Conceptual Representation for a Soccer Commentary Generator

Damian Stolarski

Adam Mickiewicz University
Faculty of Mathematics and Computer Science
ul. Umultowska 87, 61-614, Poznan
Poland
dst@amu.edu.pl

Abstract. The issue of this paper is bound with the project of building a commentary system for a soccer game. We surveyed an example of natural commentary in order to examine various sorts of comments used by a human reporter to cover the soccer match. The content and function of the messages was taken into account. This survey enabled us to create a conceptual representation for the RoboCup soccer game, which constitutes the input for our commentary generation system. We also give some details about how the representation is created from the raw quantitative data. The presented results may be interesting for human-machine communication researchers and software engineers involved in a similar project.

1 Introduction

Multi-agent system is an environment in which several autonomous individuals coexist. Such systems provide a rich problem domain and offer new challenges for artificial intelligence and robotics. Research effort focuses mainly on various interactions between the agents. This includes general topics like cooperation, competition and communication between the agents. Both new learning techniques and inventive knowledge modelling have to be developed for these problems. The issue presented in this paper is oriented more toward the aspect of the communication, though the information conveying between the agents is not directly studied. We study various forms of the commentary used for describing a situation and events in a multi-agent game. Our research is focused on the presentation techniques used by a human reporter to comment a soccer match. This research topic has been suggested in [7]. The practical purpose of this project is building a soccer commentary generator based on empirical information. The methodological steps leading from the system design to its evaluation are also our research objectives.

The number of works on the automatic generation of soccer commentary is limited. We are aware of three important projects from this area ([1], [2]) namely Soccer, Rocco, and Mike. Two of them are directly connected with the RoboCup environment. These projects are oriented toward the various topics like an explanation of complex events, description of spatial relations, and dynamic content selection. So

A. Bredenfeld et al. (Eds.): RoboCup 2005, LNAI 4020, pp. 496 – 503, 2006.

far, the commentary forms have been studied only in [6]. The obtained results are interesting, but from our point of view they are too general. A more detailed survey of a human commentary is necessary for our purposes. The approach proposed in this project depends on empirical data. We suggest analysing an example of a commentary text[1], which is a video recording transcription of the real match. By examining the example we can learn how the reporter covers various match events and situations of the game. The analysed text had been divided on small coherent blocks of comments. These blocks were next surveyed and classified. More details of this part of the research can be found in [5]. The analysis enabled us to identify the concepts within the domain of soccer game and the various forms of comments appearing in the text. Then, we tried to determine a relation between the elements of the conceptual representation and the natural language expressions. In this way, we hope to construct an automatic generation system, which could potentially achieve more natural and cohesive output. Since the inspiration for our work comes from the RoboCup program, we started to develop our commentary system for this platform [4]. There are some practical advantages using RoboCup. RoboCup offers many ready-to-use logs of simulated games. The logs contain the low-level data. On such basis we can automatically create a conceptual representation of a game, which is in essence a high-level record of the match course. One disadvantage is that many essential commentary concepts like a team history, personal information, have a rather limited use in the RoboCup competition. That's why we neglected a number of the important background concepts. However, we believe that the RoboCup games may be successfully utilize to evaluate the accuracy of the generator.

Next in the other part of this paper we present the part of our project connected with building the conceptual representation of a soccer game. First, we show how the concepts have been selected on the basis of the example survey. Then we deal with some technical details of converting the Soccer Server raw output into conceptual elements of the match representation. This problem was an important part of our work. Finally, we present a short example of the match representation.

2 The Conceptual Representation of the Game

In order to generate a human-like commentary one have to identify the concepts contained in the comments of a human reporter. To the best of our knowledge the previous projects didn't include a serious investigation of the commentary conceptual structure. Although it is difficult to evaluate, it seems that their output text incorporated many unnatural elements. For example, the statements of the Mike generator seem to contain too much explicit statistical information. The analysis of the example commentaries let us select the most suitable conceptual elements and surely increases the conceptual coverage of the system. To determine what kind of information is important for a commentator we analysed the content of the particular messages. In order to illustrate the method of gaining these elements, we present some example comments and the attributes, which can be ascribed to them.

[1] The analysed test as well as the generated output is in polish language.

Comment	**Attributes (content)**
Ronaldo	actor-of-action
Immediately on the penalty area	time-of-reaction, region-of-pitch
But only corner kick	result-of-action,
Good interception by the French team	type-of-action, team-of-action, result-of-action
There is Leonardo on the penalty area	actor-of-action,region-of-pitch
Dangerous Cafu dribble	type-of-action, chance-for-goal, actor-of-action
Long pass on the left side	type-of-action, qualitative-distance, region-of-pitch
Two defenders staying next to him	number-of-opponents-staying-close
This is his first dribble	type-of-action, order-number-of-action

To compare the distribution and establish the importance of different attributes we selected approximately 280 comments from the first 25 minutes of the commentary and examined their content. The distribution of the attributes in the comments is as follows:

Action information	**Distribution in %**
Comments containing actor of action	76 %
Comments containing type or result of action	31 %
Comments containing team of action	12 %
Spatial and temporal information	
Comments containing region of the pitch or direction	5 %
Comments containing time of player's reaction expressions	3 %
Comments containing dynamics or speed of action expressions	3 %
Comments containing distance expressions	1 %
Comments containing statistical information	**3 %**

Taking into account the result of our survey we have prepared a set of elements representing a course of the RoboCup soccer match. In the current version of our system the most fundamental elements of the conceptual representation have been implemented. First of all, the attributes representing the actions and main game events have been included. We also added the most important spatial concepts and some team performance measures. We focused on these elements, which seemed to be the most appropriate for the RoboCup domain. The current set of the attributes encompasses:

1) Attributes representing the individual action of player and match events

type-of-action : Pass, Shot, KeepBall, Dribble, Defence, KickOut, etc.
result-of-action : Receive, Loss, Out, Goal, Corner, Interception, etc.

actor-of-action	: identifier of the player
type-of-event	: Beginning-of-match, End-of-match, Break, etc.
state-of-action	: Active, Finished

2) Attributes representing the action of team

state-of-action	:Beginning, Course, Break, Continuation, End
type-of-action	: Counter-Attack, Positional-Attack, etc.
team-of-action	: name of the team

3) Attributes of the individual and team's action

region-of-pitch	: Middle, Rival-penalty-area, Left-half, etc.
distance	: appropriate qualitative value
direction	: Right, Left, Forward, Back
time-of-reaction	: Immediately, Normal, Slow
speed-of-action	: appropriate qualitative value
dynamics-of-action	: Acceleration, Constant, Slow-Down
order-number-of-action	: a counter of team actions
chance-for-goal	: None, Little, High, Very-High

4) Attributes describing player's situation

number-of-opponents-staying-close	: a number
number-of-team-mates-staying-close	: a number
player-position	: Free, Marked
distance-to-enemy-goal	: a value from {Far, Near, Very-Near}

5) Attributes describing team's performance

initiative	: a degree of team's prevail in the match
ball-possession	: a percent of ball possession in the match
dangerous-situations	: a counter of team's dangerous situations

6) Attributes representing the match situation

time-of-game	: current game time
result-of-game	: current game score

We hope that after providing some background commentary concepts this representtation would be rich enough for generating an interesting soccer commentary. The distributions of the particular attributes may be applied to defining the generation rules, so as the proportions were comparable with a human text. On the other hand the collected data can be used to evaluate quality of the generated output. After having selected the main elements of the conceptual representation, we had to cope with the problem of extracting these concepts from the Soccer Server low-level data.

3 Building the Conceptual Representation

In this section the way of converting the Soccer Server data into the conceptual representation is outlined. The raw input received directly from the Soccer Server or loaded from a game log is consisted of the records containing:

- current location, state, and orientation of the players,
- current location of the ball,
- current score, time, and mode of the game.

This is the low-level quantitative layer, because the data provides us with no conceptual information about the events and the situation of the game. Such representation can not be directly used for a commentary generating. So, this is the starting point for building a higher conceptual layer of a match representation. The raw data is processed in order to extract the necessary conceptual information. After the conversion, a file which contains the conceptual representation of the match is returned as a result. Such file can be used as an input for the generator. The conversion of the raw data can be divided into a few successive steps executed for each time frame. The main routine of the conversion is as follows:

```
t = current_time_of_game();
calculate_values_of_spatio_temporal_primitives(t);
new_mode_event_search(t);  //a new mode event search
new_state_event_search(t); //a new state event search
if (lastKickEvent->type-of-action == KeepBall) ||
   (lastKickEvent >type-of-action == Dibble)
{
  check_player_properties(currentKickEvent->actor-of-action,t);
  report_player_properties(currentKickEvent->actor-of action);
}
update_team_action(LeftTeam);
update_team_action(RightTeam);
```

1. First, we calculate the basic geometrical primitives of the representation. They represent basic properties of the objects and different spatial relations which occur between them. This includes:
   - Velocity and acceleration of the movable objects.
   - Finding a player who is nearest the ball.
   - Tracking the ball movement (velocity change and direction change).
   - Region of the pitch where player is located.

2. Then, we check for new events. The events can be divided into:
   1) Signalised by the change in the game mode (*newModeEvent*).
   2) Signalised by the change in the player's state (*newKickEvent*).

Every state event is in essence a player's action represented by a separate object of *SingleKickEvent* type. The system stores the information about every three successive events: *newKickEvent*, *currentKickEvent*, and *previousKickEvent*, because they can be utilized for recognizing the type (i.e. *pass* or *dribble*), and the result (i.e. *goal*) of the action. If a new event is detected then an appropriate action is being taken depending on the event's sort. For example when a new kick action is detected then the following code is executed:

```
newKickEvent = new SingleKickEvent(time, actor-of-action);
       //a new object is being created
if (currentKickEvent != NULL)
{
  currentKickEvent->setFinishEvent(newKickEvent);
       //the new kick event finishes the current one
  currentKickEvent->recognize();
       //the type of current event is being determined
  currentKickEvent->setActionResult();
       //the result of the current event is being determined
  currentKickEvent->report();
       //this call reports the type and the result
};
```

If a new kick action has been detected then a type of the current one is determined. We experimented with using the recognition automata [2] to model the basic actions like *pass, dribble, shot*, and etc. An interesting idea for recognizing the complex actions in a dynamic multi-agent environment has been outlined in [3], but it still remains purely theoretical because of the computational complexity. Currently we prefer to use a simpler technique that is based on matching the patterns of successive kicks and the game events. The recognition of the type and the result of an event is done by special rules which use spatio-temporal primitives and a set of helpful functions like: *team*(*player*) which returns the identifier of player's team, or *is_pass_possible*(*start_time, actor-of-action, team-mate*) which checks whether the pass was possible. For example, one of the recognition rules says that:

```
if (team(actor-of-action)==team(newKickEvent->actor-of-action))
{
  type-of-action    = Pass;
  result-of-action  = Receive;
}
else
{
  if (is_pass_possible(start_time, actor-of-action, team-mate))
     type-of-action = Pass;
  else
     type-of-action = KickBall;
  result-of-action = Loss;
};
```

The above rule just checks whether the ball was transferred between the two team-mates. If yes, the pass is recognized and its result is set to *Receive*. If no, then the pass is recognized only if it was intended by the actor. The result is set to *Loss*. The rules are triggered in a proper order. For each kind of action the appropriate attributes are calculated. For example, length is calculated for a pass, and the numeric value is transformed into a qualitative value: *short, normal, long*. Other attributes are calculated in a similar way. For a player who is in the ball possession the additional properties of his position are determined and reported. These properties include: *player-position,* the number of team-mates and opponents staying close, and *distance-to-enemy-goal.*

3. At the end, we update the data concerning the action of both teams. It encompasses setting the following attributes: *region-of-action*, *dynamics-of-action*, s*tate-of-action*, and *speed-of-action*. Also the *team-of-action* (the attacking team) is updated if necessary. Finally, the performance data are updated.

4 Example

Below, we give a short example of the RoboCup match representation created automatically from the Soccer Server output data. A similar representation of the real match fragments can be created manually, so that we could check how the generator is commenting the real match.

<*time*=1; *type-of-event* = beginning-of-match>
<*time*=6; *type-of-event* = beginning-of-game>
<*time*=6; *team-of-action* = ATTCMUnited2000; *region-of-pitch* = Middle; *state-of-action* = Beginning; *number-of-action* = 1>
<*time*=6; *type-of-action* = KeepBall; *region-of-pitch* = Middle; *actor-of-action* =Yellow8>
<*time*=7; *player-position* = Free>
<*time*=7; *distance-to-enemy-goal* = Far>
<time=7; *team-of-action* = ATTCMUnited2000; *region-of-pitch* = Middle; *state-of-action* = Course; *number-of-action* = 1>
<*time*=8; *type-of-action* = Pass; *direction* = Left; *region-of-pitch* = Middle; *actor-of-action* = Yellow8>
<*time*=11; *result-of-action* = Receive; *region-of-pitch* = Middle; *partner-of-action* = Yellow9>
<*time*=11; *type-of-action* = KeepBall; *region-of-pitch* = Middle; *actor-of-action* = Yellow9>
<*time*=12; *player-position* = Free>
<*time*=12; *distance-to-enemy-goal* = Far>
<*time*=12; *type-of-action* = Pass; *direction* = Left; *region-of-pitch* = Middle; *actor-of-action* = Yellow9>
<*time*=17; *result-of-action* = Receive; *region-of-pitch* = Right-Half; *partner-of-action* = Yellow5>
<*time*=17; *type-of-action* = KeepBall; *region-of-pitch* = Right-Half; *actor*=Yellow5>
<*time*=18; *player-position* = Marked; *number-of-opponents-staying-close* =1>
<*time*=18; *type-of-action*=Pass; *direction* = Forward; *region-of-pitch* = Left-Half; *actor-of-action* = Yellow5>
<*time*=18; *distance-to-enemy-goal* = Far>
<*time*=38; *result-of-action* = Loss>

5 Conclusion

In this paper the conceptual structure of information for a dynamic multi-agent game was presented. The representation is designed for a soccer commentary generation system. The analysis of a human soccer commentary lead us to the representation which contains the attributes representing: the actions, the events and their spatial, temporal, and statistical properties. We also introduced two levels for representing an action of the game. The one for representing the actions of individual players and other level for a team action. Next, we outlined the way the representation is created from a low-level quantitative data. The topic beyond the scope of this paper is a commentary generation. In future research, we want to add more performance measures and the attributes representing the knowledge used in background commentaries. We also want to extract more statistical data from the example and apply them to the generation rules. The evaluation procedure for a soccer commentary generator is another open question. We are planning to test the accuracy of the generator and similarity between computer and human output.

References

1. Andre, E., Binsted, K.., Tanaka-Ishii, K., Luke, S., Herzog, G., & Rist, T.: Three robocup simulation league commentary systems. AI Magazine 21 (1), 2000, 57-66
2. Andre, E., Herzog, G., Rist, T.: On the simultaneous interpretation of real world image sequences and their natural language description: The system soccer. Proceedings of the 8th ECAI, pages 449-454, Munich, 1988
3. Lison, M.: Deformation Parameters in Dynamic Event Recognition. RoMoCo '02, Poland, 2002
4. Stolarski, D.: Reprezentacja konceptualna wieloagentowej gry dynamicznej w srodowisku irtualnym RoboCup. XIII National Seminar on Artificial Intelligence, Virtual Organization, AI-18’2003, Nr22, pp 51-59
5. Stolarski, D.: An empirical approach toward building a soccer commentary generator. Proceedings of 2nd Language & Technology Conference, Poznan, 2005
6. Tanaka-Ishii, K., Frank, I.: Multi-Agent explanation strategies in real-time domains. Proceedings of ACL2000, the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, 2000
7. Tanaka-Ishii, K., Frank, I., Noda, I., Matsubara, H., Hasida, K., Nakashima, H.: Automatic Soccer Commentary and RoboCup. In M. Asada (Ed.): Proc. of the second RoboCup Workshop, RoboCup-98: Robot Soccer world Cup II