AI Magazine Volume 19 Number 3 (1998) (© AAAI)

Belief-Desire-Intention Deliberation in Artificial Soccer

Hans-Dieter Burkhard, Markus Hannebauer, and Jan Wendler

■ Many different architectures have been proposed for the design of autonomous agents. In this article, the application of the belief-desire-intention architecture to the artificial soccer domain is described. We show how it supports efficient deliberation in a highly dynamic environment.

The consideration of programs as agents focuses at first on the aspect of autonomy: Programs have to act in an appropriate way to changes in the environment; therefore, they need some input or sensor facilities and some output or actoric components. The mapping from input to output can be done in simple ways (for example, strictly reactive) or in more sophisticated ways, for example, with models that are inspired by human decision processes. We found that mental notions such as capabilities-skills, belief, goals-desires, and intentions-plans are useful pictures to make agent programming transparent. The aspect of rationality forces agents to deal efficiently with their resources, especially with time.

Many different architectures have been proposed for the design of agents. The subsumption architecture (Brooks 1990), for example, is a well-known architecture with different levels of performance. The so-called belief-desire-intention (BDI) model fits best to our concept of agents in artificial soccer (Kitano et al. 1997). The BDI approach is based on the philosophical work of Bratman (1987) and the theoretical and practical work of Rao and Georgeff (1995) and others (cf., for example, Wooldridge and Jennings [1994] and Burkhard [1996]).

Rao and Georgeff characterize typical problem domains that can successfully be solved by BDI. These characteristics fit the artificial soccer domain well: (1) The simulator and the opponents create a nondeterministic environment. (2) The agent itself reacts nondeterministically because parts of the planning process are randomized. (3) The player can have different goals at the same time, for example, reaching the ball while covering an opponent. (4) The success of the player's own commands depends strongly on the simulated environment and the opponents. (5) The whole information is local and different for every player. (6) The environment pushes bounded rationality because reasoning that is too deep is without payoff in a dynamic surrounding.

In the BDI approach, agents maintain a model of their world that is called *belief* (because it might not be true knowledge). The way from belief to actions is guided by the desires of the agent. Bratman has argued that intentions are neither desires nor beliefs but an additional independent mental category. Intentions are considered as (partial) plans for the achievement of goals by appropriate actions. Commitment to intentions is needed that has impact on the rational use of resources: The (relative) stability of committed intentions prevents overload in deliberation and useless plan changes, and it serves trustworthiness in cooperation.

All components of a BDI architecture can be identified in our deliberation process. The following sections give an overview to our realization of this kind of architecture.

Belief—The World Model

The soccer server sends only a partial noisy picture in relative coordinates to the agent, which leads to an individual belief of the world in every agent. The agent cannot rely on the accuracy of the received and interpolated data; therefore, it is belief, not knowledge.

Belief is updated in the component world model in our realization. In a dynamic and uncertain domain such as artificial soccer, a consistent (as far as possible) modeling of the environment is necessary. Short-term false information has to be corrected, imprecise information must be evaluated, and inferences are necessary for missing information. The related algorithms lead to a certain stability of the agent's belief.

To satisfy these demands, the world model provides, for example, basic classes for linear algebra and special classes for every object on the field. Inheritance is strongly used, and additional features such as timed objects, and encapsulated environments make synchronization with the simulated surrounding easier.

The agent's absolute position on the playing field is calculated using the relative visual information concerning lines, flags, and goals. The triangulation considers all possible cases (actually several hundred). The agent's velocity is estimated from the movement commands sent to the simulator. Absolute positions of all other seen objects can be computed because the own absolute position is known. A special algorithm matches new information on unnamed objects (for example, a player with a missing number) to known objects. The world model can also close the information gaps for unobservable objects by simulation.

Simulation is also used to predict future situations by using the knowledge about positions and velocities. This ability is extensively used by the planning process to estimate consequences of possible commands. For example, the component *advanced skills* can instantiate a new ball object, simulate it for some time steps, and look at the position and speed. Additional features such as wind can easily be taken into account this way. The world model logs an adjustable number of environments to track the player's history. This ability was implemented to support online learning (not used until now).

Desires—Goal Finding

In our implementation, *desires* are goals that are selected out of a fixed goal library. The list of possible goals is still small, but the set will be extended, for example, to allow joint goals (such as double passes) in the future. In the current realization, different (even opposite) goals can be achieved, but the agent selects only one of them.

The component *planning* embeds the planning process that can be initiated each time a new piece of sensor information has arrived. A new situation is classified as follows: If the ball is under control, the agent is able to pass the ball or dribble. If the player has no control of the ball, it can decide whether to intercept it, watch the game, or run to a certain position. This goal (target) finding is done by a usual decision tree. Some of the decisions are trivial ("Is the ball in the kick range or not?"), but some are really tricky ("Should I run to the ball, or should my teammate do it?"). The latter decision is done by using a distance measure: If the agent supposes to be the first of its team to reach the ball, it will run. If not, it relies on its teammates and runs back to its home position.

Intentions—The Planning

The achievement of the chosen goal is realized by an appropriate plan. Such a plan might last for a larger set of server cycles with new arriving sensor information. This sensor information can be used for the adaptation of a plan (for example, for correcting the agent's direction while it is running for the ball). However, permanent (small) corrections can cost more time than only occasional adaptations. Here we observe a trade-off between adaptation and persistence: Strict execution of predetermined long-term plans can fail because of unforeseen events in a changing and only partially observable environment. However, permanent evaluation and new deliberation for every execution step can lead to many additional actions. Such a procedure could have serious drawbacks for fixed plans with special initialization steps: If new deliberation leads to a new plan, then related initialization steps (for example, turning to a new direction) might be executed again and again. Other drawbacks of too many new deliberation processes might be the extensive use of computing resources (bounded rationality).

We have developed a special kind of deliberation strategy that regards each new sensor information but makes adaptations only in special situations. Our deliberation process can be considered as a special implementation of intention stability. Intentions are considered committed plans for achieving a goal. We have two stages of planning in our system: (1) determining the best way to achieve the intention and (2) computing concrete parameters and single actions. The first stage of the planning process produces a coarse long-term plan with some parameters (partial plan).

There are two major cases to cope with: The agent is out of the kick range, or it can control the ball (that is, has ball possession). In the first case, the player calculates an optimal

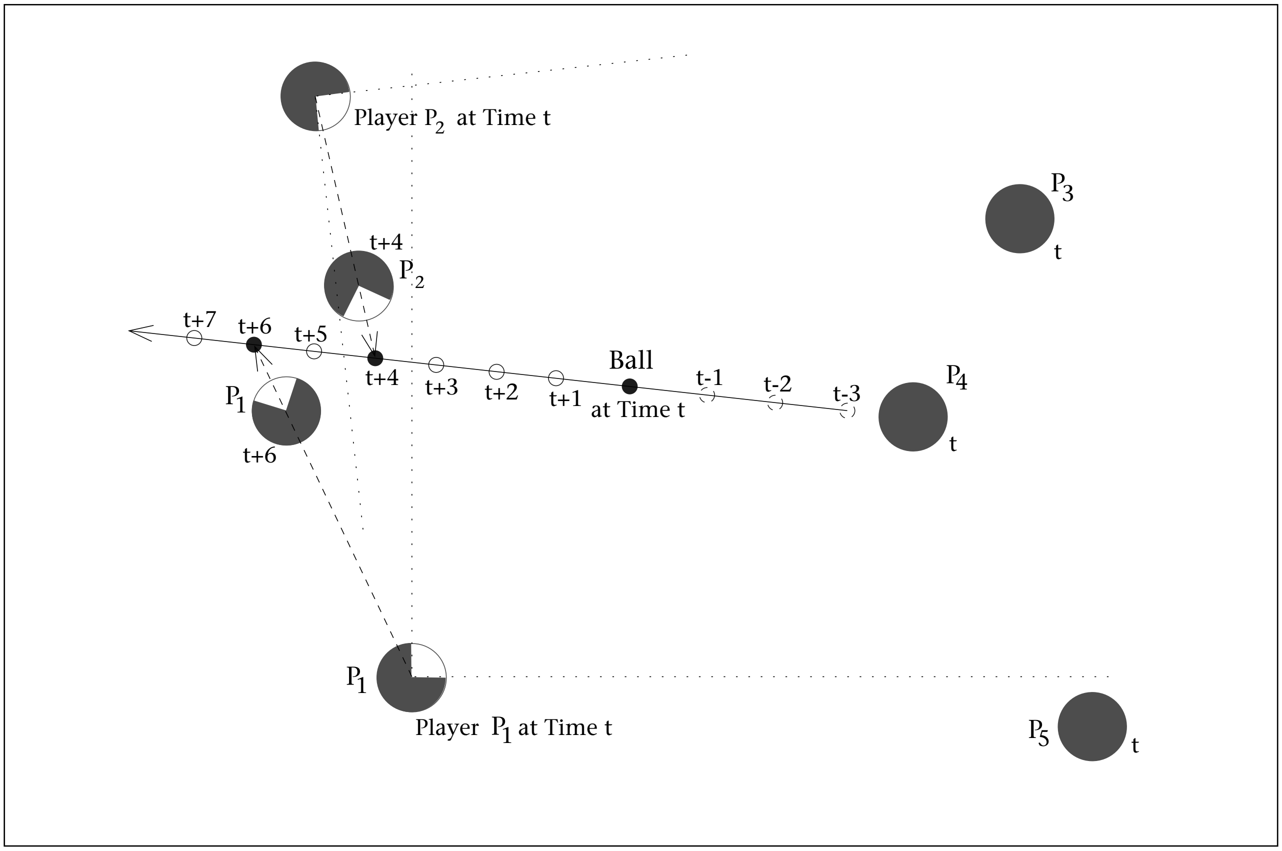


Figure 1. Decision Finding for Ball Interception.

The solid line shows the ball movement with the ball positions at the time steps t ± i. The dotted lines represent the view sectors of the different players P_1 and P_2 at time t. The dashed lines show the movements of the players. P_1 does not see P_2 and P_5. It calculates distance 26 for P_4, 29 for P_3, and 6 for itself. The agent decides to intercept the ball. P_2, whose view sector includes P_1, calculates 6 for P_1, 30 for P_3, 26 for P_4, 39 for P_5, and 4 for itself. It also decides to go for the ball. Hence, both players of the same team go for the ball. P_1 calculates the interception position at ball position t + 6. Likewise does P_2 for ball position t + 4. At time t + 3, a new sensor information comes in. P_1 cannot see the ball. It will keep its plan according to the implementation of our planning component for such situations. P_2 sees the ball and also continues intercepting the ball.

interception position, if it has decided to get the ball. For several reasons (for example, to regard the wind if necessary), we decided to use the simulation capability of the world model. The agent tries mentally to reach the ball in one step, in two steps, and so on, until it finds a certain number of steps in which it can reach the ball. This procedure also provides a distance measure because it can be applied to every player and ball instance. Figure 1 shows this process graphically and gives a commented example. If the player has decided not to intercept the ball, it returns to its home position or (if it is already there) collects information by turning and waiting.

If the player controls the ball, it has to decide whether to pass the ball or dribble. Furthermore, it has to decide in which direction to kick or dribble, respectively. It should prefer a direction with the best chances to score or pass the ball to a teammate. At the same time, it should prefer directions that promote an offensive play style. A fixed single direction is evaluated by a special function that takes into account the distance in terms of the distance measure, as described previously. The minimal distance of the player to the ball and the related mean distances for its own team and the opponent team, respectively, are combined using role-dependent weight factors and a goal-hitting bonus.

Having the function to evaluate single directions, the agent can look for a globally optimal direction. Because of its definition, the evaluation function is normalized to zero, which means that negative values indicate a good direction, and positive values indicate a bad direction. In our recent implementation, several discrete directions are evaluated, and then the best direction is taken. Figure 2 exemplifies such an evaluation process.

Values near zero are neutral; these values are

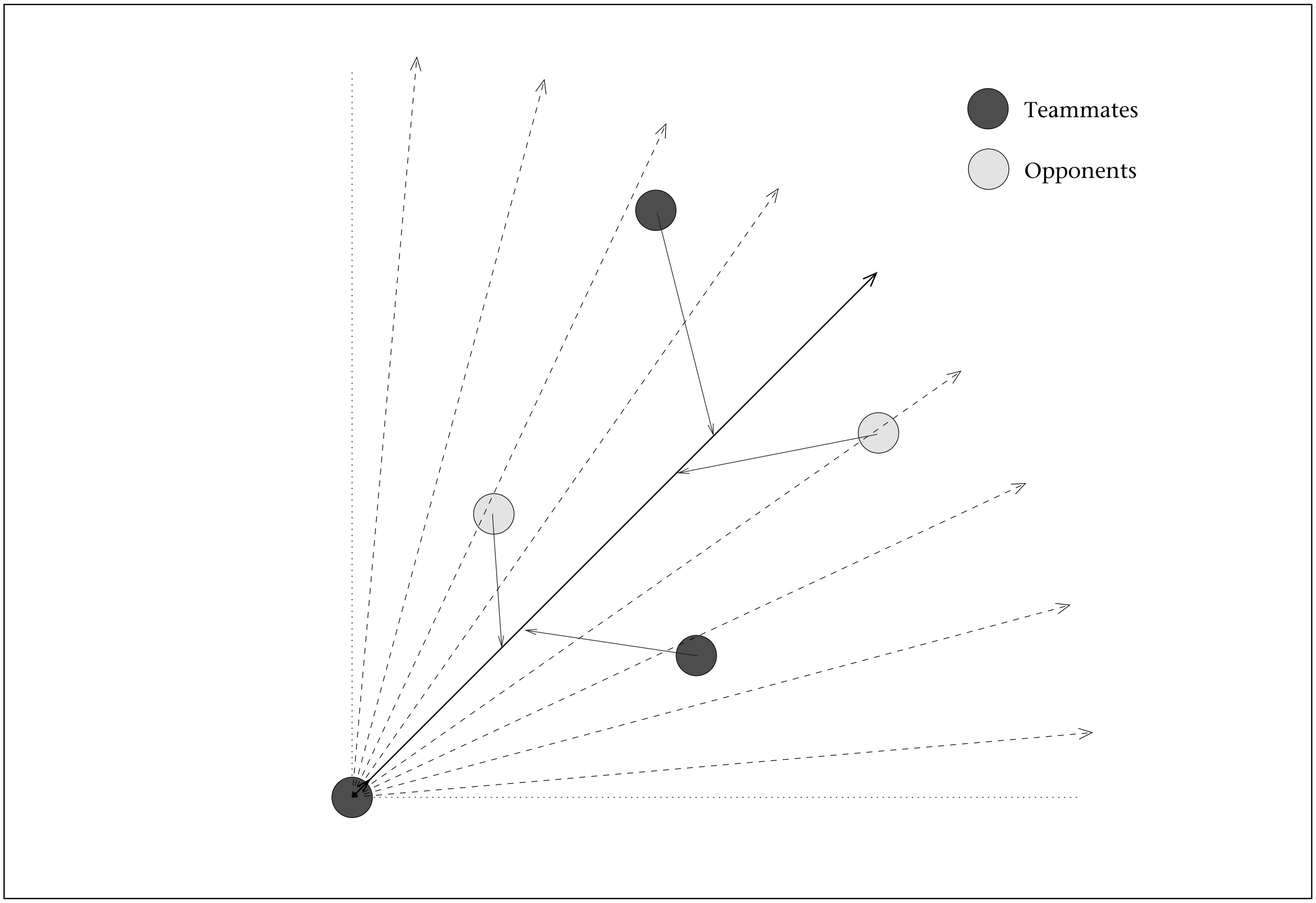


Figure 2. Evaluation of Discrete Kick Directions.

especially difficult to judge. For this purpose, a randomized function was implemented that decides afterwards whether to kick or dribble. It is a function that takes the distance value of the best direction and calculates a role-dependent probability to dribble.

These planning procedures have a certain overlapping with the finding procedure for desires. This overlapping is necessary because the decision process has to look for achievable desires, and the realization of the intentions relies on the capabilities of the agent.

The main problem of the evaluation strategy described here is that players move in unpredictable ways. Therefore, the positions of unseen players are uncertain, which is why we decided to evaluate only directions in the view area (our agents had a view angle of 90 degrees). To prevent too many backward shots, every area on the field has a defined preferred direction. Our first idea was the following rule: Turn to preferred direction, and evaluate the kick directions. If there is no good kick direction, determine a turn direction and turn, evaluate this sector, and so on. This repetitive procedure was a safe way to find the global best direction but was actually too slow because our agents had to wait 300 microseconds (ms) to get the next information. (We did not consider another sensor cycle because we missed one action by sending a view-change command. It is under discussion that such actions should not consume time, which could make the other strategy more interesting.)

Thus, we implemented the following behavior: If the agent looks approximately in the preferred direction (the definition of *approximately* is role dependent), it will do the evaluation process. If not, it will do an emergency kick directed at the opponent's goal. (We did not use communication between the players in the recent implementation. With additional information obtained by communication, we can make a full evaluation of the situation in the future.)

The described stage of planning corresponds to long-term plans with some parameters that can be also seen as partial plans. Its calculated end is the achievement of the selected goal. The second stage is to split the intention into smaller pieces that are implemented as short-term plans in the Advanced Skills component.

This component is a library of skills that facilitate efficient ball handling and optimal movement. The short-term plans produced by Advanced Skills are not longer than the interval between consecutive sensor information (with our preferences, these are actually three basic actions). This way, the long-term plans are executed by iterated calls of Advanced Skills after each sensor information.

It was one of the major decisions during development to use this strategy of plan execution. The more common strategy is to fix a long-term plan that can be adapted during execution if necessary. Such a long-term plan can start with some initial actions to achieve a well-defined situation (for example, a suitable ball position for dribbling). Afterwards, the actions are performed in the fixed sequence according to the plan such that each action relies on the successful execution of its predecessors.

In our strategy, a strictly new deliberation process can start for each new sensor information (actually every 300 ms), and in this case, we have a new long-term plan started just at this time point. If we always needed certain (new) initial actions for preparation, then we might never come to the continuation of a plan. To overcome this problem, the advanced skills are designed to deal immediately with any situation that might appear at the beginning or during the execution of a long-term plan (for example, to continue dribbling in any situation). As a side effect, the advanced skills are able to realize the fastest way for goal achievement in a flexible way from arbitrary start situations.

The main advanced skills of the player agent are Directed Kick, Go to Position, and Dribble. *Go to Position* is used to reach any absolute position on the field. It produces one basic Turn (if needed) and/or up to two/three Dashes. If demanded, this procedure avoids obstacles such as other players. *Dribble* moves the ball into a certain direction without losing contact with it, including the production of several Kick, Turn, and Dash combinations.

The *Directed Kick* skill is a good example for the planning subtask of the advanced skills; therefore, we describe it in detail. This capability allows the players to kick the ball in any direction with a demanded power (as far as possible). It handles difficult situations such as high velocities and the player itself as an obstacle for the desired direction. If the desired direction with the desired speed cannot be achieved, the skill tries to meet the demands as well as possible.

First, the skill tries to determine the kick angle and the power that is necessary to transform the current movement vector into the demanded movement vector (figure 3a). If the length of the necessary kick vector (the power) is physically impossible, the skill tries to at least keep the correct direction. A complication occurs for kick vectors that are possible but hit the player itself. In this case, an intermediate target is calculated that is at the side of the player (figure 3b). The first kick leads to this point, and further kicks are calculated from there (figure 3c). In some cases, the ball can be kicked once more (figure 3d).

As mentioned previously, Advanced Skills provides precompiled plan skeletons of a size that fits between two time points of sensor information. They have their own calculation capability that is used to compute the short-time optimal command sequence. Looking at the mentioned trade-off, these short-term plans are atomic and cannot be interfered with by sensor information. However, in composition, they build a long-term plan that is complex enough to achieve higher goals.

Persistence of Planning

The consideration of our agents as BDI constructs is appropriate because for each new piece of sensor information, we have a complete deliberation process with an update of belief, a choice of a desire, a commitment to an intention, and an execution of a plan part.

A problem arises when the commitment of intentions is mostly performed independently from the previous intentions, which might contradict the principle of stability of committed intentions, a central point in Bratman's theory. The canonical deliberation process has to maintain an old intention as long as there are no serious counterindications.

A new deliberation process is initiated every time a new sensor information comes in, and a new plan is created. It is a special task of our planning strategy to ensure stability of the long-term plans and avoid constantly changing goals or intentions. Our world model gives a good picture of the outside reality; thus, updates usually coincide with expectations. Therefore, a completely new planning process will usually lead to the same goal and a similar intention (plan). Hence, the player must only prevent this missing new belief or slightly better new intentions from leading to unsuitable changes in its behavior. To avoid this, the

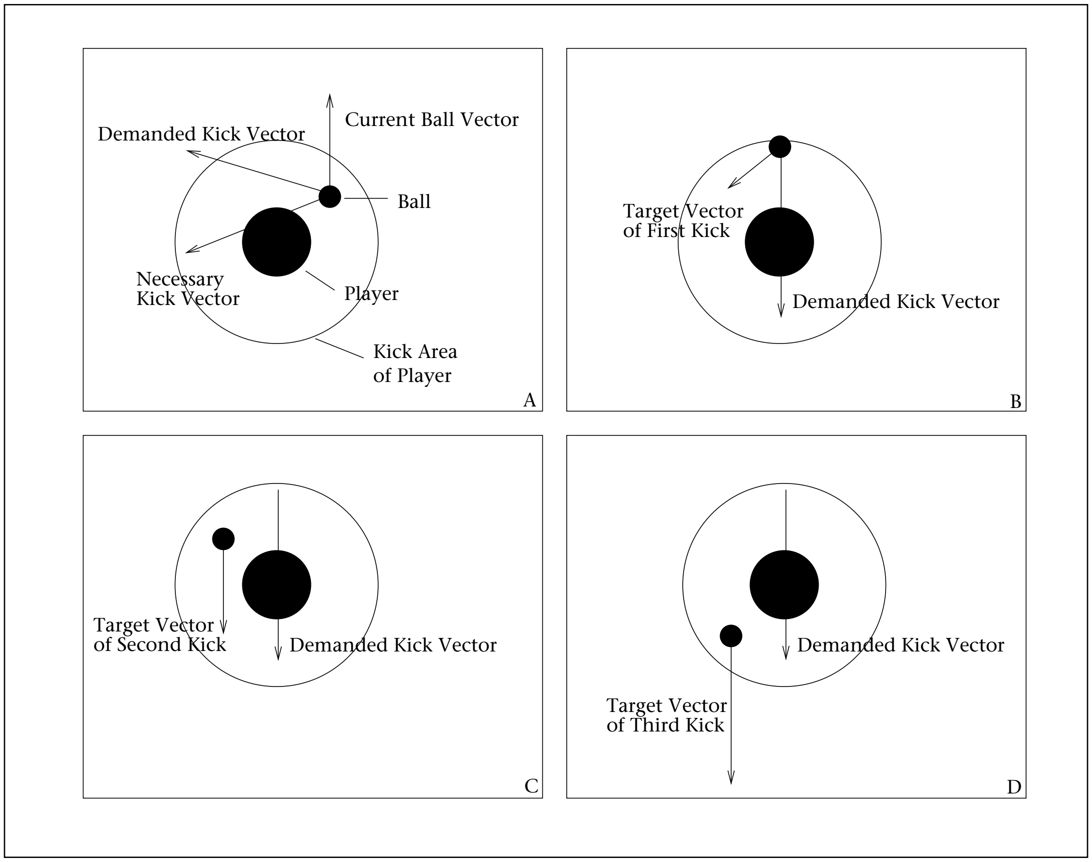


Figure 3. Several Steps of the Directed Kick Skill.

A. This view shows the actual kick that is necessary to give the ball the demanded direction (by vector addition of the kick and the current movement). B. If the player itself is an obstacle for the demanded kick direction, then a first kick must place the ball beside the player. C. The second kick gives the ball the demanded direction. D. If the ball is still in the kick range, then a third kick might give the ball additional power. (According to vector addition, the actual kicks have to regard the current movement of the ball as described in A.)

agent, for example, lets out minor turns on the way to a certain position. (The avoidance of minor turns saves time: Very often, the old direction still leads at least approximately to this position. If the small omitted changes sum up over time, then a related turn would be executed later.) The result of this strategy could be called *implicit persistence.*

The *explicit persistence* of goals and intentions in our implementation can be exemplified by the realization of the goal "Go to home position." To decide whether to intercept the ball or go home, the agent has calculated the minimal distance of all teammates and opponents to the ball and has stored these values. If the decision is to go home, the agent uses these values to determine a time interval in which it must not care for the ball because no other player is able to change the ball's known movement. The decision tree usually strongly relies on sight of the ball, but in this case, the agent won't turn for the ball in the calculated "don't care" interval on its way to its assigned position. The agent avoids the additional actions (turns) that would be necessary to look for the ball. He makes a straight run to the designated position until the "don't care" time is over. In general, the old goal and intention are kept as long as the calculated interval lasts.

We found this to be an interesting implementation of the stability principle for committed intentions without explicitly using the old intention. Our agents are able to adapt a plan to new situations if necessary (for example, a turn command with a greater change would not be dropped). It might be the case that future implementations would need an explicit treatment of previous intentions (for example, if a commitment were given to teammates in explicit cooperation).

Conclusion

An architecture that makes the agent processing transparent is important for the development of concepts and for the implementation.

It is especially valuable for fast refinements and changes of code. In this article, we showed that the BDI architecture can successfully be applied to artificial soccer because of its potentials for deliberation and adaptation in highly dynamic environments and maintenance of stable intentions.

The trade-off between in-deep planning and accuracy to fast-changing situations can efficiently be solved by the presented execution of long-term partial plans by short-term command sequences without losing the necessary persistence.

Until now, the team behavior of our agents has only been implicit. Players act according to their expectations of the behavior of their teammates. Different roles result in an efficient use of the whole playground. However, we have also identified several situations where communication would improve the behavior; so, future development might take into account explicit team play. Explicit cooperation needs an extension of our architecture for team goals, negotiation, and joint plans. Related strategies could lead to a real heterogeneous team with players having their own character but cooperating efficiently.

Another long-term goal is the use of online and offline learning. We have already implemented the storage of histories in the agents, which is to be used for online learning with methods from case-based reasoning.

Acknowledgments

Many ideas and difficult implementation work came from Pascal Müller-Gugenberger, Amin Coja-Oghlan, Adrianna Foremniak, Derrick Hepp, Heike Müller, and Kay Schröter in a practical exercise. We want to thank them for their great efforts. We want to thank all colleagues for the joy and fun that we had together in Nagoya; we especially want to thank our Japanese hosts.

Notes

1. In the future, we might deal with the commitment to concurrent goals. In such a case, we will have to regard the "scope of admissibility" (Bratman) set by previous intentions. For example, an existing intention to preserve the off-side position of an opponent might restrict the commitment to later goals for reaching special positions.

References

Bratman, M. E. 1987. *Intentions, Plans, and Practical Reason.* Cambridge, Mass.: Harvard University Press.

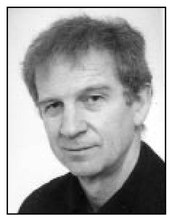
Brooks, R. A. 1990. Elephants Don't Play Chess. In *Designing Autonomous Agents,* ed. P. Maes, 3–15. Cambridge, Mass.: MIT Press.

Burkhard, H. D. 1996. Abstract Goals in Multi-Agent Systems. In *Twelfth European Conference on Artificial Intelligence (ECAI96),* ed. W. Wahlster, 524–528. New York: Wiley.

Jennings, N. R., and Wooldridge, M. 1994. *Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages.* Lecture Notes in Artificial Intelligence 890. New York: Springer-Verlag.

Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Coradeschi, S.; Osawa, E.; Matsubara, H.; Noda, I.; and Asada, M. 1997. The RoboCup Synthetic Agent Challenge '97. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, ed. M. E. Pollack, 24–29. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Rao, A. S., and Georgeff, M. P. 1995. BDI Agents: From Theory to Practice. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95),* ed. V. Lesser, 312–319. Cambridge, Mass.: MIT Press.



Hans-Dieter Burkhard is the leader of the Artificial Intelligence Group in the Institute for Informatics at the Humboldt University Berlin. He has worked on automata theory, Petri nets, distributed systems, VLSI diagnosis, and knowledge-based systems. His current interests include agent-oriented techniques, distributed AI, and case-based reasoning. His e-mail address is hdb@informatik.hu-berlin.de.

Markus Hannebauer is an undergraduate student in computer science at Humboldt University Berlin. His primary research interests are distributed AI and operations research. He was the technical project manager for the participation of AT HUMBOLDT at RoboCup97. Currently, he is working on the formal specification of flexible manufacturing systems with the use of algebraic Petri nets, the scheduling of manufacturing processes with multiagent systems, and belief-desire-intention agent architectures. His e-mail address is hannebau@informatik.hu-berlin.de.

Jan Wendler studies computer science at the Humboldt University Berlin, where he works in the Artificial Intelligence Group. He is currently writing his diploma-thesis about learning in multiagent systems. Other interests include multiagent cooperation and case-based reasoning. His e-mail address is wendler@informatik.hu-berlin.de.