abort**: used in situations in which the **Set-play** must be terminated giving up its goals;
– **Finish**: used in situations in which the **Set-play** has reached its intended goal.

The **Directives** include a list of **Actions** to perform and can be of two kinds: **Do** and **Don't**, meaning, respectively, that the contained **Actions** should, or should not, be executed.

An **Action** is a high-level concept that represents a skill (e.g. pass) that can be executed by a player.

A **Condition** deals with the whole domain of robotic soccer, imposing constraints over the world state. Spatial entities in the world are represented through **Regions** (e.g. static points, dynamic points, named regions, triangles, arcs and rectangles), **Dynamic Points** (e.g. player, ball) that refer to the location of game moving objects and **Named Regions** (e.g. opponent penalty area) that model intuitive locations which can change during the match.

All participants monitor the execution of the **Set-play** to decide whether to continue its execution. In each **Step** the **Leader** instructs other participants of the start of the **Set-play**, the entry on a **Step** and the choice of **Transition**.

This framework has mostly been tested in the RoboCup 2D soccer simulation for non-play-on situations. However, some tests have also been performed in the RoboCup 3D soccer simulation and Middle Size leagues.

4 Set-play extraction framework

The architecture of the set-play extraction framework is depicted in Fig. 2. The raw material used in the context of this work are game log files of soccer matches generated by Soccer Server, the official RoboCup 2D soccer simulator. In the context of this work a game log file consists of an ordered representation of world scenes (e.g. players and ball positions) and events (e.g. change of play-mode from play-on to kick-off) that occurred during the soccer match.

The process for extracting set-plays from these game logs, unveiled in Fig. 2, comprises five phases:

1. Selection of the source game logs from which the set-plays are to be extracted;
2. Identification of high-level game events (e.g. passes, dribbles, goals) from the previously selected game logs;
3. Filter relevant game-events (e.g. pass-chain thats lead to goal performed by a specific team) from the previously identified high-level game events;
4. Expansion of relevant high-level events (e.g. pass-chain with intermediate dribbles that leads to goal);
5. Representation of an expanded high-level event using the set-play language described in Sect. 3.

In the context of this article, the term "goal plan" shall be used to refer to an extracted cooperative behavior that starts from a set-piece which consists of a pass-chain, with possible intermediate dribbles that ultimately leads to a shoot and the scoring of a goal.

A pass-chain is a complex event which can be described as two or more consecutive and successful passes made between teammates, during which no opponent is able to intercept the ball. An opponent interception is assumed to occur whenever an opponent kicks the ball in a temporal instant in which the other team has the ball possession and deflects it to himself or another teammate. A pass-chain can

start from a set-piece or during play-on (e.g. after a ball recovery). In some situations it might be possible that during the detected pass-chain, more than one player kicks the ball. Whenever this happens it is assumed that the closest player to the ball is the intended sender, receiver or shooter as the case.

The start and end play-modes of a game event (e.g. pass, dribble) are the observed play-modes at the start and end time of its execution (e.g. in a pass-chain these are the observed play-modes at the time of execution of the sender's first kick and the receiver's first kick, respectively).

During the execution of the goal plan, pass senders and the last pass receiver might perform interleaved dribbles and ball holds prior to the start of the pass and the actual shoot at goal, respectively.

The developed framework can be parametrized in order to allow the tuning of the set-play extraction process to better suit the users needs. Currently, such parametrization can only be applied in the following scopes:

– Event detection: new detectors can be added or existing ones can be further parametrized to use different semantics for recognizing and extracting an observable game event;
– Event relevance: filtering policies can be used to decide on the importance of an extracted game event (e.g. number of passes in a pass-chain, maximum dribble duration);
– Plan representation: the definition of a plan can be tuned according to several policies that focus on set-play complexity (e.g. participants selection method, number of steps), spatial translation (e.g. assume an attack reference side) and abstraction (e.g. enforce positions, abstract concrete players with roles) and specific actions (e.g. off-ball movement, dribble, pass, forward).

Albeit this work makes use of a specific set-play plan definition language and framework (Mota et al. 2010), the process described in Fig. 2 can be extended for use with other plan-based frameworks. These plan-based frameworks must support the implementation of soccer domain-specific concepts in order to allow the adequate description of the recognized cooperative behaviors in a soccer match. The internal data structures used by these plan-based frameworks will often share many similarities but they can also contain significant differences. For this reason, the algorithms devised for the last step of set-play extraction process will most likely need to be adapted to conform with the particularities of a given plan-based framework. Examples of plan-

Fig. 2 Set-play extraction process
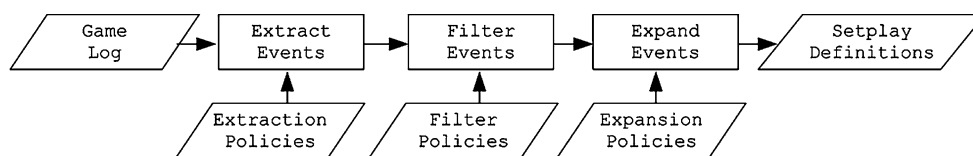
based frameworks proposed in the soccer domain include coordination graphs (CGs) (Kok et al. 2002, 2003), scenario-based teamworking (SBT) (Rad et al. 2004), hierarchical task networks (HTNs) (Obst 2006; Obst and Boedecker 2006) and Petri-net plans (PNPs) (Kontes and Lagoudakis 2007; Ziparo et al. 2008; Palamara et al. 2009).

4.1 Event extraction

Events are extracted from a set of scene data structures parsed from a game log using SoccerScope.[3] A scene is described by a time of occurrence, a play-mode, physical data (e.g. position, velocities) for the players and ball . Several meaningful events can be extracted from the set of scenes parsed from a game log file. The definitions of these events and the algorithms used for their extraction were borrowed from Abreu et al. (2011). In the scope of this work the most relevant events to detect are successful passes, successful dribbles, pass-chains and non-own goals (shoots are implicit) which are formally defined in Eqs. 1, 2, 3 and 4, respectively.

$$Pass(p_0, p_1, t_0, t_1) \leftarrow t_0 < t_1 \wedge p_0 \neq p_1 \wedge SameTeam(p_0, p_1)$$
$$\wedge KicksBall(p_0, t_0) \wedge KicksBall(p_1, t_1) \wedge \nexists (p, t_0) :$$
$$KicksBall(p, t_0) \wedge p \neq p_0$$
$$\wedge \nexists (p, t_1) : KicksBall(p, t_1) \wedge p \neq p_1 \wedge \nexists (p, t) :$$
$$t_0 < t < t_1 \wedge KicksBall(p, t) \tag{1}$$

$$Dribble(p_0, t_0, t_1) \leftarrow t_0 < t_1 \wedge KicksBall(p_0, t_0)$$
$$\wedge KicksBall(p_0, t_1)$$
$$\wedge \nexists (p, t_0) : KicksBall(p, t_0) \wedge p \neq p_0$$
$$\wedge \nexists (p, t_1) : KicksBall(p, t_1) \wedge p \neq p_0 \wedge \nexists (p, t) :$$
$$t_0 < t < t_1 \wedge KicksBall(p, t) \tag{2}$$

$$PassChain(T, t_0, t_5) \leftarrow \forall (p_0, p_1, p_2, p_3, t_1, t_4) :$$
$$t_0 < t_1 \leq t_4 < t_5$$
$$\wedge Pass(p_0, p_1, t_0, t_1) \wedge Pass(p_2, p_3, t_4, t_5)$$
$$\wedge \forall (p_4, p_5, t_2, t_3) : t_1 \leq t_2 < t_3 \leq t_4 \wedge Pass(p_4, p_5, t_2, t_3)$$
$$\vee Dribble(p_4, t_2, t_3)]$$
$$\wedge \nexists (p, t) : t_0 \leq t \leq t_5 \wedge KicksBall(p, t) \wedge Team(p) \neq T \tag{3}$$

$$Goal(T, t_3) \leftarrow \exists (p_0, t_0, t_1, t_2) : t_0 < t_1 < t_2 < t_3$$
$$\wedge KicksBall(p_0, t_0)$$
$$\wedge Team(p) = T \wedge \nexists (p, t_0) : KicksBall(p, t_0) \wedge p \neq p_0$$
$$\wedge Region(ball, t_1) = GoalieArea \wedge InterceptsBall$$
$$(OpponentGoalLine, t_2)$$
$$\wedge CrossesBall(OpponentGoalLine, t_3) \wedge \nexists (p, t) :$$
$$t_0 < t \leq t_3 \wedge KicksBall(p, t) \tag{4}$$

---

[3] More information available online at http://ne.cs.uec.ac.jp/*koji/SoccerScope2/index.htm.

where $p, p_i, i = \{0..5\}$ are players, $t, t_i, i = \{0..5\}$ are temporal instants and T is a specific team.

Some events such as short dribbles and explicit ball holds can be hard to distinguish. In this context, the use of fuzzy logic and additional internal parameters (e.g. thresholds) for event detectors can prove useful to obtain even more adequate descriptions of the recognized cooperative behaviors.

4.2 Extraction of set-plays

The set-play extraction phase requires that all relevant high-level game events have been previously detected.

The general algorithm for extracting the set-plays from a set of game logs is achieved using Algorithm 1. This algorithm essentially takes each game log in a set and selectively extracts a set of high-level events (passes, dribbles and pass-chains in this order) using the algorithms in Abreu et al. (2011). For each detected pass-chain an expansion process is applied using Algorithm 2 which looks for the occurrence of other meaningful events that took place immediately before and after the pass-chain while keeping the ball possession. If an expanded pass-chain is considered relevant, a set-play definition will be extracted from it using Algorithm 3.

Algorithm 1 $extractSetplays(gameLogs)$

Require: $gameLogs \neq \emptyset$
1: $extractedSetplays \leftarrow \emptyset$
2: **for** each $gameLog$ in $gameLogs$ **do**
3:     $passes \leftarrow detectPasses(gameLog)$
4:     **if** $passes \neq \emptyset$ **then**
5:         $dribbles \leftarrow detectDribbles(gameLog)$
6:         $passChains \leftarrow getPassChains(passes, dribbles)$
7:         $goals \leftarrow detectGoals(gameLog)$
8:         **for** each $passChain$ in $passChains$ **do**
9:             $expandedPassChain \leftarrow expandPassChain(passChain, dribbles, goals)$
10:             **if** $isRelevant(expandedPassChain)$ **then**
11:                 $extractedSetplays \leftarrow extractedSetplays \cup \{extractSetplay(expandedPassChain)\}$
12:             **end if**
13:         **end for**
14:     **end if**
15: **end for**
16: **return** $extractedSetplays$

The expansion of a detected pass-chain into a richer set of events is performed using Algorithm 2. This algorithm merges all relevant events that occurred before, during and after the pass-chain that sustain the recognized cooperative behavior of the team while the ball possession is kept. The first pass of the pass-chain is initially added to the sequence of events from which the cooperative behavior shall be extracted. Prior to this pass, it is also relevant to understand if a change in the play-mode has occurred and if the player executing this first pass held the ball explicitly to make time for other teammates to position themselves adequately. If this is the case, then events

describing these situations are prepended to the sequence of relevant events. Between each sequence of passes one or more dribbles might have been executed by the participant owning the ball. Whenever these dribble events occur, they are also added to the sequence of relevant events After the end of the last pass in the pass-chain, a goal could have been scored by the team which performed the pass-chain. If this is the case and the team kept the ball possession from the end of the pass-chain until the scoring of the goal, then the receiver of the last pass was the goal scorer and could have executed dribbles prior to shooting at the goal. If any dribble events were executed by the shooter they will also be added to the sequence of relevant events, followed by the shoot event and the goal event. During this process, some of the events gathered may be simplified in order for the associated actions or conditions to become more general (e.g. avoid defining short continuous dribbles).

Algorithm 2 $expandPassChain(passChain, dribbles, goals)$

Require: $passChain \neq \emptyset$

1: $expandedPassChain \leftarrow \{getFirstPass(passChain)\}$
2: **for** each $(pass_i, pass_{i+1})$ in $passChain$ **do**
3:     **if** $EndTime(pass_i) \neq StartTime(pass_{i+1})$ **then**
4:         $intermediateDribbles \leftarrow getDribbles(dribbles, endTime(pass_i), startTime(pass_{i+1}))$
5:         $expandedPassChain \leftarrow expandedPassChain \cup \{intermediateDribbles\}$
6:     **end if**
7:     $expandedPassChain \leftarrow expandedPassChain \cup \{pass_{i+1}\}$
8: **end for**
9: $goal \leftarrow getFirstGoalAfter(endTime(passChain_{|passChain|-1}))$
10: **if** $goal \neq \emptyset \wedge getTeam(goal) = getTeam(passChain) \wedge$
    $keptPossession(getTeam(passChain), endTime(passChain), startTime(goal)))$ **then**
11:     $lastPassEndTime \leftarrow endTime(passChain_{|passChain|-1})$
12:     $dribbleBeforesGoal \leftarrow getDribbles(dribbles, lastPassEndTime, startTime(Goal))$
13:     **if** $dribblesBeforeGoal \neq \emptyset$ **then**
14:         $expandedPassChain \leftarrow expandedPassChain \cup \{dribblesBeforeGoal\}$
15:     **end if**
16:     $expandedPassChain \leftarrow expandedPassChain \cup \{getShoot(goal)\} \cup \{goal\}$
17: **end if**
18: **return** $expandedPassChain$

The abstraction of an expanded pass-chain into a set-play definition is described in Algorithm 3. This process starts with the creation of a set of steps using Algorithm 4 that are mapped to the game scenes associated with the start and end of each game event in the expanded pass-chain. After the creation of steps, a decision is made regarding which players should participate in the set-play using Algorithm 5. Then, for each of the existing steps a detailed characterization of its start conditions, wait and abort times and transitions is made. The abort time of the first step is defined statically as the maximum time the team is allowed to execute the set-piece minus a threshold. This threshold is used to allow the player starting the set-piece to perform a reactive behavior whenever the step entry condition is not verified and prevent the referee from awarding a foul in favor of the opponent.

The time to wait before entering the first step is defined using a formation model combined with a player motion model. The formation model uses the location of the ball to predict the location of team players that are eligible to participate in the first step. The set of players which minimizes the overall distance to each of the step participants initial positions is chosen. For each of players in this set, a player motion model is applied to determine how much time each requires to reach the associated initial step position. The maximum time calculated in the previous procedure is used as the first step wait time which must be always lower than the step abort time. In subsequent steps, no wait time is currently defined because we are assuming that the next step participants should already be at the expected initial location at the start of the next step. However, this might not always be adequate due to the dynamic and stochastic nature of the soccer environment.

The decision of when to start the step of a set-play is currently defined based on the following conditions:

– verify that the positions of all step participants relative to the leader of the step are within an acceptable distance threshold;
– verify that the leader of the step owns the ball whenever the underlying game event that originated the step is a pass, dribble or a shoot.

A transition consists of one or more actions to be performed by the players participating in a given step. Each transition has a main action which is executed by the leader of that step. Currently, the condition to start the execution of a transition is characterized by checking whether that action can be safely executed.

For the time being, the set-play abort conditions are defined statically. As a general rule of thumb, the execution of a set-play will be aborted whenever one of following conditions is verified:

– The opponent intercepts the ball during its execution;
– The game (or one of its halves) ends;
– The ball goes out-of-bounds.

Algorithm 3 $extractSetplay(expandedPassChain)$

Require: $expandedPassChain \neq \emptyset$

1: $setplay \leftarrow createSetplay(expandedPassChain)$
2: $setplaySteps \leftarrow createSteps(setplay)$
3: $decideSetplayParticipants(setplay)$
4: **for** each $(step_i, step_{i+1})$ in $setplaySteps$ **do**
5:     $decideStepParticipants(step_i)$
6:     $decideStepWaitTime(step_i)$
7:     $decideStepAbortTime(step_i)$
8:     $decideStepCondition(step_i)$
9:     $createNextStepTransition(step_i, step_{i+1})$
10: **end for**
11: $createFinishTransition(setplaySteps_{|setplaySteps|-1})$
12: $decideSetplayIdentification(setplay)$
13: $decideSetplayAbortConditions(setplay)$
14: $decideSetplayInversion(setplay)$
15: **return** $setplay$

The creation of steps from an expanded pass-chain when extracting a set-play is described in Algorithm 4. This process consists of a step for each of the game events present in the expanded pass-chain. The number of steps and transitions in the extracted goal plans will be equal to the number of game events that result from the expansion of the initially detected pass-chain. Between each consecutive pair of **Steps** and for the last **Step** a **Next Step** and **Finish** transition shall be created, respectively. The players who own the ball at the start of each game event will be deemed as the Leaders of the associated **Steps**. These events consist essentially of those that require that a player has the ball possession such as holding, passing and dribbling the ball.

Algorithm 4 $createSteps(gameEvents)$

Require: Time-ordered ascendant and non-empty set of relevant game events
1: $steps \leftarrow \emptyset$
2: **while** $gameEvents \neq \emptyset$ **do**
3:     $event \leftarrow getNextEvent(gameEvents)$
4:     $gameEvents \leftarrow gameEvents \setminus \{event\}$
5:     $step \leftarrow createStepFromEvent(event)$
6:     $steps \leftarrow steps \cup \{step\}$
7: **end while**
8: **return** $steps$

The decision for choosing the players that will participate in the set-play is described in Algorithm 5. For each of the previously created steps in Algorithm 4, this algorithm sets the participants of the set-play to be the union of all players that explicitly participate in each of the game events (e.g. pass, dribble) associated with each step.

Algorithm 5 $decideSetplayParticipants(steps)$

1: $setplayParticipants \leftarrow \emptyset$
2: **for** each $step$ in $steps$ **do**
3:     **for** each $participant$ in $getEventParticipants(step)$ **do**
4:        **if** $participant \notin setplayParticipants$ **then**
5:           $setplayParticipants \leftarrow setplayParticipants \cup \{participant\}$
6:        **end if**
7:     **end for**
8: **end for**
9: **return** $setplayParticipants$

The decision of which participants should take part in a given step of a set-play is directly related to the amount of effort that players will have to employ for executing their actions. Thus if during a given detected game event (e.g. pass), a passive player (not the receiver nor the kicker) executes a motion that is irrelevant for the success of the set-play, this motion should not be incorporated into the definition of the resulting set-play. However, for simplicity the implementation used assumes for simplicity that these should be all the players that were chosen using Algorithm 5. This approach is not optimal, as it assumes that all set-play participants must do something relevant (e.g. open space for another teammate) in each of the steps of the set-play, even though some of them might not be actively involved in all the steps underlying events. The major drawback of this approach is that the possible movements that these players might have executed do not contribute to the success of the set-play, but will nonetheless waste their stamina.

More clever algorithms are currently supported by the set-play extraction framework that could have been used to choose the players that should participate in each step which include:

1. The participants of a given step should be the distinct set of players that actively participate only in the current step. The major drawback of this approach is that it neglects players that participate in subsequent steps and might do something relevant in the current step in order to allow the execution of their future actions;

2. The participants of a given step should be the distinct set of players that actively participate in the step underlying event and in the subsequent steps underlying events. This approach is able to free players from the set-play before its end and allows them to perform other useful tasks for the team. However, this approach assumes that if a set-play participant does not have an active role in any of the underlying events of the subsequent steps, he does not do anything useful to ensure the success of the set-play execution. This might not always be true, since for instance the positional moves performed by players that do not actively participate in any of the set-play subsequent events (whether they participate or not in the set-play) can be useful to mislead opponents and thus contribute to maximize the success probability of its execution;

3. Another alternative would be to always consider the players taking part in the step's underlying event as participants and to devise a measure of importance for others that considers the relevance of their actions in the context of that step, even if they do not actively participate in it. For instance, a distance based criteria can be used to discard irrelevant players, without even considering the actions they performed if they are too far away from the ball.

The previously described methods are not perfect and should not be used blindly for deciding the step participants for all steps. We believe that a more clever approach will consist of applying a mix of the previously described procedures to different steps of a set-play depending on other contextual information associated with each step and its transitions to other steps.

5 Methodological approach

In order to validate the presented set-play extraction framework two experiments were conducted:

1. Extract previously defined goal plans from 25 game logs generated by the FC Portugal team without opponent;
2. Extract goal plans from game logs of the RoboCup 2010 competition in which at least four goals were scored.

In the first experiment a set of 25 known set-plays that start when the game is stopped and lead to goal were defined based on all the possible combinations of five non-play-on modes (kick-off, goal-kick, kick-in, corner-kick, free-kick) and different number of participants (2–6 players). For instance a possible combination for a set-play would be a kick-off involving three players. In this experiment a total of 25 game logs were generated by playing matches with the FC Portugal team with no opponent with the previous set of defined set-plays. In each of the games played, only one specific set-play was active and was deliberately triggered several times during the match to promote a relevant set of examples for analysis.

The goal of the second experiment is to assess the generalization of the extraction framework in a non-controlled environment. For this purpose a set of 69 game logs were selected for analysis based on the following criteria:

– To maximize the probability of detecting set-plays that lead to goal, only games in which a combined total of four or more goals were scored were considered;
– For the sake of obtaining relevant results to be used in the short term, only the most recent games from the RoboCup 2010 competition were used.

Due to the incipiency of the FC Portugal set-play framework, set-plays cannot still be triggered in an adequate manner during play on mode and thus no set-plays that began in this play mode were defined for these games.

In the first experiment the environment is known and controlled by the authors. However, in the last experiment the environment is unknown, since the game logs have been already recorded without the authors' intervention.

All extracted goal plans were also validated by visual inspection of each game.

6 Results and discussion

6.1 Experiment 1

A total of 285 goal plans resulted from the execution of the predefined set-plays in the 25 generated game logs. The distribution of the executed set-plays per play-mode and number of participants involved is described in Table 1.

Table 1 Goal plans observed in matches played by FC Portugal with no opponent and a known set of set-plays

Participants	Goal-kick	Corner-kick	Free-kick	Kick-in	Kick-off	Totals
2 players	9	14	13	12	11	59
3 players	14	15	14	11	10	64
4 players	13	15	8	11	9	56
5 players	12	14	8	11	10	55
6 players	10	14	8	11	8	51
Totals	58	72	51	56	48	285

The disparity in the number of goal plans executions observed per combination can be explained by the fact that execution times varied among different set-plays and also within instances of the same set-play. The cause for the variation in the execution times of set-plays is due to the inaccurate perception and execution of actions enforced in the simulated environment. Also, some set-plays caused an excess of stamina consumption due to their repetition which contributed to wearing the players and affected the intended execution flow.

The extraction process supported by the algorithms described in Sect. 4 achieved an accuracy of 100 %, which was corroborated after visual inspection. This result attests the effectiveness of the developed algorithms for these controlled cases.

6.2 Experiment 2

The performance of each team in the 69 observed games and in the competition is summarized in Table 2.

A total of 607 goals were scored in the 69 analyzed matches, from which 59 (86 %) matches contained goal plans. From the 547 scored goals in these 59 matches, 150 (22 %) goals originated from a set-piece. This result reveals that many goals can be scored from such situations. An accuracy of 100 % was also obtained in the extraction of these goal plans, which was corroborated by a visual inspection.

However, two ambiguous situations occurred during the detections in which ball possession was considered to be kept:

– a player shoots at goal and an opponent kicks the ball during its traversal but merely deflects it and is unable to stop it entering the goal;
– two players of different teams kick the ball at the same time of a pass and the ball goes to a player of the team which previously had the ball possession.

From the 59 games an average of 2.5 ± 1.4 goal plans were detected per match. The team which produced the most goal plans (30), scored the most goals (116) and won the most matches (9) was WrightEagle. The winner of the competition, HELIOS2010, played one less game than WrightEagle. HELIOS2010 scored less than 50 % of the goals (54) of WrightEagle and was the third team to have produced the most

Table 2 Analysis of team's performance in light of the extracted goal plans and their rank in the competition

Rank	Team	W	D	L	Goals	Plans	Steps	Participants	Zones	Duration
2	WrightEagle	9	0	0	116	30	10 ± 4	4 ± 1	5 ± 2	76 ± 53
9	FC_Pars	8	1	1	63	21	11 ± 8	4 ± 2	6 ± 3	112 ± 99
1	HELIOS2010	6	0	2	54	14	9 ± 4	4 ± 1	5 ± 2	82 ± 43
5	Nemesis2010	6	0	2	50	12	9 ± 4	5 ± 2	5 ± 2	82 ± 53
3	Oxsy	8	0	2	99	12	6 ± 2	4 ± 1	3.8 ± 2	51 ± 28
6	Unique	3	0	3	30	11	8 ± 4	3 ± 1	4.7 ± 3	64 ± 46
8	HfutEngine	5	0	3	35	9	8 ± 5	4 ± 2	4 ± 2	84 ± 47
4	ESKILAS	7	0	1	34	7	9 ± 3	4 ± 1	5 ± 1	85 ± 35
10	opuCI_2D	4	2	4	37	7	7 ± 2	4 ± 1	5 ± 2	88 ± 38
17	Fifty-storms	2	0	5	22	6	7 ± 1	4 ± 1	4 ± 1	63 ± 9
13	Apollo	1	2	4	9	5	8 ± 5	4 ± 2	4 ± 2	83 ± 60
14	NCL10	1	1	5	16	5	7 ± 2	4 ± 1	5 ± 2	79 ± 51
16	Ri-one	0	2	5	9	4	7 ± 3	3 ± 1	5 ± 2	67 ± 36
18	AUA2010	1	0	7	8	3	6 ± 1	4 ± 1	5 ± 1	81 ± 30
12	RaiC-2010	2	0	5	16	3	7 ± 3	4 ± 1	4 ± 2	69 ± 32
7	FC Portugal	1	2	0	9	1	8 ± 0	4 ± 0	5 ± 0	80 ± 0
19	Bahia2D	0	0	9	0	0	–	–	–	–
15	Iran	0	0	4	0	0	–	–	–	–
11	KickOffTUG	0	0	2	0	0	–	–	–	–

W, *D*, *L* stand for wins, draws and lost matches, respectively

goal plans (14), which stands for 50 % less than WrightEagle. Both teams played against five common teams (Iran, Ri-one, Oxsy, opuCI_2D and Fifty-Storms) and different teams in the remaining matches. In these matches, WrightEagle was up against NCL10 (14th), Nemesis2010 (5th) and Unique (6th) (this last one twice) while HELIOS2010 was up against ESKILAS (4th), HFutEngine (8th) and FC_Pars (9th). Based on the final rankings of the teams in the competition to empirically assess their quality, no conclusions can be drawn to support the thesis that WrightEagle was up against inferior teams than HELIOS2010 on average (without a significant difference) or vice-versa.

The Apollo team achieved the highest ratio of goal plans (56 %) per scored goals, having achieved a total of 5 in the 7 matches played. However, two of these goal plans were very simple and consisted of free-kicks that started in the opponent's goalie area and involved two participants (pass and shoot).

The steps of a captured corner-kick goal plan[4] involving four participants are depicted in Fig. 3. The definition extracted for this goal plan using the set-play language is described in Appendix.

In Fig. 3 the players' movements from the team that owns the goal plan between each of the presented scenes are unanticipated by the opponent. The execution of these combinations of runs and passes allows the team to set its teammates free in

order to receive the ball and have a clear shot at goal in situations without significant opponent's pressure.

Some metrics (duration of the plan, number of steps, number of participants involved, ball travelled distance) to understand the overall complexity of the goal plans were also extracted and are summarized in Table 3.

In general, the extracted goal plans were quite complex since they consisted of the execution of 8–10 steps, involving 3–5 participants and taking around 80 cycles to complete. The most simple extracted goal plan was a free-kick in the opponent goalie area with three steps that took two participants and seven cycles (simple pass and direct shoot at goal). Contrarily to what could be expected, the most complex extracted goal plan had 37 steps and started from a kick-in in the opponent middle instead of a goal-kick for being further away from the opponent's goal. Also contrarily to what could be expected for having the highest number of steps, this goal plan was not the one that involved the highest number of participants (8), involving only 5. However, it was the one which took the most cycles (416) to complete and had the highest number of steps.

Also the goal plans that took the longest to complete were in general the ones that started furthest away from the opponent's goal. The number of participants required for the execution of a goal plan tends to increase with the distance to the goal, although some exceptions were observed, particularly in matches opposing stronger to much weaker teams.

The number of participants that took part in the extracted goal plans per play mode is described in Table 4.

---

[4] This cooperative behavior was exhibited by the WrightEagle team in the RoboCup 2010 quarter finals game against opuCI_2D at cycle 4551 and lasted 27 cycles.

Fig. 3 Game scenes of an extracted corner-kick goal plan with four participants. In each of the figures, *arrows* are used to characterize players actions (not all are considered in the extraction and representation of the goal plan). The *black*, *blue* and *red arrows* represent a pass, run and shoot at goal, respectively. **f** describes a simplified view of the mapping of game events to set-play concepts based on the use of the set-play-extraction framework described in Sect. 4. The symbol *L* in **f** refers to the left team side, which in this case WrightEagle

(a) 4 passes to 3 **(b)** 3 passes to 10 **(c)** 10 passes to 6 **(d)** 6 shoots **(e)** Goal is scored
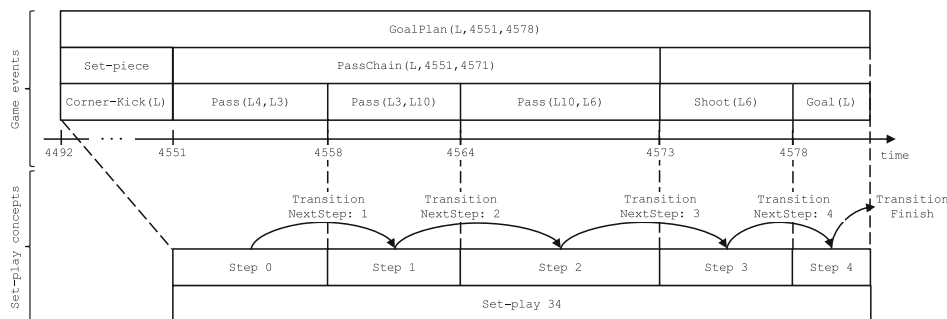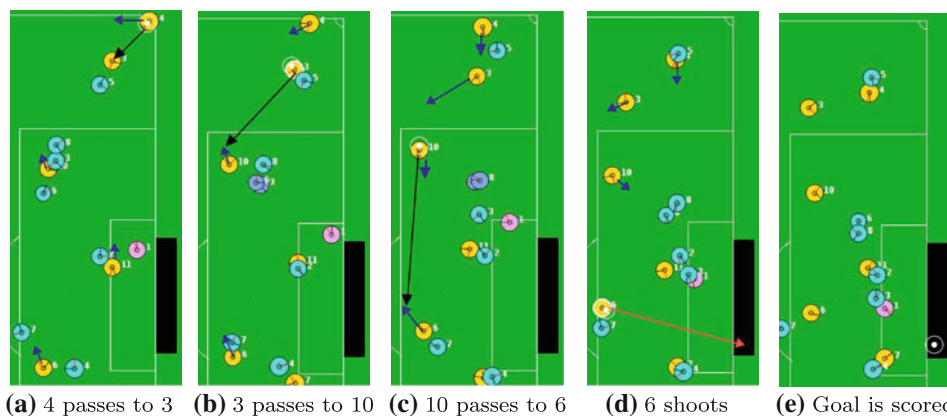
(f) Mapping of extracted game events to set-play concepts

Table 3 Complexity measurements for the extracted goal plans

Metric	Corner-kick	Free-kick	Goal-kick	Kick-in	Kick-off
Duration	47.3 ± 17.8	64.2 ± 67	140.4 ± 29.9	82.7 ± 56.1	104.6 ± 35.6
Steps	7.5 ± 1	7.1 ± 1.4	10.6 ± 1.2	8.8 ± 1.4	10.3 ± 1.3
Participants	3.9 ± 2.4	3.3 ± 4.8	5.3 ± 3.3	4.2 ± 5	4.3 ± 4.3
Traveled distance	90.9 ± 30.1	94.2 ± 70	161.7 ± 21.8	114.7 ± 61.8	135.1 ± 38.4

Table 4 Number of extracted set-plays per participants and play mode

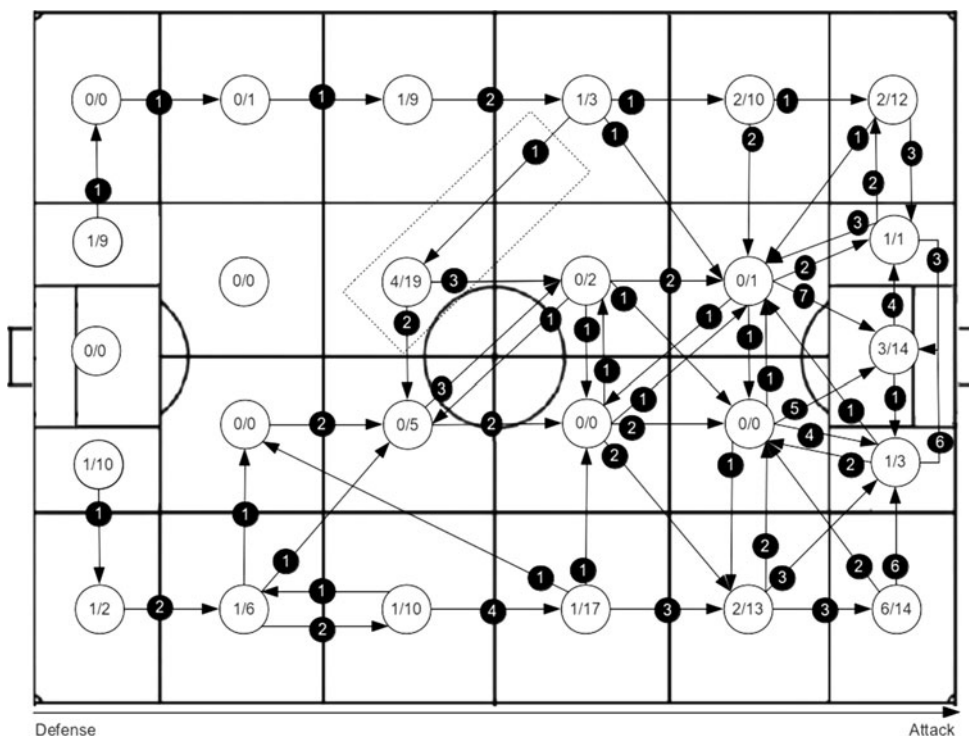| Participants	Corner-kick	Free-kick	Goal-kick	Kick-in	Kick-off	Total
2 players	1	19	0	5	2	27
3 players	5	5	0	18	2	30
4 players	4	5	2	21	7	39
5 players	5	10	4	12	4	35
6 players	0	2	2	6	2	12
7 players	0	0	0	4	1	5
8 players	0	0	1	1	0	2
Total	15	41	9	67	18	150

The majority of the extracted goal plans originated from kick-ins (45 %) and involved 3–4 players (26 %). From these kick-ins, WrightEagle was the team with the most (25 %).

The start zones and transition zones for all games and the team with the highest number of extracted goal plans (WrightEagle) are depicted in Fig. 4.

By taking a closer look at the dashed polygon it can be concluded that the highest number of extracted goal plans for WrightEagle from that zone was 4 out of a total of 19. Moreover, from this zone the goal plans continue downwards and rightwards for 2 and 3 times, respectively. It should be noted that there is 1 enter transition from one other zone to this start zone which must be considered when interpreting its exit transitions.

The extracted goal plans are distributed unevenly over 17 distinct starting zones and followed 131 distinct execution paths (linked transitions between zones). The zone from which the most goal plans (19) started is near the midfield and it includes kick-offs (18). From these, the most started from the opponent side wings (69), in particular from the right side (44). Other results obtained allowed us to conclude that 97 % of the goal plans ended with a shot at goal inside the opponent penalty area performed by players #11 (43 %), #10 (26 %) and #9 (21 %). However, it was a surprise to see that some goals (5) were scored from outside the opponent penalty area, since the distance to goal is quite high and should give a good goalie enough time to make a save.

Fig. 4 Start execution zones and zone transitions of extracted goal plans from all games and from WrigthEagle. The *left* and *right numbers* in the *white circles* separated by a slash represent the number of goal plans for all games and for WrightEagle that started from the underlying zone, respectively. The *number in the black circles* is the number of transitions that occurred between two zones for WrightEagle. The sum of zone transitions that enter a given zone is equal to the number goal plans that start and exit from that zone

Table 5 Number of extracted goal plans per game quarters and play mode

Game period	Corner-kick	Free-kick	Goal-kick	Kick-in	Kick-off	Total
First quarter	3	9	0	19	6	37
Second quarter	3	12	4	12	2	33
Third quarter	2	12	2	17	10	43
Forth quarter	7	8	3	19	0	37
Total	15	41	9	67	18	150

The majority of WrightEagle goal plans started from the opponent side wings nearer to their penalty area. A bird's eye view over the goal plans start zones and zone transitions followed by this team reveals that their goal plans tend to start in the wings and converge to the inner sides and center of the opponents penalty.

The number of extracted goal plans per game period and play mode are depicted in Table 5.

The third quarter was the period in which the most goal plans (43) were extracted, although their distribution was more or less balanced throughout the game periods. From this data, it can also be inferred that corner-kicks and kick-offs tend to be more effective in the 4th and third quarters, respectively, possibly due to the teams tiredness.

The most effective type of extracted goal plans were kick-ins (67) followed by free-kicks (41). The majority of the kick-ins started near the opponents penalty area (50 %) and near the midfield zones (28 %). Since the most frequent event that occurs during matches are kick-ins, this result suggests that we should try to take advantage of this situations with predefined set-plays.

In order to check if the participation of certain groups of players was favored in the extracted goal plans a cluster analysis was performed for each distinguished number of initial participants and its results are depicted in Table 6.

For instance, in the five goal plans that involved seven players there were only two relevant subgroups that emerged with 3 and 4 players, respectively. By taking a closer look at subgroup {7,10,11} it should be interpreted that in the smaller subgroups those specific players could be combined in all possible manners and would maintain the same coverage.

From Table 6 it can be concluded that the player, pair of players, threesome of players and foursome of players that were most used in the extracted goal plans are {11}, {10,11}, {7,10,11} and {6,7,10,11}, respectively. These results can be used to infer which particular players should be payed special attention when a goal plan is about to be triggered, particularly when there are too many possibilities of players from which to choose from.

7 Conclusions

The developed set-play extraction process revealed an optimal accuracy for identifying possible goal plans that start from a set-piece, corroborated by a visual inspection which identified all the relevant cases.

The results of the last experiment supported the relevance of performing this kind of analysis since a significant

Table 6 Representative groups of players that participated in the extracted goal plans

Participants	Groups of players	Coverage (%)
2 players	{11}	53.33
2 players	{8,10}	13.33
3 players	{11}	66.67
3 players	{10,11}, {9,10}, {9,11}	30.00
3 players	{9,10,11}	13.33
4 players	{11}	82.05
4 players	{10,11}	53.85
4 players	{8,10,11}, {7,6,11}	23.08
4 players	{6,7,10,11}	5.41
5 players	{9}, {10}	75.71
5 players	{9,10}	54.29
5 players	{7,9,10}, {9,10,11}	34.29
5 players	{6,7,10,11}	22.86
5 players	{6,7,8,9,10}	8.57
6 players	{6}, {10}, {11}	83.33
6 players	{6,7,9,10,11}	25.00
7 players	{7,10,11}	100.00
7 players	{6,7,10,11}, {9,7,10,11}, {5, 7, 10, 11}, {5,6,10,11}	80.00
8 players	{2,6,7,8,9,10,11}	100.0

The start and end brackets identify a particular group which can contain one or more player identifiers separated by a comma. Several groups of players with the same coverage are also separated by a comma. The coverage of a group of players is the percentage of times it was used in extracted goal plans relative to the underlying number of participants depicted in Table 4. Groups of players of smaller sizes were omitted whenever their coverage was equal to the next largest group described and their composition is equal to all the possible combinations of players of that group

amount of goals (25 %) originated from set-pieces. However, this also suggests more strongly that the focus of this study should be shifted to the analysis of goal plans that result from play-on situations.

Moreover, the results of the last experiment can be used to aid the preparation of a team against specific opponents and derive several guidelines for the definition of goal plans in the future. For instance the extracted goal plans or derived behaviors could be used to gain a competitive advantage over an opponent against which it was successful possibly with some minor optimizations.

The complexity associated with some of the extracted goal-plans (e.g. too many participants, steps and/or time to execute) reveals that they were not purposeful and thus they will be unlikely to have a successful outcome. There can be no certainty on whether the extracted goal plans were deliberately executed by a team or if they simply emerged as combination of players simple combined behaviors. In particular, goal plans that use many players are likely to pose higher risks to if

the opponent team recovers the ball since in most cases players will be far away from their tactical position.

Another relevant conclusion that can be drawn from these results is that goal plans that start from kick-ins were the most representative. Also, since they are generally the most frequent in a match more attention should be given to them. Also from the analysis of the results obtained in the second experiment several guidelines can be inferred to guide the future definition of set-plays that start from set-pieces and lead to goal. A set-play should involve up to five participants since 87 % of the extracted set-plays had this number of participants. A set-play should comprise from 3 to 10 steps since 77 % of the set-plays, of which the majority had 4 steps (18 %).

In the future several developments are worth pursuing to improve the usefulness of the developed framework.

A possible development already mentioned consists of analyzing behaviors that start during play-on mode, not restricting them only to set-pieces, and that do not necessarily lead to goal (e.g. enable a team to quickly progress in the field towards the opponent goal). For instance, it can be useful to extract a behavior that allows the team to keep the ball possession for a long period to waste time when it is winning. Also, the usefulness of the extracted set-plays can be improved by adding more complex methods for selecting the participants (e.g. considering opponents and passive teammates behaviors) and to decide the triggering conditions. Robustness could also be added to the extracted set-plays by trying to define virtual transitions between real steps that account for unexpected events (e.g. instead of moving linearly from step 1 to step 2 of a set-play, a direct leap could be made to a subsequent step).

The extracted set-plays that indite good results could be applied to try to improve the performance of a team (defensively or offensively). To measure the success of these set-plays an automatic procedure should be developed that in conjunction with the previously mentioned fuzzy recognition system will be able to score their impact. Moreover, the ad hoc set-plays defined by our coach (user) could also be automatically checked for its effectiveness by figuring out if they triggered any of the scored goals by a team. Finally, another interesting line of research that can be followed is to extend the current set-play extraction framework to other domains such as other RoboCup Soccer leagues (e.g. 3D, MSL) and other cooperative and competition sports (e.g. handball).

Appendix: Set-play definition of a corner-kick with four participants

A Set-play definition of a corner-kick with 4 participants

```
(setplay :id 34 :name CK_4_3_10_6 :invertable true
  :players (list (playerRole :roleName P4) (playerRole :roleName P3)
                 (playerRole :roleName P10) (playerRole :roleName P6))
  :abortCond (bowner :players (player :team opp :number 0))
  :steps
    (seq
      (step :id 0 :waitTime 100 :abortTime 107 :leadPlayer P4
        :participants (list P4
                            (at P3 (ptRel :x -4 :y 5 :pt (pt ball)))
                            (at P10 (ptRel :x -11 :y 18 :pt (pt ball)))
                            (at P6 (ptRel :x -12 :y 42 :pt (pt ball))))
        :condition (and (playm ck_our)
                        (bpos :region (rec :points (list (pt :x 52 :y -35)
                                                         (pt :x 54 :y -33)))))
        :transitions
          (nextStep :id 1
            :condition (canPassReg :from P4 :to (ptRel :x 0 :y 1 :pt (pt ball)
            :directives
              (list
                (do :players P4
                    :actions (list (bto :region (ptRel :x 0 :y 1 :pt (pt ball)
                                   (pos :region (ptRel :x 1 :y 0 :pt (pt ball)
                (do :players P3 :actions (intercept)))))
      (step :id 1 :waitTime 0 :abortTime 15 :leadPlayer P3
        :participants (list P3
                            (at P4 (ptRel :x 2 :y -5 :pt (pt ball)))
                            (at P10 (ptRel :x -7 :y 12 :pt (pt ball)))
                            (at P6 (ptRel :x -7 :y 35 :pt (pt ball))))
        :condition (and (playm play_on) (bowner :players P3))
        :transitions
          (nextStep :id 2
            :condition (canPassPl :from P3 :to P10)
            :directives (list
              (do :players P4  :actions (pos :region (ptRel :x 2 :y -5 :pt (pt
              (do :players P6  :actions (pos :region (ptRel :x -7 :y 35 :pt (p
              (do :players P3  :actions (bto :players P10))
              (do :players P10 :actions (receiveBall)))))
      (step :id 2 :waitTime 0 :abortTime 22 :leadPlayer P10
        :participants (list P10
                            (at P4 (ptRel :x 8 :y -15 :pt (pt ball)))
                            (at P3 (ptRel :x 7 :y -9 :pt (pt ball)))
                            (at P6 (ptRel :x 1 :y 22 :pt (pt ball))))
        :condition (and (playm play_on) (bowner :players P10))
        :transitions
          (nextStep :id 3
            :condition (canPassPl :from P10 :to P6)
            :directives (list
              (do :players P4  :actions (pos :region (ptRel :x 8 :y -15 :pt (pt b
              (do :players P3  :actions (pos :region (ptRel :x 7 :y -9 :pt (pt ba
              (do :players P10 :actions (bto :players P6))
              (do :players P6  :actions (receiveBall)))))
      (step :id 3 :waitTime 0 :abortTime 11 :leadPlayer P6
        :participants (list P6
                            (at P4  (ptRel :x 9 :y -31 :pt (pt ball)))
                            (at P3  (ptRel :x 3 :y -26 :pt (pt ball)))
                            (at P10 (ptRel :x 1 :y -17 :pt (pt ball))))
        :condition (and (playm play_on) (bowner :players P6))
        :transitions
          (nextStep :id 4
            :condition (canShoot :players P6)
            :directives (do :players P6 :actions (shoot))))
      (step :id 4 :waitTime 0 :abortTime 0 :leadPlayer P6
        :participants P6
        :condition (playm ag_our)
        :transitions (finish))))
```

References

Abreu PH, Faria M, Reis LP, Garganta J (2010a) Knowledge representation in soccer domain: an ontology development. In: Proceedings of the 2010 Iberian conference on information systems and technologies, pp 1–6

Abreu PH, Moura J, Silva DC, Reis LP, Garganta J (2010b) Football Scientia—an automated tool for professional soccer coaches. In: Proceedings of the 2010 IEEE international conference on cybernetics and intelligent systems (CIS 2010), June 28–30, 2010, Singapore. IEEE, pp 126–131

Abreu PH, Vinhas V, Mendes P, Reis LP, Garganta J (2010c) Real-time wireless location and tracking system with motion pattern detection. InTech, Robot localization and map building, pp 467–492

Abreu PH, Moura J, Silva DC, Reis LP, Garganta J (2011) Performance analysis in soccer: a Cartesian coordinates based approach using RoboCup data. In: Soft computing—a fusion of foundations, methodologies and applications, pp 1–15

Abreu PH, Mendes-Moreira J, Costa I, Castelão D, Reis LP, Garganta J (2012) Human versus virtual robotics soccer: a technical analysis. Eur J Sport Sci 12(1):26–35

Aler R, Valls JM, Camacho D, Lopez A (2009) Programming Robosoccer agents by modeling human behavior. Expert Syst Appl 36(2, Part 1):1850–1859

Bauckhage C, Thurau C, Sagerer G (2003) Learning human-like opponent behavior for interactive computer games. In: DAGM symposium for pattern recognition, pp 148–155

Carling C, Williams A, Reilly T (2007) Handbook of soccer match analysis: a systematic approach to improving performance, 2nd edn. Routledge

de Dios Bulos R, Dulalia C, Go P, Tan P, Uy M (2005) A data mining approach in opponent modeling. In: Zhang S, Jarvis R (eds) AI 2005: advances in artificial intelligence. LNCS, vol 3809. Springer, Berlin, pp 993–996

Druecker C, Huebner S, Neumann H, Schmidt E, Visser U, Weland H (2000) Virtualweder: using the online-coach to change team formations. Technical report, University of Bremen

Fathzadeh R, Mokhtari V, Mousakhani M, Mahmoudi F (2006) Mining Opponent behavior: a champion of RoboCup coach competition. In: Robotics symposium, 2006. LARS '06. IEEE 3rd Latin American, pp 80–83

Fathzadeh R, Mokhtari V, Kangavari MR (2008) Opponent provocation and behavior classification: a machine learning approach. In: RoboCup 2007: Robot Soccer World Cup XI. Springer, Berlin, pp 540–547

Gonzalez I, Abreu PH, Reis LP (2008) Using a datawarehouse to extract knowledge from robocup teams. In: Proceedings of the 2008 international conference on enterprise and information systems, pp 511–514

Iglesias JA, Fernández JA, Villena IR, Ledezma A, Sanchis A (2009a) The winning advantage: using opponent models in robot soccer. In: Proceedings of the 10th international conference on intelligent data engineering and automated learning, IDEAL'09. Springer, Berlin, pp 485–493

Iglesias JA, Ledezma A, Sanchis A (2009) CAOS Coach 2006 simulation team: an opponent modelling approach. Comput Inf 28(1):57–80

Kok JR, Spaan MTJ, Vlassis N (2002) An approach to noncommunicative multiagent coordination in continuous domains. In: Wiering M (ed) Proceedings of the 12th Belgian-Dutch conference on machine learning, Utrecht, The Netherlands, pp 46–52

Kok JR, Spaan MTJ, Vlassis N (2003) Multi-robot decision making using coordination graphs. In: de Almeida AT, Nunes U (eds)

Proceedings of the 11th international conference on advanced robotics, Coimbra, Portugal, pp 1124–1129

Kontes G, Lagoudakis M (2007) Coordinated team play in the four-legged RoboCup league. In: Proceedings of the 19th IEEE conference on tools with artificial intelligence (ICTAI), vol 1. IEEE Computer Society, Washington, pp 109–116

Kuhlmann G, Stone P, Lallinger J (2005) The UT Austin Villa 2003 champion simulator coach: a machine learning approach. In: Nardi D, Riedmiller M, Sammut C, Santos-Victor J (eds) RoboCup 2004: Robot Soccer World Cup VIII. LNCS, vol 3276. Springer, Berlin, pp 636–644

Ledezma A, Aler R, Sanchis A, Borrajo D (2005) Predicting opponent actions by observation. In: Nardi D, Riedmiller M, Sammut C, Santos-Victor J (eds) RoboCup 2004: Robot Soccer World Cup VIII, The RoboCup Federation. LNCS, vol 3276. Springer, Lisbon, pp 286–296

Ledezma A, Aler R, Sanchis A, Borrajo D (2009) OMBO: an opponent modeling approach. AI Community 22(1):21–35

Mota L, Lau N, Reis LP (2010) Co-ordination in RoboCup's 2D simulation league: setplays as flexible, multi-robot plans. In: 2010 IEEE Conference on robotics automation and mechatronics (RAM), pp 362 –367

Noda I, Matsubara H, Hiraki K, Frank I (1998) Soccer server: a tool for research on multiagent systems. Appl Artif Intell 12(2–3): 233–250

Obst O (2006) Using a planner for coordination of multiagent team behavior. In: Bordini RH, Dastani M, Dix JCBr, Seghrouchni AE (eds) Programming multi-agent systems: 3rd international workshop, ProMAS 2005, revised and invited papers, vol 3862. Springer, Berlin, pp 90–100

Obst O, Boedecker J (2006) Flexible coordination of multiagent team behavior using HTN planning. In: Noda I, Jacoff A, Bredenfeld A, Takahashi Y (eds) RoboCup 2005: Robot Soccer World Cup IX. Springer, Berlin, pp 521–528

Palamara PF, Ziparo VA, Iocchi L, Nardi D, Lima P (2009) Teamwork design based on Petri Net plans. In: Iocchi L, Matsubara H, Weitzenfeld A, Zhou C (eds) RoboCup 2008: Robot Soccer World Cup Xii, vol 5399. Springer, Berlin, pp 200–211

Portela J, Abreu PH, Reis LP, Oliveira EC, Garganta J (2010) An intelligent framework for automatic event detection in robotic soccer games: an auxiliar tool to help coaches improve their teams performance. In: Proceedings of the 2010 international conference on enterprise information systems, pp 244–249

Rad AA, Qaragozlou N, Zaheri M (2004) Scenario-based teamworking, how to learn, create, and teach complex plans? In: Polani D, Browning B, Bonarini A, Yoshida K (eds) RoboCup 2003: Robot Soccer World Cup VII, vol 3020. Springer, Berlin, pp 137–144

Riley P, Veloso M (2001) Coaching a simulated soccer team by opponent model recognition. In: Proceedings of the 5th international conference on autonomous agents, AGENTS '01. ACM, New York, pp 155–156

Riley P, Veloso M (2006) Coach planning with opponent models for distributed execution. Auton Agent Multi-Agent Syst 13:293–325

Riley P, Veloso M, Kaminka G (2002) Towards any-team coaching in adversarial domains. In: Proceedings of the first international joint conference on autonomous agents and multiagent systems: part 3, AAMAS '02. ACM, New York, pp 1145–1146

Steffens T (2004a) Adapting similarity-measures to agent-types in opponent-modelling. In: Bauer M, Gmytrasiewicz P, Kaminka GA, Pynadath DV (eds) Workshop on modeling other agents from observations at AAMAS, pp 125–128

Steffens T (2004b) Feature-based declarative opponent-modelling. In: Polani D, Browning B, Bonarini A, Yoshida K (eds) RoboCup 2003: Robot Soccer World Cup VII. LNCS, vol 3020. Springer, Berlin, pp 125–136

Steffens T (2005) Knowledge-intensive similarity-based opponent modelling. In: Aha DW (ed) Proceedings of the IJCAI workshop on representation, reasoning, and learning in computer games

Stone P, Riley P, Veloso M (2000) Defining and using ideal teammate and opponent agent models. In: Proceedings of the seventeenth national conference on artificial intelligence and twelfth conference on innovative applications of artificial intelligence. AAAI, San Francisco, pp 1040–1045

Ziparo VA, Iocchi L, Nardi D, Palamara PF, Costelha H (2008) Petri net plans: a formal model for representation and execution of multi-robot plans. In: Proceedings of the 7th international joint conference on international foundation for autonomous agents and multiagent systems, AAMAS'08, Richland, vol 1, pp 79–86