The magmaOffenburg 2009 RoboCup 3D Simulation Team

Klaus Dorer, Mathias Ehret, Stefan Glaser, Thomas Huber, Simon Raffeiner, Srinivasa Ragavan, Ingo Schindler[1]

Hochschule Offenburg, Elektrotechnik-Informationstechnik, Germany

Abstract. This paper describes the magmaOffenburg 3D simulation team trying to qualify for RoboCup 2009. It focuses on two distinctive features of the team: decisions making using extended behavior networks and its software architecture and implementation in Java to open the simulation for the Java community.

1 Introduction

The magmaOffenburg team is the successor team of the 2D simulation league teams magmaFreiburg (2^{nd} 1999, 5^{th} 2000), living systems (13^{th} 2001) and magmaFurtwangen (23^{rd} 2003). One of its main distinctive features has been and still is its decision making using extended behavior networks [4]. This will be described in more detail in section 2.

One main goal of our work is to simplify the creation of teams using Java. Section 3 describes the steps taken to achieve this goal. We hope that in the competition we can show that teams written in Java are well competive and that, for example, the disadvantage of Java performance is non-existing.

2 Decision Making

Decision making in the magmaOffenburg team is done using extended behavior networks (EBNs) [4]. They provide support for reactive and goal-directed behavior selection. This is done using a mechanism of activation spreading to calculate an estimate of the expected utility of each behavior rule combined with situation-dependent calculation of executability of behavior rules. An example network for the soccer domain is shown in figure 1.

EBNs extend original behavior networks [6] in numerous ways. EBNs allow the explicit representation of goals with dynamic, i.e. situation-dependent, utility function. Goals may be prioritized by a static importance value, but also by a dynamic relevance. This helps an agent to focus on the goals that are relevant in a specific situation. Experiments in the RoboCup 2D simulation domain have shown that the performance of a team of agents with dynamic goals was significantly higher compared to an identical team with static goal importance. While the static team scored 6.6 goals per game the dynamic team scored 10.8 goals on average during 30 games played against each other.

EBNs also extend Maes networks through the introduction of continuous state-propositions to exploit additional information in continuous domains. This affects the calculation of preconditions, spreading of activation and calculation of goal relevance. All of them use fuzzy connectors to combine multiple preconditions and activations respectively. Again experiments in the RoboCup domain showed a significant improvement of agents using continuous state-propositions where 'discrete' agents scored 0.7 goals playing against 'continuous' agents that scored 4.9 goals on average during 30 games.

EBNs are further able to do concurrent behavior selection [2]. I.e. the network can decide within one decision cycle to execute multiple actions concurrently if they do not interfere. In order to decide if behaviors interfere, resources are introduced into the model. Two behaviors do not interfere if they do not use any resource in common or if the resources they both use are sufficiently available. In the RoboCup domain an agent may speak, run and turn its head within one decision cycle. Since the number of cycles an agent can communicate is restricted to 4% of all cycles and because separate turning of the head relative to the body was only performed in about 8% of all cycles, concurrent behavior selection effectively only took place in 2% of the cycles. Despite this, the team using concurrent behavior selection scored significantly more goals (4.3) than the team using serial behavior selection (2.4).

Most decision mechanisms for agents only have influence on the decision which behavior the agent should perform, but not on the behavior execution itself. In biological systems, however, the determinedness of a decision has influence on the execution of a behavior. "Intensity and endurance of an activity is determined by the volition strength of the goal intention" [5]. So the more decided a human is to perform a behavior, the more intense is the execution. A measure for the decidedness of EBNs to perform a behavior is the activation value which can be used to influence the intensity of behavior execution. Experiments with influencing the "runToBall" behvior to trade off speed versus stamina showed a significantly higher performance of the parametrized team (11.2 goals) compared to the static team not using influence of decidedness on behavior execution(8.9 goals) [2].

Further experiments with the RoboCup simulator showed that when adding an increasing level of noise to the sensor input of the agents, EBNs show a graceful degradation of their performance.

Comparing EBNs with the original approach of Maes in the RoboCup domain results in a significant improvement of the agents. In direct comparison during 18 games Maes-agents scored 0.1 goals while EBN-agents scored 8.9 goals on average. This is remarkable since both agents used the same set of perceptions and behaviors. The only difference stems from improvements in decision making.

Finally EBNs are an interesting approach to model human decision making that deviates from rational decision making of decision theory [1]. EBNs have been successfully used to reproduce findings of Kahneman and Tversky on human decision making which led to the formulation of prospect theory and was awarded the Nobel prize for economic sciences in 2002 [8]. This will be needed in 2050 when playing against a human team. For more details on EBNs see [3].

The 3D simulation domain creates a couple of new challenges to decision making using EBNs. One problem is that other than in the 2D simulation domain behaviors like kicking need some time to finish. This means that on the one hand side the decision making should be more inert, not changing its decision for kicking after one cycle. On the other side it should still be possible to interrupt a behavior at certain steps if it does no longer make sense. Other extensions currently investigated concern the detection of effect deviation that should be used to influence decision making of the EBN.

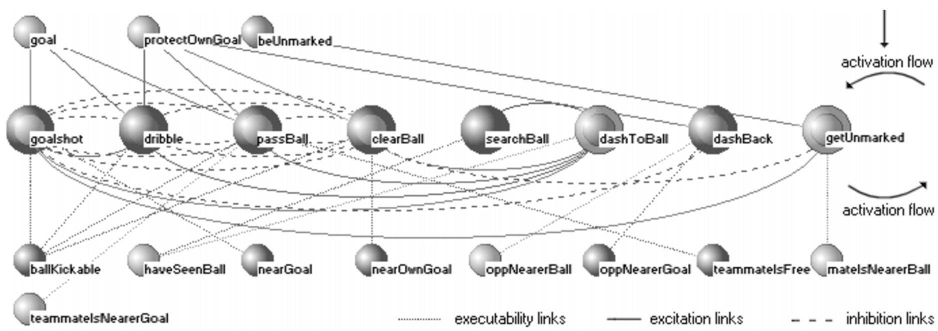

Fig. 1. Extended behavior network for the soccer domain

3 Architecture

One main purpose of our work is to open the 3D simulation for the Java community. We hope that many teams will benefit from our work as we have benefited enormously by the source code release of the Little Green Bats team during our development [7]. This should be achieved with a number of measures taken into account during designing the team:

- Layered, component-based architecture
- Java as implementation language
- JUnit tests to ensure stability
- Releasing source code after the competition

In this section we will mainly focus on the design and architecture of the team.

3.1 Layered Architecture

The layered architecture of our team is shown in figure 2. Currently our agents consist of five layers: communication, protocol, model, control and decision making.

These layers are designed and implemented to avoid dependencies from lower layers to higher layers. This has the advantage for other teams using our code that they can decide to build on any of the layers we provide. Some teams might choose to just reuse the communication layer, others will want to just implement their own decision making. The package structure corresponds to these layers so that also technically it is easy to setup a project on any layer.

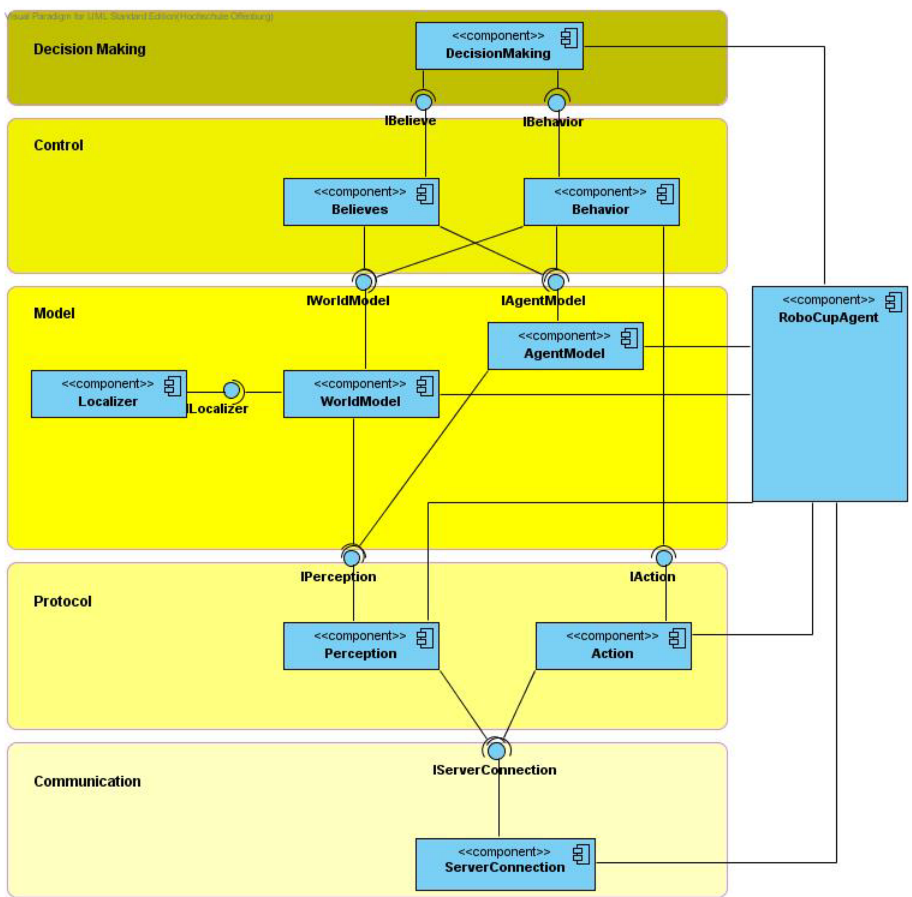

Fig. 2. Layered, component-based architecture of the magmaOffenburg team.

Information flow from lower to higher layers is achieved using the observer design pattern to keep lower layers independent from higher layers. Other components interested in state change of one of our componentes simply attach themselves as observers to the component. This way we implemented, for example, a GUI component showing the world model that is updated whenever the worldmodel is changing. The world model is completely independent of the GUI and the GUI is only loosely coupled to the world model (see next section).

3.2 Component-based Architecture

In addition to allow other teams to build on any layer of the architecture we choose a componente-based architecture to allow other teams also to exchange any of the components with their own implementation.

Interfaces are used to establish loose coupling of the components. Many components provide already loosly coupling to the other components like decision making, believe, behavior, world model, agent model, connection. Current work includes removing tight couplings of remaining components so that each component can be replaced with other team's own implementations.

Dependency injection is used to setup the components and their dependencies. This means that the overall structure of the system can simply be generated by creating the desired components and pass them to dependent components. Currently this is done in the RoboCupClient class but can easily be overwritten by other teams.

Dependency injection also simplifies creation of unit tests for components. Mocks can be used instead of dependent components to simplify the creation of test fixtures enormously. They are simply injected instead of the real components to control the behavior of components not under test. Our own unit tests make heavy use of this.

4 Team

The magmaOffenburg team:

- Klaus Dorer (Team leader)
- Mathias Ehret
- Stefan Glaser
- Thomas Huber
- Simon Raffeiner
- Srinivasa Ragavan
- Ingo Schindler

Our project started four month ago in October last year. Although progress has been excelent, we are currently not yet able to play competitively as the logfile shows. However, with another four months to go we are confident to get to a competitive level of game play.

References

1. Dorer, K.: Modeling Human Decision Making using Extended Behavior Networks. Submitted to RoboCup Symposium, Graz, Austria (2009)
2. Dorer, K.: Extended Behavior Networks for Behavior Selection in Dynamic and Continuous Domains. In: U. Visser, et al. (Eds.) Proceedings of the ECAI workshop Agents in dynamic domains, Valencia, Spain (2004)
3. Dorer, K.: Motivation, Handlungskontrolle und Zielmanagement in autonomen Agenten. PhD thesis, Albert-Ludwigs University (2000)
4. Dorer, K.: Behavior Networks for Continuous Domains using Situation–Dependent Motivations. Proceedings of the Sixteenth International Conference of Artificial Intelligence (1999) 1233–1238
5. Heckhausen, H.: Motivation und Handeln (1989).
6. Maes, P.: The Dynamics of Action Selection. Proceedings of the International Joint Conference on Artificial Intelligence (1989) 991–997
7. Veenstra, A., Neijt, B., Vermeulen, F., Veenstra, G., Prins, J., Kuypers, J., Stollenga, M., vd Sanden, M., Klomp, M., Platje, M., van Dijk, S. The Little Green Bats at http://www.littlegreenbats.nl/ (2008)
8. Kahneman, D., and Tversky, A.: Prospect Theory: An Analysis of Decision under Risk. Econometrica (47) (1979) 263–291