IEEE pervasive COMPUTING
MOBILE AND UBIQUITOUS SYSTEMS
www.computer.org/pervasive

Computerized Real-Time Analysis of Football Games

Michael Beetz, Bernhard Kirchlechner, and Martin Lames

Vol. 4, No. 3
July–September 2005

Computerized Real-Time Analysis of Football Games

Using a real-time positioning system, the FIPM game analysis system can acquire action models, infer action-selection criteria, and identify player and team strengths and weaknesses.

Michael Beetz and Bernhard Kirchlechner
Technische Universität München

Martin Lames
Universität Augsburg

Computer systems support many coaching activities performed before and after competitions,[1] such as strategy development and performance evaluation, but competent assistance during games is much more demanding. It requires real-time interpretation of sensor data, the recognition and classification of ball actions, and fast-action game analysis and assessment.

Only recently has high-precision microwave technology enabled computer systems to simultaneously track the position of players and the ball. Furthermore, sensed positional data contains an enormous amount of information that must be correctly abstracted to give coaches optimal support for controlling the competition. To make good tactical decisions, coaches need player-activity profiles, reports on frequency distributions of tactical behavior, and assessments of the opponent's tactical weaknesses. Consequently, real-time game analysis systems must be able to automatically recognize intentional activities in a multiagent system with continually acting agents.

To meet these demands, we've developed the Football Interaction and Process Model and a software system that can acquire, interpret, and analyze this model.[2,3] Our FIPM system can acquire models of player skills, infer action-selection criteria, and determine player and team strengths and weaknesses. We tested it on the RoboCup Simulation League and received promising results.

Key game analysis ideas

Two important ideas guided our FIPM's design and implementation. First, we wanted to construct appropriate models that we could use for analysis and assessment. Second, we wanted to pose model acquisition as a probabilistic inference and learning problem.

Model-based game analysis

Model building in sports requires mapping the real game process into an abstract representation formulated in features specifically designed for game analysis objectives.[4] So, we wanted to formalize the models using a representation of actions and game evolution based on a first-order interval temporal logic, enhancing statements and rules with probabilities.[5]

Recognizing tactical intentions from observable behavior requires a game model that reflects that behavior isn't just determined by the player's actions and skills. Behavior is also the result of a player's interaction with other teammates and opponents—for example, the result of a shot depends on the defensive players' reactions. To further complicate matters, the interaction between opponents changes during a game as players make adjustments. A wing-player who can't dribble against an opponent might start passing more, for example. From the computer science viewpoint,

1536-1268/05/$20.00 © 2005 IEEE ■ *Published by the IEEE CS and IEEE ComSoc*

Related Work in Action-Model Acquisition and Game Analysis

Related research on sensor-based acquisition of action models includes that of Donald J. Patterson and his colleagues,[1] who are learning probabilistic models of the daily activities of Alzheimer patients. The main difference between their work and ours is that the actions they consider aren't interactive and don't have high failure rates.

Also, Ranjit Nair and his colleagues[2] have learned predictive rules for RoboCup Simulation League games but have not yet performed deep analyses of the actions and games. Andrea Miene and Ubbo Visser have proposed formal representations of Simulation League games using first-order predicate logic.[3] However, they don't deal with uncertainty and failures in football games.

REFERENCES

1. D. Patterson et al., "Inferring High-level Behavior from Low-level Sensors," *Proc. Int'l Conf. Ubiquitous Computing*, Sprinter-Verlag, 2003, pp. 73–89.
2. R. Nair et al., "Automated Assistants for Analyzing Team Behaviors," *J. Autonomous Agents and Multiagent Systems*, vol. 8, no. 1, 2004, pp. 69–111.
3. A. Miene and U. Visser, "Recognition and Prediction of Motion Situations based on a Qualitative Motion Description," *Proc. RoboCup 2003 Symp.*, LNAI 3020, Springer-Verlag, 2003, pp. 77–88.

the interactive character of sports games requires rich models of situations, context-specific action models, and causal models of actions that represent various dimensions of player and team interaction (see the "Related Work" sidebar).

Model estimation and learning

In football, ball actions often fail and have nondeterministic effects, especially in offensive play. Nondeterministic action execution and interference with other players often yield strong variations in player behavior. To deal with the nondeterministic outcome of actions, we used probabilistic action models (which we discuss later).

Another source of uncertainty lies in recognizing ball actions based on real position data. Position senders are located in the ball and on the players' shin guards, so many situations exist in which we can't decide who has played the ball. Additionally, the sensor data is sometimes inaccurate.

To deal with these issues, the FIPM system aims to infer m^*, the most likely model of all models m, given sensor data d. In other words, $m^* = argmax_m P(m \mid d)$. We built our models by observing games and applying data-mining and automatic-learning mechanisms to model acquisition.

FIPM: System overview

Applying our system requires a real-time positioning system that can continually track the position of the players' feet and the ball with an accuracy of a few centimeters. The system must also provide position data for the ball that enables our system to accurately detect when a player comes into contact with the ball. Cairos Technologies and Fraunhofer IIS have developed such a positioning system (see the sidebar "A Position-Tracking System"); unfortunately, it won't be available until next year.

So, we instead developed our system using data from the 2003 RoboCup Simulation League's simulated soccer matches (see www.uni-koblenz.de/~fruit/orga/rc03). In this league, two teams, each comprising 11 autonomous software agents, play each other using a simulator that emulates a soccer environment. The simulated players have fairly realistic sensing and action capabilities; each simulated one can have its own action skills, individual play strategy, and physical abilities.

The data we collected comprises log files written as protocols of 48 games. The main differences between this data and that of the forthcoming positioning system are that the simulation league data is only 2D, and it contains additional events—most importantly, referee decisions. Detecting referee decisions is necessary because it indicates episodes where the game is interrupted and thus doesn't need require analysis.

A five-layer game model

Our FIPM contains five layers—*position and motion*, *action*, *situation*, *tactical*, and *assessment*—each layer permitting specific analyses and forming the basis for inferences at the next layer.

The most basic layer represents the positions and motions of all players and the ball. The Cairos real-time positioning system will produce seven to nine Gbytes of position data for a game, so we need to be able to compactly store the data to lower storage costs. More importantly, data compression and structuring accelerates data access by substantially reducing the search space for the subsequent interpretation and analysis of position information. To compress and structure position data, the FIPM segments the continuous motions of players into curve segments such that we can approximate each segment with a specified accuracy using simple curve functions.[3]

Game representation at the position and motion layer lets the FIPM system calculate movement profiles for each player, either from the beginning of the match to monitor total exertion or within a certain timeframe to determine the actual physical load. You can also draw maps of preferred player or team posi-

A Position-Tracking System

Cairos Technologies and Fraunhofer IIS have developed a position-tracking system (see figure A) that relies on tiny microwave senders placed in the ball and the players' shin guards. These senders broadcast signals including the sensor identifier with high frequency (the ball with 2000 Hz, and the shin guards with 700 Hz). Eight antennas placed at optimal positions in the stadium receive the signals. A position estimation system then determines the position of each microwave sender through triangulation. The position estimate is typically within five to eight cm of the actual position.

In April 2003, Cairos Technolgoies and Fraunhofer IIS demonstrated an operational prototype of their system on one player and a ball. This fall, they plan to apply a working system to an official competition ball, and then they hope to extend the system to operate for complete teams.

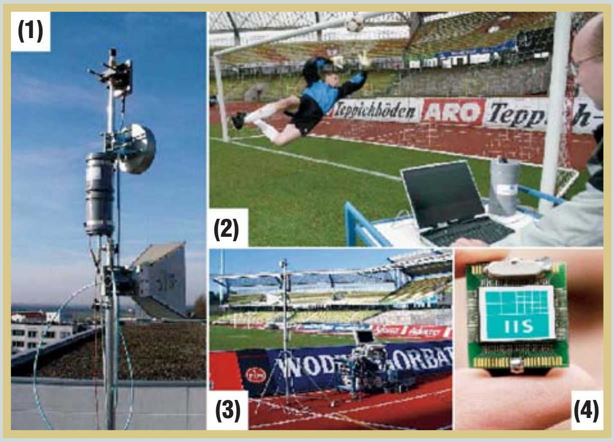


Figure A. The microwave-based position-tracking system by Cairos Technologies and Fraunhofer IIS: (1) the receiver; (2) test installation—senders are in the ball and the players' shin guards; (3) test installation of the position estimation system; and (4) a miniaturized sender.

tions so the FIPM system can analyze the tactical line-ups.

The action layer uses the positional data to help the FIPM system infer activities such as ball possession, dribbling, passing, and goal shots. The statistics of ball actions imply activity profiles, which let the FIPM system identify key players. Also, the representations of passing dyads suggest promising ways to intervene in the opponent's game.

The situation layer draws even more abstract conclusions. Identifying typical situations in football, such as standard position-oriented attacks, counterattacks, and kick-and-rush attacks, depends on the positions and actions of several players and their opponents. The situational analysis reconstructs a team's strategic preferences depending on the game time and standings.

Using a representation of a match as sequences of situation transitions, the FIPM system analyzes tactical team behavior. Transitions between situations are often direct consequences of tactical decisions. Coaches are especially interested in transitions that lead to scoring opportunities. Determining the frequency of situation-specific losses of ball possession enables more informative analyses of tactical behavior.

The highest layer of game analysis assesses tactical metastructures such as playing styles or competition strategies. At this layer, the FIPM system can identify team strengths and weaknesses, which coaches can then use to make necessary changes on the field.

Software architecture

Our FIPM system consists of four functional components: an interpreter, a game database, a situation and action model miner, and a game analyzer (see figure 1).

The FIPM interpreter receives input from the sensor system. It then structures the motions into discrete ball actions, classifies ball actions and situations, and stores the inferred action and situation models in the game model database. It interprets the position data in two steps. First, it probabilistically estimates the game's state and the players' and ball's motion to account for missing data and inaccurate measurements. The result is a complete motion and event model of the game. Then, it parses the motion data to segment it into meaningful episodes, classifying the episodes as passes, dribbles, and so forth.

The FIPM system performs position data interpretation for all the games it processes, so it eventually creates a set of databases, each of which represents a particular game's player and ball positions and motions as well as its action model.

Conceptually, we designed the game database using first-order predicate logic, so the relations stored in the database are ground instances of the predicates. The database primarily uses the following five predicates:

- *pos*(o, p, t_i) is true if object o is at position p at time t_i.
- *occurs*(*ev*, *int*) is true if and only if the event *ev* occurs in the time interval *int*.
- *eventType*(*ev, type*) is true if and only if the type of event *ev* is *type*.
- *situation*(*sit*) is true if *sit* is a situation that is a tuple of positions, one position for each player and the ball.
- *at*(*t, sit*) holds if *sit* is the game situation at time instant *t*.

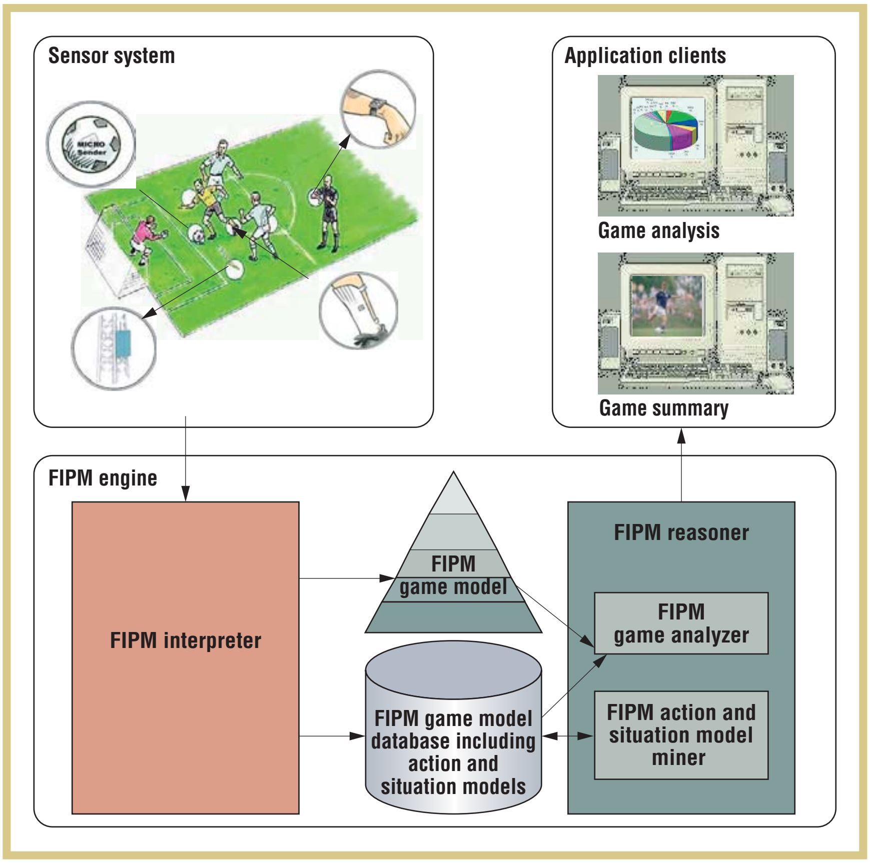


Figure 1. Software architecture of the FIPM game analysis system. The sensor system sends data to the FIPM interpreter, which models the data. The FIPM database stores the models, and the FIPM game model enhances them with background knowledge. The FIPM reasoner then analyzes the game model and displays the data using the application clients.

Situation and action model miner

Many predicates that the FIPM system uses to analyze games must be defined in terms of the game's evolution. For example, a *scoring opportunity* is a situation in which the player is likely to score; a player is *under pressure* if he or she is likely to lose possession of the ball. To help the FIPM recognize such situations, we must specify them in terms of features derivable from position data and game models.

So, the FIPM model miner takes as input a feature language, a concept to be learned, and a set of games. It then learns a set of situation-specific classification rules for the concept based on the specified set of games.

Consider the following. Say we give the FIPM model miner a feature language that contains, among other features, *distance*—the ball's distance to the goal—and *scoringAngle*—the sum of unblocked angles toward the goal. The concept to be learned is the predicate *ScoringOpportunity*. We specify this concept in terms of a data-mining task as the set of situations in which players tend to shoot, weighted by the probability of success.

This solution is elegant for two reasons. First, the model miner learns concepts based on observations of previous games and captures the effects of situations on the game process. (During the game, FIPM constructs only models that are local to the observed game and applies learned models computed from previous games to the current one.) Second, the model miner can automatically learn specific concept definitions for youth and professional football games by just changing the set of games used for learning. One example of a rule the model miner learned is

$$\forall sit. \begin{array}{l} scoringAngle(sit) \geq 35.6^{\circ} \\ \wedge\, distance(sit) \leq 16.38 \\ \xrightarrow{86\%} ScoringOpportunity(sit) \end{array}$$

This rule states that game situations in which the offensive player with the ball is less than 16.38 meters away from the goal, and in which the largest angle to the goal not blocked by a defensive player is at least 35.6°, constitutes a scoring opportunity with a probability of 86 percent.

Game analyzer

The game analyzer applies action and situation models to analyze player and team actions and performances in a given football game. So, to recognize scoring opportunities, it applies rules the FIPM model miner learned to situations occurring in the game. Thus, if the rule we just listed matches a given game situation, the game analyzer classifies the situation as a scoring opportunity.

Acquiring action models

A key FIPM capability is the automatic acquisition of action models. Computer systems that analyze actions as complex and diverse as those in football must be equipped with rich action models. Here, we analyze "shooting the ball" to show how the FIPM system can automatically acquire certain action models.

Observation models for shots let the FIPM system recognize shots, distinguishing them from dribbles or passes. Intuitively, a shot is a kick performed with the intention of scoring. However, because FIPM system can't observe the intention, it must infer it from observable features. For our observation model, we consider shots to be ball kicks that result in a goal or result in the opposing goal keeper catching the ball.

Correctly classifying failed shots is more difficult. To get an effective criterion, we assume that the ball's motion for a failed shot resembles that of a successful one: the ball moves quickly toward the goal. So, the FIPM model miner can create the observation model of failed shots by observing successful ones.[3]

Causal models of shots are specifications of the conditions under which shots typically succeed. To account for football's interactive nature, the FIPM system acquires situation-specific models of shooting skills. So, the FIPM model miner learns situations in which shots are likely to succeed and fail, taking the same approach as for learning what constitutes a scoring opportunity.

To formulate *predictive models* of successful shots, we create a language in which we define certain features:

- *Distance*—the distance between ball and goal.
- *Defenders*—the number of defenders between the ball and goal.
- *Dribbling length*—the minimum distance the attacker could carry the ball.
- *Scoring-angle*—the largest unblocked angle segment toward the goal.

We expect these situation features to be highly correlated with the success and failure of shots. We also take all situations in which the players shoot and represent them in terms of our feature language, recording those that result in a goal. The FIPM model miner applies decision-tree learning to learn predictive rules for the shot success and to obtain compact, general rules with high predictive accuracy.[6]

For example, the FIPM model miner learned the rule

$$\begin{aligned}\forall t_1,t_2,ev,p,sit.\big(&occurs(ev,[t_1,t_2])\\ &\wedge at(t_1,sit)\\ &\wedge eventType(ev,shot(p))\\ &\wedge shootingAngle(sit) \le 44.4^{o}\\ &\wedge defenders(sit) \le 1\\ &\xrightarrow{70\%} succeeds(ev,t_2)\big)\end{aligned}$$

This states that shots taken in situation *sit* occurring at start time t_1 of the shot time interval $[t_1, t_2]$ has a 70 percent chance of succeeding if the shooting angle is at most 44.4° and if only one defender is between the ball and goal.

The FIPM model miner can also acquire *action-selection models* by learning situations in which players shoot, pass, and dribble. We give, as a training set, all ball actions performed in a set of games and the situations in which they were executed. An example action-selection rule that the FIPM model miner learned is

$$\begin{aligned}\forall t_1,t_2,ev,p,sit.occurs&(ev,[t_1,t_2])\\ &\wedge at(t_1,sit)\\ &\wedge shootingAngle(sit) \le 31.2^{o}\\ &\wedge goalDistance(sit) > 13.2\\ &\xrightarrow{93\%} \neg\ eventType(ev,shot(p))\end{aligned}$$

This rule specifies that if an action is performed over the time interval $[t_1, t_2]$, and if the situation *sit* at action start t_1 is such that the ball's distance to the goal is more than 13.2 meters, and if the unblocked angle toward the goal is at most 31.2°, then the action is probably not a shot (93 percent).

Comparative action models compare properties of actions, such as the shooting skills of two teams. The model miner creates this rule by learning situations in which one team succeeds and another fails. This kind of model helps identify player and team strengths and weaknesses. The model miner learns the comparative rule from two action models that were learned themselves. A *combined action model* combines the predictions of component models. For example, a combined model might predict situations in which players could have scored had they shot the ball. This rule is learned from the two models listed earlier: the success-prediction model and the action-selection model. The action selection model offers advice on how to better select shooting situations.

A key FIPM capability is the automatic acquisition of action models. Computer systems that analyze actions as complex and diverse as those in football must be equipped with rich action models.

Many other kinds of action models exist that the FIPM model miner could acquire and use for game analysis. For example, the system could learn situation-specific rules of where and how players shoot. Or, it could learn the spatial distribution of passes.

However, note that when we learn action models by observing a set of games and apply these models to another game, we implicitly assume that the events we classify are drawn from the same distribution that the system learned from. The other caveat is that action models learned from small sets of samples might not be reliable. This implies that we shouldn't blindly apply action-analysis models to a game and that rules with high predicted accuracy, learned from big sample sets, are more reliable.

Game analysis

At the position and motion layer, the FIPM game analyzer can infer the tactical lineup of teams. For this purpose, the FIPM models a player's positions as a Gaussian distribution over all the players' positions. We then represent the inferred Gaussian using an ellipse, and

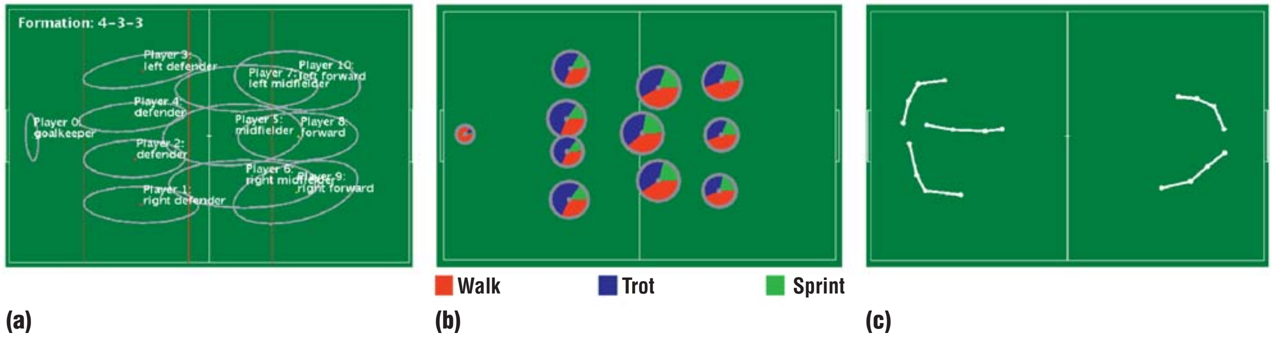


Figure 2. The FIPM game analyzer uses player positions to infer player roles: (a) a team's tactical lineup; (b) motion distances (visualized as the size of the circles) and motion profiles of the players; (c) prototypical preparation of shots. The sequences of ball actions are spatially clustered.

the mean is the mean value of the player's positions. The player's orientation is the axis of the player motion showing the largest variance. The ellipses are set to cover 65 percent of the positions.

We can further analyze players' positions by clustering them along the length of the field to obtain clusters for defensive players, midfielders, and offensive players. Using these clusters, the FIPM game analyzer infers the players' roles (see figure 2a).

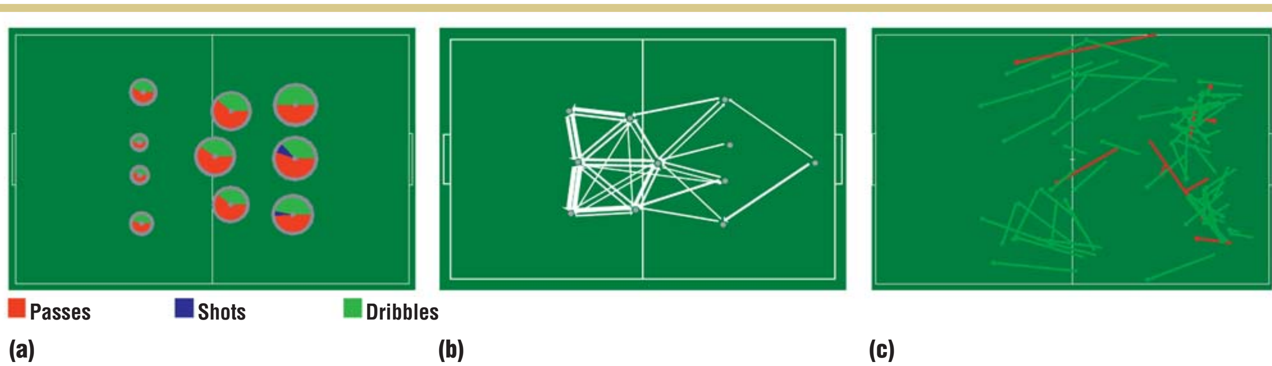
The ability to analyze a player's physical strain is also particularly useful for coaches. So, we infer the distance each player covers and classify the player's motions into standing, walking, trotting, and running. Figure 2b shows that attackers perform more sprints and walks, whereas defenders do more trotting and cover the largest distance.

At the action layer, the FIPM system infers information about the actions that players execute and their parameters. Figure 3a reveals that although midfielders and attackers carry out most ball actions, only two attackers deliver shots. Figure 3b shows the main passing channels, and figure 3c shows the spatial distributions of passes made by three players.

We've already discussed analysis mechanisms at the *situation layer* in the context of acquiring action models. Furthermore, the situation-specific rules of the action models interpret situations in the context of action execution. This close coupling between action and situation models is necessary because of the interactive nature of football games—and sports games in general. We're currently working on automatic recognition mechanisms of standard settings such as corner and free kicks.

At the tactical layer, our current research focuses on finding clusters of game-state transition sequences that lead to scoring opportunities. In figure 2c, we spatially clustered the last three ball actions that resulted in scoring opportunities using the Euclidean distance measure. We can see that the team playing from left to right attacks over both wings and the middle, whereas the other team prefers wing attacks.

Figure 3. Analyses of ball actions. (a) Players' action profiles. The pie charts show the fractions of passes, dribbles, and shots. (b) Passers and receivers and the number of passes between two players. The number of passes from one player to another is depicted by the width of the corresponding arrow. (c) Spatial distribution of the passes three players made.



the AUTHORS

Michael Beetz is a professor and the head of the Intelligent Autonomous Systems research group in the Computer Science Department at the Technische Universität München. His research interests include physically embedded intelligent systems, automatic acquisition of action models, and plan-based robot control. He received his PhD in computer science from Yale University. Contact him at Fachbereich Informatik, Technische Universität München, Boltzmannstr. 3, D-85748 Garching bei München, Germany; beetz@in.tum.de.

Bernhard Kirchlechner is a doctoral student in the computer science department at the Technische Universität München. His research interests include investigating novel computation mechanisms for the automatic acquisition of action models from position data. He received a Diploma degree in physics from the Technische Universität München. Contact him at Fachbereich Informatik, Technische Universität München, Boltzmannstr. 3, D-85748 Garching bei München, Germany; kirchlec@in.tum.de.

Martin Lames is a professor of movement science and training science at Augsburg University. His research interests include game analysis, model building and simulation in sports, and dynamic systems modeling in sports. He's also contributed to the methodology of game observation, performance analysis in sports games, computer science applications in sport, and training science. He received his PhD in sports science from Mainz University and his Habilitation from Kiel University. Contact him at the Inst. for Sport Science, Sportzentrum, Universitätsstraße 3, D-86135 Augsburg, Germany; martin.lames@sport.uni-augsburg.de.

At the assessment layer, we identify team strength and weaknesses. The following rule compares two teams in the RoboCup Simulation League with respect their shooting skills:

$$
\begin{aligned}
&\forall t_1, t_2, ev_1, p_1, p_2, team_1, team_2, sit. \\
&\quad occurs\left(ev_1, \left[t_1, t_2\right]\right) \\
&\quad \wedge occurs\left(ev_2, \left[t_1, t_3\right]\right) \\
&\quad \wedge at\left(t_1, sit\right) \\
&\quad \wedge defenders(sit) \leq 1 \\
&\quad \wedge dribblingLength(sit) \leq 5.46 \\
&\quad \wedge shootingAngle(sit) \leq 47.4^{\circ} \\
&\quad \wedge eventType\left(ev_1, shot(p_1)\right) \\
&\quad \wedge eventType\left(ev2, shot(p_2, team_2)\right) \\
&\quad \xrightarrow{100\%} succeeds(ev_1,) \wedge \neg\, succeeds(ev_2)
\end{aligned}
$$

In particular, this rule specifies situations in which good teams typically score and bad ones don't. The condition is that the way to the goal is blocked within some distance (≤ 5.46 m) and when the shooting angle is moderately big ($\leq 47.4°$).

The research we've reported on here is part of a larger project we're working on that aims to

- investigate novel computational mechanisms that enable computer systems to recognize intentional activities based on position data,
- develop an integrated software system to automate game interpretation and analysis, and
- demonstrate the impact of automatic game analysis on sports science, football coaching, and sports entertainment.

Regarding the integrated software system, we intend to showcase the FIPM system next July for the matches at the Football World Championship 2006.

Furthermore, the range of applications for real-time game analysis systems is broad. Companies that commercially perform postgame analyses of football games for professional football teams have shown an interest in the inferences drawn from and the visualizations computed by the FIPM system. Using the system, broadcasters could offer more information, including continual updates on high-scoring players regarding different variables such as completed passes or successful tackles. Additionally, this technology will promote the trend toward customized broadcasts. Future viewers will be able to specify their favorite players and game situations to receive custom-made video streams.

More importantly, this type of sensor-based analysis of intentional activity applies to a much wider range of pervasive computing applications—in particular, to various sensor-equipped intelligent environments.

ACKNOWLEDGMENTS

The Deutsche Forschungsgemeinschaft partly funded the research this article describes.

REFERENCES

1. M. Lames and G. Hansen, "Designing Observational Systems to Support Top-Level Teams in Game Sports," *Int'l J. Performance Analysis*, vol. 1, no. 1, 2001, pp. 85–91.
2. M. Beetz et al., "Watching Football with the Eyes of Experts: Integrated Intelligent Systems for the Automatic Analysis of (Simulated) Football Games," *Proc. 5th Ann. Conf. dvs-Section Computer Science in Sport*, 2004, pp. 177–181.
3. M. Beetz, S. Flossmann, and T. Stammeier, "Motion and Episode Models for (Simulated) Football Games Acquisition, Representation, and Use," *Proc. 3rd Int'l Joint Conf. Autonomous Agents & Multi Agent Systems* (AAMAS 04), ACM Press, 2004, pp. 1370–1371.
4. J. Perl et al., "Informatische Methoden und Modelle zur Lüsung Sportwissenschaftlicher und Sportpraktischer Probleme" (Methods and Models of Computer Science as Solutions for Problems in Sport Science and Sports Practice), *Leistungssport*, vol. 23, no. 6, 1993, p. 1722.
5. J. Allen and G. Ferguson, "Actions and Events in Interval Temporal Logic," *J. Logic and Computation*, vol. 4, no. 5, 1994, pp. 531–579.
6. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.